Algorithmic Approaches to Statistical Questions

Gregory Valiant

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-198
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-198.html

September 26, 2012

Algorithmic Approaches to Statistical Questions

by

Gregory John Valiant

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Christos Papadimitriou, Chair
Professor David Aldous
Professor Satish Rao

Fall 2012

Algorithmic Approaches to Statistical Questions

Abstract

Algorithmic Approaches to Statistical Questions

by

Gregory John Valiant

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Christos Papadimitriou, Chair

We live in a probabilistic world—a world full of distributions from which we sample. Learning, evolution, and much of science, rely on samples furnished by nature. This prompts the basic question: Given a sample from some unknown distribution, what can one infer? In even the simplest settings, our understanding of this question is startlingly poor. While this question is traditionally viewed as lying within the province of statistics and information theory, at its core it is an algorithmic question. The increasing size of our datasets—and perhaps more importantly, the increasing complexity of the underlying distributions that we hope to understand—are exposing issues that seem to demand computational consideration.

In this dissertation, we apply the computational perspective to three basic statistical questions which underlie and abstract several of the challenges encountered in the analysis of today's large datasets.

Estimating Statistical Properties Given a sample drawn from an unknown distribution, and a specific statistical property of the distribution that we hope to estimate, how should one compute that estimate, and what sample size is necessary to guarantee that with high probability, the computed estimate is accurate? We focus on a large and natural class of properties, which includes the number of distinct elements, entropy, and distance metrics between pairs of distributions, including total variational distance (also known as statistical distance or ℓ_1 distance). Such properties are easy to estimate if the sample size is large in comparison to the size or complexity of the underlying distribution, but what can be done given relatively few samples? Our results can be interpreted via the following three concrete problems, each defined with respect to an arbitrarily small accuracy parameter $\epsilon > 0$:

- **Distinct Elements:** Given access to n buckets, each of which contains one object that is not necessarily distinct from those in the others buckets, how many buckets must one inspect in order to estimate the total number of distinct objects to $\pm\epsilon n$, with high probability?
- **Entropy Estimation:** Given access to a sample obtained by taking independent draws from a distribution p, of support size at most n, how large does the sample need

to be to estimate the entropy of the distribution, $H(p) := -\sum_{x:p(x)>0} p(x) \log p(x)$, to within $\pm\epsilon$, with high probability?

- **Distance:** Given access to two samples obtained by taking independent draws from two distributions, p_1, p_2 of support size at most n, how large do the samples need to be to estimate the total variational distance between the distributions (also referred to as ℓ_1 distance or "statistical distance"), $D_{tv}(p_1, p_2)$, to within $\pm\epsilon$, with high probability?

We show that *sublinear* sample estimation is possible: for any constant $\epsilon > 0$, the above estimation tasks can be accomplished using $O\left(\frac{n}{\log n}\right)$-sized samples, with probability of success $1 - o(1)$. Additionally, we prove some results about the algorithmic structure of optimal estimators, and provide experimental evidence suggesting that our estimators perform very well in practice. Complementing these positive results, we prove matching information theoretic lower bounds, establishing the sample complexity of these tasks up to constant factors. Previously, no explicit sublinear sample estimators had been described for any of these tasks, and the best previous information theoretic lower bounds on the required sample size for any of these problems was $n/2^{\Theta(\sqrt{\log n})}$ [131].

As a component of the lower bound machinery, we prove two new multivariate central limit theorems: one for sums of independent (though not necessarily identical) multivariate random variables in the Wasserstein metric, and the second for "generalized multinomial distributions" (a class of distributions generalizing binomial, multinomial, and sums of such distributions) in terms of the stringent ℓ_1 distance metric. We suspect these limit theorems may have broader applications beyond the property estimation setting.

Finding Correlations and Identifying Relevant Variables: Perhaps the most basic type of structure that can be present in a dataset is correlation. How much computation is required to find correlated variables? One can certainly brute-force search through all pairs of variables, and for each pair, the correlation can be estimated very efficiently. But is there a *sub-quadratic* time algorithm for finding correlated variables? More generally, suppose one has a data set where each data sample has a label which is given as some function of a small number of the variables. If we have n total variables, perhaps there is a small number, $k = 3, 4, 5, \ldots$, of *relevant* variables which can be used to predict the labels. Such a function is termed a *k-junta*. How quickly can one find this set of k relevant variables? As above, one could simply perform a brute-force search over all possible subsets of size k, taking time roughly $O(n^k)$. Can one find the set of relevant variables significantly more efficiently?

We show that a planted pair of ρ-correlated variables can be found in a set of n otherwise uniformly random Boolean variables in time $n^{1.6} poly(1/\rho)$. This improves upon the $O(n^{2-O(\rho)})$ runtime given by locality sensitive hashing and related approaches. Extensions of this algorithm yield significantly improved algorithms for several important problems, including multiplying matrices whose product is guaranteed to be sparse, learning k-juntas, learning sparse parity with noise, and computing the approximate closest pair of points, in both Euclidean and Boolean settings.

Learning Mixtures of Gaussians A sample from a mixture model (with, for example, two components) is generated via the following process: for each data point, with some probability, w_1, the point is drawn from one distribution p_1, and with the remaining probability $1-w_1$ the point is drawn from a second distribution p_2. Supposing one is given a large sample from such a mixture of distributions, can one efficiently deduce the components, p_1 and p_2 of the mixture? Can one accurately cluster the sample points according to the distribution from which they originated? In the special case in which each component, p_1, p_2 is a Gaussian distribution, this is the problem of learning a *Gaussian mixture model* (GMM), and is, perhaps, the most natural (and practically relevant) starting point for tackling the question of recovering mixtures of more general families of distributions. We obtain a basic handle on the sample and computational complexity of this problem, and describe an algorithm which, given a sample from a GMM with any constant number of components, provably returns accurate estimates of the components, with runtime and sample size polynomial in the relevant parameters—the dimension of the space, and the inverse of the desired accuracy of the recovered components. Previously, no such algorithm was known, even in the special case of univariate mixtures with just two components.

The questions considered in this dissertation are not new: the question of efficiently finding correlations was introduced to the computer science community over 25 years ago; the effort to describe accurate estimators for entropy and the other properties that we consider originated in the statistics community nearly 75 years ago and also received significant attention from the information theory community; the question of recovering Gaussian mixture models was originally posed by Karl Pearson in the 1890's. The progress on these questions that we describe in this dissertation stems from the observation that these statistical questions are intrinsically algorithmic and hence may be amenable to the tools, techniques, and perspectives of theoretical computer science.

Acknowledgments

When I arrived at Berkeley's graduate student visit day, Christos rescued me from the rain and drove me the final block up the hill to Soda Hall. By the time we had gotten to Soda Hall, I had chosen which graduate school to attend, and selected an advisor. I could not have made a better decision. Christos' raw energy, and passionate belief in computer science theory as a vantage point from which to survey the world, are both inspiring, and contagious. I am grateful to Christos for letting me pick my own path, and even more grateful to him for the abundance of wisdom, encouragement, support, and friendship he offered as I stumbled down this path. Perhaps most importantly, Christos, thank you for your reminder that we live in an extraordinarily rich and beautiful world, and that it is best to keep ones eyes open.

I am very grateful to all the students, postdocs, and faculty in the Theory group at Berkeley, who create the atmosphere that is Berkeley. I may now understand that distant and teary-eyed look so endemic to Berkeley graduates when they reflect on their student days. Umesh, thanks for always saying the right things at precisely the right times. Satish, thank you for optimistically entertaining my ideas and for our many hours of discussions. Alistair, I consider myself extremely fortunate to have TAed for you—much of what I know about probability theory, and teaching, is due to you.

I want to thank my fellow students for all the inspiring discussions, fantastic retreats, and clandestine coffee raids: special thanks to James, Thomas, Piyush, Jonah, Siu Man, Siu On, Chris, Seung Woo, Tom, Di, Anupam, Anand, Isabelle, George, Raf, Jake, Meromit, Guy and Mayan; and to the students who served as role models and mentors in the years above me, Costis, Grant, Alexandra, Alexandre, Alex, Henry, Omid, Lorenzo, and Madhur. Anindya, Urmila, Ilias, and Yaron, thank you for all your ideas, stories, advice, and friendship.

I am extremely grateful to my collaborators: Ho-Lin Chen, Costis Daskalakis, Ilias Diakonikolas, Rafael Frongillo, Georg Gottlob, Adam Kalai, Stephanie Lee, Ankur Moitra, Noam Nisan, Christos, George Pierrakos, Tim Roughgarden, Michael Schapira, Grant Schoenebeck, Rocco Servedio, Paul, and Aviv Zohar. Thank you for sharing your special insights and unique perspectives with me; I look forward to many more discussions with each of you.

I owe much to Adam Kalai, and Vitaly Feldman for mentoring me during my summer internships at Microsoft and IBM. Adam, thank you for introducing me to learning theory, for pushing me to think clearly and deeply, and for all the fun along the way. Vitaly, thank you for many, many hours of discussing parity with noise, and for the bike rides.

Thank you to Berkeley, the National Science Foundation, and to IBM for funding my graduate education and giving me the liberty to pursue the research directions of my choosing.

Finally, none of this would have been possible without Paul and Steph and my parents Leslie and Gayle, who make me who I am and offer their constant support and love.

To Paul and Steph for all the adventures.

Contents

List of Figures

Chapter 1

Data, Computation, Statistics, and Learning

The large datasets of today are not like the datasets of the 20th century. The hardware used today to store these datasets is not like that of the 20th century. The software used today to manipulate these datasets is not like that of the 20th century. And yet, in many cases, our theoretical understanding of basic statistical tasks *is* like that of 20th century. Basic tasks, such as finding correlated variables in a dataset, or estimating the difference between two distributions, change fundamentally when one starts to consider very large datasets, or very complicated distributions. Given a dataset with a modest number of variables, a simple textbook calculation will let one compute the correlations between the variables and find any correlations that might be present. If one wishes to estimate the difference in distributions of heights of people in two demographics, one simply takes large samples from each group, and compares the two empirical distributions.

The story is quite different, however, if we are hoping to find correlations in a dataset that has millions, or billions of variables, such large genomic datasets in which each position in the genome might be interpreted as a variable. Instead of estimating the difference in distributions of heights between groups of people, consider trying to estimate the difference between the distributions of two human gut "microbiomes"—large and complex distributions (with a domain consisting of hundreds or thousands of species of bacteria, yeasts, fungi and protozoa) the majority of which are only present in tiny quantities and thus may be observed few times, if at all, in a given sample [13, 143]. How does one estimate the difference in these distributions if much of the domain is unseen in our sample?

The extreme parameters of the datasets and distributions that we are now facing reveal aspects of these very basic problems that were not apparent in more traditional settings. In many cases, these newly exposed challenges are fundamentally computational in nature. Applying algorithmic tools and ideas, and more importantly, viewing these statistical challenges through a computational lens, seems essential. Further, this is a two-way street. While these new challenges have significant practical implications, some are also extremely provoking and elegant mathematical problems. As such, they carry the potential to spawn a rich array of powerful new theoretical ideas and insights that may find applications in a range of settings both within theoretical computer science, and more generally.

1.1 Statistical Property Estimation

What can one infer about an unknown distribution based on a sample? If the distribution in question is relatively "simple" in comparison to the sample size—for example if we are given a sample consisting of 1000 independent draws from a distribution supported on 100 distinct domain elements—then the empirical distribution of the sample will likely be an accurate representation of the actual distribution. If, on the other hand, we have a relatively small sample in relation to the size and complexity of the distribution—for example if we have a sample of size 100 drawn from a distribution supported on 1000 domain elements—then the empirical distribution may be a poor approximation of the actual distribution. In this case, can one still extract accurate estimates of various properties of the actual distribution?

Many real–world machine learning and data analysis tasks face this challenge. In this age of big data, the rapid increase in the size of our datasets has, in many cases, been accompanied by a realization that the underlying distributions we hope to understand are far larger and more complex than we may have imagined. Thus despite the enormous size of some of these datasets, we are only viewing a tiny fraction of the domain of the actual distribution.

One especially relevant illustration of this point is the discovery that the rapid growth of the human population over the past 5000 years has resulted in an abundance of very rare genetic mutations. Two recent independent studies [121, 93] (appearing in *Science* in July, 2012) each considered the genetic sequences of over 14,000 individuals, and found that rare variants are extremely abundant, with over 80% of mutations observed just once in the sample [121]; the conclusion is that even with huge numbers of sequenced genomes, "rare [genetic] variant catalogs will be largely incomplete" [93]. Understanding these distributions of rare mutations provides insight into our evolution and selective pressures, as well as the potential for genetic screenings for various diseases. Highlighting the difficulty in working with such sparse data, the paper [80] (also appearing in *Science* in May, 2012) found that the discrepancy in rare mutation abundance cited in different demographic modeling studies can largely be explained by discrepancies in the sample sizes of the respective studies, as opposed to differences in the actual distributions of rare mutations across demographics. These works highlight some of the recent struggles of the genetics community to deal with this pervasive question of how to accurately infer properties of large and complex distributions given a sample that is "too small".

Similar challenges are encountered in a variety of other fields, including Biology, Ecology, Linguistics, Neuroscience, and Physics (see, for example, the discussion and extensive bibliographies in [32, 102]). Specific settings in which these problems arise that may be more familiar to the computer science community include analyzing customer data or web traffic (many customers or website users are only seen a small number of times), and text analysis (most of one's vocabulary is not represented in a given writing sample). Additionally, many database management tasks employ sampling techniques to optimize query execution; improved estimators would allow for either smaller sample sizes or increased accuracy, leading to improved efficiency of the database system (see, e.g. [95, 65]).

But what can one hope to infer about a distribution if we are given such a small sample that much of the domain of the distribution has not been seen? We can not know which domain elements we have not seen, but we might still hope to estimate certain properties of the distribution that depend on this unseen component of the distribution.

To give a simple example, suppose one reaches into a large bag of marbles, and pulls out a handful consisting of 10 marbles that are each a different color. If none of these 10 marbles are yellow, we certainly should not conjecture that there is a yellow marble in the bag. Nevertheless, based on this sample of size 10, we might be inclined to suspect that the bag of marbles contains many colors of marbles that we did not see. Indeed, if the bag only contained 10 different colors, then the probability that we would see all 10 colors in a (random) sample of size 10 is very small—less than 1 in 2,500—and we could safely conclude that the bag likely contained at least 20 colors of marbles, since if this were not the case, the probability of having drawn 10 distinct colors in our sample of size 10 would be very small (< 0.1, in fact). We have not made any assumptions on the distribution of marbles—these conclusions were not made from a Bayesian standpoint—and yet we have used our sample to infer something about the portion of the distribution from which we have drawn no elements.

In the above example, we were reasoning about the support size of the distribution; this property of the distribution was amenable to such speculation, in part, because we did not need to know the labels of the unseen elements in order to reason about their potential contributions to this property. The class of distribution properties that we consider in this dissertation is precisely characterized by this independence from the labels of the support of the distribution. We term such properties *symmetric*. Formally, a property of a distribution (or set of distributions) with discrete support is *symmetric* if the property is invariant to relabeling the support elements.

Many natural and practically relevant properties are symmetric, including measures of the amount of structure or diversity of the distribution, such as the support size or entropy of the distribution. For properties of pairs of distributions, the class of symmetric properties contains measures of how similar two distributions are, such as total variational distance (and more generally, ℓ_k distance metrics), and KL-divergence. Our results apply to a large subclass of symmetric properties; for clarity, we summarize our results in terms of the following three concrete questions, which are parameterized by an arbitrarily small constant error parameter $\epsilon > 0$:

- **Distinct Elements:** Given access to n buckets, each of which contains one object that is not necessarily distinct from those in the other buckets, how many buckets must one inspect in order to estimate the total number of distinct objects to $\pm\epsilon n$, with high probability?[1]

[1]We phrase our results for estimating the support size of a distribution in terms of this 'distinct elements' problem. Slightly more generally, one could also frame this as the problem of estimating the support size of a distribution given the promise that all domain elements occur with probability at least $1/n$. Estimation is impossible without such a lower bound simply because otherwise, an arbitrarily large number of domain elements can occupy an arbitrarily small amount of probability mass.

- **Entropy Estimation:** Given access to a sample obtained by taking independent draws from a distribution p, of support size at most n, how large does the sample need to be to estimate the entropy of the distribution, $H(p) := -\sum_{x:p(x)>0} p(x) \log p(x)$, to within $\pm\epsilon$, with high probability?

- **Distance:** Given access to two samples obtained by taking independent draws from two distributions, p_1, p_2 of support size at most n, how large do the samples need to be to estimate the total variational distance between the distributions (also referred to as ℓ_1 distance or "statistical distance"), $D_{tv}(p_1, p_2) := \sum_{x:p_1(x)+p_2(x)>0} \frac{1}{2}|p_1(x) - p_2(x)|$, to within $\pm\epsilon$, with high probability?

For all three problems, it is clear that the required sample size will be some increasing function of n. As described above, if the sample size is much larger than the support size of the actual distribution, we expect the empirical distribution defined by the sample to be close to the actual distribution, and thus one can recover an accurate estimate of the distribution and estimate the property in question by returning the property value of the empirical distribution defined by the sample. These arguments can be used to show that given a sample of size $O(n/\epsilon^2)$, both entropy and total variational distance can be estimated to $\pm\epsilon$, with high probability. Of course, for the distinct elements problem, simply looking at all n buckets will yield the total number of distinct elements. The question is whether one can improve upon these trivial sample complexities: in particular, can one estimate these properties using a *sublinear* sized sample? We show that the answer is "yes".

> *For any constant $\epsilon > 0$, to estimate the number of distinct elements given n buckets to accuracy $\pm\epsilon n$, or estimate the entropy of a distribution of support size at most n to $\pm\epsilon$, or total variational distance between two distributions of support size at most n to $\pm\epsilon$, a sample consisting of $O(\frac{n}{\log n})$ independent draws from the distribution in question is sufficient (or two such samples in the case of total variational distance).*

Prior to this work, despite a simple nonconstructive argument showing the existence of an $o(n)$ sample estimator for entropy [102, 101], the best proposed explicit estimators for any of these properties required $\Omega(n)$-sized samples to produce an estimate with constant error.

Perhaps not unexpectedly, the crux of our estimators is a new approach to characterizing the unobserved portion of a distribution—the portion of the distribution from which we have drawn no examples.

This effort to infer properties of the unseen portion of a distribution is not new; both Alan Turing and R.A. Fisher, the respective fathers of computer science and statistics, independently considered aspects of this problem. Working with I.J. Good during WWII to understand the distribution of the German enigma machine ciphers, Turing was interested in estimating the total probability mass in the distribution that is composed of domain elements

that have not been observed. Stated in a different fashion, Turing wanted to predict the probability that the next element drawn is a new (previously unobserved) element of the support. This work is now known as the *Good–Turing frequency estimation scheme*, which has since been analyzed and extended in a long line of work by both computer scientists and statisticians [86, 96, 97, 139, 138]. Fisher was interested in a related parameter: the number of new elements that one expects to discover in a given time period.

In contrast to the work of Fisher and Turing, rather than simply trying to estimate a single parameter of the unseen portion of the distribution, we try to characterize the entire shape of the distribution in this region. We can never reconstruct the labels of the unseen portion of the support of the distribution, but we can hope to recover estimates of the number of domain elements that occur within various probability ranges. For the purposes of estimating symmetric properties, this representation of the "shape" of the distribution contains all the relevant information.

Lower Bounds

Complementary to our positive results, we prove matching information theoretic lower bounds showing that, up to constant factors, our estimators are optimal; together, this settles the question of the sample complexities of the distinct elements problem, estimating the entropy of a distribution, and estimating total variational distance between distributions.

> *For the distinct elements problem, no algorithm that looks at $o(\frac{n}{\log n})$ entries can estimate the number of distinct elements to within $\pm 0.1n$, with any probability greater than 0.51. Similarly, for the problems of entropy and total variational distance estimation, no algorithm that takes an $o(\frac{n}{\log n})$-sized sample can estimate these properties of distributions of support at most n to within ± 0.1 with probability of success greater than 0.51.*

The challenge in proving an information theoretic lower bound is that one must argue about the distribution of a set of independent draws from a distribution. These are complex, discrete, high dimensional distributions, and there are relatively few tools available to analyze these distributions. To enable the analysis of our lower bound construction, we develop two new tools.

We prove two new multivariate central limit theorems, one via Stein's method in terms of the Wasserstein (earthmover's) metric, and one in terms of total variational distance (the ℓ_1 metric). While multivariate central limit theorems are known (for example, [61]), the Wasserstein metric is an especially useful and natural metric in this setting, and it is surprising that such a limit theorem was not previously known. Our second limit theorem is useful both because the bound has a very modest dependence on the dimension (a linear dependence), and because the characterization is in terms of the stringent ℓ_1 metric—two distributions with small ℓ_1 distance are, information theoretically, indistinguishable given a small sample. Such limit theorems seem especially rare in the multivariate setting. We hope

(and suspect) that these limit theorems will find applications beyond the property estimation setting.

The Structure of Estimators

Finally, we consider the structure of estimators as algorithmic objects. Nearly all the estimators proposed by the statistics community for the properties that we consider can be expressed as formulae that map the parameters of a sample in a transparent fashion to an estimate of the desired property. Our estimators, in sharp contrast, are "canonical" and extremely algorithmic: the sample is used to formulate a linear program. The set of feasible points with low objective function values roughly correspond to the set of "shapes" of distributions from which the sample could plausibly have been drawn. Given a solution to this linear program, to estimate a specific property, one then simply evaluates the property value of the distribution represented by that solution. It is worth stressing that the majority of the algorithmic work is done independently of the specific property one wishes to estimate.

This contrast between our estimators and the long line of proposed estimators from the statistics community that require significantly larger samples to achieve the same level of accuracy as our estimators, prompted two questions: 1) Is the full algorithmic power of linear programming necessary to achieve this level of sample efficiency? 2) Given a specific property of interest, is there a more direct estimator; namely, is there an estimator that directly estimates (say) entropy *without* first estimating the "shape" of the distribution? We show that there *do* exist near-optimal *linear* estimators—estimators that compute the vector of collision statistics of the sample, $\mathcal{F}_1, \mathcal{F}_2, \ldots$, where $\mathcal{F}_i$ represents the number of domain elements seen exactly i times in the sample, and then simply return the dot product between this vector, and a vector of precomputed coefficients.

Our proof that near-optimal linear estimators exist establishes a correspondence between the problem of finding worst-case lower bound instances and the problem of finding good linear estimators. Thus these optimal linear estimators, in a rigorous sense, are tailored to worst-case distributions. Our linear programming based estimators achieve the same worst-case performance, yet do not seem to be directly related to any lower bound constructions, perhaps suggesting that they might perform better than the linear estimators on typical or "easy" instances.

As a conclusion to the section of this dissertation on estimating symmetric properties, we implemented practical variants of these estimators, and experimentally evaluated them against a variety of estimators from the literature on a range of synthetic data, and real text data. While these experiments should not be construed as a comprehensive evaluation of these estimators, the performance of our linear programming based estimator is extremely compelling.

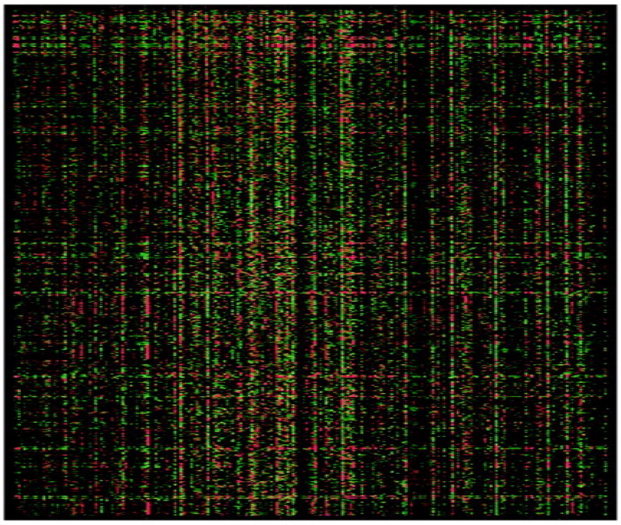

Figure 1.1: A DNA microarray: each row corresponds to a cell sample, each column corresponds to a gene, and the intensity of each entry corresponds to the level of gene expression. Computationally, how does one efficiently find genes whose expressions are correlated? (Image from L. Liu et al. [82]).

1.2 Finding Correlations and Relevant Variables

The first section of this dissertation considers the problem of estimating properties of distributions without any assumption on the structure of the true distribution. The remaining two sections consider the problem of finding the structure in a distribution, given that such structure exists. Perhaps the most basic type of structure that might be present is correlation between pairs of variables, or the analogs of correlation for sets of three or more variables. The algorithmic task of finding such relationships in a database is pervasive, both as a component within larger data analysis programs, and as an ends in itself.

We begin by describing two concrete problems in biology for which the computational complexity of finding correlations and sets of related variables has taken center stage. Figure 1.1 depicts a DNA microarray; the rows of the microarray correspond to cell samples, and the columns correspond to genes. For each cell sample/gene pair, the corresponding entry of the microarray depicts the level of gene expression in that cell sample. Typical microarrays can involve on the order of a hundred samples, and thousands of genes [82]. The most basic information that biologists hope to glean from such data is an understanding of which pairs of genes are coregulated (have expressions that are correlated or anti–correlated). To find such pairs of genes, must one perform a brute-force search over all pairs of columns of the microarray, or does there exist a significantly faster algorithm?

In many cases, in addition to the microarray, each cell sample has a label: for example,

suppose that the top half of the samples correspond to healthy cells, whereas the bottom half of the samples correspond to cells exhibiting some disease. The hope, which often proves well founded, is that there might be a very small number ($k = 2, 3, 4, \ldots$) of genes from whose expressions one can glean some indication as to whether the sample is healthy or not. Such information may help suggest biomarkers of the disease or even possible avenues of disease control. If one has a microarray with $n > 1000$ genes, must one search all $O(n^k)$ subsets of k genes to find such small sets of *relevant* genes? Even for very modest values of k, such a brute–force search is computationally infeasible.

The recent surge of *genome wide association studies* (GWAS) provide another compelling potential application of improved algorithms for finding relevant variables. Such studies involve datasets consisting of several million SNPs (single-nucleotide polymorphisms, which account for the portion of the genome that seems likely to account for differences between humans) that have been sequenced for thousands of people. The task is then to try to explain the presence or absence of certain traits using the genetic information. Currently, many of the findings are for single SNPs that correlate with traits. Presumably, there are many traits for which no single SNP has significant explanatory value, but for which a pair, or triple of SNPs does have explanatory value. Can one find such sets of relevant SNPs without performing a brute-force search over the quadratic or cubic number of such potential hypotheses?

We begin to tackle these questions from the most basic setting in which one can consider the problem of finding correlations:

> *Given a set of n d-dimensional Boolean vectors with the promise that the vectors are chosen uniformly at random with the exception of two vectors that have Pearson–correlation ρ (i.e. each index of the pair of vectors agree with probability $\frac{1+\rho}{2}$), how quickly can one find the correlated pair?*

This problem was, apparently, first posed by Leslie Valiant in 1988 as the *light bulb problem* [130]. This name owes itself to its original phrasing in terms of n light bulbs that each blink on and off randomly at each time step, with the exception of a pair of lightbulbs that are correlated.

In the case that $\rho = 1$, the pair of correlated vectors is identical. In such a setting, provided the dimension d is slightly larger than $\log n$, then with high probability the true correlated pair will be the only pair of vectors that are identical, and one can find such a pair in near linear time by the following simple approach: consider each length d vector as a d-digit Boolean number, sort the set of n such numbers, and then perform a single pass through the sorted list to see if any two adjacent numbers are identical. Such an algorithm runs in time $O(n \log n)$, improving upon the trivial quadratic time brute-force approach. This algorithm, however, relies crucially on the assumption that the pair of correlated vectors are identical. For $\rho < 1$, it is not clear how to obtain such improvements in runtime.

It is worth stressing that the issue for $\rho < 1$ is computational rather than information theoretic. The number of indices one requires to information theoretically determine the

correlated pair decays modestly as a function of ρ. A simple Chernoff bound shows that as long as $d = \Omega\left(\frac{\log n}{\rho^2}\right)$, with high probability the pair of vectors that differ in the fewest locations will be the true correlated pair.

The earliest results for the light bulb problem are due to Paturi et al. [104], and give an algorithm whose runtime is $O(n^{2-O(\rho)})$. The Locality Sensitive Hashing approach of Indyk and Motwani [71], and the Bucketing Codes approach of Dubiner [50] also give algorithms that run in time $O(n^{2-O(\rho)})$, with the approach of Dubiner achieving the best constants, with a runtime of $O(n^{2-2\rho})$, in the limit as ρ gets small.

In these previous works, because ρ appears in the exponent of n, for small values of ρ these approaches do not yield appreciable savings over the trivial $O(n^2 \log n)$ runtime of the brute-force search. This small-ρ regime is especially relevant in practice because correlations frequently do degrade with the dimension of the space that one is working in, for the simple reason that random vectors in high dimensional space will be nearly orthogonal with high probability. Our first result is a sub-quadratic algorithm that has an inverse polynomial dependence on the correlation, ρ:

> *Given n random Boolean vectors in sufficiently large dimension with the promise that there is a pair of vectors that is ρ-correlated, the correlated pair can be found in time $n^{\frac{5-\omega}{4-\omega}} poly(1/\rho) < n^{1.62} poly(1/\rho)$, where $\omega < 2.38$ is the exponent of matrix multiplication.*

We note that an extension of this algorithm for the light bulb problem can also be viewed as an improved algorithm for approximating the product of two matrices given the promise that their product has a small number of large entries.

More generally, the light bulb problem is a special case of the Boolean Approximate Closest Pair problem: given a set of Boolean vectors, how can one quickly find two vectors with near-minimal Hamming distance (i.e. that differ in the fewest number of indices)? The Locality Sensitive Hashing approach of Indyk and Motwani [71] can find a pair of vectors whose Hamming distance is at most a factor of $(1+\epsilon)$ times that of the distance between the closest pair, and achieves runtime $O(n^{1+\frac{1}{1+\epsilon}})$, which tends to $O(n^{2-\epsilon})$ for small ϵ. Subsequent work on Locality Sensitive Hashing improves this dependence for other metrics—specifically, Andoni and Indyk [10] show that this problem can be solved in time $O(n^{1+\frac{1}{(1+\epsilon)^2}}) \approx O(n^{2-2\epsilon})$ for ℓ_2 distance, as opposed to Hamming distance. The main ideas used in our algorithm for the light bulb problem can be extended to yield an improved algorithm for the $(1+\epsilon)$ approximate closest pair problem in both the Boolean (Hamming) and Euclidean settings:

> *Given n points in $\mathbb{R}^d$, for any constants $\epsilon > 0$, with high probability, our algorithm finds a pair of vectors whose distance is at most a factor of $(1+\epsilon)$ larger the distance between the closest pair. Additionally, the runtime is*
>
> $$O(n^{2-\Theta(\sqrt{\epsilon})}) + nd \cdot poly(\log n).$$

The best previously proposed algorithms for this problem achieve runtime $O(n^{2-\Theta(\epsilon)} + nd)$.

All previous approaches to the light bulb and closest pair problems take the following rough approach: first project the vectors into a lower-dimensional space, then try to cleverly hash or cluster the resulting vectors in such a way that vectors that are close have a higher probability of being assigned to the same bucket or cluster. In sharp contrast to these approaches, we perform a metric embedding that carefully projects the vectors to a slightly *higher-dimensional* space. This projection has the property that vectors that are correlated will remain relatively strongly correlated after the projection, however vectors that are very weakly correlated, will end up projecting to nearly orthogonal vectors. After this projection step, sets of vectors are aggregated so as to effectively reduce the number of vectors that must be considered. Finally, we leverage fast matrix multiplication algorithms. Our results for the light bulb problem use the fact that $n \times n$ matrices may be multiplied in time $O(n^\omega)$, for $\omega < 3$. The best bound on ω is due to Virginia Vassilevska Williams [141], who showed that $\omega < 2.372$. Our results for the approximate closest pair problem rely on a fact shown by Coppersmith, that for any $\epsilon > 0$, for $\alpha < 0.29$, the product of an $n \times n^\alpha$ and $n^\alpha \times n$ matrix may be computed in time $O(n^{2+\epsilon})$ [42].

Learning Juntas and Parities

The problem of identifying relevant variables is related to the light bulb problem via the problem of learning parity with noise, which we now describe. Suppose one is given access to a sequence of examples (x, y), where $x \in \{-1, +1\}^n$ is chosen uniformly at random, and $y \in \{-1, +1\}$ is set so that $y = z \prod_{j \in S} x_i$, for some fixed, though unknown set $S \subset [n]$, where $z \in \{-1, +1\}$ is chosen to be -1 independently for each example with probability $\eta \in [0, 1/2)$. In the case where the *noise rate* $\eta = 0$, the problem of recovering the set S is easy: given n such examples, with high probability one can recover the set S by Gaussian elimination—translating this problem into a problem over $\mathbf{F}_2$, S is given simply as the solution to a set of linear equations. From an information theory standpoint, the addition of some nonzero amount of noise ($\eta > 0$) does not change the problem significantly; for constant η, given $O(n)$ examples, with high probability there will only be one set $S \subset [n]$ where the parities of the corresponding set of indices of the examples are significantly correlated with the labels. From a computational standpoint, the addition of the noise seems to change the problem entirely. In contrast to the simple polynomial-time algorithm for the noise-free case, when given a small constant amount of noise the best known algorithm, due to Blum et al. [27] takes time $2^{O(\frac{n}{\log n})}$, which is a super–polynomial improvement over brute-force search, though still a far cry from polynomial–time.

This problem of learning parity with noise is, increasingly, understood to be a central problem in learning theory. Beyond learning theory, this problem reoccurs in various forms throughout theoretical computer science, including coding theory (as the problem of decoding random binary linear codes) [85], and cryptography via its relation to the "learning with errors" problem whose assumed hardness is the basis for many cryptosystems, including the recent work on fully homomorphic encryption (see e.g. [4, 110, 30]).

Our results for learning parities apply to the setting in which the true parity set S is much smaller than n, say $k := |S| = O(\log n)$. This problem of learning k-parities is especially relevant to learning theory, as was revealed by a series of reductions given in work of Feldman et al. [54], showing that the problem of learning k-parities (under the uniform distribution, with random classification noise) is at least as hard as the problems of learning Boolean functions of k variables (termed k-juntas), learning 2^k-term DNF from uniformly random examples, and the variants of these problems in which the noise is adversarial (rather than random).

This reduction has a natural interpretation: the problem of learning a parity with noise is the problem of finding a set of k indices whose parity is correlated with the labels, given that such a set exists. In other words, it is the problem of finding a heavy Fourier coefficient, given that one exists. In the case of learning an arbitrary function of just k variables—a *k-junta*—basic Fourier analysis shows that there will be at most 2^k sets of indices whose parity is significantly correlated with the label: i.e. there will be at most 2^k heavy Fourier coefficients. Intuitively, the presence of more heavy low-degree Fourier coefficients should, if anything, facilitate the task of finding such a coefficient.

We show the following result for learning sparse parities with noise:

> *For any constant $\epsilon > 0$, the problem of learning parities of size k from uniformly random length n strings, with noise rate η can solved in time*
>
> $$n^{\frac{\omega+\epsilon}{3}k} poly(\frac{1}{1/2-\eta}) < n^{0.80k} poly(\frac{1}{1/2-\eta}),$$
>
> *where ω is the exponent of matrix multiplication.*

All previous algorithms for this problem had a runtime with the noise rate η appearing in the exponent of n. This result, via the reduction of Feldman et al. [54] yields the following corollaries for learning k-juntas (identifying sets of k relevant variables):

> *Given labeled uniformly random binary examples of length n, where the label is given as a function of $k << n$ of the indices, for any constant $\epsilon > 0$, the set of relevant indices can be found in time*
>
> $$O(n^{\frac{\omega+\epsilon}{4}k}) < O(n^{0.60k})$$
>
> *Additionally, if an $\eta < 1/2$ fraction of the labels are corrupted, the runtime is*
>
> $$n^{\frac{\omega+\epsilon}{3}k} poly(\frac{1}{1/2-\eta}) < n^{0.80k} poly(\frac{1}{1/2-\eta}).$$

These results improve upon the algorithm of Mossel et al. [92] showing that size k juntas can be learned (in the absence of noise) in time $O(n^{\frac{\omega k}{\omega+1}}) \approx O(n^{0.70k})$. In the setting with classification noise $\eta > 0$, no algorithm running in time $O(n^{ck})$ for any constant $c < 1$ was previously known.

1.3 Learning Mixtures of Gaussians

For some datasets, either by assumption or because of knowledge of the underlying process that generates the data, one can assume, *a priori*, the family of distributions from which the data were drawn. The problem then is to estimate which member of the family gave rise to the data.

To illustrate, perhaps one knows that each data point arises as the aggregate sum of many independent, nearly identical random variables (such is the case for many data sets encountered in physics, biology, and the social sciences); in such settings, via the central limit theorem, one can assume that the data will be roughly distributed according to a Gaussian distribution. The goal then, would be to estimate the mean and covariance matrix of the distribution; in the case of a Gaussian distribution, this estimation task is trivial—the mean and covariance of the data samples will converge quickly to the true mean and covariance of the actual underlying Gaussian distribution. For many common families of distributions, however, constructing good algorithms for learning the distributions is much more opaque, and in many cases we know very little about how the necessary sample size scales with basic parameters such as the desired accuracy, dimensionality of the space, or the computational power of the estimator.

Given the ease with which one can estimate Gaussians, it is natural to consider the problem of learning *Gaussian mixture models* (GMMs). Gaussian mixture models are one of the oldest, and most widely used statistical model, and consist of a weighted combination of heterogeneous Gaussians, with probability density given as the weighted sum of the densities of the component Gaussians. To give a simple one–dimensional example, consider the distribution of the heights of adults; this distribution can be closely modeled as a Gaussian mixture with two components, one corresponding to the heights of men, and the other corresponding to the heights of women, as is depicted in Figure 1.3. Can one recover accurate estimates of the distributions of the heights of men and women given only the aggregate data without gender labels?

The study of reconstructing the parameters of the Gaussian mixture model dates back to work from the 1890s of Karl Pearson [105]. More recently, motivated by the need to analyze large, high-dimensional data sets, the question of learning GMMs has been revisited. Dasgupta formally introduced the problem of learning GMMs to the theoretical computer science community in a paper which described a polynomial-time algorithms for learning GMMs, under the assumption that the component Gaussians are extremely far apart, in comparison to their covariances [43]. Given such an assumption, the algorithm proceeds by first trying to consistently cluster the sample points according to which component generated that data point. Given an accurate such clustering of the sample, estimating the distribution is easy: one can simply return the empirical mean, covariance, and weight of each cluster. While the task of accurately clustering the sample is quite easy in one or two dimensions, in very large dimension, even if the components are sufficiently far apart so as to have little overlap in their probability density, it is certainly not obvious how to perform such clustering. In contrast to the low-dimensional setting in which one expects to see many data

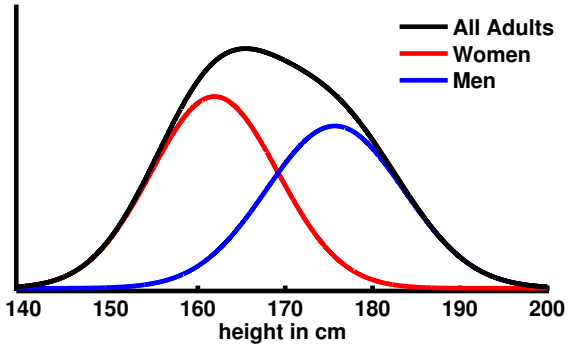

Figure 1.2: The Gaussian approximations of the heights of adult women (red) and men (blue). Can one recover estimates of these Gaussian components given only the aggregate data without gender labels (black)? (Data from the National Health and Nutrition Examination Surveys [87].)

points near the mean of each component, in the high dimensional setting, almost all points will lie on a thin spherical shell, and hence one will see almost no points near the mean of each component, significantly complicating the task of clustering. To facilitate clustering, in Dasgupta's original paper the separation between components was assumed to be very large in comparison to the dimension of the points, and additional assumptions were placed on the components. These conditions were relaxed in a long line of work [3, 12, 31, 76, 134], though any algorithm that proceeds by clustering must assume considerable separation between components.

Despite considerable attention to this problem of learning GMMs, Pearson's original question of learning a one dimensional GMM in which the components might overlap substancially, remained. In particular, it was not known whether the sample size required to estimate the parameters to within a desired accuracy increased polynomially with the inverse of the desired accuracy, or exponentially. Phrased in the language of statistics, it was not known whether an optimal estimator of the components had polynomial, or exponential convergence rate.

> *Our algorithm takes a sample from a GMM in d dimensional space, and outputs approximations of the mixing weights accurate to $\pm\epsilon$ and estimates of the constituent components, accurate to within $\pm\epsilon$ in total variational distance (ℓ_1 distance). Additionally, the required sample size and runtime are bounded by $poly(d, \frac{1}{\epsilon})$.*

Rather than attempting to first cluster the sample points, our approach is based on the *algebraic* structure of the GMM. We returned to Pearson's original one-dimensional setting, and analyze a variant of the method of moments that he had employed, to show that

1-dimensional (univariate) GMMs are *robustly identifiable*. Namely, we show that if the components of one mixture differ by ϵ from those of another mixture, then one of the low-order moments of the two mixtures must be significantly different—must differ by at least $poly(\epsilon)$. Such an argument guarantees that if we have accurate estimates of the low-order moments of the true mixture, then any mixture whose moments closely match the true moments must have the property that the components of the recovered mixture closely match the components of the true mixture. The problem then is to simply come up with such a mixture; this task can be performed by a simple brute-force search through a polynomially–coarse net of mixtures.

Given such an algorithm for recovering mixtures in one dimension, we reduce the general problem of learning mixtures in high dimension to a series of one-dimensional learning problems—leveraging the property of Gaussians that the projection of a high-dimensional Gaussian distribution onto any vector is a one-dimensional Gaussian. While our algorithm has a super-exponential dependence on the number of mixture components, we prove that at least an exponential dependence is information theoretically necessary to recover the components.

1.4 Thesis Organization

Part I: Estimating Statistical Properties

Chapter 2–Definitions, and Related Work. We give an introduction to property estimation, and discuss approaches from both the statistics and computer science communities, beginning with the early work of Turing, Fisher, and Good on the problem of inferring properties of distributions given a sample that seems "too small". We define the concepts and notation that are used throughout Part I.

Chapter 3–Estimating the Unseen: Sublinear Sample Estimators for Entropy, Support Size, and Other Properties. We describe an algorithm for accurately estimating the shape of the unseen portion of a distribution given a relatively small sample, and prove that this algorithm yields sublinear-sample estimators for a class of symmetric distribution properties. In terms of entropy estimation, we show that for any constant $\epsilon > 0$, our algorithm when given a sample consisting of $O(\frac{n}{\log n})$ independent draws from a discrete distribution of support size at most n, will output an estimate of the entropy of the distribution that is accurate to within $\pm\epsilon$, with high probability over the randomness of the sample. Similarly, given an instance of the distinct elements problem with n buckets, our algorithm will query $O(\frac{n}{\log n})$ buckets and return an estimate of the number of distinct elements to within $\pm\epsilon n$, with high probability. Finally, we show that the entire property estimation approach can be extended to estimate properties of pairs of distributions, including distance metrics. As an illustration, we describe an estimator that takes as input a pair of samples, and estimates the ℓ_1 distance (total variational distance) between the two distributions from

which the samples were drawn. We prove that for any constant ϵ, given $O(\frac{n}{\log n})$-sized samples from each of two distributions of support size at most n, our estimator will return an estimate of their ℓ_1 distance that is accurate to within $\pm\epsilon$, with high probability over the randomness of the samples. These are the first explicit sublinear-sample estimators for any of these properties, and the performance of these estimators matches the lower bounds of Chapter 5, to constant factors.

Chapter 4–Two Multivariate Central Limit Theorems In this chapter, we prove two new multivariate central limit theorems. Our first limit theorem is very general, and compares the sum of independent (though not necessarily identical) multivariate random variables to the Gaussian distribution of corresponding mean and covariance in the Wasserstein distance metric (also known as the "Earthmover" metric). We prove this limit theorem directly via Stein's method. We leverage this general central limit theorem to prove our second limit theorem, which is more specific and stronger. Our second limit theorem applies to "generalized multinomial distributions"—a class of distributions that generalizes binomial and multinomial distributions, and any sums of such distributions. We show that such distributions are close in total variational distance (ℓ_1 distance) to the *discretized* Gaussian of corresponding mean and covariance (defined by the process of drawing a point from the Gaussian and then rounding the point to the nearest point in the integer lattice).

Chapter 5–Lower Bounds for Property Estimation We prove information theoretic lower bounds on the sample size required to estimate entropy, the number of distinct elements, and total variational distance between distributions, establishing the optimality of the estimators of Chapter 3, up to constant factors. For any n, we describe an ensemble of distributions, half of which are close in variational distance to a uniform distribution over $n/2$ elements, and half of which are close to a uniform distribution over n elements, yet which have the property that given $o(\frac{n}{\log n})$ independent draws from one of the distributions of the ensemble, it is information theoretically impossible to distinguish the two cases. Both the construction, and proof of correctness are rather technical; as a final step in our proof of correctness, we apply the central limit theorem for "generalized multinomial distributions" of Chapter 4 to characterize the distribution of summary statistics defined by the process of drawing a sample of size k from a fixed distribution.

Chapter 6–The Power of Linear Estimators Most proposed estimators for entropy have the following form: given a sample, the estimators compute the vector $\mathcal{F} = \mathcal{F}_1, \mathcal{F}_2, \ldots,$ where $\mathcal{F}_i$ is the number of elements that occur exactly i times in the sample, and then output the dot product of $\mathcal{F}$ with some fixed vector of coefficients. We term such estimators *linear*. The estimators of Chapter 3 do not take this form, instead of computing a dot product, they solve a linear program. In this chapter, we consider the question of whether the additional computational power of linear programming is necessary to effectively estimate these properties using small samples. We show that for a broad class of symmetric properties, including entropy, there exist near-optimal linear estimators. Specifically, for the properties

in question, we show if there is *any* algorithm that is capable of obtaining an ϵ-accurate estimate with probability at least 0.51 when given a sample of size k (drawn independently) from a distribution of support size at most n, then there exists a linear estimator that takes a sample of size $1.01 \cdot k$ and estimates the property to within accuracy 2ϵ, and succeeds with probability 0.99. Our proof is constructive, and exposes a duality between the search for good lower bound instances and the search for good linear estimators. As our proof is via duality, unsurprisingly, it does not yield any bounds on the sample complexity of these estimation tasks, and hence the results of this chapter complement rather than subsume those of Chapters 3 and 5.

Chapter 7–Explicit Linear Estimators In this chapter we describe machinery for constructing and analyzing the performance of explicit linear estimators. We describe a linear estimator for entropy, which has an inverse linear convergence rate: it estimates the entropy of a distribution of support size at most n to within error ϵ using a sample of size $O(\frac{n}{\epsilon \log n})$, provided $\epsilon > 1/n^{\alpha}$ for some small constant α. This inverse linear rate of convergence is rather surprising, and matches the lower bounds of Chapter 5 both in terms of the dependence on the support size n, and the dependence on ϵ. We also construct an estimator for "distance to uniformity", which estimates the total variational distance (ℓ_1 distance) to the closest uniform distribution of support m, for some specified value m, and show that for any constant error ϵ, given $O(\frac{m}{\log m})$-sized samples from a distribution of any support size, the distance to the uniform distribution of support m can be estimated to error ϵ, with high probability over the randomness of the sample.

Chapter 8–Estimating Properties in Practice In this chapter we describe a practical adaptation of the estimators of Chapter 3, and experimentally evaluate their performance on both computer generated and real datasets. For entropy estimation, we compare our estimator to five estimators from the literature, including both standard, and more recently proposed estimators; in all settings considered, our estimator performs at least as well as the best previously proposed estimator that we consider, and significantly outperforms each of these estimators in some settings.

Part II: Correlations, Parities, and Juntas

Chapter 9–Finding Correlations, and the Closest Pair Problem We begin by describing an algorithm for finding a pair of correlated variables from among n independent Boolean random variables. If the correlated pair of variables has Pearson correlation ρ, and the dimension of the vectors (i.e. the sample size) is sufficiently large, our algorithm runs in time $n^{\frac{5-\omega}{4-\omega}} poly(1/\rho) < n^{1.62} poly(1/\rho)$, where $\omega < 2.38$ is the exponent of fast matrix multiplication. Previously, no subquadratic time algorithm with a polynomial dependence on ρ had been described for this problem. We then extend this result to give a subquadratic time algorithm with a slightly worse exponent for a more general setting, which can be interpreted

as efficiently computing the product of an $n \times d$ and $d \times n$ matrix given the promise that the result has a small number of large entries. Finally, we further extend this approach to the general $(1+\epsilon)$ *approximate closest pair problem*, to yield an algorithm that, given n arbitrary vectors, finds a pair of vectors whose distance is at most a factor of $1 + \epsilon$ larger than that of the closest pair of vectors; the runtime of this algorithm is $O(n^{2-\Theta(\sqrt{\epsilon})})$, improving on the $O(n^{2-O(\epsilon)})$ runtime given by locality sensitive hashing approaches. This second algorithm relies on fast *rectangular* matrix multiplication.

Chapter 10–Learning Parities and Juntas We explain the connection between the problem of finding correlated variables and the problem of learning parity with noise, and sketch how the first result of Chapter 9 could be used to obtain an algorithm for learning sparse parities—parities of at most k bits from examples of size $n >> k$—with noise rate $\eta < 1/2$ in time $n^{\frac{5-\omega}{2(4-\omega)}k} poly(\frac{1}{\frac{1}{2}-\eta}) \approx n^{0.81k} poly(\frac{1}{\frac{1}{2}-\eta})$. We then describe an alternative approach for this problem, which simulates perturbing the distribution of examples slightly, and yields the exponent $\frac{\omega}{3}k < 0.80k$. These are the first algorithms for this problem with a polynomial dependence on the noise with runtime $O(n^{ck})$ for any $c < 1$. The polynomial dependence on the noise rate allows this result to be leveraged to obtain new results for the problems of learning k-juntas—the problem of identifying the small set of relevant variables from among many possible variables—both in the presence, and absence of noise. For learning k-juntas without noise over random length n instances, we obtain a runtime of $O(n^{0.60k})$, improving upon the $O(n^{0.70})$ result of Mossel et al. [92].

Part III: Learning Mixtures of Gaussians

Chapter 11–Learning Univariate Mixtures of Gaussians In this chapter we consider the problem of recovering the parameters of a one-dimensional Gaussian mixture model (GMM). For any constant k, we show that a $poly(1/\epsilon)$ runtime and sample size suffice to recover ϵ-accurate estimates of the means, variances, and mixing weights of each component of a GMM with k Gaussian components. We prove this by establishing what we term the *polynomially robust identifiability* of GMMs: for any two GMMs whose components differ significantly, there will be a discrepancy in one of the low order moments of the mixtures whose magnitude is polynomially related to the discrepancy in the components. The dependence of the runtime and sample size on the number of Gaussian components, k, is severe, though we also give a lower bound construction proving that at least an exponential dependence on k is information theoretically necessary.

Chapter 12–Learning Mixtures of Gaussians in High Dimension We consider the problem of learning high dimensional GMMs; we show that the runtime and sample size required to obtain accurate estimates of the mixture parameters is polynomial in both the inverse of the desired accuracy, and the dimension, d. Specifically, given a sample (drawn independently) from a GMM $F = \sum_{i=1}^{k} w_i F_i$, where each F_i is a d-dimensional Gaussian

distribution, for any $\epsilon, \delta > 0$, with probability at least $1 - \delta$ our algorithm returns a mixture $\hat{F} = \sum_{i=1}^{\hat{k}} \hat{w}_i \hat{F}_i$ such that the total variational distance (ℓ_1 distance) between F and $\hat{F}$ is at most ϵ; additionally, if for all i, $w_i > \epsilon$, and for all i, j the total variational distance between F_i and F_j is at least ϵ, then $\hat{k} = k$ and there exists some permutation π of the integers $[k] = \{1, \ldots, k\}$ such that for all i, $|w_i - \hat{w_{\pi(i)}}| < \epsilon$, and the total variational distance between F_i and $\hat{F_{\pi(i)}}$ is at most ϵ. The algorithm requires runtime and sample size $poly(\epsilon, d, \log \frac{1}{\delta})$.

1.5 Bibliographics

The results presented in the first part of this dissertation, on estimating properties, are joint work with Paul Valiant. The results of Chapter 3, and those of Chapters 4 and 5 were posted as two separate papers on the Electronic Colloquium on Computational Complexity [127, 126], and appeared together in STOC'11 [128]. A preliminary version of the results of Chapters 6 and 7 appeared in FOCS'11 [129].

The results contained in the second part of the dissertation, on finding correlations and relevant variables, will appear in FOCS'12 [124], with an earlier version containing a portion of the results posted on the Electronic Colloquium on Computational Complexity [125].

The results of the third section of the dissertation, on learning GMMs, was presented in two papers, one with Adam Kalai and Ankur Moitra that appeared in STOC'10 [74], and the second that was joint with Ankur Moitra that appeared in FOCS'10 [91]. An overview of this work also appeared in the Communications of the ACM [73].

Part I

Estimating Symmetric Properties

Chapter 2

Definitions, and Related Work

2.1 Definitions and Examples

The estimation tasks that we consider have been studied by the statistics, information theory, and computer science communities, and the terminology varies between these communities. Before discussing the historical background and related work, it will be helpful to establish a consistent terminology. The definitions we give below will be used in Chapters 3 through 8.

Definition 2.1. *A* distribution *on* $[n] = \{1, \ldots, n\}$ *is a function* $p : [n] \to [0, 1]$ *satisfying* $\sum_i p(i) = 1$. *Let* $\mathcal{D}^n$ *denote the set of distributions over domain* $[n]$.

We will be dealing exclusively with *symmetric* properties of distributions with discrete support. Informally, symmetric properties are those that are invariant to renaming the domain elements.

Definition 2.2. *A* property *of a distribution is a function* $\pi : \mathcal{D}^n \to \mathbb{R}$. *Additionally, a property is* symmetric *if, for all distributions* $p \in \mathcal{D}^n$, *and all permutations* $\sigma : [n] \to [n]$, $\pi(p) = \pi(p \circ \sigma)$, *where* $p \circ \sigma$ *denotes the distribution obtained by permuting the labels of* p *according to the permutation* σ.

Since symmetric properties cannot depend on the labels of the domain, it will prove convenient to have a "symmetrized" representation of a distribution. We thus define the *histogram of a distribution*:

Definition 2.3. *The* histogram *of a distribution* $p \in \mathcal{D}^n$ *is a mapping* $h_p : (0, 1] \to \mathbb{N} \cup \{0\}$, *where* $h_p(x) = |\{i : p(i) = x\}|$. *When the distribution in question is clear, we drop the subscript, and simply refer to the histogram* h.

To see the motivation for calling such a representation a "histogram" of a distribution, consider representing a distribution p by the unordered list of probabilities with which the domain elements arise: the *histogram* h_p is simply the histogram, in the traditional sense, of that list.

Any symmetric property is a function of only the histogram of the distribution. For example:

- The *entropy* $H(p)$ of a distribution $p \in \mathcal{D}^n$ is defined to be
$$H(p) := -\sum_{i:p(i)\neq 0} p(i)\log p(i) = -\sum_{x:h_p(x)\neq 0} h_p(x)x\log x.$$
- The *support size* is the number of domain elements that occur with positive probability:
$$|\{i : p(i) \neq 0\}| = \sum_{x:h_p(x)\neq 0} h_p(x).$$

Additionally, the total probability mass in the distribution at probability x—namely the probability of drawing a domain element whose probability is x—is $x \cdot h(x)$ and thus $\sum_{x:h(x)\neq 0} x \cdot h(x) = 1$, as distributions have total probability mass 1.

Throughout, we will use n to denote the size of the domain of the distribution (provided the distribution in question has finite support size), and k to denote the size of the sample to which we have access. We assume throughout that each sample consists of independent draws from a fixed distribution.

In analogy with the histogram of a distribution, we define the *fingerprint* of a sample:

Definition 2.4. *Given a sample $S = (s_1, \ldots, s_k)$, the associated* fingerprint, *$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$, is the "histogram of the frequency counts" of the sample. Formally, $\mathcal{F}$ is the vector whose i^{th} component, $\mathcal{F}_i$, is the number of elements that occur exactly i times in S. If the sample in question is ambiguous, we use a superscript, $\mathcal{F}^S$ to denote the fingerprint corresponding to S.*

For estimating entropy, or any other symmetric property, the fingerprint of a sample contains all the relevant information (see [20], for a formal proof of this fact). Throughout, we will be representing distributions by their histograms, and samples by their fingerprints. We note that in some of the literature, the fingerprint is alternately termed the *pattern*, *histogram of the histogram*, *collision statistics*, or *partition* of the sample.

We provide an example illustrating the above definitions:

Example 2.5. *Consider a sequence of fish species, drawn from a certain lake,*

$$S = (trout, salmon, trout, cod, cod, whale, trout, eel, salmon).$$

We have $\mathcal{F} = (2, 2, 1)$, indicating that two species occurred exactly once (whale and eel), two species occurred exactly twice (salmon and cod), and one species occurred exactly three times (trout).

Suppose that the true distribution of fish is the following:

$$Pr(trout) = 1/2, \quad Pr(salmon) = 1/4,$$
$$Pr(cod) = Pr(whale) = Pr(eel) = Pr(shark) = 1/16.$$

The associated histogram *of this distribution is* $h : \mathbb{R}^+ \to \mathbb{Z}$ *defined by* $h(1/16) = 4$, $h(1/4) = 1$, $h(1/2) = 1$, *and for all* $x \notin \{1/16, 1/4, 1/2\}$, $h(x) = 0$.

Poisson Samples

Before proceeding, it will be helpful to have an intuitive understanding of the distribution of the fingerprint corresponding to a sample of size k drawn from a distribution with histogram h. This distribution intimately involves the Poisson distribution. Throughout, we use $Poi(\lambda)$ to denote the Poisson distribution with expectation λ, and for a nonnegative integer j,

$$poi(\lambda, j) := \frac{\lambda^j e^{-\lambda}}{j!}$$

denotes the probability that a random variable distributed according to $Poi(\lambda)$ takes value j. Additionally, for integers $i \geq 0$, we refer to the function $poi(x, i)$, viewed as a function of the variable x, as the ith *Poisson function*.

Given a fingerprint corresponding to a sample of size k drawn from a distribution p, the number of occurrences of any two elements are not independent; however, if instead of taking k samples, we chose $k' \leftarrow Poi(k)$ according to a Poisson distribution with expectation k and then take a sample of size k' from p, the number of occurrences of each domain element $i \in [n]$ will be independent random variables with distributions $Poi\left(k \cdot p(i)\right)$. This independence is quite helpful when arguing about the structure of the distribution of such fingerprints.

We provide a clarifying example:

Example 2.6. *Consider the uniform distribution on* $[n]$, *which has histogram* h *such that* $h(\frac{1}{n}) = n$, *and* $h(x) = 0$ *for* $x \neq \frac{1}{n}$. *Let* $k' \leftarrow Poi(5n)$ *be a Poisson-distributed random number, and let* X *be the result of drawing a sample of size* k' *from the distribution. The number of occurrences of each element of* $[n]$ *will be independent, distributed according to* $Poi(5)$. *Note that* $\mathcal{F}_i$ *and* $\mathcal{F}_j$ *are* not *independent (since, for example, if* $\mathcal{F}_i = n$ *then it must be the case that* $\mathcal{F}_j = 0$, *for* $i \neq j$). *A fingerprint of a typical trial will look roughly like* $\mathcal{F}_i \approx n \cdot poi(5, i)$.

Since $k' \leftarrow Poi(k)$ is closely concentrated around k (see, for example, the standard tail bounds for Poisson distributions given in Appendix A.2), as one might expect, there is little difference between considering samples of size exactly k, and $Poi(k)$-sized samples. Thus we will be able to prove statements about k-sample fingerprints by considering the structurally more simple $Poi(k)$-sample fingerprints.

We conclude this section by considering the distribution of the ith entry of a $Poi(k)$-sample fingerprint, $\mathcal{F}_i$. Since the number of occurrences of different domain elements are independent, $\mathcal{F}_i$ is distributed as the sum of n independent $0, 1$ random variables $Y_1, \ldots, Y_n$, where $\Pr[Y_j = 1] = poi(k \cdot p(j), i)$ is the probability that the jth domain element occurs exactly i times in a sample of size $k' \leftarrow Poi(k)$. By linearity of expectation,

$$E[\mathcal{F}_i] = \sum_{j \in [n]} poi(k \cdot p(j), i) = \sum_{x : h(x) \neq 0} h(x) \cdot poi(kx, i), \tag{2.1}$$

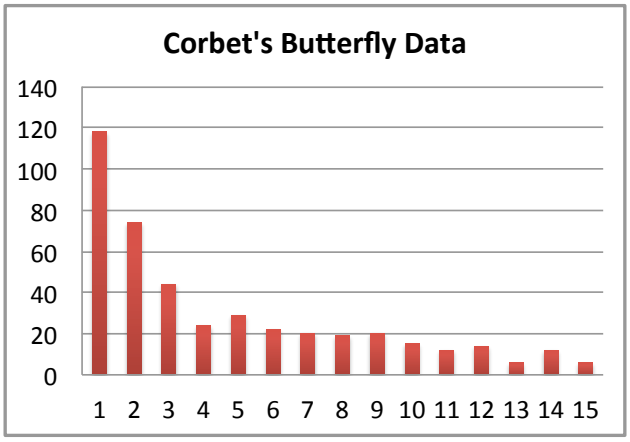

Figure 2.1: A plot of Corbet's butterfly data, depicting the number of butterfly species for which $1, 2, 3, \ldots$ specimens were obtained during a 2 year expedition in Malaysia. Good and Toulmin showed that the alternating sum of these statistics—in this case $118 - 74 + 44 - 24 + \ldots \approx 75$—yields an unbiased estimator for the number of *new* species that would be discovered over another 2 year period. [55, 60]

and from the independence, we will have Chernoff–style concentration about this expectation.

2.2 Historical Background: The Work of Turing, Fisher, and Good

The problem of inferring properties of an unknown discrete distribution from "too small" samples has a very rich history of study in statistics and computer science, with early contributions from both R.A Fisher, and Alan Turing. In the early 1940's, R. A. Fisher was approached by a naturalist, Corbet, who had just returned from two years of collecting butterflies in the Malay peninsula. Corbet presented Fisher with data on his butterfly collections—specifically, he indicated the number of species for which he had only seen a single specimen (118 species), the number of species for which he had two specimens (74 species), three specimens (44 species), and so on (see Figure 2.1). Corbet hoped that from this data, Fisher would be able to deduce some properties of the true distribution of butterflies in Malaysia, and in particular, he wanted an estimate of the number of new species he might discover if he were to return to the Malay jungle for another 2 years.

Fisher approached this problem from a parametric Bayesian standpoint, and assumed that the distribution of butterfly species could be accurately approximated by a gamma distribution, and fit the parameters of the gamma distribution to the butterfly data [55]. A decade later, I.J. Good and Toulmin [60] returned to this problem, and offered a nonpara-

metric alternate analysis that makes no assumptions on the form of the true distribution of butterfly species, only relying on the assumptions that the number of butterflies of a given species, s, that are caught during a specified time period is given by $Poi(\lambda_s)$, for some unknown λ_s dependent on the species. Given this assumption, the probability that one discovers species s during the second time period is given by the following expression:

$$Pr(s \text{ not seen in 1st period}) \cdot Pr(s \text{ seen in 2nd period}) = poi(\lambda_s, 0)\left(1 - poi(\lambda_s, 0)\right).$$

Thus the expected number of species discovered during the second time period is obtained by simply summing this expression over all species: $\sum_s poi(\lambda_s, 0)\left(1 - poi(\lambda_s, 0)\right)$. Good and Toulmin then argued that one could expand the second term as a power series, and re-express this sum in terms of the fingerprint expectations:

$$\begin{aligned}
\sum_s poi(\lambda_s, 0)\left(1 - poi(\lambda_s, 0)\right) &= \sum_s e^{-\lambda_s}\left(1 - e^{-\lambda_s}\right) \\
&= \sum_s e^{-\lambda_s} \sum_{i=1}^{\infty} \frac{(-1)^{i+1}\lambda_s^i}{i!} \\
&= \sum_{i \geq 1} (-1)^{i+1} \sum_s poi(\lambda_s, i) = \sum_{i \geq 1} (-1)^{i+1} E[\mathcal{F}_i].
\end{aligned}$$

Thus the number of new species we expect to discover in the second time period can be expressed as the alternating sum of the expected fingerprint entries: the number of species one expects to see once, twice, etc, in any given time period. The crucial observation, as was argued after Equation 2.1, is that the fingerprint values will be closely concentrated about their expectations, and thus the data furnished during the first visit provides accurate (and unbiased) estimates of these fingerprint expectations. In the case of Corbet's butterfly data, the alternating sum $\mathcal{F}_1 - \mathcal{F}_2 + \mathcal{F}_3 - \ldots = 118 - 74 + 44 - 24 + \ldots \approx 75$ yields the conclusion that roughly 75 new species would be discovered in a second 2-year expedition to the jungle. (To the best of my knowledge, Corbet did not conduct another expedition.)

At roughly the same time as Fisher's work on this problem, at the height of WWII, Alan Turing and I.J. Good were working on a similar problem in the rather different context of the pivotal British war–effort to analyze the statistics of the German Enigma Machine ciphers. Turing and Good were interested in estimating the total probability mass accounted for by the "unseen" portion of the distribution: what is the probability that the next Enigma codeword is a previously unseen codeword? This question can be tackled via the same approach as Corbet's butterfly question: namely, representing the desired quantity in terms of the fingerprint expectations. Making the assumption that there is some distribution over ciphertexts, with each ciphertext s occurring with probability p_s, and each observation yielding an independent draw from this distribution, the number of times ciphertext s is seen in a sample of size k will be roughly distributed according to $Poi(k \cdot p_s)$. Thus the expected unseen probability mass in the distribution given a sample of size k is simply $\sum_s p_s \cdot poi(k \cdot p_s, 0)$, since each domain element will be unseen with probability $poi(k \cdot p_s, 0)$

and accounts for p_s units of probability mass. We can reexpress this sum in terms of the expected fingerprints as follows:

$$\sum_s p_s \cdot poi(k \cdot p_s, 0) = \frac{1}{k} \sum_s poi(k \cdot p_s, 1) = \frac{E[\mathcal{F}_1]}{k}.$$

Thus the ratio $\mathcal{F}_1/k$ provides an estimate of the unseen probability mass.

This result is a component of what is now known as the *Good-Turing frequency estimation scheme*, and was published after the war in 1953 by Good [59]. In addition to the many practical applications of the Good-Turing estimates, there has been considerable recent work from the computer science community analyzing and extending variants of this estimation scheme [86, 96, 97, 139, 138].

More broadly, this early work of Fisher, Turing, Good and Toulmin demonstrated that one could very easily and accurately estimate some specific parameters of the "unseen" portion of a distribution. The challenge of constructing sublinear-sample estimators for entropy and other symmetric properties is that such estimators must take into account the contribution towards the property value of the unseen portion of the distribution; this early work, in some sense, hinted that sublinear-sample estimators might exist for the properties we consider in this dissertation.

2.3 Estimating Properties and Estimating Distributions

In contrast to the parameters that Fisher and Turing were concerned with—the number of new domain elements that would be seen in a second sample, or the probability that the next draw is a previously–unseen domain element—for which they devised simple unbiased estimators with small variance—for entropy and the other properties we study, there are *no* unbiased estimators. An arbitrarily small amount of probability mass in a given distribution can contribute an arbitrarily large amount of entropy, or support size, and thus for any fixed estimator, one can construct a distribution for which the estimator has an arbitrarily large bias. This unfortunate state of affairs complicates the problem of designing good estimators, and opened the door for an enormous body of work, spanning the past 75 years, describing and analyzing heuristic estimators for these properties that seem to perform well on real-world data. Below we describe the most commonly used estimators for entropy, and two more recently proposed estimators.

There is also a large literature on the "unseen species" problem and the closely related "distinct elements" problems, including the efforts of Efron and Thisted to estimate the total number of words that Shakespeare knew (though may not have used in his extant works) [52]. That work employs a heuristic linear programming approach that is related to our approach, though is designed for the single purpose of estimating the number of words Shakespeare knew, and is based on heuristics, some of which are specifically tuned to that

data. Much of the later work on the "unseen elements" problem is also based heavily on heuristic arguments or strong assumptions on the true distribution from which the sample is drawn, and thus lies beyond the scope of our work; we refer the reader to [33] and to [32] for several hundred references on this problem.

Practical Estimators for Entropy

Because of the practical importance of estimating entropy, both as a natural measure of the "biodiversity" of the distribution, and as the central quantity that must be estimated to compute the *mutual information* between two signals, there has been an especially long line of work focussing on this property. Below, we discuss five estimators for entropy from the literature. We begin with three classical estimators, which are, perhaps the most commonly used estimators for entropy [102].

The "naive" estimator: The entropy of the empirical distribution, namely, given a fingerprint $\mathcal{F}$ derived from a sample of size k, $H^{naive}(\mathcal{F}) := \sum_i \mathcal{F}_i \cdot \frac{i}{k} |\log \frac{i}{k}|$.

The Miller-Madow corrected estimator [90]: The naive estimator H^{naive} corrected to try to account for the second derivative of the logarithm function, namely $H^{MM}(\mathcal{F}) := H^{naive}(\mathcal{F}) + \frac{(\sum_i \mathcal{F}_i)-1}{2k}$, though we note that the numerator of the correction term is sometimes replaced by various other quantities, see [103].

The jackknifed naive estimator [144, 51]:

$$H^{JK}(\mathcal{F}) := k \cdot H^{naive}(\mathcal{F}) - \frac{k-1}{k} \sum_{j=1}^{k} H^{naive}(\mathcal{F}^{-j}),$$

where $\mathcal{F}^{-j}$ is the fingerprint given by removing the contribution of the jth element of the sample.

These estimators and their many variants generally perform very well *provided that all of the elements of the support occur with large probability*. The problem with these estimators can be summarized as their inability to appropriately deal with a sample from a distribution for which a significant portion of the probability mass of the distribution *is not* represented in the sample. For example, given a sample of size $o(n)$ drawn from the uniform distribution on support n, these estimators generally fail. In particular, these estimators make no attempt to understand the (potentially significant) contribution towards the entropy of the true distribution that comes from the "unseen" portion of the distribution.

The following estimator, proposed in 2003 by Chao and Shen [36], is specifically designed to apply to settings in which there is a significant component of the distribution that is unseen. It is heuristic, though there is some evidence that it performs well in some practical settings [137].

The coverage adjusted estimator (CAE) [36]: We briefly motivate the definition of H^{CAE} (see [137] for a more detailed discussion). Consider a domain element α that occurs

i times in a sample of size k. The naive estimator associates a probability $p_\alpha = i/k$, and a contribution towards the entropy of $-p_\alpha \log p_\alpha$. The Good-Turing frequency estimation scheme [59, 113] provides an (unbiased) estimate of the total "unseen" probability mass in the distribution, P_{unseen}, and thus it seems intuitive to adjust p_α to take this unseen mass into account, namely to use $\hat{p_\alpha} := p_\alpha(1 - P_{unseen})$. Finally, if the true probability of α were $\hat{p_\alpha}$, the probability that we actually observe α in our sample of size k is only $1-(1-\hat{p_\alpha})^k$, and thus we might be inclined to believe that for each domain element α that we observe, there are $\frac{1}{1-(1-\hat{p_\alpha})^k}$ similar domain elements that we do not observe, and thus we should multiply the contribution of α by this amount (this approach is recognized as the Horvitz–Thompson estimator for population totals [68]). This yields the following: Given a fingerprint $\mathcal{F}$ derived from a sample of size k, let $P_s := 1 - \mathcal{F}_1/k$, represent the Good–Turing estimate of the probability mass of the "seen" portion of the distribution.

$$H^{CAE}(\mathcal{F}) := -\sum_i \mathcal{F}_i \frac{(i/k)P_s \log\left((i/k)P_s\right)}{1-\left(1-(i/k)P_s\right)^k}.$$

One weakness of the CAE which arises from its attempt to account for the probability mass of the unseen portion of the distribution is that in some simple settings, the CAE overcompensates, leading to disappointing performance. For example, given a sample of size k from a uniform distribution over k elements, it is not hard to show that the bias of the CAE is $O(\log k)$. This error is not even bounded; for comparison, even the naive estimator has error bounded by a constant in the limit as $k \to \infty$ in this setting.

In a different direction, Paninski gave a simple though non–constructive proof of the existence of a sublinear sample estimator for additively approximating the entropy to within a constant; the proof is via a direct application of the Stone–Weierstrass theorem to the set of Poisson functions [102, 101]. Namely, this work gave a nonconstructive proof of the existence of a set of coefficients a_i, such that the associated linear estimator $\sum_i a_i \mathcal{F}_i$, performs well. The "Best Upper Bound" [BUB] estimator of Paninski, briefly discussed below, is a practical variant of this approach:

The *Best Upper Bound* estimator [102]: This estimator is obtained by numerically searching for a minimax linear estimator, with respect to a certain error metric. Roughly, this estimator seeks to minimize the bias of the estimator while keeping a heuristic sense of the variance in check. In order to compute such an estimator, the BUB requires, as input, an upper bound on the support size of the distribution from which the sample is drawn.

Estimating Distributions

Given a property of interest, there are two general approaches to designing an estimator: the property-specific "direct" approach, and the "canonical" approach. All the estimators discussed thus far, with the arguable exception of the coverage adjusted estimator for entropy, take the direct approach, and return an estimate of the property in question without revealing

any additional insights into the structure of the true distribution beyond the returned value of the specific property. In the canonical approach, rather than directly estimating the property in question, one first attempts to recover a distribution that is close, in some metric, to the true distribution from which the sample was drawn. Given such a distribution, one can then simply return the property value of that hypothetical distribution. While one cannot know the labels of the unseen elements of a distribution, one could still hope to reconstruct an approximation of the histogram of the true distribution—capturing, for each interval of probability, the approximate number of unseen domain elements whose true probability lies within that interval. For symmetric properties, such a histogram would contain all the relevant information to estimate the property.

Orlitsky et al. have been pursuing one approach to reconstructing such a histogram from a sample [98, 1, 2]. They posed the following question:[1]

> *Given a sample of size k with fingerprint $\mathcal{F}$, what distribution maximizes the likelihood of yielding fingerprint $\mathcal{F}$ from a sample of size k?*

Such a distribution need not be unique, though the question is well-defined. To illustrate the distinction between the distribution that maximizes the likelihood of the sample, and the distribution that maximizes the likelihood of the fingerprint, we give two basic examples:

Example 2.7. *Consider the following sample of size $k = 3$: $X = a, b, a$, with fingerprint $\mathcal{F} = (1, 1, 0, \ldots)$. The maximum likelihood distribution of the sample, p_{ML}, assigns probability $2/3$ to a and probability $1/3$ to b. The probability that a sample consisting of 3 (independent) draws from p_{ML} has fingerprint $\mathcal{F}$ is simply the probability that one sees either a or b exactly once, and thus is given by $1 - p_{ML}(a)^3 - p_{ML}(b)^3 = 2/3$. It is not hard to show that the distribution that maximizes the likelihood of the fingerprint $(1, 1, 0, \ldots)$, is the uniform distribution over support 2, assigning probability $1/2$ to each of two elements. The probability that 3 draws from such a distribution yield fingerprint $\mathcal{F}$ is $3/4$, which is optimal.*

Example 2.8. *Consider a sample of size k with fingerprint $\mathcal{F} = (k, 0, \ldots)$, that is, each element is seen only once. The distribution maximizing the likelihood of that fingerprint is the continuum, with infinite support, and a sample of size k from the continuum will have k distinct elements with probability 1.*

The initial work on this question of Orlitsky focussed on this combinatorially rich likelihood landscape, rather than on property estimation. While it seems unclear how to prove that such a likelihood maximizing distribution would, necessarily, have similar property values to the true distribution, at least intuitively one might hope that this is true: since symmetric properties are invariant to relabeling the support the distribution, it seems natural to hope that the distribution maximizing the likelihood of the fingerprint of the sample might yield a better estimate of the property in questions than, say, the distribution maximizing the likelihood of the sample (i.e. the empirical distribution of the sample). Recently,

[1] What we term the *fingerprint* of a sample, Orlitsky *et al.* term the *pattern*

Acharya *et al.* showed that this maximum likelihood approach could be used to yield a near-optimal algorithm for deciding whether two samples were drawn from *identical* (or very similar) distributions, versus distribution that have large distance [2]

From a computational standpoint, Orlitsky *et al.* showed that such fingerprint likelihood maximizing distributions can be found in some specific simple or small settings [1]. The problem of finding or approximating such distributions for typical fingerprints derived from large samples, however, seem daunting.

The results, and approach we take in Chapter 3 were directly motivated by this question of Orlitsky. Rather than attempting to find the distribution maximizing the likelihood of the fingerprint, we eschewed the computational and analytical difficulties of dealing with this complicated combinatorial likelihood landscape, and instead analyzed the set of distributions whose *expected* fingerprints are close to the given fingerprint $\mathcal{F}$.

2.4 Property Testing, and the Computer Science Perspective

The interest in property estimation from the computer science community has its roots in work on property *testing*. The property-testing framework was originally developed to provide a rigorous framework for studying the minimum amount of information necessary to determine, with high probability, whether an object in question possesses a certain property. This framework was applied in many settings, perhaps most successfully to testing algebraic properties [29, 115], and testing graph properties [7, 58].

As adapted to the statistical property estimation setting, the framework asks the following question:

> *Given a property of a distribution or set of distributions, a desired accuracy ϵ, a probability of failure δ, and access to a sample or samples consisting of independent draws, how large must the samples be in order to guarantee that the estimate of the property differs from the true value by at most ϵ, with probability at least $1 - \delta$?*

This framework offers a natural alternative to the standard metrics used by the statistics community for evaluating property estimators. The literature on property estimation from the statistics community almost exclusively analyzes the asymptotic *rate of convergence* of the estimator in question (given a fixed distribution, how does the expected deviation between the true property value and estimate decrease as the sample size goes to infinity). The obvious shortcoming of such asymptotic analysis is that it says little about the many settings in which one does not have sufficiently large samples for the asymptotic behavior to be relevant. In particular, such analysis does not inform our understanding of the performance of estimators in the regime in which large portions of the distribution are unseen in the sample.

In contrast to the classical asymptotic analysis, the property testing framework allows one to quantitatively formulate the question of the existence of estimators achieving certain performance guarantees given a fixed (finite) sample size. Beginning with pioneering work of Rubinfeld, who was one of the first to adapt the property testing framework to the statistical property setting, the theoretical computer science community began tackling these questions in the early 2000's. This work had an eye both towards describing estimators, and proving information theoretic lower bounds on the performance of any estimator.

One of the first problems considered was the problem of *identity testing* (also referred to as *closeness testing*): Given access to samples from two distribution, A and B, how large must the samples be in order to accurately distinguish the case that A and B are identical or nearly identical, from the case that A and B are far apart (have total variational distance $D_{tv}(A, B) > \epsilon$)? As for the problems we consider, the difficulty of this decision task is parameterized by n, an upper bound on the support size of the distributions. In 2000, Batu et al. showed that this task can be accomplished using $\tilde{O}(n^{2/3})$-sized sample, for constant $\epsilon > 0$ [20]. This result matches the lower bound, of P. Valiant [131, 132], to logarithmic factors. In contrast, determining (with high probability) whether a sample was drawn from some explicitly described distribution of support $[n]$, versus a distribution that has constant ℓ_1 distance from the described distribution requires a sample of size $\tilde{\Theta}(n^{1/2})$ [19].

For the problem of estimating entropy. Batu *et al.* [18, 21, 22], Guha *et al.* [63], and Valiant [131, 132] considered the problem of multiplicatively estimating the entropy of a distribution; in all these works, the estimation algorithm has the following basic form: given a sample, discard the species that occur infrequently and return the entropy of the empirical distribution of the frequently–occurring elements, adjusted by some function of the amount of missing probability mass. In particular, no attempt is made to understand the portion of the true distribution consisting of infrequently occurring elements—the unseen, or little–seen, portion. To achieve constant additive error, these estimators all require $O(n)$-sized samples.

For the problem of estimating the support size of a distribution in which all elements occur with probability at least $1/n$, or the distinct elements problem, tight *multiplicative* bounds of $\Theta(n/\alpha^2)$ on the sample size required to approximate the number of distinct elements to a multiplicative factor of α (given a total number of n buckets) are given in [15, 37] though they are somewhat unsatisfying as the worst-case instance is distinguishing the case that there is *one* distinct element, from the case that there are α^2. For additively estimating the number of distinct elements to within $\pm(\frac{1}{2} - \epsilon)n$, to the best of our knowledge there were no improvements upon the trivial $\Omega(n)$ algorithm for any constant $\epsilon > 0$.

Lower Bounds

Prior to the results of this dissertation, the best lower bound for additively estimating the support size or entropy of a distribution was due to P. Valiant, who showed that for any constant $\epsilon > 0$, any estimator that obtains additive error at most $(1/2 - \epsilon)n$ for the number of distinct elements (or constant error in entropy estimation) with probability of success at

least .51 requires a sample of size at least $n/2^{\Theta(\sqrt{\log n})}$ [131, 132]. This bound improved upon the slightly weaker bound of $n/2^{\Theta(\sqrt{\log n}\cdot\log\log n)}$ of Rashodnikova *et al.* [107].

The difficulty in constructing lower bounds on the sample size required to estimate symmetric properties, is that one must argue that there are two distribution with quite different property values, but for which the distribution of fingerprints derived from a samples of size k are *indistinguishable*, with high probability. This condition of indistinguishability is very stringent. The distribution over fingerprints defined by the process of drawing a sample of size k from a fixed distribution, is a high dimensional distribution supported on the integer lattice. The condition of indistinguishability means that one must show that two such distributions are close in the ℓ_1 metric; such characterizations of high-dimensional distributions are rare.

The lower bounds of [131, 132] characterized this distribution over fingerprints via a theorem of Roos [114] that essentially characterized these distributions in terms of the vector of fingerprint expectations. To obtain the tighter lower bounds of this dissertation, we instead characterize these distributions in terms of both the expectations as well as the covariance matrix, via a new central limit theorem that we prove in Chapter 4. While we go on to show that the second moments of the distribution of fingerprints can be inferred from the expectations, and thus considering only the expectations suffices, the improvement of our lower bounds over the previous lower bounds, in some sense, stems from the fact that our central limit theorem uses both the first and second moments of the distribution, as opposed to only the first moments, as in the theorem of Roos.

The Streaming Setting

The problems of estimating the support size (and the general problem of estimating frequency moments) and estimating the entropy, have also been considered by the computer science community in the setting of *streaming*. In the streaming setting, one has access to the entire distribution (as represented by a very large database), though one has very little memory, and can only perform a single (or several) passes over the database [5, 16, 25, 35, 66, 72, 75, 142].

One might hope that the streaming setting could be tackled by filling the limited memory with a small (random) sample of the database, then applying the estimation techniques of this dissertation, for example. Unsurprisingly, this approach is very far from optimal—if one has access to the entire distribution, even if one has limited memory and can only perform a single pass, one can obtain far better estimates than if one only considers a small sample.

Chapter 3

Estimating the Unseen: Sublinear Sample Estimators for Entropy, Support Size, and Other Properties

We introduce a new approach to characterizing the unobserved portion of a distribution, which provides sublinear–sample additive estimators for a class of properties that includes entropy and distribution support size. Additionally, we demonstrate that this approach can be extended to estimate properties of pairs of distributions, such as estimating the total variational distance (ℓ_1 distance) between a pair of distributions, based on samples drawn from each distribution. Together with the lower bounds presented in Chapter 5, this settles the longstanding open questions of the sample complexities of these estimation problems (up to constant factors). Our algorithms estimate these properties up to an arbitrarily small additive constant, using a sample of size $O(n/\log n)$, and can be made to run in in time *linear* in the sample size. Our lower bounds show that no algorithm that uses a sample of size $o(n/\log n)$ can achieve this. In the case of estimating entropy and estimating total variational distance between pairs of distributions, n is a bound on the support size of the distributions, and is a natural parameterization of the difficulty of these task. For the problem of estimating the distribution support size, as is standard, we assume that all elements in the support occur with probability at least $1/n$, since without such a lower bound it is impossible to estimate support size.[1]

For statistical properties of a single distribution, our algorithm can estimate any property that is symmetric (invariant to relabeling the domain) and sufficiently smooth. Rather than directly trying to estimate a specific property of the distribution, we instead take the *canonical* approach and return to the basic question "*what can we infer about the true distribution?*" given a sublinear sample size? Our algorithm returns a distribution that is, with high probability, "close," in a particular metric, to the true distribution. Specifically, we re-

[1]This setting is a strict generalization of the "distinct elements" problem, which is equivalent to the problem of estimating the support size of a distribution under the condition that each element in the support occurs with probability j/n, for some positive integer j.

turn a distribution $\hat{p}$ with the property that if we had taken our sample from the hypothetical $\hat{p}$ instead of from the unknown true distribution, then with high probability the fingerprint of the sample (the number of domain elements seen once, twice, etc.) will closely match the corresponding parameters of the actual sample. How does one find such a distribution? Via linear programming, the computer scientist's battle-axe—bringing this powerful tool to bear on these problems opens up results that withstood previous approaches to constructing such estimators. The fingerprint of the sample is used to formulate a linear program, whose feasible points correspond to histograms of distributions, and whose objective function at feasible point v captures the degree to which the fingerprint of the sample deviates from the expected fingerprint had the sample been drawn from a distribution corresponding to v. Given the distribution $\hat{p}$ corresponding to a solution to the linear program, to obtain an estimate of some property, we may simply evaluate the property on $\hat{p}$. Unsurprisingly, this yields a very good estimate; surprisingly, one can actually prove this.

Our proof decomposes into two main parts. In the first part we argue that with high probability over the randomness of the sample, the actual distribution from which the sample was drawn, minimally modified, corresponds to a feasible point for the linear program with small objective function value. This part, though slightly tedious, is technically and intuitively straight forward. The second part of the proof argues that *any* pair of feasible points that have small objective function values must correspond to distributions with similar values of entropy, support size, etc. This second part of the proof is the main technical challenge, and it relies on a Chebyshev polynomial construction. It is worth pointing out that there seems to be no intuitive explanation for the $O(n/\log n)$ sample complexity that we achieve (and hence it is unsurprising that such a sample complexity had not been previously conjectured). The $\log n$ factor comes from our Chebyshev polynomial construction; roughly, we are able to use Chebyshev polynomials up to degree $O(\log n)$ (above which the coefficients of the polynomials become too large for our purposes) and the degree $O(\log n)$ polynomials yield an $O(\log n)$ factor tighter analysis than the $O(n)$ sample complexity that could be argued via more basic approaches.

Definitions

In addition to the basic definitions of the *histogram* of a distribution and the *fingerprint* of a sample (Definitions 2.3 and 2.4) given in Chapter 2, we require two additional definitions in order to express our main theorem.

We start by defining what it means for two distributions to be "close"; because the values of symmetric properties depend only upon the histograms of the distributions, we must be careful in defining this distance metric so as to ensure that it will be well–behaved with respect to the properties we are considering. In particular, "close" distributions should have similar values of entropy and support size.

Definition 3.1. *For two distributions p_1, p_2 with respective histograms h_1, h_2, we define the* relative earthmover distance *between them, $R(p_1, p_2) := R(h_1, h_2)$, as the minimum over all*

schemes of moving the probability mass of the first histogram to yield the second histogram, of the cost of moving that mass, where the per-unit mass cost of moving mass from probability x to y is $|\log(x/y)|$. Formally, for $x, y \in (0, 1]$, the cost of moving $x \cdot h(x)$ units of mass from probability x to y is $x \cdot h(x)|\log \frac{x}{y}|$.

One can also define the relative earthmover distance via the following dual formulation (given by the Kantorovich-Rubinstein theorem [77], which yields exactly what one would expect from linear programming duality):

$$R(h_1, h_2) = \sup_{f \in \mathcal{R}} \sum_{x: h_1(x)+h_2(x)\neq 0} f(x) \cdot x \left(h_1(x) - h_2(x)\right),$$

where $\mathcal{R}$ is the set of differentiable functions $f : (0, 1] \to \mathbb{R}$, s.t. $|\frac{d}{dx} f(x)| \leq \frac{1}{x}$.

We provide two examples:

Example 3.2. *Letting $Unif(q)$ denotes the uniform distribution supported on q elements,*

$$R\left(Unif(m), Unif(\ell)\right) = |\log m - \log \ell|,$$

since all of the probability mass in the first histogram at probability $\frac{1}{m}$ must be moved to probability $\frac{1}{\ell}$, at a per-unit-mass cost of $|\log \frac{m}{\ell}| = |\log m - \log \ell|$.

Example 3.3. *Consider the following distribution of fish species in a given lake:*

$$Pr(trout) = 1/2, \quad Pr(salmon) = 1/4,$$
$$Pr(cod) = Pr(whale) = Pr(eel) = Pr(shark) = 1/16.$$

The associated histogram *of this distribution is $h : \mathbb{R}^+ \to \mathbb{Z}$ defined by $h(1/16) = 4$, $h(1/4) = 1$, $h(1/2) = 1$, and for all $x \notin \{1/16, 1/4, 1/2\}$, $h(x) = 0$. If we now consider a second distribution over $\{a, b, c\}$ defined by the probabilities $Pr(a) = 1/2, \quad Pr(b) = 1/4, \quad Pr(c) = 1/4$, and let h' be its associated histogram, then the relative earthmover distance $R(h, h') = \frac{1}{4}|\log \frac{1/4}{1/16}|$, since we must take all the mass that lies at probability $1/16$ and move it to probability $1/4$ in order to turn the first distribution into one that yields a histogram identical to h'.*

In this chapter we will restrict our attention to properties that satisfy a weak notion of continuity, defined via the relative earthmover distance.

Definition 3.4. *A symmetric distribution property π is* (ϵ, δ)-continuous *if for all distributions p_1, p_2 with respective histograms h_1, h_2 satisfying $R(h_1, h_2) \leq \delta$ it follows that $|\pi(p_1) - \pi(p_2)| \leq \epsilon$.*

We note that both entropy and support size are easily seen to be continuous with respect to the relative earthmover distance.

Fact 3.5. *For a distribution $p \in \mathcal{D}^n$, and $\delta > 0$*

- *The entropy, $H(p) := -\sum_i p(i) \cdot \log p(i)$ is (δ, δ)-continuous, with respect to the relative earthmover distance.*
- *The support size $S(p) := |\{i : p(i) > 0\}|$ is $(n\delta, \delta)$-continuous, with respect to the relative earthmover distance, over the set of distributions which have no probabilities in the interval $(0, \frac{1}{n})$.*

Summary of Results

The main theorem of the chapter is the following:

Theorem 3.1. *For sufficiently large n, and any $c \in [1, \log n]$, given a sample of $c\frac{n}{\log n}$ independent draws from a distribution $p \in \mathcal{D}^n$, with probability at least $1 - e^{-n^{\Omega(1)}}$ over the randomness in the draws of the sample, our algorithm returns a distribution $\hat{p}$ such that the relative-earthmover distance between p and $\hat{p}$ satisfies*

$$R(p, \hat{p}) \leq O\left(\frac{1}{\sqrt{c}}\right).$$

For entropy and support size, Theorem 3.1 together with Fact 3.5 yields:

Corollary 3.6. *There exists a constant b such that for any $\epsilon \in [\frac{1}{\sqrt{\log n}}, 1]$ and sufficiently large n, given $\frac{b}{\epsilon^2}\frac{n}{\log n}$ independent draws from distribution $p \in \mathcal{D}^n$, our estimator will output a pair of real numbers $(\hat{H}, \hat{s})$ such that with probability $1 - e^{-n^{\Omega(1)}}$,*

- *$\hat{H}$ is within ϵ of the entropy of p, and*
- *$\hat{s}$ is within $n\epsilon$ of the support size of p, provided none of the probabilities in p lie in $(0, \frac{1}{n})$.*

This estimator has the optimal dependence on n, up to constant factors; in Chapter 5 we show the following lower bound:

Theorem 5.1. *For any positive constant $\phi < \frac{1}{4}$, there exists a pair of distributions p^+, p^- that are $O(\phi|\log \phi|)$-close in the relative earthmover distance, respectively, to the uniform distributions on n and $\frac{n}{2}$ elements, but which are information theoretically indistinguishable for any constant probability greater than $1/2$ when given fingerprints derived from samples of size $k = \frac{\phi}{32} \cdot \frac{n}{\log n}$.*

Specifically, estimating entropy to any constant error less than $\frac{\log 2}{2}$ requires a sample of size $\Theta(\frac{n}{\log n})$, as does estimating support size to any error that is a constant factor less than $\frac{n}{4}$.

Phrased differently, letting $\hat{H} : [n]^k \to \mathbb{R}$ denote an estimator that takes as input a sample of size k, and outputs an estimate of the entropy of the distribution from which the sample was drawn, and letting $S \underset{k}{\leftarrow} p$ denote a sample of size k consisting of independent draws from distribution $p \in \mathcal{D}^n$, we have the following: there exists a constant b' such that for $k = b' \frac{n}{\log n}$,

$$\inf_{\hat{H}} \sup_{p \in \mathcal{D}^n} \Pr_{S \underset{k}{\leftarrow} p} \left[|\hat{H}(S) - H(p)| > 0.3 \right] > 0.49,$$

where the infimum is taken over all possible estimators.

Further, by choosing a positive $\epsilon < 1$ and then constructing the distributions $p_\epsilon^+, p_\epsilon^-$ that, with probability ϵ draw an element according to p^+, p^- respectively and otherwise return another symbol, $\perp$, the entropy gap between p_ϵ^+ and p_ϵ^- is an ϵ fraction of what it was originally, and further, distinguishing them requires a factor $\frac{1}{\epsilon}$ larger sample. That is,

Corollary 3.7. *For large enough n and small enough ϵ, the sample complexity of estimating entropy to within ϵ grows as $\Omega(\frac{n}{\epsilon \log n})$.*

While the positive results of Theorem 3.1 match these lower bounds in their dependence on n, there is a gap in the dependence on the desired accuracy, ϵ, with a $\frac{1}{\epsilon}$ dependence in the lower bounds and a $\frac{1}{\epsilon^2}$ dependence in the upper bound. This prompts the following question, which we resolve in Chapter 6:

> *For an optimal entropy estimator, as the sample size increases, does the error decay linearly, or with the square root of the sample size?*

In Section 3.3, we generalize our entire framework to estimate properties of pairs of distributions. As in the setting described above for properties of a single distribution, given a pair of samples drawn from two (different) distributions, we can characterize the performance of our estimators in terms of returning a representation of the pair of distributions. For clarity, we state our performance guarantees for estimating total variational distance (ℓ_1 distance); see Theorem 3.3 in Section 3.3 for the more general formulation.

Theorem 3.2. *There exists a constant b such that for any positive $\epsilon < 1$ and sufficiently large n, given a pair of samples of size $\frac{b}{\epsilon^2} \frac{n}{\log n}$ drawn, respectively, from $p, q \in \mathcal{D}^n$, our estimator will output a real number, $\hat{d}$, such that with probability $1 - e^{-n^{\Omega(1)}}$*

$$|\hat{d} - D_{tv}(p, q)| \leq \epsilon,$$

where $D_{tv}(p, q) = \sum_i \frac{1}{2}|p(i) - q(i)|$ is the ℓ_1 distance between distributions p and q.

This estimator has the optimal dependence on n, up to constant factors; in Chapter 5 we also extend our lower bounds for estimating properties of single distributions to lower bounds for estimating properties of pairs of distributions:

Theorem 5.2. *For any constants $0 < a < b < \frac{1}{2}$, and probability of failure $\delta < 1/2$, for sufficiently large n, given samples from a pair of distributions of support at most n, distinguishing whether their total variational distance (ℓ_1 distance) is less than a or greater than b with probability of success at least $1 - \delta$, requires samples of size $O(\frac{n}{\log n})$.*

3.1 An LP–Based Canonical Estimator

Given the fingerprint $\mathcal{F}$ of a sample of size k drawn from a distribution with histogram h, the high-level approach is to use linear programming to find a histogram $\hat{h}$ that has the property that if one were to take k samples from a distribution with histogram $\hat{h}$, the fingerprint of the resulting samples would be similar to the observed fingerprint $\mathcal{F}$. Thus in some sense $\hat{h}$ "could plausibly have generated" $\mathcal{F}$. The hope is then that h and $\hat{h}$ will be similar, and, in particular, have similar entropies, support sizes, etc.

For general fingerprints, how does one obtain the histogram that could have "most plausibly" generated the fingerprint, in a principled fashion? The intuition will come from first understanding the structure of the map from histograms to fingerprints, as this is the map that we are effectively inverting. See Figure 3.1 for several examples of histograms and a typical fingerprint of a sample drawn from the corresponding distribution.

Given a distribution p, and some domain element j occurring with probability $x = p(j)$, the probability that it will be drawn exactly i times in a sample of size k drawn from p is $Pr[Binomial(k, x) = i]$. By linearity of expectation, the expected ith fingerprint entry will be

$$E[\mathcal{F}_i] = \sum_{x:h_p(x)\neq 0} h_p(x) Pr[Binomial(k, x) = i]. \tag{3.1}$$

As an illustration of our approach, consider a sample of size $k = 500$ drawn from the uniform distribution on 1000 elements. The expected fingerprint of this sample would be $\mathrm{E}[\mathcal{F}_i] = 1000 \cdot \Pr[Binomial(k, \frac{1}{1000}) = i] \approx (303.5, 75.8, 12.6, 1.6, \ldots)$. Thus if we are given a sample of size $k = 500$, with fingerprint $\mathcal{F} = (301, 78, 13, 1, 0, 0, \ldots)$, one might note that the uniform distribution on 1000 elements could plausibly have generated the observed fingerprint, and thus, although the observed sample only contained 391 unique domain elements, we might be justified in concluding that the true distribution from which the sample was drawn is "close" to $Unif[1000]$, and, for example, guess that the entropy of the true distribution is close to $H(Unif(1000)) = \log 1000$.

Our approach rests on the following two observations: 1) the mapping (described by Equation 3.1) between histograms and expected fingerprints is *linear* in the histogram entries, with coefficients given by the binomial probabilities. 2) fingerprint entries will be tightly concentrated about their expected value. These observations motivate a "first moment" approach. We will describe a linear program that inverts the "roughly linear" map from histograms to fingerprint entries, to yield a map from observed fingerprints, to plausible histograms $\hat{h}$. Namely, we use linear programming to find the generalized histogram $\hat{h}$

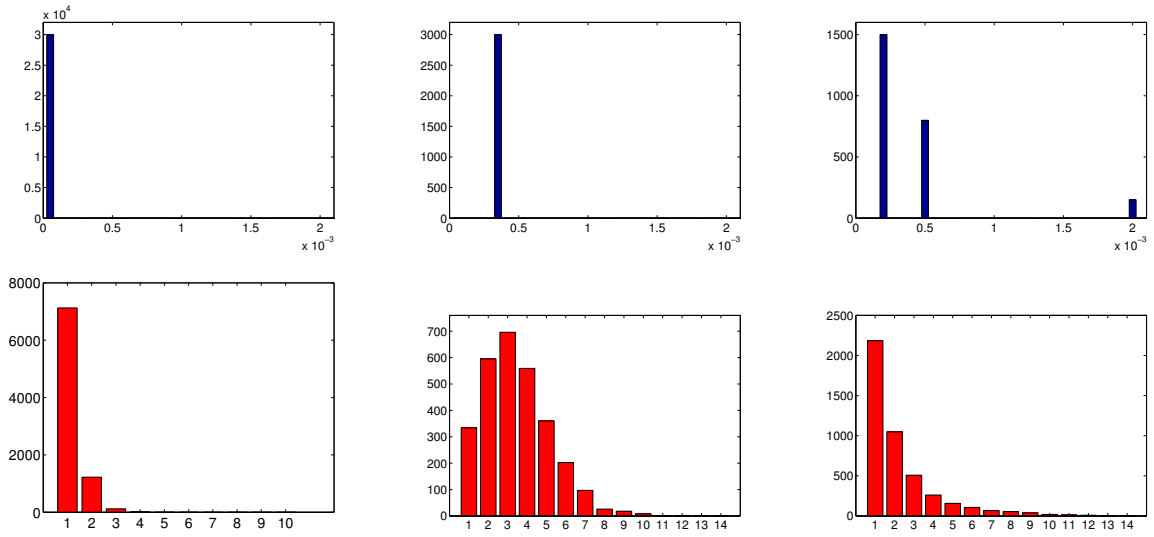

Figure 3.1: Three fingerprints (bottom row) derived from samples of size 10,000, together with the corresponding histograms (top row) of the distributions from which each sample was drawn. Intuitively, our estimator is solving the inversion problem: given a fingerprint, it finds a histogram from which the sample could, plausibly, have been drawn.

that minimizes the discrepancy between the observed fingerprint values and the expected fingerprint values if the sample had been drawn from $\hat{h}$, given by Equation 3.1.

To make the linear programs finite, we consider a fine mesh of values $x_1, \ldots, x_\ell \in (0, 1]$ that between them discretely approximate the potential support of a histogram. The variables of the linear program, $v_{x_1}, \ldots, v_{x_\ell}$ will correspond to the histogram values at these mesh points, with variable v_{x_i} representing the number of domain elements that occur with probability x_i, namely $\hat{h}(x_i)$.

As it turns out, we will only solve the linear program on the "infrequently occurring" portion of the distribution—that is, the mesh $\{x_1, \ldots, x_\ell\}$ of variables for the linear program will stop considerably before 1. This linear programming approach is specifically designed to tackle the regime in which many domain elements occur infrequently; for the "frequently occurring" region of the distribution, the empirical distribution of the sample does an adequate job of capturing the shape of the distribution (i.e. if some domain element is seen $k/2$ times, it is likely that this domain element has probability $\approx 1/2$). Thus our estimator will use the linear program to recover the "infrequently occurring" portion of the distribution, and then simply append the empirical distribution of the frequently occurring portion. To avoid the issues which may arise near the threshold between the "low probability" and "high probability" regimes , we choose the location of this threshold so as to have relatively little probability mass in the nearby region. We note that a unified approach is possible, though, computationally, it is certainly preferable to only apply the linear programming approach

to the portion of the fingerprint for which the naive empirical estimates fail. Finally, to simplify the analysis, we replace the binomial probabilities, with the corresponding Poisson approximation: $\Pr[Binomial[k, x] = i] \approx poi(kx, i)$.

We now formally define our algorithm. In the first step, we pick a cutoff value c which serves as the threshold between the linear program regime, and the regime for which we use the empirical distribution. For clarity of exposition, we state the algorithm in terms of three positive constants, $\mathcal{B}, \mathcal{C}$, and $\mathcal{D}$, which can be defined arbitrarily provided $\mathcal{B} > \mathcal{C} > \frac{\mathcal{B}}{2}$, $\frac{\mathcal{B}}{2} > \mathcal{D}$, and $0.8 + \mathcal{B} + \mathcal{D} < 1$.

Algorithm 3.8. ESTIMATE UNSEEN
Input: k-sample fingerprint $\mathcal{F}$, upper bound on the support size, n:
Output: Generalized histogram g_{LP}.

- *Let* $c := \min\{i : i \in [k^{\mathcal{B}}, 2 \cdot k^{\mathcal{B}}]$ *and* $\sum_{j=i}^{i+k^{\mathcal{C}}} j\mathcal{F}_j \leq k^{1-\mathcal{B}+\mathcal{C}}\}$.
- *Let* $v = (v_{x_1}, v_{x_2}, \ldots)$ *be the solution to Linear Program 3.9, on input* $\mathcal{F}, c,$ *and* n.
- *Let* g_{LP} *be the generalized histogram formed by setting* $g_{LP}(x_i) = v_{x_i}$ *for all* i, *and then for each integer* $j \geq c + k^{\mathcal{C}}$, *incrementing* $g_{LP}(\frac{j}{k})$ *by* $\mathcal{F}_j$.

Linear Program 3.9.
Given a k-sample fingerprint $\mathcal{F}$, integer c, and upper bound n on the support size:

- *Define the set* $X := \{\frac{1}{nk}, \frac{2}{nk}, \frac{3}{nk}, \ldots, \frac{c+k^{\mathcal{C}}/2}{k}\}$.
- *For each* $x \in X$, *define the associated LP variable* v_x.

The linear program is defined as follows:

$$\textit{Minimize} \sum_{i=1}^{c} \left| \mathcal{F}_i - \sum_{x \in X} poi(kx, i) v_x \right|,$$

Subject to:

- $\sum_{x \in X} x \cdot v_x + \sum_{i=c+k^{\mathcal{C}}}^{k} \frac{i}{k} \mathcal{F}_i = 1$ *(total prob. mass $= 1$.)*
- $\sum_{x \in X} v_x \leq n + k$ *(support size is not too big)*
- $\forall x \in X, v_x \geq 0$ *(histogram entries are non-negative)*

The following restatement of Theorem 3.1 describes the performance of the above algorithm.

Theorem. *For sufficiently large n and any $c \in [1, \log n]$, given a sample of size $c\frac{n}{\log n}$ consisting of independent draws from a distribution $p \in \mathcal{D}^n$, with probability at least $1 - e^{-n^{\Omega(1)}}$ over the randomness in the selection of the sample, Algorithm 3.8 returns a generalized histogram g_{LP} such that*

$$R(p, g_{LP}) \leq O\left(\frac{1}{\sqrt{c}}\right).$$

We stress that the above formulation of the linear program and algorithm were chosen to provide the simplest analysis. While the proof of the above theorem requires considerable technical machinery, the machinery can, at the expense of clarity, be adapted to prove analogous results for a number of variants of Linear Program 3.9. In particular, analogous results hold for various natural rescalings of the objective function, including $\sum_i \frac{i}{k} \left|\mathcal{F}_i - \sum_x poi(kx, i)v_x\right|$, or $\sum_i \frac{1}{\sqrt{\mathcal{F}_i+1}} \left|\mathcal{F}_i - \sum_x poi(kx, i)v_x\right|$. This latter objective function is particularly natural, as $\sqrt{\mathcal{F}_i}$ is a reasonable approximation for the standard deviation of the distribution of $\mathcal{F}_i$. In Chapter 8, we use this scaling in the linear program that we employ in our experimental tests. In a different direction, we note that the mesh X of variables for which we solve the linear program can also be varied considerably, while preserving the guarantees of the above theorem. In particular, one could employ a coarser quadratically spaced mesh, e.g. $X = \frac{1}{k^{1.1}}\{1^2, 2^2, 3^2, \ldots, k^{0.2}\}$, which would reduce the number of variables in the linear program to $k^{0.1}$, in which case the linear program can be solved (using Karmarkar's algorithm [78]) in time linear in the sample size, k. We provide details of this modification in Section 3.2. A number of other aspects of Linear Program 3.9 can also be varied, including the selection of the cutoff, c, between the "high" and "low" probability regimes (and the mechanics of the transition region), and, of course, the constants.

We note that Algorithm 3.8 returns a generalized histogram; specifically, the histogram values will not be integral, as is the case for an actual histogram corresponding to a distribution. For the purpose of estimating relative-earthmover continuous properties, a generalized histogram suffices. For example, in the case of entropy, given a generalized histogram g, one can compute $H(g) := \sum_{x:g(x)\neq 0} g(x) \cdot x|\log x|$, irrespective of whether $g(x)$ is an integer. Nevertheless, if one desires an actual distribution corresponding to a histogram (as opposed to a generalized histogram), the following algorithm and lemma characterizing its performance, show one way to easily round the generalized histogram to obtain a histogram that is close in relative earthmover distance.

Algorithm 3.10. ROUND TO HISTOGRAM
Input: Generalized histogram g.
Output: Histogram h.

- *Initialize h' to be identically 0, let $x_1, x_2, \ldots, x_m$ be the support of g, and set the variable $diff := 0$.*
- *Define* $\alpha = \max(x_i : g(x_i) \notin \mathbb{N} \cup \{0\})$.
- *For each* $i = 1, \ldots, m$ *s.t.* $x_i \leq \alpha$, *do the following:*
  - *If* $diff < 0$ *set* $h'(x_i) = \lceil g(x_i) \rceil$, *otherwise, if* $diff \geq 0$ *set* $h'(x_i) = \lfloor g(x_i) \rfloor$.
  - *Increment* $diff$ *by* $x_i(h(x_i) - g(x_i))$.
- *For each* i *s.t.* $x_i > \alpha$, *set* $h'(x_i) = g(x_i)$.
- *Define histogram* h *by setting* $h(\frac{x_i}{1+diff}) = h'(x_i)$.

Lemma 3.11. *Let h be the output of running Algorithm 3.10 on generalized histogram g. The following conditions hold:*

- *For all x, $h(x) \in \mathbb{N} \cup \{0\}$, and $\sum_{x:h(x)\neq 0} xh(x) = 1$, hence h is a histogram of a distribution.*
- $R(h, g) \leq 5\alpha \left|\log \min(x : g(x) \neq 0)\right|$. *where* $\alpha := \max(x : g(x) \notin \mathbb{N} \cup \{0\})$.

Proof. Let h' be as defined in Algorithm 3.10, and note that for any y,

$$\sum_{x \leq y : g(x) \neq 0} x(h'(x) - h(x)) \in [-y, y],$$

and thus at the termination of the algorithm, $|diff| \leq \alpha$. Intuitively, the lemma follows from observing that aside from at most $|diff|$ probability mass that we might need to move anywhere, at a cost of $\log \min(x : g(x) + h(x) \neq 0)$, we can smear the probability mass in g to yield h by only moving mass locally, at a total cost of roughly α.

To make this intuition rigorous, will explicitly describe an earthmoving scheme, in two stages. In the first stage, consider the generalized histogram h'' that is defined to be h' augmented by adjusting $diff$ probability mass (without altering the support) so that h'' has total probability mass 1. We first show that $R(h, h'') \leq 2\alpha |\log \min(x : h(x) \neq 0)|$. Consider the earthmoving scheme that, for all i, moves $x_i h''(x_i)$ units of probability mass from x_i to $\frac{x_i}{1+diff}$. Such an earthmoving scheme has cost at most $|\log(1 + diff)| \leq \alpha$, and the generalized histogram h''' resulting from such a scheme has the property that $\sum_{x:h(x)\neq 0} |x(h(x) - h'''(x))| \leq |diff|$, and hence this discrepancy can be removed by moving at most $|diff|$ units of probability mass, at a total cost of $|diff| \cdot |\log \min(x : h(x) \neq 0)| \leq \alpha |\log \min(x : h(x) \neq 0)|$.

We now bound $R(g, h'')$, from which the lemma will follow from the triangle inequality. Let $X = \{x_1, x_2, \ldots\}$ denote the support of g. Consider the following earthmoving scheme

that yields h'' from g : begin by defining g' to be identical to g, and move $x_1(g(x_1) - h''(x_1))$ units of probability mass from x_1 to $x_2, \ldots, x_j$, in such a way that for all $j' < j, g'(x_{j'}) = h''(x_{j'})$, and $h''(x_j) > g(x_j)$. Continue iteratively: for $i \geq 2$, if $h''(x_i) < g(x_i)$ then move $x_i(g'(x_i) - h''(x_i))$ units of probability mass to $x_{i+1}, \ldots, x_j$, in such a way that for all $j' < j, g'(x_{j'}) = h''(x_{j'})$, and $h''(x_j) > g(x_j)$; if $h''(x_i) > g(x_i)$ then move $x_i(g'(x_i) - h''(x_i))$ units of probability mass from $x_{i+1}, \ldots, x_j$, in such a way that for all $j' < j, g'(x_{j'}) = h''(x_{j'})$, and $h''(x_j) < g(x_j)$.

By construction, the above process will yield $g' = h''$. We now analyze the cost of the scheme. Define the set

$$Y := \{x_i : sign\left(\sum_{j<i} x_j(g(x_j) - h'(x_j))\right) \neq sign\left(\sum_{j\leq i} x_j(g(x_j) - h'(x_j))\right)\},$$

and denote the elements of Y by $y_1, \ldots, y_m$, with $y_i < y_{i+1}$. The cost of the above scheme is bounded by

$$\alpha|\log y_m| + \sum_{i=1}^{m} y_i \log(\frac{y_{i+1}}{y_i}),$$

where the first term is the contribution from performing the scheme for $x_i \in [y_m, \alpha]$, and the second term is the cost of the remaining portion of the scheme. To simplify the above expression, consider the following:

$$\begin{aligned} y_i \log \frac{y_{i+1}}{y_i} &= y_i \int_{y_i}^{y_{i+1}} \frac{1}{x} dx \\ &\leq \int_{y_i}^{y_{i+1}} x \frac{1}{x} dx \qquad \text{(since } x \geq y_i \text{ for } x \in [y_i, y_{i+1}]) \\ &= y_{i+1} - y_i. \end{aligned}$$

Hence we have

$$R(g, h'') \leq \alpha|\log y_m|.$$

Putting together the pieces, we have:

$$\begin{aligned} R(g, h) &\leq R(g, h'') + R(h'', h) \\ &\leq \alpha|\log y_m| + \alpha + \alpha + \alpha|\log \min(x : h(x) \neq 0)| \\ &\leq 2\alpha + \alpha|\log \min(x : g(x) \neq 0)|(2 + \alpha) \\ &\leq 5\alpha|\log \min(x : h(x) \neq 0)|. \end{aligned}$$

□

The following condition defines what it means for a sample from a distribution to be "faithful"; roughly a sample is "faithful" if it is representative of a typical sample from the distribution—no domain element occurs too much more frequently than one would expect,

and the fingerprint entries are reasonably close to their expected values. To prove Theorem 3.1, we first show that a sample consisting of random draws from a fixed distribution is "faithful" with probability $1-e^{-k^{\Omega(1)}}$. This will follow easily from basic tail bounds on Poisson random variables (see Appendix A.2), and Chernoff bounds. Having thus compartmentalized the probabilistic component of our theorem, we will then argue that our algorithm will be successful whenever it receives a faithful sample as input.

Definition 3.12. *A sample of size k with fingerprint $\mathcal{F}$, drawn from a distribution p with histogram h, is said to be* faithful *if the following conditions hold:*

- *For all i,*
$$\left|\mathcal{F}_i - \sum_{x:h(x)\neq 0} h(x)\cdot poi(kx,i)\right| \leq k^{\frac{1}{2}+\mathcal{D}}.$$
- *For all domain elements i whose true probability $p(i) \geq k^{-1+\mathcal{B}}$, the number of times i occurs differs from its expectation of $k\cdot p(i)$ by at most $(k\cdot p(i))^{\frac{1}{2}+\mathcal{D}}$.*
- *Defining $c = \min\{i : i \in [k^{\mathcal{B}}, 2\cdot k^{\mathcal{B}}] \text{ and } \sum_{j=i}^{i+k^{\mathcal{C}}} j\mathcal{F}_j \leq k^{1-\mathcal{B}+\mathcal{C}}\}$, as in Algorithm 3.8,*
$$\sum_{x\in\left[\frac{c}{k},\frac{c+k^{\mathcal{C}}}{k}\right]} x\cdot h(x) \leq 4k^{\mathcal{C}-\mathcal{B}}.$$
- *Additionally,*
$$\sum_{i\geq c+k^{\mathcal{C}}} \frac{i}{k}\mathcal{F}_i + \sum_{x\leq\frac{c+k^{\mathcal{C}}/2}{k}:h(x)\neq 0} x\cdot h(x) \;\leq\; 1+k^{-\frac{1}{2}+\mathcal{D}}.$$

Lemma 3.13. *There is a constant $\gamma > 0$ such that for sufficiently large k, the probability that a sample of size k consisting of independent draws from a fixed distribution is "faithful" is at least $1-e^{-k^{\gamma}}$.*

Proof. We first analyze the case of a $Poi(k)$-sized sample drawn from a distribution with histogram h. Thus
$$\mathrm{E}[\mathcal{F}_i] = \sum_{x:h(x)\neq 0} h(x)poi(kx,i).$$
Additionally, the number of times each domain element occurs is independent of the number of times the other domain elements occur, and thus each fingerprint entry $\mathcal{F}_i$ is the sum of independent random 0/1 variables, representing whether each domain element occurred exactly i times in the sample (i.e. contributing 1 towards $\mathcal{F}_i$). By independence, Chernoff bounds apply, showing that for any i,
$$\Pr\left[|\mathcal{F}_i - \mathrm{E}[\mathcal{F}_i]| \geq k^{\frac{1}{2}+\mathcal{D}}\right] \leq 2e^{-\frac{k^{1+2\mathcal{D}}}{3\mathrm{E}[\mathcal{F}_i]}} \leq 2e^{-\frac{k^{2\mathcal{D}}}{3}}.$$

A union bound over the first k fingerprints shows that the probability that any of the first k fingerprints exceed their expectations by this amount is at most $k \cdot 2e^{-\frac{k^{2\mathcal{D}}}{3}} \leq e^{-k^{\Omega(1)}}$.

For the second condition of "faithful", by basic tail bounds for the Poisson distribution, we have $\Pr[|Poi(x) - x| > x^{\frac{1}{2}+\mathcal{D}}] \leq e^{-x^{\Omega(1)}})$, hence for $x > k^{\mathcal{B}}$, the probability is bounded by $e^{-k^{\Omega(1)}}$.

We now show that the third condition is implied by the first. For any $x \in [\frac{c}{k}, \frac{c+k^{\mathcal{C}}}{k}]$, $\mathrm{E}[I_{[c,c+k^{\mathcal{C}}]}(Poi(x))] \geq \frac{x}{2} - o(1)$, where $I_{[c,c+k^{\mathcal{C}}]}(y)$ is the function that is equal to y if $y \in [c, c+k^{\mathcal{C}}]$, and is 0 otherwise. Specifically, assuming for the sake of contradiction that $\sum_{x \in \left[\frac{c}{k}, \frac{c+k^{\mathcal{C}}}{k}\right]} xh(x) > 4k^{\mathcal{C}-\mathcal{B}}$, then $\sum_{i=c}^{c+k^{\mathcal{C}}} \mathrm{E}[\mathcal{F}_i] > 2k^{1+\mathcal{C}-\mathcal{B}} - o(1)$. On the other hand $\sum_{i=c}^{c+k^{\mathcal{C}}} \mathcal{F}_i \leq k^{1+\mathcal{C}-\mathcal{B}}$, yet the disparity between these $k^{\mathcal{C}}$ fingerprints and expected fingerprints, by the first condition, is bounded by $k^{\mathcal{C}} k^{\frac{1}{2}+\mathcal{D}}$, yielding the claimed condition.

We now consider the final condition. By a union bound over tail bounds on Poissons, with probability $1 - e^{-k^{\Omega(1)}}$ every domain element with true probability less than $\frac{c+k^{\mathcal{C}}/2}{k}$ will occur fewer than $c + k^{\mathcal{C}}$ times. Given that this happens, such domain elements will not contribute to the $\sum_{i \geq c+k^{\mathcal{C}}} \frac{i}{k}\mathcal{F}_i$ term in the statement of the fourth condition; thus if the fourth condition is violated, then the total empirical probability mass of these domain elements must exceed its expectation by at least $k^{-\frac{1}{2}+\mathcal{D}}$. The probability this occurs is bounded by $\Pr[Poi[k] > k + k^{\frac{1}{2}+\mathcal{D}}] \leq e^{-k^{\Omega(1)}}$.

Thus we have shown that provided we are considering a sample of size $Poi(k)$, the probability that the conditions hold is at least $1 - e^{-k^{\Omega(1)}}$. To conclude, note that $\Pr[Poi(k) = k] > \frac{1}{3\sqrt{k}}$, and hence the probability that the conditions do not hold for a sample of size exactly k (namely, the probability that they do not hold for a sample of size $Poi(k)$, conditioned on the sample size being exactly k), is at most a factor of $3\sqrt{k}$ larger, and hence this probability of failure is still $e^{-k^{\Omega(1)}}$, as desired. □

Lemma 3.14. *Given a distribution of support size at most n with histogram h, and a "faithful" sample of size k with fingerprint $\mathcal{F}$, if c is chosen as prescribed in Algorithm 3.8 then Linear Program 3.9 has a feasible point v' with objective value at most $O(k^{\frac{1}{2}+\mathcal{B}+\mathcal{D}})$. Additionally,*

$$R(h, h_{v'}) \leq O(k^{-\mathcal{B}+\mathcal{C}} + k^{-\mathcal{B}(\frac{1}{2}-\mathcal{D})} = O(\frac{1}{k^{\Omega(1)}}),$$

where $h_{v'}$ is the histogram that would be returned by Algorithm 3.8 if v' were used in place of the solution to the linear program, v.

Recall that the linear program aims to find distributions that "could reasonably have generated" the observed fingerprint $\mathcal{F}$. Following this intuition, we will show that provided the sample is faithful, the true distribution, h, minimally modified, will in fact be the desired feasible point v'.

Roughly, v' will be defined by taking the portion of h with probabilities at most $\frac{c+k^{\mathcal{D}}/2}{k}$ and rounding the support of h to the closest multiple of $1/nk$, so as to be supported at points in the set $X = \{1/nk, 2/nk, \ldots\}$. We will then need to adjust the total probability

mass accounted for in v' so as to ensure that the first condition of the linear program is satisfied; this adjusting of mass must be accomplished while ensuring that the fingerprint expectations do not change significantly.

To argue that the linear program objective function value of v' is small, we note that the mesh X is sufficiently fine so as to guarantee that the rounding of the mass to integer multiples of $1/nk$ does not greatly change the expected fingerprints, and hence the expected fingerprint entries associated with v' will be close to those of h. Our definition of "faithful" guarantees that all fingerprint entries are close to their expectations, and hence the objective function value will be small.

To bound the relative earthmover distance between the true histogram h and the histogram $h_{v'}$ associated to v', we first note that the portion of $h_{v'}$ corresponding to probabilities below $\frac{c+k^c/2}{k}$ will be extremely similar to h, because it was created from h. By our choice of c, and the definition of "faithful", there is little mass in either the empirical or actual histograms between probabilities $\frac{c}{k}$ and $\frac{c+k^c}{k}$, and hence the discrepancy in this region will not contribute significantly to the cost of an earthmoving scheme. Finally, for probabilities above $\frac{c+k^c/2}{k}$, $h_{v'}$ and h will be similar because these frequently-occurring elements will appear close to their expected number of times, by the second condition of "faithful", and hence the relative earthmover distance between the empirical histogram and the true histogram in this frequently-occurring region will also be small. Below we make the details of this argument rigorous.

Proof of Lemma 3.14. We explicitly define v' as a function of the true histogram h and fingerprint of the sample, $\mathcal{F}$, as follows:

- Define h' such that $h'(x) = h(x)$ for all $x \leq \frac{c+k^c/2}{k}$, and $h'(x) = 0$ for all $x > \frac{c+k^c/2}{k}$, where c is as defined in Algorithm 3.8.
- Initialize v' to be 0, and for each $x \geq 1/nk$ s.t. $h'(x) \neq 0$ increment v'_y by $h'(x)$, where $y = \max(z \in X : z \leq x)$ is x rounded down to the closest point in set $X = \{1/nk, 2/nk, \ldots\}$.
- Let $m := \sum_{x \in X} x v'_x + m_{\mathcal{F}}$, where $m_{\mathcal{F}} := \sum_{i \geq \frac{c+k^c}{k}} \frac{i}{k} \mathcal{F}_i$. If $m < 1$, increment v'_y by $(1-m)/y$, where $y = \frac{c+k^c/2}{k}$. Otherwise, if $m \geq 1$, decrease v' arbitrarily until the total probability mass $m_{\mathcal{F}} + \sum_{x \in X} x v'_x = 1$.

We first note that the above procedure is well-defined, since $m_{\mathcal{F}} \leq 1$, and hence the probability mass in v' can always be modified so as to result in the total mass $m_{\mathcal{F}} + \sum_{x \in X} x v'_x = 1$, while keeping all entries of v' nonnegative.

To see that v' is a feasible point of the linear program, note that by construction, the first and third conditions of the linear program are trivially satisfied. The second condition of the linear program is satisfied because the true distribution has support at most n, and, crudely, in the final step of the construction of v', we increment v'_y by less than k.

We now consider the objective function value of v'. Note that $\sum_{i \leq c} poi(c + k^{\mathcal{C}}/2, i) = o(1/k)$, so the fact that we are truncating h at probability $\frac{c+k^{\mathcal{C}}/2}{k}$ in the first step in our construction of v', has little effect on the first c "expected fingerprints": specifically, for all such i,

$$\sum_{x:h(x)\neq 0} (h'(x) - h(x))\, poi(kx, i) = o(1).$$

Together with the first condition of the definition of faithful, by the triangle inequality, for each $i \leq c$,

$$\left| \mathcal{F}_i - \sum_{x:h'(x)\neq 0} h'(x) poi(kx, i) \right| \leq k^{\frac{1}{2}+\mathcal{D}} + o(1).$$

We now bound the contribution of the discretization to the objective function value. To this order, note that $\left|\frac{d\, poi(kx,i)}{d\, x}\right| \leq k$, and hence we have

$$\left| \sum_{x:h'(x)\neq 0} h'(x) poi(kx, i) - \sum_{x \in X} v'_x poi(kx, i) \right| \leq n\frac{k}{kn} = 1.$$

In the case that $m \leq 1$, where m is the amount of mass in v' before the final adjustment (as defined in the final step in the construction of v'), mass is added to v'_y, where $y = \frac{c+k^{\mathcal{C}}/2}{k}$, and thus since $\sum_{i \leq c} poi(ky, i) = o(1/k)$, this added mass alters $\sum_{x \in X} v'_x poi(kx, i)$ by at most $o(1)$. In the case that $m \geq 1$, by the fourth condition of "faithful", and the fact that h' is generated from h by rounding the support down, which only decreases the amount of probability mass, $m \leq 1 + k^{-\frac{1}{2}+\mathcal{D}}$, and hence the removal of this extra probability mass from v' will decrease the expected fingerprints, in total, by at most $k \cdot k^{-\frac{1}{2}+\mathcal{D}} = k^{\frac{1}{2}+\mathcal{D}}$. Thus in both the cases that $m \leq 1$ and $m > 1$ the objective function value associated to v' is bounded by

$$c\left(k^{\frac{1}{2}+\mathcal{D}} + 1 + o(1)\right) + k^{\frac{1}{2}+\mathcal{D}} \leq 3k^{\frac{1}{2}+\mathcal{B}+\mathcal{D}},$$

for sufficiently large k.

We now turn to analyzing the relative earthmover distance $R(h, h_{v'})$. Let g denote the generalized histogram defined by $g(x) = v'_x$ for all $x \in X$, and $g(\frac{i}{k}) = \mathcal{F}_i$ for all $i \geq c + k^{\mathcal{C}}$; in particular, $h_{v'}$ is the result of applying Algorithm 3.10 to the generalized histogram g. By Lemma 3.11, $R(h_{v'}, g) \leq 5(2k^{\mathcal{B}-1}) \log nk$. We now bound $R(g, h)$, from which our claim will follow by the triangle inequality.

We proceed in two steps. Let p denote the true distribution, with $p(i)$ representing the true probability with which domain element i occurs. Consider the following earthmoving scheme that will yield generalized histogram t from the true histogram h: first, for each $x \in [\frac{1}{nk}, \frac{c+k^{\mathcal{C}}/2}{k}]$ for which $h(x) > 0$, move $yh(x)$ units of probability mass from probability x to probability y, where $y = \max(z \in X : z \leq x)$. Next, for each domain element i, for which $p(i) \geq \frac{c+k^{\mathcal{C}}/2}{k}$ and for which element i appears $j \geq c + k^{\mathcal{C}}$ times in the sample, move $\frac{j}{k}$ units of

probability mass from probability $p(i)$ to probability $\frac{j}{k}$. We now argue that the generalized histogram t, yielded by this process satisfies $R(t,h) \leq 2k^{-\mathcal{B}(1/2-\mathcal{D})} + \frac{2}{k}$: To bound the cost of the first step of the scheme, note that since the support size is bounded by n, and the total probability mass in the distribution is at most 1, we have that the relative earthmover cost incurred by the first step of the above scheme is at most

$$\max_{x \geq 1/nk} \left(\min(nx \log \frac{x}{\lfloor nkx \rfloor / nk}, \log \frac{x}{\lfloor nkx \rfloor / nk}) \right) < \frac{2}{k}.$$

To bound the second phase of this earthmoving scheme, by the second condition of "faithful", each domain element with true probability $p(i) \geq k^{-1+\mathcal{B}}$ will occur j times, where $|j - kp(i)| \leq (kp(i))^{\frac{1}{2}+\mathcal{D}}$, and hence the total cost of this portion of the scheme is bounded by

$$\log \frac{k^{-1+\mathcal{B}}}{k^{-1+\mathcal{B}} - k^{-1+\mathcal{B}(\frac{1}{2}+\mathcal{D})}} \leq 2k^{-\mathcal{B}(1/2-\mathcal{D})}.$$

To complete our proof, we now argue that $R(t,g)$ is small. Indeed, the portions of g and t above probability $\frac{c+k^{\mathcal{C}}}{k}$ are *identical*. The portions of these generalized histograms corresponding to probabilities in the interval $[1/nk, \frac{c+k^{\mathcal{C}}/2}{k})$ would also be identical, except for the possible decreasing of the probability mass in the final step of constructing v'. By the fourth condition of "faithful", this removal step modifies at most $k^{-1/2+\mathcal{D}}$ units of mass, and hence the relative earthmover cost of moving this mass to any probability above $1/nk$ is bounded by $k^{-1/2+\mathcal{D}} \log nk$. The probability mass in t below probability $1/nk$ can be moved to any probability above $1/nk$ incurring a cost of at most $\max_{x \leq 1/nk} nx|\log x| \leq \frac{1}{k} \log nk$, (since the support is bounded by n.) Finally, the remainder of the discrepancy between t and g in due to the discrepancy in the range of probabilities $[\frac{c+k^{\mathcal{C}}/2}{k}, \frac{c+k^{\mathcal{C}}}{k}]$. By the third condition of"faithful", the mass in this region of t is bounded by $4k^{\mathcal{C}-\mathcal{B}}$; putting together these bounds, we have

$$R(t,g) \leq \left(k^{-1/2+\mathcal{D}} + \frac{1}{k} + k^{\mathcal{C}-\mathcal{B}} \right) \log nk.$$

Thus we have:

$$\begin{aligned} R(h, h_{v'}) &\leq R(h_{v'}, g) + R(g,t) + R(t,h) \\ &\leq O(k^{\mathcal{B}-1} \log k) + O\left((k^{-1/2+\mathcal{D}} + k^{\mathcal{C}-\mathcal{B}}) \log k\right) + O(k^{-\mathcal{B}(1/2-\mathcal{D})}). \end{aligned}$$

□

3.2 Similar Expected Fingerprints Imply Similar Histograms: A Chebyshev "Bump" Scheme

In this section we argue that if two histograms, h_1, h_2 corresponding to distributions with support size at most $O(n)$ have the property that their expected fingerprints derived from

$Poi(k)$-sized samples are very similar, then $R(h_1, h_2)$ must be small. This will guarantee that any two feasible points of Linear Program 3.9 that both have small objective function values correspond to histograms that are close in relative earthmover distance. The previous section established the existence of a feasible point with small objective function value that is close to the true histogram, hence by the triangle inequality, all such feasible points must be close to the true histogram; in particular, the optimal point—the solution to the linear program—will correspond to a histogram that is close to the true histogram of the distribution from which the sample was drawn, completing our proof of Theorem 3.1.

We define a class of earthmoving schemes, which will allow us to directly relate the relative earthmover cost of two distributions to the discrepancy in $Poi(k)$-sized sample fingerprint expectations corresponding to the two distributions. Our main technical tool is a Chebyshev polynomial construction.

Definition 3.15. *For a given k, a* β-bump earthmoving scheme *is defined by a sequence of positive real numbers $\{c_i\}$, the* bump centers*, and a sequence of functions $\{f_i\} : (0, 1] \to \mathbb{R}$ such that $\sum_{i=0}^{\infty} f_i(x) = 1$ for each x, and each function f_i may be expressed as a linear combination of Poisson functions, $f_i(x) = \sum_{j=0}^{\infty} a_{ij} poi(kx, j)$, such that $\sum_{j=0}^{\infty} |a_{ij}| \leq \beta$.*

Given a generalized histogram h, the scheme works as follows: for each x such that $h(x) \neq 0$, and each integer $i \geq 0$, move $xh(x) \cdot f_i(x)$ units of probability mass from x to c_i. We denote the histogram resulting from this scheme by $(c, f)(h)$.

Definition 3.16. *A bump earthmoving scheme (c, f) is* ϵ-good *if for any generalized histogram h, the relative earthmover distance between h and $(c, f)(h)$ is at most ϵ.*

The crux of the proof of correctness of our estimator is the explicit construction of a surprisingly good earthmoving scheme. We will show that for any k and $n = \delta k \log k$ for some $\delta \in [1, \log k]$, there exists an $O(\sqrt{\delta})$-good $k^{0.3}$-bump earthmoving scheme. In fact, we will construct a single scheme for all δ. Before describing our earthmoving scheme, we define a very simple scheme that provides some intuition as to how one can create a scheme whose cost is related to fingerprint expectations.

Perhaps the most natural bump earthmoving scheme is where $f_i(x) = poi(kx, i)$ and $c_i = \frac{i}{k}$. For $i = 0$, we may, for example, set $c_0 = \frac{1}{2k}$ so as to avoid a logarithm of 0 when evaluating relative earthmover distance. This is clearly a valid earthmoving scheme since $\sum_{i=0}^{\infty} f_i(x) = 1$ for any x.

The motivation for this construction is the fact that, for any i, the amount of probability mass that ends up at c_i in $(c, f)(h)$ is exactly c_i times the expectation of the ith fingerprint in a $Poi(k)$-sample from h; namely

$$\sum_{x:h(x)\neq 0} h(x)x \cdot poi(kx, i) = \sum_{x:h(x)\neq 0} h(x) \cdot poi(kx, i+1)\frac{i+1}{k}.$$

Thus if we apply this earthmover scheme to two histograms h, g derived from solutions to the linear program, their fingerprint expectations will closely match, and the result of applying

the earthmoving scheme to h, g will result in a pair of generalized histograms h', g', supported at the bump centers c_i, such that $R(h', g')$ is small.

The problem with this "Poisson bump" earthmoving scheme is that it incurs a very large relative earthmover cost, particularly for small probabilities. This is due to the fact that most of the mass that starts at a probability below $\frac{1}{k}$ will end up in the zeroth bump, no matter if it has probability nearly $\frac{1}{k}$, or the rather lower $\frac{1}{n}$. Phrased differently, the problem with this scheme is that the first few "bumps" are extremely fat. The situation gets significantly better for higher Poisson functions: most of the mass of $Poi(i)$ lies within relative distance $O(\frac{1}{\sqrt{i}})$ of i, and hence the scheme is relatively cheap for larger probabilities $x >> \frac{1}{k}$. We will therefore construct a scheme that uses Poisson functions $poi(kx, i)$ for $i \geq O(\log k)$, but takes great care to construct "skinnier" bumps below this region.

The main tool of this construction is the Chebyshev polynomials. For each integer $i \geq 0$, the ith Chebyshev polynomial, denoted $T_i(x)$, is the polynomial of degree i such that $T_i(\cos(y)) = \cos(i \cdot y)$. Thus, up to a change of variables, any linear combination of cosine functions up to frequency s may be re-expressed as the same linear combination of the first s Chebyshev polynomials. Given this, constructing a frugal earth-moving scheme is an exercise in trigonometric constructions.

Before formally defining our bump earthmoving scheme, we give a rough sketch of the key features. We define the scheme with respect to a paramter $s = O(\log k)$. For $i > s$, we use the fat Poisson bumps: that is, we define the bump centers $c_i = \frac{i}{k}$ and functions $f_i = poi(kx, i)$. For $i \leq s$, we will use skinnier "Chebyshev bumps"; these bumps will have roughly quadratically spaced bump centers $c_i \approx \frac{i^2}{k \log k}$, with the width of the ith bump roughly $\frac{i}{k \log k}$. At a high level, the $\log n$ factor in our $O(\frac{n}{\log n})$ bound on the sample size necessary to achieve accurate estimation, arises because the first few Chebyshev bumps have width $O(\frac{1}{k \log k})$, in contrast to the first Poisson bump, $poi(kx, 1)$, which has width $O(\frac{1}{k})$.

Definition 3.17. *The* Chebyshev bumps *are defined in terms of k as follows. Let $s = 0.2 \log k$. Define $g_1(y) = \sum_{j=-s}^{s-1} \cos(jy)$. Define*

$$g_2(y) = \frac{1}{16s}\left(g_1(y - \frac{3\pi}{2s}) + 3g_1(y - \frac{\pi}{2s}) + 3g_1(y + \frac{\pi}{2s}) + g_1(y + \frac{3\pi}{2s})\right),$$

and, for $i \in \{1, \ldots, s-1\}$ define $g_3^i(y) := g_2(y - \frac{i\pi}{s}) + g_2(y + \frac{i\pi}{s})$, and $g_3^0 = g_2(y)$, and $g_3^s = g_2(y + \pi)$. Let $t_i(x)$ be the linear combination of Chebyshev polynomials so that $t_i(\cos(y)) = g_3^i(y)$. We thus define $s+1$ functions, the "skinny bumps", to be $B_i(x) = t_i(1 - \frac{xk}{2s}) \sum_{j=0}^{s-1} poi(xk, j)$, for $i \in \{0, \ldots, s\}$. That is, $B_i(x)$ is related to $g_3^i(y)$ by the coordinate transformation $x = \frac{2s}{k}(1 - \cos(y))$, and scaling by $\sum_{j=0}^{s-1} poi(xk, j)$.

See Figure 3.2 for a plot of $g_2(y)$, illustrating, up to coordinate transformations, a "skinny Chebyshev bump." The Chebyshev bumps of Definition 3.17 are "third order"; if, instead, we had considered the analogous less skinny "second order" bumps by defining $g_2(y) := \frac{1}{8s}\left(g_1(y - \frac{\pi}{s}) + 2g_1(y) + g_1(y + \frac{\pi}{s})\right)$, then the results would still hold, though the proofs are slightly more cumbersome.

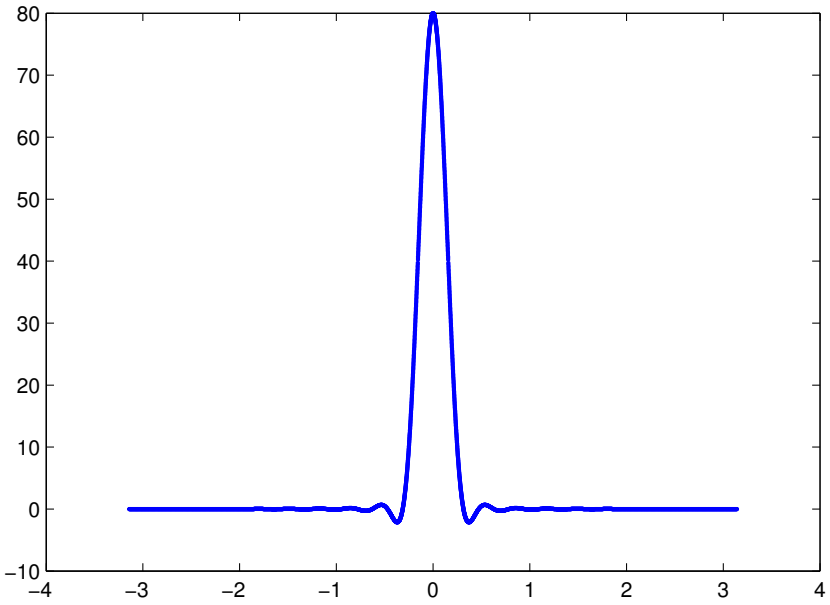

Figure 3.2: A plot of the "skinny" function $g_2(y)$ (without the scaling factor). This is the main ingredient in the Chebyshev bumps construction of Definition 3.17.

Definition 3.18. *The* Chebyshev earthmoving scheme *is defined in terms of k as follows: as in Definition 3.17, let $s = 0.2 \log k$. For $i \geq s+1$, define the ith bump function $f_i(x) = poi(kx, i-1)$ and associated bump center $c_i = \frac{i-1}{k}$. For $i \in \{0, \ldots, s\}$ let $f_i(x) = B_i(x)$, and for $i \in \{1, \ldots, s\}$, define their associated bump centers $c_i = \frac{2s}{k}(1 - \cos(\frac{i\pi}{s}))$, and let $c_0 := c_1$.*

The following two lemmas, together, show that the Chebyshev earthmoving scheme is a $2k^{0.3}$-bump earthmoving scheme.

Lemma 3.19. *Each $B_i(x)$ may be expressed as $\sum_{j=0}^{\infty} a_{ij} poi(kx, j)$ for a_{ij} satisfying*

$$\sum_{j=0}^{\infty} |a_{ij}| \leq 2k^{0.3}.$$

Proof. Consider decomposing $g_3^i(y)$ into a linear combination of $\cos(jy)$, for $j \in \{0, \ldots, s\}$. Since $\cos(-jy) = \cos(jy)$, $g_1(y)$ consists of one copy of $\cos(sy)$, two copies of $\cos(jy)$ for each j between 0 and s, and one copy of $\cos(0y)$; $g_2(y)$ consists of 8 copies of $g_1(y)$, with some shifted so as to introduce sine components, but these are canceled out in the formation of $g_3^i(y)$, which is a symmetric function for each i. Thus, together with the normalization

by $1/16s$, each $g_3^i(y)$ may be regarded as a linear combination $\sum_{j=0}^{s} \cos(yj)b_{ij}$ where the sth term has coefficient at most $1/s$, and all the remaining terms have coefficients at most $2/s$.

Next, under the coordinate transformation $x = \frac{2s}{k}(1 - \cos(y))$, the function $\cos(yj)$ becomes the Chebyshev polynomial $T_j(1 - \frac{xk}{2s})$. We note that each term $\alpha_\ell (xk)^\ell$ from this polynomial will ultimately be multiplied by $\sum_{m=0}^{s-1} poi(xk, m)$. We reexpress this as

$$x^\ell \sum_{m=0}^{s-1} \frac{x^m e^{-x}}{m!} = \sum_{m=\ell}^{s+\ell-1} poi(xk, m) \frac{m!}{(m-\ell)!}.$$

We have thus expressed our function as a linear combination of Poisson functions. As we aim to bound the sum of the coefficients of these Poisson functions, we consider this now: $\sum_{m=\ell}^{s+\ell-1} \frac{m!}{(m-\ell)!}$ which we note equals $\frac{1}{l+1}\frac{(s+\ell)!}{s!}$ since, in general, $\sum_{m=i}^{j} \binom{m}{i} = \binom{j+1}{i+1}$. Expressing $T_j(z)$ as $\sum_{i=0}^{j} \beta_{ij} z^i$, we note that, since we evaluate Chebyshev polynomials at $1 - \frac{xk}{2s}$, a term $\beta_{ij} z^i$ becomes $\beta_{ij} \sum_{\ell=0}^{i} \binom{i}{\ell} \frac{1}{(2s)^\ell} x^\ell$, which, by the previous calculation, contributes $\beta_{ij} \sum_{\ell=0}^{i} \binom{i}{\ell} \frac{1}{(2s)^\ell} \frac{1}{l+1} \frac{(s+\ell)!}{s!}$ to the total Poisson coefficients. Since $\ell \le i \le s$, we have $s + \ell \le 2s$, from which we see $\frac{1}{(2s)^\ell} \frac{(s+\ell)!}{s!} \le 1$. We thus bound $\beta_{ij} \sum_{\ell=0}^{i} \binom{i}{\ell} \frac{1}{(2s)^\ell} \frac{1}{l+1} \frac{(s+\ell)!}{s!} \le \beta_{ij} \sum_{\ell=0}^{i} \binom{i}{\ell} = \beta_{ij} 2^i$.

We thus desire, for any $j \le s$, to bound $\sum_{i=0}^{j} \beta_{ij} 2^i$, where β_{ij} are the coefficients of the jth Chebyshev polynomial. Chebyshev polynomials have coefficients whose signs repeat in the pattern $(+, 0, -, 0)$, thus we can evaluate this sum exactly as $|T_j(2\mathbf{i})|$, for $\mathbf{i} = \sqrt{-1}$. Explicitly,

$$|T_j(2\mathbf{i})| = \frac{1}{2}\left[(2 - \sqrt{5})^j + (2 + \sqrt{5})^j\right] \le (2 + \sqrt{5})^j.$$

Since, as we showed above, in each $g_3^i(y)$ the coefficient of each $\cos(jy)$ term is at most $2/s$, and thus our final bound on the sum of Poisson coefficients is at most $2(2 + \sqrt{5})^s < 2e^{\frac{3}{2}s} = 2k^{0.3}$. □

Lemma 3.20. *For any x*

$$\sum_{i=-s+1}^{s} g_2(x + \frac{\pi i}{s}) = 1, \text{ and } \sum_{i=0}^{\infty} f_i(x) = 1.$$

Proof. $g_2(y)$ is a linear combination of cosines at integer frequencies j, for $j = 0, \ldots, s$, shifted by $\pm\pi/2s$ and $\pm 3\pi/s2$. Since $\sum_{i=-s+1}^{s} g_2(x + \frac{\pi i}{s})$ sums these cosines over all possible multiples of π/s, we note that all but the frequency 0 terms will cancel. The $\cos(0y) = 1$ term will show up once in each g_1 term, and thus $1 + 3 + 3 + 1 = 8$ times in each g_2 term, and thus $8 \cdot 2s$ times in the sum in question. Together with the normalizing factor of $16s$, the total sum is thus 1, as claimed.

For the second part of the claim,

$$\begin{aligned}\sum_{i=0}^{\infty} f_i(x) &= \left(\sum_{j=-s+1}^{s} g_2(\cos^{-1}\left(\frac{xk}{2s}-1\right)+\frac{\pi j}{s})\right)\sum_{j=1}^{s-1} poi(xk,j)+\sum_{j\geq s} poi(xk,j)\\ &= 1\cdot\sum_{j=1}^{s-1} poi(xk,j)+\sum_{j\geq s} poi(xk,j)=1.\end{aligned}$$

□

We now turn to the main thrust of the argument, showing that the scheme is $O(\sqrt{\delta})$-good, where $n=\delta k\log k$, and $\delta\geq\frac{1}{\log k}$; the following lemma, quantifying the "skinnyness" of the Chebyshev bumps is the cornerstone of this argument.

Lemma 3.21. *$|g_2(y)|\leq\frac{285}{y^4s^4}$ for $y\in[-\pi,\pi]\setminus(-3\pi/s,3\pi/s)$, and $|g_2(y)|\leq 1/2$ everywhere.*

Proof. Since $g_1(y)=\sum_{j=-s}^{s-1}\cos jy=\sin(sy)\cot(y/2)$, and since $\sin(\alpha+\pi)=-\sin(\alpha)$, we have the following:

$$\begin{aligned}g_2(y) &= \frac{1}{16s}\left(g_1(y-\frac{3\pi}{2s})+3g_1(y-\frac{\pi}{2s})+3g_1(y+\frac{\pi}{2s})+g_1(y+\frac{3\pi}{2s})\right)\\ &= \frac{1}{16s}\left(\sin(ys+\pi/2)\left(\cot(\frac{y}{2}-\frac{3\pi}{4s})-3\cot(\frac{y}{2}-\frac{\pi}{4s})\right.\right.\\ &\qquad\left.\left.+3\cot(\frac{y}{2}+\frac{\pi}{4s})-\cot(\frac{y}{2}+\frac{3\pi}{4s})\right)\right).\end{aligned}$$

Note that $\left(\cot(\frac{y}{2}-\frac{3\pi}{4s})-3\cot(\frac{y}{2}-\frac{\pi}{4s})+3\cot(\frac{y}{2}+\frac{\pi}{4s})-\cot(\frac{y}{2}+\frac{3\pi}{4s})\right)$ is bounded in magnitude by $(\pi/2s)^3$ times the maximum magnitude of $\frac{d^3}{dx^3}\cot(x/2)$ in the range $x\in[y-3\pi/2s, y+3\pi/2s]$. Since the magnitude of this third derivative is decreasing for $x\in(0,2\pi)$, we can simply evaluate the magnitude of this derivative at $y-3\pi/2s$. We thus have $\frac{d^3}{dx^3}\cot(x/2)=\frac{-(2+\cos(x))}{4\sin^4(x/2)}$, whose magnitude is at most $\frac{3}{4(x/\pi)^4}$ for $x\in(0,\pi]$. For $y\in[3\pi/s,\pi]$, we trivially have that $y/2\leq y-3\pi/2s$, and thus we have the following bound:

$$|\cot(\frac{y}{2}-\frac{3\pi}{4s})-3\cot(\frac{y}{2}-\frac{\pi}{4s})+3\cot(\frac{y}{2}+\frac{\pi}{4s})-\cot(\frac{y}{2}+\frac{3\pi}{4s})|\leq\left(\frac{\pi}{2s}\right)^3\frac{3}{4(y/2\pi)^4}\leq\frac{3\pi^7}{2y^4s^3}.$$

Since $g_2(y)$ is a symmetric function, the same bound holds for $y\in[-\pi,-3\pi/s]$. Thus $|g_2(y)|\leq\frac{3\pi^7}{16s\cdot 2y^4s^3}<\frac{285}{y^4s^4}$ for $y\in[-\pi,\pi]\setminus(-3\pi/s,3\pi/s)$. To conclude, note that $g_2(y)$ attains a global maximum at $y=0$, with $g_2(0)=\frac{1}{16s}\left(6\cot(\pi/4s)-2\cot(3\pi/4s)\right)\leq\frac{1}{16s}\frac{24s}{\pi}<1/2$. □

Lemma 3.22. *The Chebyshev earthmoving scheme of Definition 3.18 is $O(\sqrt{\delta})$-good, where $n=\delta k\log k$, and $\delta\geq\frac{1}{\log k}$.*

Proof. We split this proof into two parts: first we will consider the cost of the portion of the scheme associated with all but the first $s+1$ bumps, and then we consider the cost of the skinny bumps f_i with $i \in \{0, \ldots, s\}$.

For the first part, we consider the cost of bumps f_i for $i \geq s+1$; that is the relative earthmover cost of moving $poi(xk, i)$ mass from x to $\frac{i}{k}$, summed over $i \geq s$. By definition of relative earthmover distance, the cost of moving mass from x to $\frac{i}{k}$ is $|\log \frac{xk}{i}|$, which, since $\log y \leq y - 1$, we bound by $\frac{xk}{i} - 1$ when $i < xk$ and $\frac{i}{xk} - 1$ otherwise. We thus split the sum into two parts.

For $i \geq \lceil xk \rceil$ we have $poi(xk, i)(\frac{i}{xk} - 1) = poi(xk, i-1) - poi(xk, i)$. This expression telescopes when summed over $i \geq \max\{s, \lceil xk \rceil\}$ to yield $poi(xk, \max\{s, \lceil xk \rceil\} - 1) = O(\frac{1}{\sqrt{s}})$.

For $i \leq \lceil xk \rceil - 1$ we have, since $i \geq s$, that $poi(xk, i)(\frac{xk}{i} - 1) \leq poi(xk, i)((1 + \frac{1}{s})\frac{xk}{i+1} - 1) = (1 + \frac{1}{s})poi(xk, i+1) - poi(xk, i)$. The $\frac{1}{s}$ term sums to at most $\frac{1}{s}$, and the rest telescopes to $poi(xk, \lceil xk \rceil) - poi(xk, s) = O(\frac{1}{\sqrt{s}})$. Thus in total, f_i for $i \geq s+1$ contributes $O(\frac{1}{\sqrt{s}})$ to the relative earthmover cost, per unit of weight moved.

We now turn to the skinny bumps $f_i(x)$ for $i \leq s$. The simplest case is when x is outside the region that corresponds to the cosine of a real number — that is, when $xk \geq 4s$. It is straightforward to show that $f_i(x)$ is very small in this region. We note the general expression for Chebyshev polynomials: $T_j(x) = \frac{1}{2}\left[(x - \sqrt{x^2-1})^j + (x + \sqrt{x^2-1})^j\right]$, whose magnitude we bound by $|2x|^j$. Further, since $2x \leq \frac{2}{e}e^x$, we bound this by $(\frac{2}{e})^j e^{|x|j}$, which we apply when $|x| > 1$. Recall the definition $f_i(x) = t_i(1 - \frac{xk}{2s}) \sum_{j=0}^{s-1} poi(xk, j)$, where t_i is the polynomial defined so that $t_i(\cos(y)) = g_3^i(y)$, that is, t_i is a linear combination of Chebyshev polynomials of degree at most s and with coefficients summing in magnitude to at most 2, as was shown in the proof of Lemma 3.19. Since $xk > s$, we may bound $\sum_{j=0}^{s-1} poi(xk, j) \leq s \cdot poi(xk, s)$. Further, since $z \leq e^{z-1}$ for all z, letting $z = \frac{x}{4s}$ yields $x \leq 4s \cdot e^{\frac{x}{4s}-1}$, from which we may bound $poi(xk, s) = \frac{(xk)^s e^{-xk}}{s!} \leq \frac{e^{-xk}}{s!}(4s \cdot e^{\frac{xk}{4s}-1})^s = \frac{4^s s^s}{e^s \cdot e^{3xk/4} s!} \leq 4^s e^{-3xk/4}$. We combine this with the above bound on the magnitude of Chebyshev polynomials, $T_j(z) \leq (\frac{2}{e})^j e^{|z|j} \leq (\frac{2}{e})^s e^{|z|s}$, where $z = (1 - \frac{xk}{2s})$ yields $T_j(z) \leq (\frac{2}{e^2})^s e^{\frac{xk}{2}}$. Thus $f_i(x) \leq poly(s) 4^s e^{-3xk/4} (\frac{2}{e^2})^s e^{\frac{xk}{2}} = poly(s)(\frac{8}{e^2})^s e^{-\frac{xk}{4}}$. Since $\frac{xk}{4} \geq s$ in this case, f_i is exponentially small in both x and s; the total cost of this earthmoving scheme, per unit of mass above $\frac{4s}{k}$ is obtained by multiplying this by the logarithmic relative distance the mass has to move, and summing over the $s+1$ values of $i \leq s$, and thus remains exponentially small, and is thus trivially bounded by $O(\frac{1}{\sqrt{s}})$.

To bound the cost in the remaining case, when $xk \leq 4s$ and $i \leq s$, we work with the trigonometric functions g_3^i, instead of t_i directly. For $y \in (0, \pi]$, we seek to bound the per-unit-mass relative earthmover cost of, for each $i \geq 0$, moving $g_3^i(y)$ mass from $\frac{2s}{k}(1 - \cos(y))$ to c_i. For $i \geq 1$, this contribution is at most

$$\sum_{i=1}^{s} |g_3^i(y)(\log(1 - \cos(y)) - \log(1 - \cos(\frac{i\pi}{s}))|.$$

To simplify the analysis, we compare $\log(1-\cos(y))$ with $2\log y$ when $y \in (0,\pi]$, noting that their derivatives respectively are $\frac{\sin(y)}{1-\cos(y)}$ and $\frac{2}{y}$, and we claim that the second expression is always greater. To compare the two expressions, cross-multiply and take the difference, to yield $y\sin y - 2 + 2\cos y$, which we show is always at most 0 by noting that it is 0 when $y = 0$ and has derivative $y\cos y - \sin y$, which is negative since $\cot y \leq \frac{1}{y}$. Thus we have that $|\log(1-\cos(y)) - \log(1-\cos(\frac{i\pi}{s}))| \leq 2|\log y - \log\frac{i\pi}{s}|$; we use this bound in all but the last step of the analysis. Additionally, we ignore the $\sum_{j=0}^{s-1} poi(xk,j)$ term as it is always at most 1.

Case 1: $y \geq \frac{\pi}{s}$.

Not that for such y, the contribution of f_0, c_0 to the relative earthmover cost is bounded by the contribution of f_1, c_1, thus it suffices to bound $\sum_{i=1}^{s} |g_3^i(y)(\log y - \log\frac{i\pi}{s})|$. For i such that $y \in (\frac{(i-3)\pi}{s}, \frac{(i+3)\pi}{s})$, by the second bounds on $|g_2|$ in the statement of Lemma 3.21, $g_3^i(y) < 1$, and for such i, $|(\log y - \log\frac{i\pi}{s})| < \frac{1}{sy}$, to yield a bound of $\frac{1}{sy}$.

For the contribution from i such that $y \leq \frac{(i-3)\pi}{s}$ or $y \geq \frac{(i-3)\pi}{s}$, the first bound of Lemma 3.21 yields $|g_3^i(y)| = O(\frac{1}{(ys-i\pi)^4})$. We split up our sum over $i \in [s] \setminus [\frac{ys}{\pi} - 3, \frac{ys}{\pi} + 3]$ into two parts according to whether $i > ys/\pi$:

$$\begin{aligned}\sum_{i \geq \frac{ys}{\pi}+3}^{s} \frac{1}{(ys-i\pi)^4}|(\log y - \log\frac{i\pi}{s})| &\leq \sum_{i \geq \frac{ys}{\pi}+3}^{\infty} \frac{\pi^4}{(\frac{ys}{\pi}-i)^4}(\log i - \log\frac{ys}{\pi}) \\ &\leq \pi^4 \int_{w=\frac{ys}{\pi}+2}^{\infty} \frac{1}{(\frac{ys}{\pi}-w)^4}(\log w - \log\frac{ys}{\pi}). \quad (3.2)\end{aligned}$$

Since the antiderivative of $\frac{1}{(\alpha-w)^4}(\log w - \log\alpha)$ with respect to w is

$$\frac{-2w(w^2 - 3w\alpha + 3\alpha^2)\log w + 2(w-\alpha)^3\log(w-\alpha) + \alpha(2w^2 - 5w\alpha + 3\alpha^2 + 2\alpha^2\log\alpha)}{6(w-\alpha)^3\alpha^3},$$

the quantity in Equation 3.2 is equal to the above expression evaluated with $\alpha = \frac{ys}{\pi}$, and $w = \alpha + 2$, to yield

$$O(\frac{1}{ys}) - \log\frac{ys}{\pi} + \log(2 + \frac{ys}{\pi}) = O(\frac{1}{ys}).$$

A nearly identical argument applies to the portion of the sum for $i \leq \frac{ys}{\pi} + 3$, yielding the same asymptotic bound of $O(\frac{1}{ys})$.

Case 2: $\frac{ys}{\pi} < 1$.

The per-unit mass contribution from the 0th bump is trivially at most $|g_3^0(y)(\log\frac{ys}{\pi} - \log 1)| \leq \log\frac{ys}{\pi}$. The remaining relative earthmover cost is $\sum_{i=1}^{s} |g_3^i(y)(\log\frac{ys}{\pi} - \log i)|$. To bound this sum, we note that $\log i \geq 0$, and $\log\frac{ys}{\pi} \leq 0$ in this region, and thus split the above sum into the corresponding two parts, and bound them separately. By Lemma 3.21, we have:

$$\sum_{i=1}^{s} g_3^i(y)\log i \leq O\left(1 + \sum_{i=3}^{\infty} \frac{\log i}{\pi^4(i-1)^4}\right) = O(1).$$

$$\sum_{i=1}^{s} g_3^i(y) \log \frac{ys}{\pi} \leq O(\log ys) \leq O(\frac{1}{ys}),$$

since for $ys \leq \pi$, we have $|\log ys| < 4/ys$.

Having concluded the case analysis, recall that we have been using the change of variables $x = \frac{2s}{k}(1 - \cos(y))$. Since $1 - \cos(y) = O(y^2)$, we have $xk = O(sy^2)$. Thus the case analysis yielded a bound of $O(\frac{1}{ys})$, which we may thus express as $O(\frac{1}{\sqrt{sxk}})$.

For a distribution with histogram h, the cost of moving earth on this region, for bumps f_i where $i \leq s$ is thus $O(\sum_{x:h(x)\neq 0} h(x) \cdot x \cdot \frac{1}{\sqrt{sxk}})$. Because $\frac{1}{z}$ is a decreasing function, for $x \geq \frac{1}{n} = \frac{1}{5\delta sk}$ we have that $\min_{x \geq 1/n} \left(\frac{1}{\sqrt{sxk}}\right) = \sqrt{\frac{5\delta sk}{sk}} = O(\sqrt{\delta})$. Since $\sum_{x:h(x)\neq 0} h(x) \cdot x = 1$, the total relative earthmover cost from the mass of the histogram for $x \geq 1/n$ is thus bounded by $O(\sqrt{\delta})$, as desired. We now consider the portion of the cost from histogram entries $h(x)$ for $x \leq 1/n$. While the per-unit mass cost of this region goes to infinity as $x \to 0$, the bound on the histogram support allows one to still bound this contribution. Formally, since $x \cdot \frac{1}{\sqrt{sxk}}$ is an increasing function, it is maximized when $x = \frac{1}{n}$. Since the remaining term, $h(x)$, sums to at most n, by assumption, we thus have

$$\sum_{x \leq \frac{1}{n}: h(x)\neq 0} h(x) \cdot x \cdot \frac{1}{\sqrt{sxk}} \leq n\frac{1}{n} \cdot \frac{n}{\sqrt{sk}} = O(\sqrt{\delta}),$$

as above. As we have already bounded the relative earthmover cost for bumps f_i for $i > s$ at least this tightly, this concludes the proof. □

We are now equipped to prove Theorem 3.1.

Proof of Theorem 3.1. Let h, g be two generalized histograms corresponding to feasible points of Linear Program 3.9 that each have objective function value at most α. Specifically, h, g are obtained by appending the empirical distribution of the fingerprint to feasible points of the LP. Let h', g' be the generalized histograms that result from applying the Chebyshev earth-moving scheme of Definition 3.18 to h and g, respectively. By Lemma 3.22, $R(h, h') = O(\sqrt{\delta})$, and $R(g, g') = O(\sqrt{\delta})$. We now consider the discrepancy between h' and g'.

By definition, h', g' are generalized histograms supported at the bump centers c_i. Since all but the first $s + 1$ bumps are simply the standard Poisson bumps $f_i(x) = poi(xk, i)$, for $i > s$, we have

$$\begin{aligned} |h'(c_i) - g'(c_i)| &= \left| \sum_{x:h(x)+g(x)\neq 0} (h(x) - g(x)) x poi(kx, i-1) \right| \\ &= \left| \sum_{x:h(x)+g(x)\neq 0} (h(x) - g(x)) poi(kx, i) \frac{i}{k} \right|. \end{aligned}$$

Since $h(x) = g(x)$ for all $x > \frac{c+k^{\mathcal{C}}/2}{k}$, by the tail bounds for Poissons, the total relative earthmover cost of equalizing h' and g' for all bump centers c_i with $i > c + k^{\mathcal{C}}$ is trivially bounded by $o(\frac{\log k}{k})$.

Next, we consider the contribution of the discrepancies in the Poisson bumps with centers c_i for $i \in [s+1, c+k^{\mathcal{C}}]$. Since $\sum_{i \leq c} poi(kx, i) = o(1/k)$ for $x > \frac{c+k^{\mathcal{C}}}{k}$, the fact the empirical fingerprints are appended to the LP points has negligible effect on these fingerprints, and hence the assumption on the LP objective function values corresponding to h, g imply that for $i \leq c$, $\left|\sum_{x:h(x)+g(x)\neq 0}(h(x) - g(x))poi(kx, i)\frac{i}{k}\right| \leq \alpha\frac{c}{k}$. We now argue that the contribution from the bump centers in the intermediate range $i \in [c, c+k^{\mathcal{C}}]$ is also small, by virtue of there being little probability mass in this region in the empirical fingerprints. Again, because $\sum_{i \geq c+\frac{3}{4}k^{\mathcal{C}}} poi(kx, i) = o(1/k)$ if $x \leq \frac{c+k^{\mathcal{C}}/2}{k}$, and thus since $h(x) = g(x)$ for all such x, the discrepancy in fingerprints for $i \geq c + \frac{3}{4}k^{\mathcal{C}}$ is negligible, contributing $o(1/k)$ to the relative earthmover distance between h' and g'. We now argue that

$$\sum_{i \in [c, c+\frac{3}{4}k^{\mathcal{C}}]} \sum_{x:h(x)\neq 0} h(x)ipoi(xk, i) \leq 2k^{1-\mathcal{B}+\mathcal{C}},$$

and similarly for g. Indeed, if this were not the case, then

$$\sum_{i \leq c} \left(i\mathcal{F}_i - \sum_{x:h(x)\neq 0} h(x)ipoi(xk, i) \right) \geq k^{1-\mathcal{B}+\mathcal{C}} - o(1),$$

contradicting the assumption that the objective value corresponding to h is at most $\alpha < k^{1-\mathcal{B}+\mathcal{C}} - o(1)$. The analogous statement holds for g. Hence the relative earthmover cost associated with bump centers in the range $[c, c+k^{\mathcal{C}}]$ is at most $O(k^{-\mathcal{B}+\mathcal{C}} \log k)$.

Finally, we consider the contribution of the discrepancies in the first $s + 1 = O(\log k)$ centers, corresponding to the skinny Chebyshev bumps. Note that for such centers, c_i, by definition the corresponding functions $f_i(x) = \sum_{j \geq 0} a_{ij} poi(xk, j)$, for some coefficients a_{ij}, where $\sum_{j \geq 0} \alpha_{ij} \leq \beta$. Thus we have the following, where $\sum_x$ is shorthand for $\sum_{x:h(x)+g(x)\neq 0}$:

$$\begin{aligned}
|h'(c_i) - g'(c_i)| &= \left| \sum_x (h(x) - g(x)) x f_i(x) \right| \\
&= \left| \sum_x (h(x) - g(x)) x \sum_{j \geq 0} a_{ij} poi(xk, j) \right| \\
&= \left| \sum_{j \geq 0} a_{ij} \sum_x (h(x) - g(x)) x poi(xk, j) \right| \\
&= \left| \sum_{j \geq 1} a_{ij} \frac{j}{k} \sum_x (h(x) - g(x)) poi(xk, j) \right|.
\end{aligned}$$

Since $a_{ij} = 0$ for $j > \log k$, since each Chebyshev bump is a linear combination of only the first $2s < \log k$ Poisson functions, the total cost of equalizing h' and g' at each of these Chebyshev bump centers is bounded by

$$\beta \left| \sum_{i=1}^{\log k} \frac{i}{k} \sum_x (h(x) - g(x)) poi(xk, j) \right| |\log c_0| \log k \leq \beta\alpha \frac{\text{polylog}\, k}{k}$$

where the $|\log c_0|$ term is a crude upper bound on the per-unit mass relative earthmover cost of moving the mass to a probability above c_0, and the final factor of $\log k$ is because there are at most $s < \log k$ centers corresponding to "skinny" bumps.

We now plug in the bound of $\beta = O(k^{0.3})$ of Lemma 3.19, and the bound on the objective function value $\alpha = O(k^{\frac{1}{2}+\mathcal{B}+\mathcal{D}})$ given by the feasible point v' constructed in Lemma 3.14 whose corresponding generalized histogram $h_{v'}$ is close to the true distribution from which the sample was drawn. Trivially, this upper bounds the objective function value of the actual solution to the linear program used by Algorithm 3.8 to construct the returned generalized histogram, g_{LP}. Letting $g'_{LP}, h'_{v'}$ denote the generalized histograms resulting from applying the Chebyshev earthmoving scheme to g_{LP} and $h_{v'}$, respectively, we thus have

$$R(g'_{LP}, h'_{v'}) = O(k^{0.3} k^{\frac{1}{2}+\mathcal{B}+\mathcal{D}} \frac{\text{polylog}\, k}{k}) = O(\frac{1}{k^{\Omega(1)}}).$$

Hence, by the triangle inequality, letting h denote the histogram of the distribution of support at most $n = \delta k \log k$, from which the sample was drawn, we have

$$\begin{aligned} R(g_{LP}, h) &\leq R(h, h_{v'}) + R(g_{LP}, g'_{LP}) + R(h_{v'}, h'_{v'}) + R(g'_{LP}, h'_{v'}) \\ &= O(\sqrt{\delta} + \frac{1}{k^{\Omega(1)}}) = O(\sqrt{\delta}), \end{aligned}$$

since, by assumption, $\delta > \frac{1}{\log k}$. □

Reducing the Runtime

In this section we briefly outline the modifications required to reduce the number of variables and constraints in the linear program to $O(k^{0.1})$. Given so few variables and constraints, we could then use Karmarkar's algorithm which solves linear programs with m variables and L bits of input in time $O(n^{3.5} L^2 \text{polylog}\, L)$ [78]. Thus the total runtime of our algorithm will be *linear* in the sample size, k. This shows that our estimators are essentially optimal in both their sample complexity and computational complexity.

To decrease the number of variables in the linear programming, we must create a more coarse mesh than the extremely fine set $X = \{\frac{1}{nk}, \frac{2}{nk}, \ldots\}$ employed above. The only step in the above proof that depends crucially on the size of this mesh, is the construction of the feasible point of Linear Program 3.9 with low objective function, described in Lemma 3.14. Recall that in this construction, we started with the true histogram, h, and obtained the

feasible point of the linear program by taking the "low probability" region of h, and rounding each nonzero histogram entry to the closest point in the mesh of probabilities X. We then argued that because the mesh is sufficiently fine, this rounding of the support of h does not alter the expected fingerprint entries, and hence the objective function value will be small whenever the fingerprint is "faithful" (i.e. close to the expected fingerprint values of the actual distribution).

In order to make similar guarantees with a significantly more coarse set of grid-points, we will choose the locations of the points more delicately, and perform a tighter analysis. In particular, we will choose the set X to be a *quadratically* spaced mesh of points $X = \{x_1, \ldots, x_{k^{0.1}}\}$, where $x_i := \frac{i^2}{k^{1.1}}$. Rather than constructing our feasible point of low objective function value by simply rounding the support of the histogram h to the closest point in X as was done in the proof of Lemma 3.14, we instead linearly partition the mass at $h(y)$ between the points $x_i, x_{i+1} \in X$ such that $x_i \leq y \leq x_{i+1}$.

Formally, the feasible point v' of the linear program corresponding to Linear Program 3.9 in which the variables are indexed by the coarser mesh X is defined as follows, as a function of the true histogram h and fingerprint of the sample, $\mathcal{F}$:

- Define h' such that $h'(x) = h(x)$ for all $x \leq \frac{c+k^{c}/2}{k}$, and $h'(x) = 0$ for all $x > \frac{c+k^{c}/2}{k}$, where c is chosen appropriately.
- Initialize v' to be 0, and for each y s.t. $h'(y) \neq 0$ choose i such that $x_i \leq y \leq x_{i+1}$ for $x_i, x_{i+1} \in X$. If $y < x_1$, we consider an $x_0 = 0$ and let $i = 0$ (though we do not return any v'_0 associated to x_0.)
- Modify v' by increasing $v'_{x_i} \leftarrow v'_{x_i} + h(y)\frac{x_{i+1}-y}{x_{i+1}-x_i}$, and increasing $v'_{i+1} \leftarrow v'_{i+1} + h(y)\frac{y-x_i}{x_{i+1}-x_i}$.
- Let $m := \sum_{x \in X} x v'_x + m_{\mathcal{F}}$, where $m_{\mathcal{F}} := \sum_{i \geq \frac{c+k^{c}}{k}} \frac{i}{k}\mathcal{F}_i$. If $m < 1$, increment v'_y by $(1-m)/y$, where $y = \frac{c+k^{c}/2}{k}$. Otherwise, if $m \geq 1$, decrease v' arbitrarily until the total probability mass $m_{\mathcal{F}} + \sum_{x \in X} x v'_x = 1$.

We now argue that the above interpolative discretization of the support of h does not significantly alter the expected fingerprints, and thus for $i \leq c$, $\sum_x poi(kx, i)h(x) \approx \sum_{x \in X} poi(kx, i)v'_x$. Consider v' as it is at the end of the third step: for each y such that $h(y) > 0$, we have "replaced" value $h(y)$ at probability y with the pair of values $h(y)\frac{x_{i+1}-y}{x_{i+1}-x_i}, h(y)\frac{y-x_i}{x_{i+1}-x_i}$ respectively at the corresponding discretized probabilities x_i and x_{i+1}, and we aim to bound

$$h(y)\left|\left(\frac{x_{i+1}-y}{x_{i+1}-x_i}poi(x_i k, j) + \frac{y-x_i}{x_{i+1}-x_i}poi(x_{i+1}k, j)\right) - poi(yk, j)\right| \tag{3.3}$$

We note the basic calculus fact that for an arbitrary twice-differentiable function $g : \mathbb{R} \to \mathbb{R}$ and real numbers $a < y < b$, the linear interpolation $\frac{b-y}{b-a}g(a) + \frac{y-a}{b-a}g(b)$ approximates $g(y)$ to within $\frac{1}{8}(b-a)^2 \max_{z \in [a,b]} |g''(z)|$. Thus Equation 3.3 is bounded by $h(y)\frac{1}{8}(x_{i+1} -$

$x_i)^2 \max_{z\in[x_i,x_{i+1}]} \left|\frac{d^2}{dz^2} poi(zk,j)\right|$. By Proposition A.18 we see that $\left|\frac{d^2}{dz^2} poi(zk,j)\right| \leq 2k^2 \min\{1, \frac{1}{zk}\}$, yielding a bound on Equation 3.3 of

$$h(y)\frac{k^2}{4}(x_{i+1}-x_i)^2 \min\{1, \frac{1}{x_i k}\}.$$

Thus we have

$$\begin{aligned}
\max_{i:x_i\leq 1/k}\left(\frac{k^2}{4}(x_{i+1}-x_i)^2\min\{1,\frac{1}{x_ik}\}\right) &= \max_{i:x_i\leq 1/k}\left(\frac{k^2}{4}(x_{i+1}-x_i)^2\right)\\
&\leq \frac{k^2}{4}(\frac{2k^{0.05}+1}{k^{1.1}})^2\\
&\leq 2k^{-0.1}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\max_{i:x_i\geq 1/k}\left(\frac{k^2}{4}(x_{i+1}-x_i)^2\min\{1,\frac{1}{x_ik}\}\right) &= \max_{i:x_i\geq 1/k}\left(\frac{k^2}{4x_i}(x_{i+1}-x_i)^2\right)\\
&\leq \frac{k^{-0.1}(2i+1)^2}{4i^2}\\
&\leq 2k^{-0.1}.
\end{aligned}$$

The remainder of the proof of Theorem 3.1 will proceed as in the previous two sections, with the very minor modification that one will need to pick the constants $\mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ such that $\mathcal{B} < k^{0.1}$ which ensures that the number of linear program constraints is also at most $2k^{\mathcal{B}} = O(k^{0.1})$, and also that the objective function value of v' is at most $O(k^{\mathcal{B}}k^{-0.1}) = O(\frac{1}{k^{\Omega(1)}})$,

3.3 Properties of Pairs of Distributions

Perhaps unsurprisingly, our general approach for constructing constant-factor optimal estimators for symmetric properties of distributions can also be extended to yield constant-factor optimal estimators for symmetric properties of *pairs* of distributions, including total variational distance (ℓ_1 distance).

For properties of pairs of distributions, one is given as input two samples, one drawn from the first distribution, and one drawn from the second distribution. In analogy with the analysis of estimators for properties of a single distribution, we begin by extending our definitions of *fingerprints* and *histograms* to this two-distribution setting.

Definition 3.23. *The* fingerprint $\mathcal{F}$ *of a sample of size k_1 from distribution p_1 and a sample of size k_2 from distribution p_2 is a $k_1 \times k_2$ matrix, whose entry $\mathcal{F}(i,j)$ is given by the number of domain elements that are seen exactly i times in the sample from p_1 and exactly j times in the sample from p_2.*

Definition 3.24. *The* histogram $h_{p_1,p_2} : [0,1]^2 \setminus \{(0,0)\} \to \mathbb{N} \cup 0$ *of a pair of distributions* p_1, p_2 *is defined by letting* $h_{p_1,p_2}(x,y)$ *be the number of domain elements that occur with probability* x *in distribution* p_1 *and probability* y *in distribution* p_2*.*

Thus in any two-dimensional histogram h corresponding to a pair of distributions, we have

$$\sum_{x,y:h(x,y)\neq 0} x \cdot h(x,y) = \sum_{x,y:h(x,y)\neq 0} y \cdot h(x,y) = 1.$$

In our analysis, it will prove convenient to also consider "generalized histograms" in which the entries need not be integral, and for which the "probability masses" $\sum_{x,y:h(x,y)\neq 0} x \cdot h(x,y)$ and $\sum_{x,y:h(x,y)\neq 0} y \cdot h(x,y)$ do not necessarily equal 1.

As in the case with symmetric properties of single distributions, symmetric properties of pairs of distributions are functions of only the histogram of the pair of distributions, and given any estimator that takes as input the actual pair of samples, there is an estimator of equivalent performance that takes as input the fingerprint $\mathcal{F}$ derived from such a pair of samples.

Many distance metrics, including total variational distance and Kullback–Leibler divergence are symmetric properties of pairs of distributions:

Example 3.25. *Consider pair of distributions* p_1, p_2 *with histogram* h*:*

- *The total variational distance (*ℓ_1 *distance) is given by*

$$D_{tv}(p_1, p_2) = \frac{1}{2} \sum_{(x,y):h(x,y)\neq 0} h(x,y) \cdot |x - y|.$$

- *The Kullback–Leibler divergence is given by*

$$D_{KL}(p_1||p_2) = \sum_{(x,y):h(x,y)\neq 0} h(x,y) \cdot x \log \frac{x}{y}.$$

We will use the following two-dimensional earthmover metric on the set of two-dimensional generalized histograms. Note that it does not make sense to define a strict analog of the relative earthmover distance of Definition 3.1, since a given histogram entry $h(x,y)$ does not correspond to a single quantity of probability mass—it corresponds to $xh(x,y)$ mass in one distribution, and $yh(x,y)$ mass in the other distribution. Thus the following metric is in terms of moving *histogram entries* rather than probability mass.

Definition 3.26. *Given two two-dimensional generalized histograms* h_1, h_2*, their* histogram distance*, denoted* $W(h_1, h_2)$*, is defined to be the minimum over all schemes of moving the histogram values in* h_1 *to yield* h_2*, where the cost of moving histogram value* c *at location*

x, y to location x', y' is $c\,(|x - x'| + |y - y'|)$. To ensure that such a scheme always exists, in the case that $\sum_{x,y:x+y>0} h_1(x,y) < \sum_{x,y:x+y>0} h_2(x,y)$, one proceeds as if

$$h_1(0,0) = \sum_{x,y:x+y>0} h_2(x,y) - \sum_{x,y:x+y>0} h_1(x,y),$$

and analogously for the case in which h_2 contains fewer histogram entries.

We provide an example of the above definitions:

Example 3.27. *Define distributions $p_1 = Unif[n]$, and $p_2 = Unif[n/2]$, where the $n/2$ support elements of distribution p_2 are contained in the support of n. The corresponding histogram h_{p_1,p_2}, is defined as $h_{p_1,p_2}(\frac{1}{n}, \frac{2}{n}) = \frac{n}{2}$, $h_{p_1,p_2}(\frac{1}{n}, 0) = \frac{n}{2}$, and $h_{p_1,p_2}(x,y) = 0$ for all other values of x, y.*

Considering a second pair of distribution, $q_1 = q_2 = Unif[n/4]$, with histogram $h_{q_1,q_2}(\frac{4}{n}, \frac{4}{n}) = \frac{n}{4}$, we have

$$\begin{aligned} W(h_{p_1,p_2}, h_{q_1,q_2}) &= \frac{n}{4}(|\frac{1}{n} - \frac{4}{n}| + |\frac{2}{n} - \frac{4}{n}|) + \frac{n}{4}(|\frac{1}{n} - 0| + |\frac{2}{n} - 0|) \\ &\qquad + \frac{n}{2}(|\frac{1}{n} - 0| + |0 - 0|) \\ &= \frac{5}{2}, \end{aligned}$$

since the optimal scheme is to move $n/4$ histogram entries in h_{p_1,p_2} from $(1/n, 2/n)$ to location $(4/n, 4/n)$, and all the remaining histogram entries must be moved to $(0,0)$ to yield histogram h_{q_1,q_2}.

We note that ℓ_1 distance is 1-Lipschitz with respect to the above distance metric:

Fact 3.28. *For any pair of two-dimensional generalized histograms, h, h'*

$$W(h,h') \geq \left| \sum_{x,y:h(x,y)\neq 0} h(x,y)|x-y| - \sum_{x,y:h'(x,y)\neq 0} h'(x,y)|x-y| \right|.$$

Hence if $h = h_{p_1,p_2}$ and $h' = h_{q_1,q_2}$ are histograms corresponding to pairs of distributions, $W(h_{p_1,p_2}, h_{q_1,q_2}) \geq |D_{tv}(p_1,p_2) - D_{tv}(q_1,q_2)|$.

We now formally define our algorithm. Both our algorithm, and its analysis closely parallel their analogs in the previous section. For simplicity, we restrict our attention to the setting in which one obtains samples of size k from both distributions—this approach extends in the obvious fashion to the setting in which one obtains samples of different sizes from the two distributions. As in the above section, we state the algorithm in terms of three positive constants, $\mathcal{B}, \mathcal{C}$, and $\mathcal{D}$, which can be defined arbitrarily provided $\mathcal{B} > \mathcal{C} > \frac{\mathcal{B}}{2}$, $\frac{\mathcal{B}}{2} > \mathcal{D}$, and $0.8 + 2\mathcal{B} + \mathcal{D} < 1$.

Algorithm 3.29. ESTIMATE UNSEEN–TWO DISTRIBUTIONS
Input: Two-dimensional fingerprint $\mathcal{F}$, derived from two samples of size k, an upper bound on the support sizes of the two distributions, n:
Output: Generalized two-dimensional histogram g_{LP}.

- *Let* $c_1 := \min\{i : i \in [k^{\mathcal{B}}, 2 \cdot k^{\mathcal{B}}]$ *and* $\sum_{j=i}^{i+k^{\mathcal{C}}} \sum_{\ell \geq 0} (j+\ell)\mathcal{F}(j,\ell) \leq 2k^{1-\mathcal{B}+\mathcal{C}}\}$.
- *Let* $c_2 := \min\{i : i \in [k^{\mathcal{B}}, 2 \cdot k^{\mathcal{B}}]$ *and* $\sum_{j=i}^{i+k^{\mathcal{C}}} \sum_{\ell \geq 0} (j+\ell)\mathcal{F}(\ell,j) \leq 2k^{1-\mathcal{B}+\mathcal{C}}\}$.
- *Let* $v = (\ldots, v_{x_i,y_j}, \ldots)$ *be the solution to Linear Program 3.9, on input* $\mathcal{F}, c_1, c_2,$ *and* n.
- *Let* g_{LP} *be the generalized histogram formed by setting* $g_{LP}(x_i, y_j) = v_{x_i,y_j}$ *for all* i, j, *and then for all pairs* i, j *with either* $i \geq c_1 + k^{\mathcal{C}}$ *or* $j \geq c_2 + k^{\mathcal{C}}$, *incrementing* $g_{LP}(\frac{i}{k}, \frac{j}{k})$ *by* $\mathcal{F}(i,j)$.

Linear Program 3.30.
Given a two-dimensional fingerprint $\mathcal{F}$, derived from two samples of size k, an upper bound on the support sizes of the two distributions, n, and two integers c_1, c_2:

- *Define the sets*

$$X := \{0, \frac{1}{nk}, \frac{2}{nk}, \ldots, \frac{c_1 + k^{\mathcal{C}}/2}{k}\}, \text{ and } Y := \{0, \frac{1}{nk}, \frac{2}{nk}, \ldots, \frac{c_2 + k^{\mathcal{C}}/2}{k}\}.$$

- *For each pair* $(x,y) \neq (0,0)$ *with* $x \in X$ *and* $y \in Y$ *define the associated LP variable* $v_{x,y}$.

The linear program is defined as follows:

$$\text{Minimize} \sum_{i \in [c_1], j \in [c_2] : i+j \neq 0} \left| \mathcal{F}(i,j) - \sum_{x \in X, y \in Y} poi(kx, i) poi(ky, j) v_{x,y} \right|,$$

Subject to:

- $\sum_{x \in X, y \in Y} x \cdot v_{x,y} + \sum_{i=c_1+k^{\mathcal{C}}}^{k} \sum_{j \geq 0} \frac{i}{k} \mathcal{F}(i,j) = 1$ *(prob. mass = 1.)*
- $\sum_{x \in X, y \in Y} y \cdot v_{x,y} + \sum_{j=c_2+k^{\mathcal{C}}}^{k} \sum_{i \geq 0} \frac{j}{k} \mathcal{F}(i,j) = 1$ *(prob. mass = 1.)*
- $\sum_{x \in X, y \in Y} v_{x,y} \leq 2(n+k)$ *(support size is not too big)*
- $\forall x \in X, y \in Y,\ v_{x,y} \geq 0$ *(histogram entries are non-negative)*

The following theorem describes the performance of the above algorithm. Together with Fact 3.28, this implies Theorem 3.2.

Theorem 3.3. *For any constant c, for sufficiently large n, given a sample of size $c\frac{n}{\log n}$ consisting of independent draws from two distributions, $p, q \in \mathcal{D}^n$ with two-dimensional*

histogram $h_{p,q}$, with probability at least $1 - e^{-n^{\Omega(1)}}$ over the randomness in the selection of the sample, Algorithm 3.29 returns a two-dimensional generalized histogram g_{LP} such that

$$W(g_{LP}, h_{p,q}) \leq O\left(\frac{1}{\sqrt{c}}\right).$$

The structure of the proof of the above theorem is identical to that of its one-distribution analog, Theorem 3.1. The details differ slightly, as we will need to define two-dimensional analogs of the Chebyshev bumps of Definition 3.18, though we can reuse much of the same machinery. The second difference between the above theorem, and Theorem 3.1 is in terms of the distance metric. In the one-distribution setting, we used relative earthmover distance, and in this setting we are using a histogram-moving metric. We note that provided we employ a Poisson-function based earthmover or histogram-mover scheme, there are not significant differences in the arithmetic of analyzing the cost of moving the probability mass in a one dimensional histogram, $x \cdot h(x)$, versus moving the corresponding histogram entry $h(x)$, since $xh(x)poi(kx, i) = h(x)poi(kx, i+1)\frac{i+1}{k}$.

As in the previous section, we begin our proof by compartmentalizing the probabilistic component of our theorem by defining a "faithful" pair of samples:

Definition 3.31. *A pair of samples of size k drawn, respectively, from distributions p, q with histogram $h = h_{p,q}$, with two-dimensional fingerprint $\mathcal{F}$, is said to be* faithful *if the following conditions hold:*

- *For all i, j,*

$$\left| \mathcal{F}(i,j) - \sum_{x,y:h(x,y)\neq 0} h(x,y) \cdot poi(kx, i)poi(xy, j) \right| \leq k^{\frac{1}{2}+\mathcal{D}}.$$

- *For all domain elements i, the number of times i occurs in the sample from p differs from its expectation of $k \cdot p(i)$ by at most*

$$\max\left\{ (k \cdot p(i))^{\frac{1}{2}+\mathcal{D}}, k^{\mathcal{B}(\frac{1}{2}+\mathcal{D})} \right\}.$$

  Analogously for the number of times i occurs in the sample from q.

- *Defining c_1, c_2 as in Algorithm 3.29,*

$$\sum_{x\in\left[\frac{c_1}{k}, \frac{c_1+k^{\mathcal{C}}}{k}\right], y\geq 0} x \cdot h(x,y) \leq 4k^{\mathcal{C}-\mathcal{B}}, \text{ and } \sum_{x\geq 0, y\in\left[\frac{c_2}{k}, \frac{c_2+k^{\mathcal{C}}}{k}\right]} y \cdot h(x,y) \leq 4k^{\mathcal{C}-\mathcal{B}}.$$

- *Additionally,*

$$1 - \sum_{i<c_1+k^{\mathcal{C}}, j<c_2+k^{\mathcal{C}}} \frac{i}{k}\mathcal{F}(i,j) + \sum_{x\leq\frac{c_1+k^{\mathcal{C}}/2}{k}, y\leq\frac{c_2+k^{\mathcal{C}}/2}{k}} x \cdot h(x,y) \leq 1 + k^{-\frac{1}{2}+\mathcal{D}},$$

and

$$1 - \sum_{i<c_1+k^{\mathcal{C}}, j<c_2+k^{\mathcal{C}}} \frac{j}{k}\mathcal{F}(i,j) + \sum_{x\le\frac{c_1+k^{\mathcal{C}}/2}{k}, y\le\frac{c_2+k^{\mathcal{C}}/2}{k}} y\cdot h(x,y) \;\le\; 1 + k^{-\frac{1}{2}+\mathcal{D}},$$

The proof of the following lemma follows from basic tail bounds on Poisson random variables, and Chernoff bounds, and is analogous to that of Lemma 3.13.

Lemma 3.32. *There is a constant $\gamma > 0$ such that for sufficiently large k, the probability that a pair of samples of size k consisting of independent draws from two fixed distribution is "faithful" is at least $1 - e^{-k^\gamma}$.*

Lemma 3.33. *Given two distributions of support size at most n with histogram h, and a "faithful" pair of samples of k with fingerprint $\mathcal{F}$, if c_1, c_2 are chosen as prescribed in Algorithm 3.29 then Linear Program 3.30 has a feasible point v' with objective value at most $O(k^{\frac{1}{2}+2\mathcal{B}+\mathcal{D}})$. Additionally,*

$$W(h, h_{v'}) \le O(k^{-\frac{\mathcal{B}}{2}+\mathcal{D}} + k^{-\mathcal{B}+\mathcal{C}})$$

where $h_{v'}$ is the generalized histogram that would be returned by Algorithm 3.29 if v' were used in place of the solution to the linear program, v.

Proof. We explicitly define v' as a function of the true histogram h and fingerprint of the sample, $\mathcal{F}$, as follows:

- Define h' such that $h'(x,y) = h(x,y)$ for all x, y satisfying $x \le \frac{c_1+k^{\mathcal{C}}/2}{k}$ and $y \le \frac{c_2+k^{\mathcal{C}}/2}{k}$, and for all other x, y set $h'(x,y) = 0$, where c_1, c_2 are as defined in Algorithm 3.29.
- Initialize v' to be identically 0, and for each pair x, y with either $x \ge 1/nk$ or $y \ge 1/nk$ such that $h'(x,y) \ne 0$ increment $v'_{x',y'}$ by $h'(x,y)$, where x', y' are defined to be x, y rounded down to the closest elements of the set $X = \{0, 1/nk, 2/nk, \ldots\}$.
- Let $m_1 := \sum_{x,y\in X} x v'_{x,y} + m_{1,\mathcal{F}}$ and $m_2 := \sum_{x,y\in X} y v'_{x,y} + m_{2,\mathcal{F}}$, where

  $$m_{1,\mathcal{F}} := 1 - \sum_{i<c_1+k^{\mathcal{C}}, j<c_2+k^{\mathcal{C}}} \frac{i}{k}\mathcal{F}(i,j) \text{ and } m_{2,\mathcal{F}} := 1 - \sum_{i<c_1+k^{\mathcal{C}}, j<c_2+k^{\mathcal{C}}} \frac{j}{k}\mathcal{F}(i,j).$$

  Assume without loss of generality that $m_1 > m_2$. If $m_1 > 1$, decrease the entries of v' arbitrarily until $m_1 = 1$. If the (recalculated) $m_2 < 1$, increase $v'_{0,y}$ by $(1-m_2)/y$, where $y = \frac{c_2+k^{\mathcal{C}}/2}{k}$. Otherwise, if $m_1, m_2 < 1$, increase $v'_{x,0}$ by $(1-m_1)/x$, where $x = \frac{c_1+k^{\mathcal{C}}/2}{k}$, and increase $v'_{0,y}$ by $(1-m_2)/y$, where $y = \frac{c_2+k^{\mathcal{C}}/2}{k}$.

To see that v' is a feasible point of the linear program, note that by construction, the first, second, and fourth conditions of the linear program are satisfied. The third condition of the linear program is satisfied because each of the true distributions has support at most

n, and, crudely, in the final step of the construction of v', we increment v' by less than $2k$—with one k corresponding to the increment we make for each of the two distributions.

We now consider the objective function value of v'. Note that $\sum_{i \le c_i} poi(c_1 + k^{\mathcal{C}}/2, i) = o(1/k)$, and analogously with c_2, hence the fact that we are truncating $h(x, y)$ at probability $x \le \frac{c_1 + k^{\mathcal{C}}/2}{k}$ and $y \le \frac{c_2 + k^{\mathcal{C}}/2}{k}$ in the first step in our construction of v', has little effect on the "expected fingerprints" $\mathcal{F}(i, j)$ for $i \le c_1, j \le c_2$: specifically, for all such i, j,

$$\sum_{x,y:h(x,y)\neq 0} (h'(x, y) - h(x, y))\, poi(kx, i) poi(ky, j) = o(1).$$

Together with the first condition of the definition of faithful, by the triangle inequality, for each such i, j

$$\left| \mathcal{F}(i, j) - \sum_{x,y:h'(x,y)\neq 0} h'(x, y) poi(kx, i) poi(ky, j) \right| \le k^{\frac{1}{2}+\mathcal{D}} + o(1).$$

We now bound the contribution of the discretization to the objective function value. As in the proof of Lemma 3.14, $\left| \frac{d\, poi(kx,i)}{d\, x} \right| \le k$, and hence we have

$$\left| \sum_{x,y:h'(x,y\neq 0)} h'(x, y) poi(kx, i) poi(ky, j) - \sum_{x,y\in X} v'_{x,y} poi(kx, i) poi(ky, j) \right| \le 4n \frac{k}{kn},$$

where the factor of 4 arises because the sum of the histogram entries is at most $2n$, and hence discretizing the support in two stages, by first discretizing the x component, and then discretizing the y component, each yields a contribution of at most $2n\frac{k}{kn}$.

In the final adjustment of mass in the final step of the creation of v', if any mass is added to v' again because $\sum_{i \le c_i} poi(c_i + k^{\mathcal{C}}/2, i) = o(1/k)$, this added mass alters the objective function value by at most $o(1)$. In the case that mass must be removed, by the fourth condition of "faithful", and the fact that h' is generated from h by rounding the support down, which only decreases the amount of probability mass, the removal of this mass will decrease the expected fingerprints by at most $2k \cdot k^{-\frac{1}{2}+\mathcal{D}} = 2k^{\frac{1}{2}+\mathcal{D}}$. Thus, putting together the above pieces, the objective function value associated to v' is bounded by

$$c_1 c_2 \left(k^{\frac{1}{2}+\mathcal{D}} + 4 + o(1) \right) + 2k^{\frac{1}{2}+\mathcal{D}} \le 7k^{\frac{1}{2}+2\mathcal{B}+\mathcal{D}},$$

for sufficiently large k.

We now turn to analyzing the distance $W(h, h_{v'})$, where $h_{v'}$ is the generalized histogram obtained by appending the empirical fingerprint entries $\mathcal{F}(i, j)$ for $i \ge c_1 + k^{\mathcal{C}}$ or $j \ge c_2 + k^{\mathcal{C}}$ to v'. Our scheme for moving the histogram entries of $h_{v'}$ to yield h will have three stages. In the first stage, we consider the portion of $h_{v'}$ consisting of the empirical fingerprint—namely, $h_{v'}(\frac{i}{k}, \frac{j}{k})$, where either $i \ge c_1 + k^{\mathcal{C}}$ or $j \ge c_2 + k^{\mathcal{C}}$. In the second stage, we consider the

portions corresponding to probability $x \leq \frac{c_1}{k}, y \leq \frac{c_2}{k}$, and in the third stage we consider the intermediate region (corresponding to the region of the fingerprint in which there are few entries).

For the first stage, for each domain element α contributing to histogram entry $h_{v'}(\frac{i}{k}, \frac{j}{k})$, we move one histogram entry to location (x, y), where x, y are the true probabilities with which α occurs in the two distributions. Considering the case that $i \geq j$, by the second condition of "faithful",

$$|\frac{i}{k} - x| + |\frac{j}{k} - y| \leq \frac{2}{k}(kx)^{\frac{1}{2}+\mathcal{D}},$$

and there can be at most $1/x$ such domain elements. Since $x > \frac{k^{\mathcal{B}}}{k}$, as it must contribute to the empirical fingerprint portion of $h_{v'}$, the total cost is at most

$$2\frac{1}{k^{\mathcal{B}}/k}\frac{2}{k}(k^{\mathcal{B}})^{\frac{1}{2}+\mathcal{D}} \leq 4k^{-\frac{\mathcal{B}}{2}+\mathcal{D}}.$$

where the first factor of 2 is the contribution from the setting in which $j \geq i$.

For the second stage, note that the rounding of h to yield $h_{v'}$ has a cost, per histogram entry of at most $\frac{1}{nk}$. There are at most $2n$ histogram entries, thus the total cost, neglecting the extra mass that might be added or removed in the final step of constructing v', is at most $\frac{2}{k}$. By the fourth condition of "faithful", in the final step of creating v' in which the total amount of mass is adjusted, at most $k^{-\frac{1}{2}+\mathcal{D}}$ units of mass will be removed from each distribution, which alters the above cost by at most $4k^{\frac{-1}{2}+\mathcal{D}}$, as the removal of a histogram element at (x, y) can contribute $x+y$ towards the cost, and $\max(x, y)$ towards the probability mass of one of the distributions.

Thus after the first two histogram-moving stages, $h(x, y)$ and $h_{v'}(x, y)$ are equal everywhere, except for (x, y) such that $x \leq \frac{c_1+k^{\mathcal{C}}}{k}$ and $y \leq \frac{c_2+k^{\mathcal{C}}}{k}$ and either $x \geq \frac{c_1+k^{\mathcal{C}}/2}{k}$ or $y \geq \frac{c_2+k^{\mathcal{C}}/2}{k}$. By the third condition of "faithful", there are at most $8k^{1-2\mathcal{B}+\mathcal{C}}$ histogram entries of h in this intermediate region. These can be moved so as to equalize the histogram entries in this region to those of $h_{v'}$ at a per-histogram entry cost of at most $4\frac{k^{\mathcal{B}}}{k}$, where the factor of 4 is because $x, y \leq 2k^{\mathcal{B}}$, and the cost is at most $x+y$, as these histogram entries will be sent to $(0, 0)$. Hence the contribution towards the cost is at most $4\frac{k^{\mathcal{B}}}{k} \cdot 8k^{1-2\mathcal{B}+\mathcal{C}} = 32k^{-\mathcal{B}+\mathcal{C}}$. Summing these bounds yields the lemma. □

We now define the two-dimensional analog of the earthmover schemes of Section 3.2. As we are working with a distance metric between two-dimensional generalized histograms that is in terms of the histogram entries, rather than the probability mass, our scheme will describe a manner of moving histogram entries. We repurpose much of the "Chebyshev bump" machinery of Section 3.2.

Definition 3.34. *For a given k, a* β-bump histogram-moving scheme *is defined by a sequence of pairs of positive real numbers $\{(r_i, r_i)\}$, the* bump centers*, and a sequence of corresponding functions $\{f_i\} : [0, 1]^2 \to \mathbb{R}$ such that $\sum_{i=0}^{\infty} f_i(x, y) = 1$ for all x, y, and each function*

f_i may be expressed as a linear combination of products of Poisson functions, $f_i(x,y) = \sum_{j,\ell=0}^{\infty} a_{ij\ell} poi(kx,j)poi(kx,\ell)$, such that $\sum_{j,\ell=0}^{\infty} |a_{ij\ell}| \leq \beta$.

Given a generalized histogram h, the scheme works as follows: for each x, y such that $h(x,y) \neq 0$, and each integer $i \geq 0$, move $h(x,y) \cdot f_i(x,y)$ histogram entries from (x,y) to the corresponding bump center (r_i, r_i). We denote the histogram resulting from this scheme by $(r,f)(h)$.

Definition 3.35. *A bump histogram-moving scheme (r,f) is* ϵ-good *if for any generalized histogram h, the histogram distance $W(h,(r,f)(h)) \leq \epsilon$.*

The histogram-moving scheme we describe will use a rectangular mesh of bump centers, and thus it will prove convenient to index the bump centers, and corresponding functions via two subscripts. Thus a bump center will be denoted (r_i, r_j), and the corresponding function will be denoted f_{ij}.

Definition 3.36. *Let $s = 0.1 \log k$, and let $B_i(x)$ denote the (one dimensional) Chebyshev bumps of Definition 3.17, corresponding to $s = 0.1 \log k$ (as opposed to $0.2 \log k$ as in Definition 3.17). We define functions f_{ij} for $i, j \in [s-1] \cup \{0\}$, by*

$$f_{ij}(x,y) = B_i(x)B_j(y).$$

Definition 3.37. *The* Chebyshev histogram-moving scheme *is defined in terms of k as follows: let $s = 0.1 \log k$. For $i \geq s$ or $j \geq s$, define the i,jth bump function $f_{ij}(x,y) = poi(kx,i)poi(ky,j)$ and associated bump center $(r_i, r_j) = (\frac{i}{k}, \frac{j}{k})$. For $i, j < s$ let $f_{i,j}(x,y) = B_i(x)B_j(y)$ and define their associated bump centers $(r_i, r_j) = \left(\frac{2s}{k}(1 - \cos(\frac{i\pi}{s})), \frac{2s}{k}(1 - \cos(\frac{j\pi}{s}))\right)$.*

The following lemma follows relatively easily from the corresponding lemmas in the one-dimensional setting (Lemmas 3.19 and 3.20), and shows that the above bump scheme is a $4k^{0.3}$-bump histogram-moving scheme.

Lemma 3.38. *Each $f_{ij}(x,y)$ may be expressed as*

$$f_{ij}(x,y) = \sum_{\ell,m=0}^{\infty} a_{ij,\ell,m} poi(kx,\ell)poi(ky,m)$$

for coefficients satisfying $\sum_{\ell,m=0}^{\infty} |a_{ij,\ell,m}| \leq 4k^{0.3}$. Additionally, for any x, y

$$\sum_{i,j \geq 0} f_{ij}(x,y) = 1.$$

Proof. To prove the first claim, in the proof of Lemma 3.19, we showed that $B_i = \sum_{j=0}^{\infty} a_{ij} poi(kx,j)$ with $\sum_{j \geq 0} |a_{ij} \leq 2e^{\frac{3}{2}s}$. Thus in our setting, as $s = 0.1k$, we have that $\sum_{\ell,m=0}^{\infty} |a_{ij,\ell,m}| \leq (2e^{\frac{3}{2}s})^2 = 4k^{0.3}$, as desired.

To prove the second claim, by Lemma 3.20, we have the following: for $i \geq s$, we have $\sum_{j\geq 0} f_{ij}(x,y) = poi(kx,i)\sum_{j\geq 0} poi(ky,j) = poi(kx,i)$. For $i < s$,

$$\begin{aligned}\sum_{j\geq 0} f_{ij}(x,y) &= \sum_{j<s} f_{ij}(x,y) + \sum_{j\geq s} f_{ij}(x,y) \\ &= B_i(x)\sum_{j=0}^{s-1} poi(ky,j) + poi(kx,i)\sum_{j\geq s} poi(ky,j).\end{aligned}$$

Hence

$$\begin{aligned}\sum_{i,j\geq 0} f_{ij} &= \left(\sum_{i=0}^{s-1} poi(kx,i)\right)\left(\sum_{j=0}^{s-1} poi(ky,j)\right) \\ &\quad + \left(\sum_{i=0}^{s-1} poi(kx,i)\right)\left(\sum_{j\geq s} poi(ky,j)\right) + \sum_{i\geq s} poi(kx,i) \\ &= \sum_{i<s} poi(kx,i) + \sum_{i\geq s} poi(kx,i) = 1.\end{aligned}$$

□

We now show that the scheme is $O(\sqrt{\delta})$-good, where $n = \delta k \log k$, and $\delta \geq \frac{1}{\log k}$. As in the one-distribution setting, the proof relies on the "skinnyness" of the Chebyshev bumps, as shown in Lemma 3.21, together with the bound on the support size.

Lemma 3.39. *The Chebyshev histogram-moving scheme of Definition 3.37 is $O(\sqrt{\delta})$-good, where $n = \delta k \log k$, and $\delta \geq \frac{1}{\log k}$.*

Proof. We begin by analyzing the contribution towards the cost of $h(x,y)$ for $x, y \leq \frac{s}{k}$. Note that we can decompose the cost of moving the histogram entry at (x,y) to the bump centers (r_i, r_j) into the component due to the movement in each direction. For the skinny bumps, the per-histogram entry cost of movement in the x direction is simply given by $\sum_{i=0}^{s-1} B_i(x)|x - r_i|$, which from Lemma 3.21 as employed in the proof of Lemma 3.22, is bounded by $O(\sqrt{\frac{x}{ks}})$. As $n = \delta k \log k$, and $h(x,y) \leq \min(\frac{1}{x}, \frac{1}{y}, 2n)$, the total cost of the skinny bumps is thus bounded by $O(n \cdot \sqrt{\frac{1/n}{ks}}) = O(\frac{1}{\sqrt{\delta}})$. For the wide bumps, the per-histogram entry cost is bounded by the following telescoping sum

$$\sum_{i\geq s} poi(kx,i)(|\frac{i}{k} - x|) = \sum_{i\geq s} poi(kx,i)\frac{i}{k} - \sum_{i\geq s} poi(kx,i+1)\frac{i+1}{k} = poi(kx,s)\frac{s}{k}.$$

And hence the total cost is at most $\sup_{x\leq s/k}\left(\frac{1}{x} poi(kx,s)\frac{s}{k}\right) = O(1/\sqrt{s})$.

For (x, y) such that either $x > \frac{s}{k}$ or $y > \frac{s}{k}$, by the analysis of the skinny bumps above, the contribution to the cost from the skinny bumps is trivially seen to be $O(1/\sqrt{s})$. For the wider bumps, as above we have the following telescoping sum

$$\begin{aligned}\sum_{i \geq kx} poi(kx, i)(|\frac{i}{k} - x|) &= \sum_{i \geq kx}^{\infty} poi(kx, i)\frac{i}{k} - \sum_{i \geq kx}^{\infty} poi(kx, i+1)\frac{i+1}{k} \\ &= poi(kx, \lceil kx \rceil)\frac{\lceil kx \rceil}{k}.\end{aligned}$$

Similarly,

$$\sum_{i < kx} poi(kx, i)(|\frac{i}{k} - x|) = poi(kx, \lfloor kx \rfloor)\frac{\lfloor kx \rfloor}{k}.$$

Thus the cost of the wide bumps, per histogram entry, is at most $O(\sqrt{x/k})$. From our lower bounds on either x or y, the histogram entry at (x, y) can be at most k/s, and hence the total cost of this portion of the histogram moving scheme is at most $O(\frac{k}{s}\sqrt{s/k^2}) = O(1/\sqrt{s})$, as desired. □

We are now equipped to assemble the pieces, and prove the performance guarantee of our ℓ_1 distance estimator. The proof mirrors that of Theorem 3.1; we leverage the fact that each Chebyshev bump can be expressed as a low-weight linear combination of Poisson functions, and hence given two generalized histograms corresponding to feasible points of Linear Program 3.30 that have low objective function, after applying the Chebyshev histogram-moving scheme, the resulting generalized histograms will be extremely similar. Together with Lemma 3.33 showing the existence of a feasible point that is close to the true histogram, all generalized histograms corresponding to solutions to the linear program (with low objective function) will be close to the true histogram, and in particular, will have similar ℓ_1 distance.

Proof of Theorem 3.3. Let h denote the generalized histogram of the pair of distributions from which the samples were drawn. Let g_1 denote the generalized histogram whose existence is guaranteed by Lemma 3.33, satisfying $W(g_1, h) \leq O(k^{-\frac{\mathcal{B}}{2}+\mathcal{D}} + k^{-\mathcal{B}+\mathcal{C}})$, corresponding to a feasible point of the linear program with objective function at most α. Let g_2 denote the generalized histogram output by Algorithm 3.29, and hence corresponds to a solution to the linear program with objective function at most α. Let g_1', g_2' denote the generalized histograms that result from applying the Chebyshev histogram-moving scheme of Definition 3.37 to g_1 and g_2, respectively. By Lemma 3.39, $W(g_i, g_i') = O(\sqrt{\delta})$. We now show that $W(g_1', g_2') = O(k^{-\mathcal{B}+\mathcal{C}})$.

The proof is nearly identical to that of Theorem 3.1, and we simply summarize the contributions to the distance from each region of bump centers, and the arguments are analogous to those of the one-distribution setting: the contribution to $W(g_1', g_2')$ from the bump centers (r_i, r_j) for $i, j \leq s$ is bounded by $O(s^2 \alpha k^{0.3}\frac{s}{k})$. The additional contribution from bump centers (r_i, r_j) with $i \leq c_1$, and $j \leq c_2$, not including the already counted bumps

with $i, j \leq s$ is bounded by $O(c_1 c_2 \alpha \frac{c_1+c_2}{k}) = O(k^{3\beta-1}\alpha)$. The contribution from (r_i, r_j) for either $i \geq c_1 + \frac{3}{4}k^{\mathcal{C}}$ or $j \geq c_2 + \frac{3}{4}k^{\mathcal{C}}$ is $o(1/k)$, as h and g are identical in this region. The remaining contribution, from the intermediate zone corresponding to bump centers (r_i, r_j) with $i \in [c_1, c_1 + \frac{3}{4}k^{\mathcal{C}}]$ and $j \leq c_2 + \frac{3}{4}k^{\mathcal{C}}$, or with $j \in [c_2, c_2 + \frac{3}{4}k^{\mathcal{C}}]$ and $i \leq c_1 + \frac{3}{4}k^{\mathcal{C}}$, contributes at most $O(k^{1-2\mathcal{B}+\mathcal{C}}\frac{k^{\mathcal{B}}}{k}) = O(k^{-\mathcal{B}+\mathcal{C}})$. □

Chapter 4

Two Multivariate Central Limit Theorems

Our information theoretic lower bounds on the sample size required to accurately estimate entropy, support size, and variational distance will rely on a characterization of the distributions of the "fingerprint" derived from a sample consisting of k independent draws from a discrete distribution. Our characterization of these distributions of fingerprints—high dimensional discrete distributions—will rely on two new multivariate central limit theorems. We devote a separate chapter to these central limit theorems, as we suspect that these limit theorems will have many applications beyond property estimation, and may be of broader interest to the computer science, information theory, and statistics communities.

Despite the increasing understanding of the various settings for which central limit theorems apply, most of the attention has been on univariate formulations. And as one might expect, the number of useful formulations of the central limit theorem seems to grow with the dimension; it is, perhaps, not surprising that the particularly natural and useful versions we prove here are absent from the statistics literature [38].

Our first central limit theorem relates the sum of independent random variables to the multivariate Gaussian of corresponding mean and covariance. As with the Berry-Esseen bound, and the classic multivariate central limit theorem of Götze[61], our bound is in terms of what may be considered the third moments of the distribution, under a suitable change of basis. We note that our bounds have an extra logarithmic term, though we suspect this could be removed with a tighter analysis.

The Berry-Esseen theorem bounds convergence to the Gaussian in terms of the maximum discrepancy between their respective cumulative distribution functions. Multiplying by two, this metric may be seen as a stand-in for the following: the maximum, over all intervals in $\mathbb{R}$, of the discrepancy between the probabilities of that interval under the two distributions. Götze's limit theorem can be thought of as generalizing this notion in the natural way to higher dimensions: convergence is shown relative to the discrepancy between the probabilities of any *convex set* ([61], and see [26] for discussion). Applying this result, intuitively, seems to require decomposing some high-dimensional set into small convex pieces, which,

unfortunately, tends to weaken the result by exponential factors. It is perhaps for this reason that, despite much enthusiasm for Götze's result, there is a surprising absence of applications in the literature, beyond small constant dimension.

For our purposes, and, we suspect, many others, convergence with respect to a more versatile distance metric is desired. The bound in our first central limit theorem is in terms of (Euclidean) earthmover distance (also known as the Wasserstein metric). Our proof of this central limit theorem is via Stein's method—a robust and elegant approach to proving central limit theorems—to which we provide a brief introduction in Section 4.2.

In Section 4.4 we then leverage this earthmover central limit theorem to prove a stronger but more specific central limit theorem for "generalized multinomial" distributions—a large class of discrete distributions (supported on the points of the integer lattice), parameterized by matrices, that generalize binomial and multinomial distributions and describe many distributions encountered in computer science (for example, the distributions considered in [47, 48, 114, 131]). We show that such distributions are close in total variational distance (ℓ_1 distance) to the Gaussian of corresponding mean and covariance that has been discretized so as to be supported on the integer lattice. This second central limit theorem, in terms of the extremely stringent ℓ_1 metric, will be one of the keystones of the proof of our lower bounds for property estimation given in Chapter 5.

4.1 Definitions and Discussion of Results

Our first central limit theorem, proved directly via Stein's method, applies to the general setting of sums of multivariate independent random variables. Given a random variable S_n that is the sum of n independent random variables $X_1, \ldots, X_n$ in $\mathbb{R}^k$, we aim to bound the earthmover distance (also known as the *Wasserstein distance*) between the distribution of S_n and the multivariate Gaussian G, which we will denote as $d_W(S_n, G)$. Intuitively, this distance $d_W(A, B)$ is defined as "the minimum, over all schemes of moving the probability mass of A to make B, of the cost of moving this mass, where the per-unit cost of moving mass from point x to point y is simply the (Euclidian) distance between x and y." It is often easier to define and work with the *dual* formulation of earthmover distance (this is the Kantorovich-Rubinstein theorem, [77], and may be intuitively seen as what one would expect from linear programming duality):

Definition 4.1. *Given two distributions A, B in $\mathbb{R}^k$, then, letting* $\mathrm{Lip}(\mathbb{R}^k, 1)$ *denote the set of functions $h : \mathbb{R}^k \to \mathbb{R}$ with Lipschitz constant 1, that is, where for any $x, y \in \mathbb{R}^k$ we have $|h(x) - h(y)| \leq ||x - y||$, then the* earthmover distance *between A and B is defined as*

$$d_W(A, B) = \sup_{h \in \mathrm{Lip}(\mathbb{R}^k, 1)} E[h(A)] - E[h(B)].$$

Our first central limit theorem is the following:

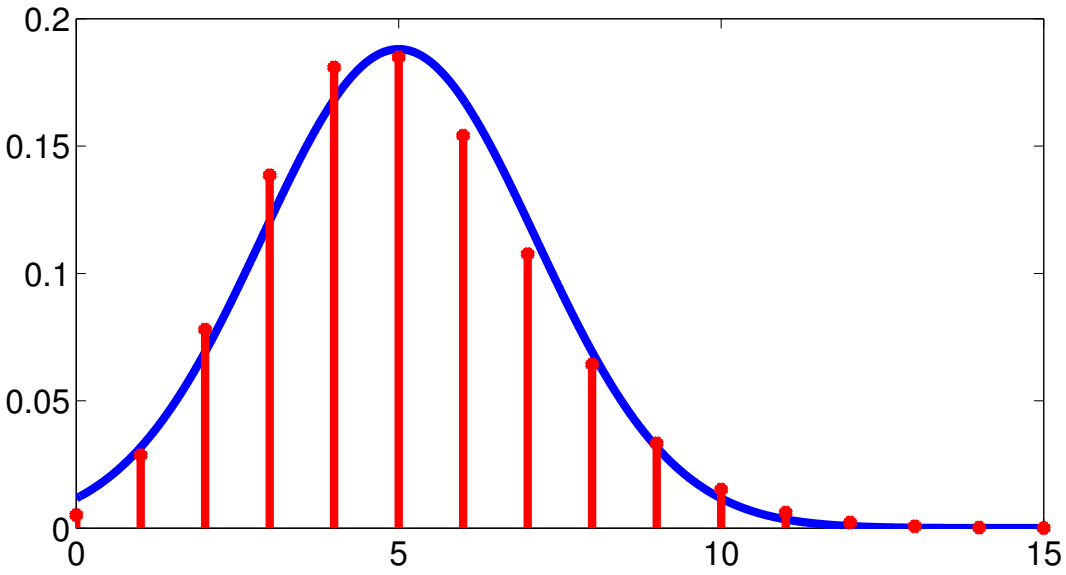

Figure 4.1: The binomial distribution with $p = 0.1$ and 50 draws (red bars), compared with the Gaussian distribution of matching mean and variance (blue curve). Theorem 4.1, implies that the earthmover distance between these distributions is at most $0.9(2.7 + 0.83 \log 50)$.

Theorem 4.1. *Given n independent distributions $\{Z_i\}$ of mean 0 in $\mathbb{R}^k$ and a bound β such $||Z_i|| < \beta$ for any i and any sample, then the earthmover distance between $\sum_{i=1}^{n} Z_i$ and the normal distribution of corresponding mean (0) and covariance is at most $\beta k(2.7+0.83 \log n)$.*

Figure 4.1 provides a simple illustration of Theorem 4.1, in the univariate setting ($k = 1$); of course, in the univariate setting, such central limit theorems are standard (see [17]).

We note the parameters of Theorem 4.1: as more and more random variables are added in, the performance of the approximation only gets very mildly worse, increasing with the logarithm of the number of random variables, n. In fact, we strongly suspect that, in analogy with univariate results, there should be no dependence on n in the theorem. The linear dependence on k, the dimension, is more fundamental; it is not hard to show that this dependence must be of order at least $\sqrt{k}$, so one might conjecture a tight form of the theorem's bound to be $\Theta(\beta\sqrt{k})$.

We note that it is somewhat more standard for central limit theorems of this type to be stated in terms of third moments, instead of a bound β on each random variable, and our approach can obtain such bounds, though we favor the clarity of Theorem 4.1, which is sufficient for our applications.

To provide algorithmic lower-bounds, we must work with a much more stringent distance metric than earthmover distance. In our second central limit theorem, we work with *total variational distance* (sometimes referred to as "statistical distance" or ℓ_1 distance). Funda-

mentally, if distributions A and B have total variational distance 0.1, then *any* algorithm taking an input drawn from A must behave identically at least 90% of the time to the algorithm run on an input drawn from B.

We first note that the conditions of Theorem 4.1 are not strong enough to imply any sort of total variational distance bound: the discrete distribution illustrated in Figure 4.1 has (maximal) total variational distance 1 from its Gaussian approximation. However, the intuition for our second central limit theorem is the observation that the total variational distance between the two distributions of Figure 4.1 is in fact very small if we first round the Gaussian distribution to be supported on the lattice points. We now define the class of distributions to which our second limit theorem will apply.

Definition 4.2. *The* generalized multinomial distribution *parameterized by a nonnegative matrix ρ each of whose rows sum to at most 1, is denoted M^ρ, and is defined by the following random process: for each row $\rho(i, \cdot)$ of matrix ρ, interpret it as a probability distribution over the columns of ρ—including, if $\sum_{j=1}^{k} \rho(i, j) < 1$, an "invisible" column 0—and draw a column index from this distribution; return a row vector recording the column sums (i.e. the ith index is the total number of rows that selected the ith column).*

The "invisible" column is used for the same reason that the binomial distribution is taken to be a univariate distribution; while one could consider it a bivariate distribution, counting heads and tails separately, it is convenient to consider tails "invisible", as they are implied by the number of heads.

Definition 4.3. *The k-dimensional* discretized *Gaussian distribution, with mean μ and covariance matrix Σ, denoted $\mathcal{N}^{disc}(\mu, \Sigma)$, is the distribution with support $\mathbb{Z}^k$ obtained by picking a sample according to the Gaussian $\mathcal{N}(\mu, \Sigma)$, then rounding each coordinate to the nearest integer.*

Our second central limit theorem, that we leverage for our property estimation lower bounds in Chapter 5, is the following:

Theorem 4.2. *Given a generalized multinomial distribution M^ρ, with k dimensions and n rows, let μ denote its mean and Σ denote its covariance matrix, then*

$$D_{tv}\left(M^\rho, \mathcal{N}^{disc}(\mu, \Sigma)\right) \leq \frac{k^{4/3}}{\sigma^{1/3}} \cdot 2.2 \cdot (3.1 + 0.83 \log n)^{2/3},$$

where σ^2 is the minimum eigenvalue of Σ.

The above theorem implies that if $\sigma^2 = \omega(k^8 \log^4 n)$ then the multinomial distribution is well-approximated by the natural discrete Gaussian approximation.

We overview some of the key ideas of the proof. Note that even among distributions over the lattice points, bounds on the earthmoving distance do not necessarily translate into bounds on total variational distance—consider a distribution supported on the even integers,

versus one supported only on the odd integers, or some much worse high-dimensional analogue. However, one elementary and completely general way to convert earthmover distance bounds, such as those of Theorem 4.1, into total variational distance bounds is to convolve the distributions by a smooth distribution that is "wide enough".

Thus the total variational distance between *convolved* versions of these distributions is small. We must, however, "deconvolve" to achieve the desired result. Deconvolution, in general, is very poorly behaved and can blow up badly. The saving grace in our setting is the fact that any multinomial distribution is in fact *unimodal* in each coordinate direction. (Intuitively, at least for the one-dimensional case, unimodality is what prevents one distribution from being supported on, say, only the even integers.) Specifically, we prove a "deconvolution lemma" that has good bounds when the result of deconvolution is unimodal.

While binomial distributions are trivially unimodal, the analysis rapidly becomes complicated. The general result for the univariate case is known as *Newton's inequalities*. The multivariate case, which we rely on in our proof of Theorem 4.2, was proven only recently in a 2008 work of Gurvits—see Fact 1.10:2 of [64].

4.2 Stein's Method

Since Stein's seminal paper [118], presented in 1970, describing an alternative proof approach—what became known as "Stein's method"— for proving Berry-Esseen-style central limit theorems, there has been a blossoming realization of its applicability to different settings. In particular, there have been several successful applications of Stein's method in multivariate settings [39, 61, 111]. To prove our first central limit theorem, we closely follow the treatment for the multivariate limit theorem given in Bhattacharya and Holmes' exposition (and slight correction) of the result of Götze [26, 61] . For a more general introduction to Stein's method, see [40]. In the remainder of this section, we provide a very basic overview of Stein's method, and illustrate its application by proving a very simple univariate central limit theorem.

The goal of central limit theorems is to argue that some peculiar distribution X (perhaps a sum of independent random variables), is close, with respect to some specified metric, to a "nice" distribution, G (typically a Gaussian). In its most general form, the basic approach of Stein's method is to consider some transformation T whose unique fixed point is the distribution G; thus if one applies T to the distribution X, and finds that $T(X) = X$, one can conclude that X is, in fact, *identical* to G. Intuitively, it is tempting to hope that if $T(X)$ is very similar to X, then X should be close to G (in some metric), and that if $T(X)$ is drastically different than X, X must be very far from the fixed point of the map T, and thus X and G will be far apart. Thus the hope is that rather than needing to compare the distribution X directly to the distribution G, it will suffice to simply compare X to the transformed distribution $T(X)$. Conveniently, rather than even comparing X to $T(X)$, at least in the case of the Gaussian, we will be able to *simulate* performing the transformation T while only altering the set of "test functions". To summarize, rather than comparing X

to G, it will suffice to compare X to $T(X)$, and furthermore, we will be able to perform that comparison by only altering the set of test functions.

In the case where the target distribution, G, is a Gaussian with isotropic covariance, the corresponding transformation T is known as the Ornstein–Uhlenbeck process, and corresponds to the procedure of adding a small amount of random noise to the distribution, and then scaling the distribution back towards the origin in such a manner so as to preserve the variance of the distribution (by compensating for the increase in variance effected by adding the small amount of random noise). The unique fixed point of such a transformation is the Gaussian, and it is not hard to see that after repeated applications of such a transformation, the resulting distribution will tend to a Gaussian.

Stein's method is especially suited to the Wasserstein metric because the Ornstein–Uhlenbeck process is a continuous transformation, and thus will interact best with a set of continuous test function. For the reminder of this section, we will take $\mathcal{H}$ to be the set of smooth functions with Lipschitz constant 1.

Let us now consider the effect that the Ornstein–Uhlenbeck process will have on $E[f(X)]$, for some function $f \in \mathcal{H}$. It is not hard to see that the addition of some small amount of random noise to X will increase $E[f(X)]$ in proportion to the second derivative $E[f''(X)]$; the rescaling of X towards the origin will decrease $E[f(X)]$ in proportion to $E[Xf'(X)]$. If our reasoning is correct, for the Gaussian distribution G, we should have $E[f''(G) - Gf'(G)] = 0$, for any smooth function f with bounded derivative. A simple integration shows that this identity holds; letting $g(x) = f'(x)$, integration by parts yields:

$$\begin{aligned} E[g'(G)] &= \int_{x\in\mathbb{R}} g'(x)e^{-x^2/2}dx \\ &= \left[g(x)e^{-x^2/2}\right]_{-\infty}^{\infty} + \int_{x\in\mathbb{R}} xg(x)e^{-x^2/2}dx \\ &= \int_{x\in\mathbb{R}} xg(x)e^{-x^2/2}dx = E[Gg(G)]. \end{aligned}$$

Thus if X is a gaussian, $E[g'(X) - Xg(X)] = 0$. We now show that the degree to which this expectation deviates from zero is related to how far X is from the Gaussian. A variant of the above integration can also be used to show the following identity:

$$E[g(X)] - E[g(G)] = E[h_g'(X) - Xh_g(X)],$$

where the functions g and h_g are related by $h_g(x) := e^{x^2/2}\int_{-\infty}^{x}(g(t) - E[g(G)])\,e^{-t^2/2}dt$. It is not hard to show that for well-behaved functions g, h_g will also be well-behaved. In particular, it is not hard to show that $||h_g''|| \leq ||g'||$. Given this identity, we now prove a simple univariate central limit theorem.

Let $X = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} X_i$, where the X_i are identical and independent random variables with expectation 0, and unit variance, and thus X has mean 0 and unit variance. We wish to prove a central limit theorem in terms of Wasserstein distance, thus we will take $\mathcal{H} = Lip(\mathbb{R}, 1)$. By

the above identity $D_W(X,G) = \sup_{g\in\mathcal{H}} |E[g(X)] - E[g(G)]| = \sup_{g\in\mathcal{H}} |E[h'_g(X) - Xh_g(X)]|$. We first evaluate the second term on the right, and then evaluate the first term. Since X_i, X_j are identical, $E[Xh_g(X)] = nE[\frac{X_1}{\sqrt{n}}h_g(X)]$. Letting $X' = \frac{1}{\sqrt{n}}\sum_{i=2}^n$, and Taylor expanding $h_g(X)$ about X' yields the following:

$$\begin{aligned} E[Xh_g(X)] &= \sqrt{n}E[X_1 h_g(X)] \\ &= \sqrt{n}E[X_1(h_g(X') + \frac{X_1}{\sqrt{n}}h'_g(X'))] + \alpha, \text{ for some } \alpha \text{ with } |\alpha| \leq \frac{1}{2\sqrt{n}}||h''_g||E[|X_1^3|] \\ &= E[h'_g(X')] + \alpha, \end{aligned}$$

where the last line followed by using the facts that X_i, X_j are independent for $i \neq j$ and the fact that $E[X_i] = 0$. To conclude, we evaluate $E[h'_g(X)]$ by Taylor expanding $h'_g(X)$ about X', to yield $E[h'_g(X)] = E[h'_g(X')] + \beta$, for $|\beta| \leq \frac{1}{\sqrt{n}}||h''_g||E[|X_1|]$. Thus, recalling that for $g \in Lip(1)$, $||h''_g|| \leq 2$, we obtain the usual third moment bound, with the correct factor of $\frac{1}{\sqrt{n}}$:

$$D_W(X,G) = \sup_{g\in\mathcal{H}} |E[h'_g(X) - Xh_g(X)]| \leq \frac{2 + E[|X_1|^3]}{\sqrt{n}}.$$

While the analysis becomes considerably more involved in the multivariate case, the basic framework and intuition of Stein's method remains the same.

4.3 A Multivariate Central Limit Theorem via Stein's Method

We now begin our proof of Theorem 4.1. We prove this as a consequence of the following theorem, which is somewhat tighter though more unwieldy. As it turns out, if the variance of $\sum_{i=1}^n Z_i$ is much larger in a certain direction than in others, then the earthmover bound is more forgiving of draws from Z_i that are large in that direction.

Theorem 4.3. *Given n independent distributions $\{Z_i\}$ in $\mathbb{R}^k$, each having mean 0, and having total covariance equal to $k \times k$ matrix Σ, let T be the Cholesky factorization of Σ—that is, a $k \times k$ matrix such that $TT^\intercal = \Sigma$, making $T^{-1}\sum_{i=1}^n Z_i$ have covariance equal to the $k \times k$ identity matrix. Then the earthmover distance between $\sum_{i=1}^n Z_i$ and the normal distribution of mean 0 and covariance Σ is at most*

$$\begin{aligned} \sum_{i=1}^n 1.16E\left[||Z_i|| \cdot ||T^{-1}Z_i||\right] \cdot E\left[||T^{-1}Z_i|| \log\left(1 + \frac{2.76}{||T^{-1}Z_i||}\right)\right] \\ + 0.49E\left[||Z_i|| \cdot ||T^{-1}Z_i||^2 \cdot \log\left(1 + \frac{9.41}{||T^{-1}Z_i||}\right)\right]. \end{aligned} \quad (4.1)$$

We prove this theorem using an adaptation of Stein's method as implemented for the multivariate case in [61]. (See also [26].) Before proving Theorem 4.3, we first show that it implies the more simple limit theorem of Theorem 4.1.

Proof of Theorem 4.1. We prove this from Theorem 4.3. In Equation 4.1 we note that both the first and second term have exactly one factor of $||Z_i||$, which we may upper-bound by β. Further, since the function $f(x) = x\log(1+\frac{1}{x})$ is increasing for positive x, the rearrangement inequality implies that the first term is bounded by the corresponding expression with all parts put inside a single expectation. Thus Equation 4.1 is bounded by

$$\beta \sum_{i=1}^{n} E\left[||T^{-1}Z_i||^2 \left(1.16\log\left(1+\frac{2.76}{||T^{-1}Z_i||}\right)+0.49\log\left(1+\frac{9.41}{||T^{-1}Z_i||}\right)\right)\right] \tag{4.2}$$

Define a new distribution Y such that for every vector x,

$$\Pr[Y=x] = \frac{1}{c}||x|| \sum_{i=1}^{n} Pr[T^{-1}Z_i = x],$$

where $c = \sum_{i=1}^{n} E[||T^{-1}Z_i||]$ is chosen so that Y is a valid distribution (that is, having total probability mass 1). (If the Z_i are continuous random variables, we define the distribution Y correspondingly.) We note that, letting $g(x) = x\cdot(1.16\log(1+\frac{2.76}{x})+0.49\log(1+\frac{9.41}{x}))$, we have that Equation 4.2 equals $\beta c \cdot E[g(||Y||)]$. The concavity of f implies the concavity of g, which implies by Jensen's inequality that $E[g(||Y||)] \le g(E[||Y||])$. We have that $E[||Y||] = \frac{1}{c}\sum_{i=1}^{n} E[||T^{-1}Z_i||^2] = E\left[||T^{-1}\sum_{i=1}^{n} Z_i||\right] = \frac{k}{c}$, since covariance adds for independent distributions, and T is the matrix that transforms $\sum_{i=1}^{n} Z_i$ to have covariance the identity matrix.

Thus the earthmover distance is bounded by $\beta k(1.16\log(1+\frac{2.76c}{k})+0.49\log(1+\frac{9.41c}{k}))$. As this is an increasing function of c, it remains to bound c. We can crudely bound c by defining the distribution W that uniformly picks $i \in \{1,\ldots,n\}$ and then draws a sample from $T^{-1}Z_i$; we note that $c = n\cdot E[||W||]$. We bound c by observing that $E[||W||^2] = \frac{k}{n}$, from which, by the convexity of the squaring function and Jensen's inequality, we have that $c = nE[||W||] \le n\sqrt{E[||W||^2]} = \sqrt{nk} \le k\sqrt{n}$. Thus the earthmover distance is bounded by $\beta k(1.16\log(1+2.76\sqrt{n})+0.49\log(1+9.41\sqrt{n}))$, which, for $n \ge 1$ is easily seen to be less than the desired bound of $\beta k(2.7+0.83\log n)$. □

We now begin our proof of Theorem 4.3. It will be convenient for us to assume that our test functions, h, in addition to being Lipschitz continuous, are also differentiable. We note that even restricting the test functions to be *smooth* does not affect the distance defined with respect to such a class of functions, as, for any Lipschitz-continuous function h, letting h_ϵ be the convolution of h with a Gaussian of radius ϵ for any $\epsilon > 0$, we note that h_ϵ is smooth, and $|h(x)-h_\epsilon(x)| \le \epsilon\sqrt{k}$; thus for any random variables A, $\lim_{\epsilon\to 0} E[h_\epsilon(A)] = E[h(A)]$, and the earthmover distance definition remains unaltered.

Proof of Theorem 4.3. We let $X_i = T^{-1}Z_i$ and work with X_i instead of Z_i throughout. While the earthmover distance in the original basis is defined via the supremum over differentiable test functions in $\mathrm{Lip}(\mathbb{R}^k,1)$, when we work with X_i, the test functions instead range over $T\circ\mathrm{Lip}(\mathbb{R}^k,1)$, that is, for $\ell \in \mathrm{Lip}(\mathbb{R}^k,1)$, we take $h(x) = \ell(Tx)$.

The heart of Stein's method consists of constructing a simple transformation $h \to f_h$ that takes test functions $h \in T \circ \mathrm{Lip}(\mathbb{R}^k, 1)$ and transforms them to appropriate functions f_h such that for any distribution S_n, we have

$$E[h(S_n)] - E[h(\Phi)] = E[S_n \cdot \nabla f_h(S_n) - \triangle f_h(S_n)], \tag{4.3}$$

where $\triangle f_h$ represents the Laplacian of f_h and ∇f_h the gradient of f_h. When one takes Taylor expansions of each of the two terms on the right hand side, one can arrange to have a pair of terms that have second-order dependence on S_n cancel, leaving only third-order terms remaining, which is what will yield the third-order dependence in the theorem.

We cite [26] for the result that Equation 4.3 is satisfied when, letting $\phi_r(x) \triangleq (2\pi r^2)^{-k/2} e^{-\frac{||x||}{2r^2}}$ be the k-dimensional Gaussian of mean 0 and radius r, we define

$$f_h(x) \triangleq \int_0^\infty (h * \phi_{\sqrt{1-e^{-2s}}})(e^{-s}x) - E[h(\Phi)]\, ds, \tag{4.4}$$

where we consider $h * \phi_0 = h$.

We take $S_n = \sum_{i=1}^n X_i$, and let S_{-i} denote $S_n - X_i$, that is, the sum of samples from all but one of the distributions; by definition S_{-i} is independent of X_i. We use the first-order expansion $f(x+y) = f(x) + \int_0^1 y \cdot \nabla f(x+ty)\, dt$, where $y \cdot \nabla f(x+ty)$ is simply the directional derivative of f in the direction y evaluated at $x + ty$. In coordinates, this is

$$f(x+y) = f(x) + \int_0^1 \sum_{a=1}^k y(a) D_a f(x+ty)\, dt,$$

where we use D_a to denote the partial derivative in the ath coordinate. Similarly, the second-order expansion is

$$f(x+y) = f(x) + y \cdot \nabla f(x) + \int_0^1 (1-t) \sum_{a,b=1}^k y(a) y(b) D_{ab} f(x+ty)\, dt,$$

where as above, $\sum_{a,b=1}^k y(a)y(b)D_{ab}f(x+ty)$ is just the "directional second derivative" of f, in the direction y, evaluated at $x + ty$. Thus we may expand $S_n \cdot \nabla f(S_n) = \sum_{i=1}^n X_i \cdot \nabla f(S_{-i} + X_i) = \sum_{i=1}^n \sum_{a=1}^k X_i(a) D_a f(S_{-i} + X_i)$ to second order as

$$\sum_{i=1}^n \sum_{a=1}^k X_i(a) \left(D_a f(S_{-i}) + \left(\sum_{b=1}^k X_i(b) D_{ab} f(S_{-i}) \right) \right.$$
$$\left. + \left(\int_0^1 (1-t) \sum_{b,c=1}^k X_i(b) X_i(c) D_{abc} f(S_{-i} + t \cdot X_i)\, dt \right) \right). \tag{4.5}$$

We note that since X_i has mean 0 and is independent of S_{-i}, the first term has expectation 0. We now aim to cancel the expectation of the second term against an expansion of

$\triangle f(S_n)$. Note that the expected value of the factor $X_i(a)X_i(b)$ in the second term is just the (a,b)th component of the covariance matrix of X_i, which we write as $\mathrm{Cov}(X_i)(a,b)$. Since by assumption, the sum over i of the covariance matrices $\mathrm{Cov}(X_i)$ equals the identity matrix, we may rewrite $\triangle f(S_n) = \sum_{(a=b)=1}^{k} D_{ab}f(S_n) = \sum_{i=1}^{n}\sum_{a,b=1}^{k}\mathrm{Cov}(X_i)(a,b)D_{ab}f(S_n)$. Expanding the ith term of this to first order centered at S_{-i}, for each i, yields

$$\sum_{i=1}^{n}\sum_{a,b=1}^{k}\mathrm{Cov}(X_i)(a,b)\left(D_{ab}f(S_{-i}) + \int_0^1 \sum_{c=1}^{k} X_i(c)D_{abc}f(S_{-i}+t\cdot X_i)\,dt\right), \tag{4.6}$$

where the expectation of the first term above is seen to be exactly the expectation of the second term of Equation 4.5, and thus the difference between the expectations of Equations 4.5 and 4.6, which for $f = f_h$ equals $E[h(S_n)] - E[h(\Phi)]$ by construction, will consist only of the last, third-order terms from each expression.

Let ζ_i denote the expectation of the last term of Equation 4.5 for the corresponding i, and η_i denote the expectation of the last term of Equation 4.6 for the corresponding i. By the above, $d_W(S_n,\Phi)$ is thus bounded by the supremum over $h \in T\circ\mathrm{Lip}(\mathbb{R}^k,1)$ of $\sum_{i=1}^{n}|\zeta_i|+|\eta_i|$. We thus turn to bounding ζ_i,η_i. We assume throughout that $X_i \neq 0$, as, when $X_i = 0$ the corresponding terms of Equations 4.5 and 4.6 are trivially seen to be 0.

Defining $g_s(x) = h(e^{-s}x)$, we note that we may reexpress the first term in the definition of f_h as $(h * \phi_{\sqrt{1-e^{-2s}}})(e^{-s}x) = (g_s * \phi_{\sqrt{e^{2s}-1}})(x)$. Letting $\widetilde{X}_i$ denote an independent sample from the distribution X_i, we note that we may replace $\mathrm{Cov}(X_i)(a,b)$ in Equation 4.6 by $E[\widetilde{X}_i(a)\widetilde{X}_i(b)]$, thus yielding that η_i equals the expectation of

$$\int_0^\infty\int_0^1\sum_{a,b,c=1}^{k}\widetilde{X}_i(a)\widetilde{X}_i(b)X_i(c)D_{abc}(g_s * \phi_{\sqrt{e^{2s}-1}})(S_i+t\cdot X_i)\,dt\,ds,$$

where we note that the final term $E[h(\Phi)]$ of Equation 4.4 is constant, and hence its third derivative does not contribute to η_i, and is thus omitted in the above equation.

We note that the expression $\sum_{a,b,c=1}^{k}\widetilde{X}_i(a)\widetilde{X}_i(b)X_i(c)D_{abc}$ is just a third directional derivative, with two differentiations in the direction of the vector $\widetilde{X}_i$ and one in the direction X_i, which we may denote as $D_{\widetilde{X}_i}D_{\widetilde{X}_i}D_{X_i}$. Since convolution commutes with differentiation, η_i thus equals the expectation of

$$\begin{aligned}
&\int_0^\infty\int_0^1 (D_{\widetilde{X}_i}g_s * D_{\widetilde{X}_i}D_{X_i}\phi_{\sqrt{e^{2s}-1}})(S_i+t\cdot X_i)\,dt\,ds\\
&\qquad = \int_0^\infty\int_0^1\int_{\mathbb{R}^k} D_{\widetilde{X}_i}g_s(x)D_{\widetilde{X}_i}D_{X_i}\phi_{\sqrt{e^{2s}-1}}(S_i+t\cdot X_i-x)\,dx\,dt\,ds\\
&\qquad = \int_0^\infty\int_{\mathbb{R}^k} D_{\widetilde{X}_i}g_s(x)\int_0^1 D_{\widetilde{X}_i}D_{X_i}\phi_{\sqrt{e^{2s}-1}}(S_i+t\cdot X_i-x)\,dt\,dx\,ds
\end{aligned}$$

Because h, by definition, is the composition of matrix T with a differentiable function of Lipschitz constant 1, g_s is the composition of T with a function of Lipschitz constant e^{-s}

and thus we can bound the absolute value of this last expression by

$$\int_0^\infty ||T\widetilde{X}_i||e^{-s} \int_{\mathbb{R}^k} \left| \int_0^1 D_{\widetilde{X}_i} D_{X_i} \phi_{\sqrt{e^{2s}-1}}(t \cdot X_i + x)\, dt \right| dx\, ds, \tag{4.7}$$

where we have made the substitution $S_i - x \to x$. We bound the integral over $\mathbb{R}^k$ in two ways. First, since a univariate Gaussian of variance r^2 is unimodal, the integral of the absolute value of its derivative is simply twice its maximum, namely $2 \cdot \frac{1}{\sqrt{2\pi r^2}}$. Since ϕ_r can be expressed as the product of k univariate Gaussians along orthogonal basis directions, each of variance r^2, and having integral 1, we have that $\int_{\mathbb{R}^k} |D_{\widetilde{X}_i} \phi_{\sqrt{e^{2s}-1}}|\, dx = \frac{2||\widetilde{X}_i||}{\sqrt{2\pi(e^{2s}-1)}}$, just the corresponding univariate expression in the basis direction $\frac{\widetilde{X}_i}{||\widetilde{X}_i||}$. Since integration is the inverse of differentiation, we have that $\int_0^1 D_{\widetilde{X}_i} D_{X_i} \phi_{\sqrt{e^{2s}-1}}(t \cdot X_i + x)\, dt = D_{\widetilde{X}_i} \phi_{\sqrt{e^{2s}-1}}(X_i + x) - D_{\widetilde{X}_i} \phi_{\sqrt{e^{2s}-1}}(x)$, and by the triangle inequality we may thus bound the $\mathbb{R}^k$ integral of Equation 4.7 as twice what we just computed: $\frac{4||\widetilde{X}_i||}{\sqrt{2\pi(e^{2s}-1)}}$.

For large s, however, this bound is not effective, and in this case we instead take

$$\begin{aligned}\int_{\mathbb{R}^k} \left| \int_0^1 D_{\widetilde{X}_i} D_{X_i} \phi_{\sqrt{e^{2s}-1}}(t \cdot X_i + x)\, dt \right| dx &\leq \int_{\mathbb{R}^k} \int_0^1 \left| D_{\widetilde{X}_i} D_{X_i} \phi_{\sqrt{e^{2s}-1}}(t \cdot X_i + x) \right| dt\, dx \\ &= \int_{\mathbb{R}^k} \left| D_{\widetilde{X}_i} D_{X_i} \phi_{\sqrt{e^{2s}-1}}(x) \right| dx\end{aligned}$$

Letting $y_i = \frac{X_i}{||X_i||}$ denote the unit vector in the X_i direction, and z_i denote an orthogonal unit vector such that, for real numbers u, v we have $\widetilde{X}_i = u \cdot y_i + v \cdot z_i$, we thus have $D_{\widetilde{X}_i} D_{X_i} = ||X_i||(u \cdot D_{y_i}^2 + v \cdot D_{z_i} D_{y_i})$, and by the triangle inequality we may bound

$$\int_{\mathbb{R}^k} \left| D_{\widetilde{X}_i} D_{X_i} \phi_{\sqrt{e^{2s}-1}}(x) \right| dx \leq ||X_i|| \int_{\mathbb{R}^k} \left| u \cdot D_{y_i}^2 \phi_{\sqrt{e^{2s}-1}}(x) \right| + \left| v \cdot D_{y_i} D_{z_i} \phi_{\sqrt{e^{2s}-1}}(x) \right| dx, \tag{4.8}$$

where we may now leverage the orthogonality of y_i and z_i.

As above, we note that since the Gaussian can be expressed as the product of one-dimensional Gaussians along any orthogonal basis, and since y_i and z_i are orthogonal unit vectors, we have that $\int_{\mathbb{R}^k} |D_{y_i} D_{z_i} \phi_{\sqrt{e^{2s}-1}}(x)|\, dx = \left(\frac{2}{\sqrt{2\pi(e^{2s}-1)}} \right)^2 = \frac{2}{\pi(e^{2s}-1)}$, just the square of the univariate case we computed above. Similarly, $\int_{\mathbb{R}^k} |D_{y_i}^2 \phi_{\sqrt{e^{2s}-1}}(x)|\, dx$ equals the corresponding expression for a univariate Gaussian, the integral of the absolute value of its second derivative, which by definition is the total variation of its first derivative. As the derivative of a univariate Gaussian of variance r^2 takes maximum and minimum values at $\pm r$, at which locations it has values respectively $\mp \frac{e^{-1/2}}{r^2\sqrt{2\pi}}$, and has no other local optima, its total variation is just four times this, which, for $r^2 = e^{2s} - 1$ gives us $\int_{\mathbb{R}^k} |D_{y_i}^2 \phi_{\sqrt{e^{2s}-1}}(x)|\, ds = \frac{4e^{-1/2}}{(e^{2s}-1)\sqrt{2\pi}}$.

Thus, since $|u|^2+|v|^2 = ||\widetilde{X}_i||^2$, we bound Equation 4.8 as $\frac{||X_i||}{e^{2s}-1}$ times $|u|\frac{4e^{-1/2}}{\sqrt{2\pi}}+|v|\frac{2}{\pi}$. We bound this last expression by the Cauchy-Schwarz inequality as $||\widetilde{X}_i||\sqrt{\left(\frac{4e^{-1/2}}{\sqrt{2\pi}}\right)^2+\left(\frac{2}{\pi}\right)^2} = ||\widetilde{X}_i||\frac{2}{\pi}\sqrt{1+2\pi e^{-1}}$. Equation 4.8 is thus bounded by $||X_i||\cdot||\widetilde{X}_i||\frac{1}{e^{2s}-1}\frac{2}{\pi}\sqrt{1+2\pi e^{-1}}$. Combining this bound with the bound computed above yields

$$|\eta_i| \leq E\left[||T\widetilde{X}_i||\cdot||\widetilde{X}_i||\int_0^\infty e^{-s}\min\left\{\frac{4}{\sqrt{2\pi(e^{2s}-1)}}, \frac{||X_i||}{e^{2s}-1}\frac{2}{\pi}\sqrt{1+2\pi e^{-1}}\right\}ds\right] \tag{4.9}$$

Because the expression for ζ_i will be similar, we derive a general bound for

$$\int_0^\infty e^{-s}\min\{\frac{1}{\sqrt{e^{2s}-1}}, \frac{\alpha}{e^{2s}-1}\}ds.$$

Note that the first term is less than the second term when $\sqrt{e^{2s}-1} < \alpha$, namely, when $s < \log\sqrt{\alpha^2+1}$. Further, it is straightforward to check that $\int\frac{e^{-s}}{\sqrt{e^{2s}-1}}ds = e^{-s}\sqrt{e^{2s}-1}$, and $\int\frac{e^{-s}}{e^{2s}-1}ds = e^{-s} - \log\frac{e^s+1}{\sqrt{e^{2s}-1}}$. Thus we evaluate

$$\begin{aligned}\int_0^\infty e^{-s}\min\{\frac{1}{\sqrt{e^{2s}-1}}, \frac{\alpha}{e^{2s}-1}\}ds &= \int_0^{\log\sqrt{\alpha^2+1}}\frac{e^{-s}}{\sqrt{e^{2s}-1}}ds + \alpha\int_{\log\sqrt{\alpha^2+1}}^\infty\frac{e^{-s}}{e^{2s}-1}ds \\ &= \frac{\alpha}{\sqrt{\alpha^2+1}} + \alpha\left[\log\frac{\sqrt{\alpha^2+1}+1}{\alpha} - \frac{1}{\sqrt{\alpha^2+1}}\right] \\ &= \alpha\log\frac{\sqrt{\alpha^2+1}+1}{\alpha} \leq \alpha\log\left(1+\frac{2}{\alpha}\right) \end{aligned} \tag{4.10}$$

We may thus bound $|\eta_i|$ from Equations 4.9 and 4.10 by setting $\alpha = \frac{1}{\sqrt{2\pi}}||X_i||\sqrt{1+2\pi e^{-1}}$. Since $\frac{2}{\pi}\sqrt{1+2\pi e^{-1}} < 1.16$ and $2\cdot\frac{4}{\sqrt{2\pi}}/(\frac{2}{\pi}\sqrt{1+2\pi e^{-1}}) < 2.76$, we have that

$$\begin{aligned}|\eta_i| &< 1.16E\left[||T\widetilde{X}_i||\cdot||\widetilde{X}_i||||X_i||\log\left(1+\frac{2.76}{||X_i||}\right)\right] \\ &= 1.16E\left[||TX_i||\cdot||X_i||\right]E\left[||X_i||\log\left(1+\frac{2.76}{||X_i||}\right)\right] \end{aligned} \tag{4.11}$$

We now turn to bounding the last term of Equation 4.5, whose expectation we have

denoted as ζ_i. Similarly to above, we have

$$\begin{aligned}
&\sum_{a,b,c=1}^{k} \int_0^1 (1-t) X_i(a) X_i(b) X_i(c) D_{abc} f_h(S_{-i} + t \cdot X_i)\, dt \\
&\qquad = \int_0^1 (1-t) D^3_{X_i} f_h(S_{-i} + t \cdot X_i)\, dt \\
&\qquad = \int_0^\infty \int_0^1 (1-t) D^3_{X_i} (g_s * \phi_{\sqrt{e^{2s}-1}})(S_{-i} + t \cdot X_i)\, dt\, ds \\
&\qquad = \int_0^\infty \int_0^1 (1-t) (D_{X_i} g_s * D^2_{X_i} \phi_{\sqrt{e^{2s}-1}})(S_{-i} + t \cdot X_i)\, dt\, ds \\
&\qquad = \int_0^\infty \int_{\mathbb{R}^k} D_{X_i} g_s(x) \int_0^1 (1-t) D^2_{X_i} \phi_{\sqrt{e^{2s}-1}}(S_{-i} + t \cdot X_i - x)\, dt\, dx\, ds \\
&\qquad \leq ||TX_i|| e^{-s} \int_0^\infty \int_{\mathbb{R}^k} \left| \int_0^1 (1-t) D^2_{X_i} \phi_{\sqrt{e^{2s}-1}}(t \cdot X_i + x)\, dt \right| dx\, ds
\end{aligned}$$

As above, if we take an orthonormal basis that includes a vector in the direction of X_i then we can decompose $D^2_{X_i} \phi_{\sqrt{e^{2s}-1}}$ into the product of the corresponding expression for a univariate Gaussian in the direction of X_i, and univariate Gaussians along all the other basis directions. Thus, if we let $\bar{\phi}_r$ denote the univariate version of ϕ_r, namely, $\bar{\phi}_r(x) = \frac{1}{r \cdot \sqrt{2\pi}} e^{-\frac{x^2}{2r^2}}$, then the above integral over $\mathbb{R}^k$ equals exactly

$$||X_i||^2 \int_{-\infty}^{\infty} \left| \int_0^1 (1-t) \bar{\phi}''_{\sqrt{e^{2s}-1}}(x + ||X_i|| t)\, dt \right| dx \tag{4.12}$$

As above, we bound this expression in two ways. First, we bound it by moving the absolute values inside the integral, swapping the order of integration, and then making the substitution $y = x + ||X_i|| t$ to yield

$$||X_i||^2 \int_0^1 \int_{-\infty}^{\infty} \left| (1-t) \bar{\phi}''_{\sqrt{e^{2s}-1}}(y) \right| dy\, dt$$

The integral may thus be expressed as the product of separate integrals over t and y: since $\int_0^1 1 - t\, dt = \frac{1}{2}$, and as we computed above, $\int_{-\infty}^{\infty} |\bar{\phi}''_{\sqrt{e^{2s}-1}}(y)|\, dy = \frac{4e^{-1/2}}{(e^{2s}-1)\sqrt{2\pi}}$, we have that Equation 4.12 is at most $||X_i||^2 \frac{2e^{-1/2}}{(e^{2s}-1)\sqrt{2\pi}}$.

For the second bound, we first note that we may simplify slightly by replacing $(1-t)$ by t in Equation 4.12 (this is the change of variables $t \to (1-t)$, $x \to -x - ||X_i||$, relying on the fact that $\bar{\phi}''$ is symmetric about 0). It will be convenient to consider the inner integral as being over $\mathbb{R}$ instead of just $[0,1]$, and we thus introduce the notation $(t)_{[0,1]}$ to represent

t if $t \in [0,1]$ and 0 otherwise. Thus we bound Equation 4.12 as

$$\begin{aligned}
&||X_i||^2 \int_{-\infty}^{\infty} \left| \int_{-\infty}^{\infty} (t)_{[0,1]} \bar{\phi}''_{\sqrt{e^{2s}-1}}(x + ||X_i||t)\, dt \right| dx \\
&\quad = ||X_i||^2 \int_{-\infty}^{\infty} \left| \int_{-\infty}^{\infty} \left((t)_{[0,1]} - \left(-\frac{x}{||X_i||} \right)_{[0,1]} \right) \bar{\phi}''_{\sqrt{e^{2s}-1}}(x + ||X_i||t)\, dt \right| dx \\
&\quad \leq ||X_i||^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left| \left((t)_{[0,1]} - \left(-\frac{x}{||X_i||} \right)_{[0,1]} \right) \bar{\phi}''_{\sqrt{e^{2s}-1}}(x + ||X_i||t) \right| dx\, dt \\
&\quad = ||X_i||^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left| \left((t)_{[0,1]} - \left(t - \frac{y}{||X_i||} \right)_{[0,1]} \right) \bar{\phi}''_{\sqrt{e^{2s}-1}}(y) \right| dy\, dt \\
&\quad = ||X_i||^2 \int_{-\infty}^{\infty} \left| \bar{\phi}''_{\sqrt{e^{2s}-1}}(y) \right| \int_{-\infty}^{\infty} \left| (t)_{[0,1]} - \left(t - \frac{y}{||X_i||} \right)_{[0,1]} \right| dt\, dy
\end{aligned}$$

where the first equality holds since ϕ'' has integral 0, and hence we can add any multiple of it (independent of t) to the inner integral; the second equality is just the substitution $x \to y - ||X_i||t$.

To bound this integral, we note the general fact that, if a function f has total variation a, then $\int_{-\infty}^{\infty} |f(x) - f(x-b)|\, dx \leq a|b|$. Thus since the function $(t)_{[0,1]}$ has total variation 2, the inner integral is bounded by $2\frac{y}{||X_i||}$. Since $\bar{\phi}''_r$ crosses 0 at $\pm r$, and integration by parts yields $\int y\bar{\phi}''_r(y)\, dy = y\bar{\phi}'_r(y) - \int \bar{\phi}'_r(y)\, dy = -\bar{\phi}_r(y)(1 + \frac{y^2}{r^2})$ and hence $\int_{-\infty}^{\infty} |y\bar{\phi}''_r(y)|\, dy = -2\int_0^r y\bar{\phi}''_r(y)\, dy + 2\int_r^{\infty} y\bar{\phi}''_r(y) = -2\bar{\phi}_r(0) + 8\bar{\phi}_r(r) = \frac{8e^{-1/2}-2}{r\cdot\sqrt{2\pi}}$ we may thus bound Equation 4.12 by $||X_i||\frac{16e^{-1/2}-4}{\sqrt{2\pi(e^{2s}-1)}}$.

Thus, similarly to above, we have

$$|\zeta_i| \leq ||TX_i|| \cdot ||X_i|| \int_0^{\infty} e^{-s} \min\left\{ \frac{16e^{-1/2}-4}{\sqrt{2\pi(e^{2s}-1)}}, \frac{||X_i|| \cdot 2e^{-1/2}}{(e^{2s}-1)\sqrt{2\pi}} \right\} ds.$$

Since $\frac{2e^{-1/2}}{\sqrt{2\pi}} < 0.49$ and $2 \cdot \frac{16e^{-1/2}-4}{\sqrt{2\pi}} \Big/ \frac{2e^{-1/2}}{\sqrt{2\pi}} < 9.41$, we have from Equation 4.10 that $|\zeta_i| < 0.49 \cdot E[||TX_i|| \cdot ||X_i||^2 \log(1 + \frac{9.41}{||X_i||})]$. Combining this and Equation 4.11 yields the theorem. □

4.4 A Central Limit Theorem for Generalized Multinomial Distributions

In this section we leverage the central limit theorem of Theorem 4.1 to show our second central limit theorem that bounds the *total variational distance*, denoted by D_{tv} between

generalized multinomial distributions and (discretized) Gaussian distributions. While Theorem 4.1 certainly applies to generalized multinomial distributions, the goal of this section is to derive a bound in terms of the rather more stringent total variational distance. The main hurdle is relating the "smooth" nature of the Gaussian distribution and earthmover distance metric to the "discrete" setting imposed by a total variational distance comparison with the discrete generalized multinomial distribution.

The analysis to compare a Gaussian to a generalized multinomial distribution proceeds in two steps. Given the earthmover distance bound provided by Theorem 4.1, we first smooth both sides via convolution with a suitably high-variance distribution to convert this bound into a total variational distance bound, albeit not between the original two distributions but between convolved versions of them. The second step is via a "deconvolution" lemma (Lemma 4.6) that relies on the unimodality in each coordinate of generalized multinomial distributions.

We begin by showing this unimodality via a result about homogeneous polynomials that generalizes the classic Newton inequalities.

Given a polynomial p in k variables, and a nonnegative integer vector $v \in \mathbb{Z}^k$, we denote by $p_{(v)}$ the coefficient of the term $x_1^{v(1)} x_2^{v(2)} \cdot \ldots \cdot x_k^{v(k)}$ in p.

Fact: Multivariate Newton Inequalities (Fact 1.10:2 of [64]). *Given a homogeneous polynomial p of degree n in k variables, with nonnegative coefficients, if it is the case that for any complex $x_1, \ldots, x_k$ with strictly positive real parts, $p(x_1, \ldots, x_k) \neq 0$, then for any nonnegative integer vector v and letting $\Delta = (1, -1, 0, \ldots, 0) \in \mathbb{Z}^k$, we have $p_{(v)}^2 \geq p_{(v+\Delta)} p_{(v-\Delta)}$.*

(We note that the actual result from [64], in analogy with Newton's inequalities, is tighter by a factor $\prod_i v(i)!^2 / \prod_i (v+\Delta)(i)!(v-\Delta)(i)! = \frac{v(1)v(2)}{(1+v(1))(1+v(2))}$, though for our purposes we need only the simpler bound.)

Definition 4.4. *A function $f : \mathbb{Z} \to \mathbb{R}^+$ is log-concave if its support is an interval, and $\forall i \in \mathbb{Z}, f(i)^2 \geq f(i-1)f(i+1)$.*

The logarithm of a log-concave function is concave (interpreting $\log 0$ as $-\infty$); thus any log-concave function is unimodal (i.e., monotonically increasing to the left of some point, and monotonically decreasing to the right). We note that we consider "unimodal" in the non-strict sense, so that, for example, the constant function is unimodal.

Lemma 4.5. *Generalized multinomial distributions are log-concave, and hence unimodal, in any coordinate.*

Proof. Given a generalized multinomial distribution parameterized by ρ, where ρ has n rows and k columns, we define $\bar{\rho}$ to be the matrix whose columns are indexed 0 through k, and which consists of ρ extended so that for each $i \in \{1, \ldots n\}$, $\sum_{j=0}^{k} \bar{\rho}(i, j) = 1$.

Let p be the homogeneous polynomial of degree n in k variables defined as $p(x_1, \ldots, x_k) = \prod_{i=1}^{n} (\bar{\rho}(i, 0)x_0 + \ldots + \bar{\rho}(i, k)x_k)$. We note that for any nonnegative integer vector v, the coefficient $p_{(v)}$ equals, by definition, the probability of drawing v from the multinomial distribution (ignoring the implicit "0th coordinate").

We invoke the multivariate Newton inequalities (with the coordinates renumbered as necessary) by noting that, first, p clearly has nonnegative coefficients, and second, if $x_0, \ldots, x_k$ are complex numbers with strictly positive real parts, then each term $(\bar{\rho}(i,0)x_0 + \ldots + \bar{\rho}(i,k)x_k)$ will have strictly positive real part, and hence be nonzero, which implies that $p(x_0, \ldots, x_k) \neq 0$. Thus the multivariate Newton inequalities imply that the multinomial distribution (with its "0th coordinate" ignored) is log-concave in its first coordinate; by symmetry, it is log-concave in every coordinate. □

Given this general structural result about the distributions at hand, we now construct the second ingredient of our proof, the "deconvolution" lemma. What this shows is that, given a convolution $f * g$ that closely approximates a third function h, we can leverage the unimodality of f under certain conditions to "deconvolve" by g and relate f and h directly. We will apply this univariate result in the proof of the central limit theorem by applying it inductively along lines in each of the k coordinate directions.

Lemma 4.6. *Given an integer $\ell > 0$, a unimodal function $f : \mathbb{Z} \to \mathbb{R}^+$, a function $g : \{-\ell, -\ell+1 \ldots, \ell-1, \ell\} \to \mathbb{R}^+$ with $\sum_i g(i) = 1$, and an arbitrary bounded function $h : \mathbb{Z} \to \mathbb{R}^+$ then, letting $f * g$ denote the convolution of f and g, we have*

$$\sum_{i=-\infty}^{\infty} |f(i) - h(i)| \leq 10\ell \left(\sup_i h(i) \right) + \sum_{i=-\infty}^{\infty} |(f * g)(i) - h(i)|.$$

Proof. Assume that we have scaled f and h so that $\sup_i h(i) = 1$. Let f^- denote the function that is the (pointwise) minimum of f and 1, and let f^+ denote $f - f^-$. We note that f^+ and f^- are unimodal. For the following inequality, we let $[[0, j]]$ denote the set of integers $\{0, \ldots, j-1\}$ when $j > 0$, the set $\{j, \ldots, -1\}$ when $j < 0$, and the empty set when $j = 0$: by the definition of convolution, two applications of the triangle inequality, and a rearrangement of terms we have

$$\begin{aligned}
\sum_{i=-\infty}^{\infty} |f^-(i) - (f^- * g)(i)| &= \sum_{i=-\infty}^{\infty} \left| \sum_{j=-\ell}^{\ell} g(j)(f^-(i) - f^-(i-j)) \right| \\
&\leq \sum_{i=-\infty}^{\infty} \sum_{j=-\ell}^{\ell} g(j) |f^-(i) - f^-(i-j)| \\
&\leq \sum_{i=-\infty}^{\infty} \sum_{j=-\ell}^{\ell} \sum_{k \in [[0,j]]} g(j) |f^-(i-k) - f^-(i-k+1)| \\
&= \left(\sum_{j=-\ell}^{\ell} g(j)|j| \right) \sum_{i=-\infty}^{\infty} |f^-(i) - f^-(i+1)| \\
&\leq \ell \sum_{i=-\infty}^{\infty} |f^-(i) - f^-(i+1)|.
\end{aligned}$$

Since f^- is unimodal and bounded between 0 and 1, $\sum_i |f^-(i) - f^-(i+1)| \leq 2$, and we thus bound the above inequality by 2ℓ.

We note that since f is unimodal, it exceeds 1 on a contiguous (possibly empty) interval, which we denote $[u, v]$. Since $f * g = f^- * g + f^+ * g$, we have the triangle inequality $|(f * g)(i) - h(i)| \leq |(f^+ * g)(i)| + |(f^- * g)(i) - h(i)|$. Since $f^- * g = 1$ on the interval $[u + \ell, v - \ell]$, and $f^+ * g$ is confined to the interval $[u - \ell, v + \ell]$, then we actually have equality everywhere *except* the intervals $[u - \ell, u + \ell - 1]$ and $[v - \ell + 1, v + \ell]$. On these intervals, we consider the reverse inequality (another triangle inequality) $|(f * g)(i) - h(i)| \geq |(f^+ * g)(i)| - |(f^- * g)(i) - h(i)|$ which, since $(f^- * g)(i) \in [0, 1]$, we bound as being at least $|(f^+ * g)(i)| + |(f^- * g)(i) - h(i)| - 2$ on these intervals. Thus

$$\begin{aligned}
\sum_{i=-\infty}^{\infty} |(f * g)(i) - h(i)| &\geq \sum_{i=-\infty}^{\infty} |(f^+ * g)(i)| + \sum_{i=-\infty}^{\infty} |(f^- * g)(i) - h(i)| + \sum_{i=u-\ell}^{u+\ell-1} (-2) \\
&\qquad + \sum_{i=v-\ell+1}^{v+\ell} (-2) \\
&= -8\ell + \sum_{i=-\infty}^{\infty} |f^+(i)| + \sum_{i=-\infty}^{\infty} |(f^- * g)(i) - h(i)| \\
&\geq -10\ell + \sum_{i=-\infty}^{\infty} |f^+(i)| + \sum_{i=-\infty}^{\infty} |f^-(i) - h(i)| \\
&= -10\ell + \sum_{i=-\infty}^{\infty} |f(i) - h(i)|,
\end{aligned}$$

where the last inequality is what we proved above, and the last equality is true term-by-term since the region where f^+ is nonzero is exactly the region where $f^-(i) = 1 \geq h(i)$, and thus we have the lemma. □

We are now equipped to assemble the components and prove the central limit theorem. Our central limit theorem related the generalized multinomial distribution to the "discretized" version of the Gaussian distribution of identical mean and covariance (see Definition 4.3). For convenience, we restate the theorem below:

Theorem 4.2 *Given a generalized multinomial distribution M^ρ, with k dimensions and n rows, let μ denote its mean and Σ denote its covariance matrix, then*

$$D_{tv}\left(M^\rho, \mathcal{N}^{disc}(\mu, \Sigma)\right) \leq \frac{k^{4/3}}{\sigma^{1/3}} \cdot 2.2 \cdot (3.1 + 0.83 \log n)^{2/3},$$

where σ^2 is the minimum eigenvalue of Σ.

Proof. Adopting the notation of Theorem 4.1, we let Z_i denote the distribution induced by the ith row of ρ, that is, a distribution over $(0, \ldots, 0)$, $(1, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, $\ldots$,

$(0, \ldots, 0, 1)$, where M^ρ is thus distributed as $\sum_{i=1}^n Z_i$. Since the range of Z_i has diameter $\sqrt{2}$, each sample from Z_i is within $\sqrt{2}$ of its mean. Theorem 4.1 implies that $d_W\left(M^\rho, \mathcal{N}(\mu, \Sigma)\right) < k\sqrt{2}(2.7 + 0.83 \log n)$.

For notational convenience, let $\phi = \mathcal{N}(\mu, \Sigma)$, and let $\phi^{disc} = \mathcal{N}^{disc}(\mu, \Sigma)$ denote the corresponding discretized Gaussian of Definition 4.3. We note that, since every point in $\mathbb{R}^k$ is within distance $\frac{\sqrt{k}}{2}$ from a lattice point, $d_W(\phi, \phi^{disc}) \leq \frac{\sqrt{k}}{2} \leq \frac{k}{2}$. Thus the triangle inequality yields $d_W(M^\rho, \phi^{disc}) < k\sqrt{2}(3.1 + 0.83 \log n)$.

Given positive integers d, ℓ, let $R_{d,\ell}$ denote the distribution over $\mathbb{Z}^k$ where the first d coordinates are each independent draws from the binomial distribution $B(2\ell, \frac{1}{2})$, shifted by $-\ell$ so as to lie in $\{-\ell, \ldots, \ell\}$ and the rest of the coordinates are 0.

The binomial distribution $B(2\ell, \frac{1}{2})$ is unimodal, with the probability of hitting its mode bounded by $\frac{1}{\sqrt{\pi\ell}}$, which implies that the total variational distance between $B(2\ell, \frac{1}{2})$ and a version shifted by an integer c is at most $\frac{c}{\sqrt{\pi\ell}}$; thus the same holds for shifting $R_{k,\ell}$ by c along any coordinate axis, since each coordinate is distributed as an independent (shifted) copy of $B(2\ell, \frac{1}{2})$. By the triangle inequality, if we shift by an integer vector x, then the total variational distance is at most $\frac{1}{\sqrt{\pi\ell}} \sum_{i=1}^k |x(i)|$. The Cauchy-Schwarz inequality yields $\sum_{i=1}^k |x(i)| \leq \sqrt{k}||x||$, yielding a bound on the total variational distance of $\frac{\sqrt{k}}{\sqrt{\pi\ell}}||x||$.

We are now prepared to make the key transformation from stating our central limit theorem in terms of earthmover distance, to stating a central limit theorem for total variational distance.

Consider a particular component of a "scheme to move earth" from M^ρ to ϕ^{disc}; for example, "move probability mass m from x to y". The bound of the previous paragraph implies that the total variational distance between copies of $R_{k,\ell}$ centered at x, and at y, respectively, is at most $\frac{\sqrt{k}}{\sqrt{\pi\ell}}||x - y||$. Thus, in this sense, convolution by $R_{k,\ell}$ converts earthmover bounds to total variational distance bounds, losing a factor of $\frac{\sqrt{k}}{\sqrt{\pi\ell}}$. We conclude that

$$d_{TV}(M^\rho * R_{k,\ell},\ \phi^{disc} * R_{k,\ell}) \leq \frac{\sqrt{2k} \cdot k}{\sqrt{\pi\ell}}(3.1 + 0.83 \log n). \tag{4.13}$$

Were it not for the convolution by $R_{k,\ell}$ in the above expression, we could conclude here. We now consider how to "remove" these convolutions.

Consider ϕ (not ϕ^{disc}) shifted by a vector x. Since ϕ has variance at least σ^2 in every direction, then, when restricted to any line in the direction of x, ϕ will be a univariate normal distribution of variance at least σ^2. We may thus bound the total variational distance of ϕ and its shifted version by the corresponding univariate bound. Note that the univariate Gaussian is unimodal, and thus the total variational distance between itself and a version shifted $||x||$ is at most $||x||$ times the pdf at its mode, which is at most $\frac{1}{\sigma\sqrt{2\pi}}$. Applying this bound for each x drawn from $R_{k,\ell}$, where for each such x, $||x|| \leq \ell\sqrt{k}$ we have $d_{TV}(\phi, \phi * R_{k,\ell}) \leq \frac{\ell\sqrt{k}}{\sigma\sqrt{2\pi}}$. Since $R_{k,\ell}$ is a distribution on the lattice points, taking $\phi * R_{k,\ell}$ and rounding samples to the nearest integer is distributed identically to $\phi^{disc} * R_{k,\ell}$. Thus we have $d_{TV}(\phi^{disc}, \phi^{disc} * R_{k,\ell}) \leq \frac{\ell\sqrt{k}}{\sigma\sqrt{2\pi}}$,

yielding, by the triangle inequality, $d_{TV}(M^\rho * R_{k,\ell}, \phi^{disc}) \leq \frac{\sqrt{2k}\cdot k}{\sqrt{\pi\ell}}(3.1 + 0.83\log n) + \frac{\ell\sqrt{k}}{\sigma\sqrt{2\pi}}$

Having "removed" the second convolution by $R_{k,\ell}$ in Equation 4.13, we now turn to the first. Recalling that $R_{i,\ell}$ is the distribution whose first i coordinates are distributed as (shifted) versions of the binomial distribution $B(2\ell, \frac{1}{2})$ where the remaining $k-i$ coordinates are 0, we aim to "deconvolve" by this binomial, coordinate-by-coordinate, so that when i reaches 0 we will have the desired central limit theorem. Our tool is Lemma 4.6, which we will use to show by induction that for every $i \in \{0, \ldots, k\}$ we have

$$d_{TV}(M^\rho * R_{i,\ell}, \phi^{disc}) \leq (k-i)\frac{5\ell}{\sigma\sqrt{2\pi}} + \frac{\ell\sqrt{k}}{\sigma\sqrt{2\pi}} + \frac{\sqrt{2k}\cdot k}{\sqrt{\pi\ell}}(3.1 + 0.83\log n) \tag{4.14}$$

Letting $i = 0$ and $\ell = \frac{1}{6^{2/3}}\sigma^{2/3}k^{1/3}(3.1 + 0.83\log n)^{2/3}$ yields the theorem.

To prove Equation 4.14, we assume as our induction hypothesis that it holds for some $i > 0$ and will derive it for $i-1$. Consider $M^\rho * R_{i,\ell}$, $M^\rho * R_{i-1,\ell}$, and ϕ^{disc} restricted to a line L in the ith coordinate direction. We note that the pdf of ϕ restricted to this line will be a multiple of a univariate normal distribution of variance at least σ^2, and thus has the property that its maximum is at most $\frac{1}{\sigma\sqrt{2\pi}}$ times its integral; as this is true for every such line, it is also true in expectation for a distribution of lines, and is thus true for the distribution of lines that will be rounded to L. Thus ϕ^{disc} restricted to the line L has the property that its maximum is at most $\frac{1}{\sigma\sqrt{2\pi}}$ times its total. With a view towards applying Lemma 4.6, we note that $R_{i-1,\ell}$ is itself a generalized multinomial distribution, and hence so is $M^\rho * R_{i-1,\ell}$, from which we invoke Lemma 4.5 to see that $M^\rho * R_{i-1,\ell}$ is unimodal along L. We thus apply Lemma 4.6 with f equal to the restriction of $M^\rho * R_{i-1,\ell}$ to L, g equal to the binomial $B(2\ell, \frac{1}{2})$ shifted so as to have support on $\{-\ell, \ldots, \ell\}$, and h equal to the restriction of ϕ^{disc} to L. Since $f * g$ is the restriction of $M^\rho * R_{i,\ell}$ to L, we conclude that,

$$\begin{aligned}\sum_{x\in L} |(M^\rho * R_{i-1,\ell})(x) - \phi^{disc}(x)| &\leq 10\ell\left(\max_{x\in L}\phi^{disc}(x)\right) + \sum_{x\in L}|(M^\rho * R_{i,\ell})(x) - \phi^{disc}(x)| \\ &\leq \frac{10\ell}{\sigma\sqrt{2\pi}}\sum_{x\in L}\phi^{disc}(x) + \sum_{x\in L}|(M^\rho * R_{i,\ell})(x) - \phi^{disc}(x)|\end{aligned}$$

Summing over all such lines L yields the induction (since total variational distance has a normalizing factor of $\frac{1}{2}$). □

Chapter 5

Lower Bounds for Property Estimation

In this chapter we leverage the central limit theorem for "generalized multinomial distributions", Theorem 4.2, to prove an information theoretic lower bound for property estimation which shows that the estimators for entropy, distinct elements/support size, and total variational distance described in Chapter 3 are optimal, up to constant factors for any sufficiently small constant error.

The connection between our central limit theorem for generalized multinomial distributions, and estimating symmetric properties of distributions, such as entropy and support size, is that generalized multinomial distributions capture the distribution of fingerprints, $(\mathcal{F}_1, \mathcal{F}_2, \ldots)$, where $\mathcal{F}_i$ is the number of domain elements for which we see i representatives in a sample. To see why, recall that we may assume that we draw $k' \leftarrow Poi(k)$ and then draw a sample of size k'. In such a setting, the number of times each domain element is observed is independent of the number of times each other domain element is observed. Thus the distribution of $Poi(k)$-sample fingerprints is given as the generalized multinomial distribution (see Definition 4.2) defined by the matrix M, where there is one row of M for each domain element, and entry $M_{i,j}$ is the probability that the ith domain element occurs exactly j times in a $Poi(k)$–sized sample.

Our central limit theorem allows us to cleanly reason about the total variational distance between these distributions of fingerprints. Specifically, this will allow us to argue that there are pairs of very different distributions p, p'—different in terms of entropy, or support size, for example—such that there is small total variational distance between the distribution of fingerprints derived from samples of size k drawn from p and the distribution of fingerprints derived from samples of size k drawn from p'.

Such a pair of distributions is not, by itself, a lower bound instance; the labels of the data points (which are not represented in the fingerprints) will be helpful in distinguishing whether the sample was drawn from p or p'. We can, however, easily construct a lower bound instance from the pair of distributions, p, p'. Let n be an upper bound on the support size of distributions, and without loss of generality assume that p and p' are distributions over

$[2n]$. Consider the ensemble of distributions $\mathcal{T}$ defined by the following process: select a random permutation π of $[2n]$, and with probability $1/2$ output p_π, and with the remaining probability output p'_π, where p_π is defined to be the distribution obtained by permuting the labels of distribution p according to permutation π, and analogously for p'_π. The lower bound instance derived from the pair of distributions p, p' will simply be the task of estimating the property value of a distribution drawn from the ensemble $\mathcal{T}$. In the case of entropy, for example, to show that no algorithm on samples of size k samples can estimate the entropy of a distribution to within error $\epsilon = \frac{|H(p)-H(p')|}{2}$ with probability of success at least $1-\delta$, it suffices to show that no algorithm when given a sample of size k from a distribution drawn from $\mathcal{T}$ can distinguish whether the sample was drawn from a distribution p_π obtained from p versus a distribution p'_π obtained from p' with probability more than $1-\delta$. As we are permuting the supports via a random permutation, the labels are meaningless, and the only useful information from the samples is the fingerprint, to which we may then apply our central limit theorem.

Our information theoretic lower bound is the following:

Theorem 5.1. *For any positive constant $\phi < \frac{1}{4}$, there exists a pair of distributions p^+, p^- that are $O(\phi|\log\phi|)$-close in the relative earthmover distance, respectively, to the uniform distributions on n and $\frac{n}{2}$ elements, but whose fingerprints derived from $k = \frac{\phi}{32} \cdot \frac{n}{\log n}$-sized samples cannot be distinguished with any constant probability greater than $1/2$.*

Because of the continuity of entropy, and support size (of a distribution all of whose domain elements occur with probability at least $1/n$) with respect to relative earthmover distance (Fact 3.5), the above theorem yields the following corollary:

Corollary 5.1. *For any constant probability of failure $\delta < 1/2$, for sufficiently large n the following hold:*

- *For any constant $\epsilon < \frac{\log 2}{2}$, any algorithm that estimates the entropy of distributions of support at most n to within additive error ϵ with probability of success at least $1-\delta$ requires a sample of size $\Omega(\frac{n}{\log n})$.*
- *For any constant $\epsilon < \frac{1}{4}$, any algorithm that estimates the support size of distributions all of whose domain elements occur with probability at least $1/n$ to within additive error ϵn with probability of success at least $1-\delta$ requires a sample of size $\Omega(\frac{n}{\log n})$.*

Further, by choosing a positive $\epsilon < 1$ and then constructing the distributions $p^+_\epsilon, p^-_\epsilon$ that, with probability ϵ draw samples from p^+, p^- respectively and otherwise return another symbol, $\perp$, we note that the entropy gap between p^+_ϵ and p^-_ϵ is an ϵ fraction of what it was originally, and further that distinguishing them requires a factor $\frac{1}{\epsilon}$ larger sample. That is,

Corollary 5.2. *For large enough n and small enough ϵ, the sample complexity of estimating entropy to within ϵ grows as $\Omega(\frac{n}{\epsilon \log n})$.*

For the task of estimating total variational distance between a pair of distributions, in Section 5.6 we leverage Theorem 5.1 to show the following lower bound:

Theorem 5.2. *For any constants $0 < a < b < \frac{1}{2}$, and probability of failure $\delta < 1/2$, for sufficiently large n, given two samples, drawn from a pair of distributions of support at most n, distinguishing whether their total variational distance (ℓ_1 distance) is less than a or greater than b with probability of success at least $1 - \delta$, requires samples of size $\Omega(\frac{n}{\log n})$.*

Both these lower bounds improve upon the previous best lower bound of $n/2^{O(\sqrt{\log n})}$ shown by P. Valiant [131].

We will construct the p^+, p^- of Theorem 5.1 explicitly, via Laguerre polynomials. The key intuition to the construction is that the most naive way to try to distinguish samples from p^+ versus from p^- is via their fingerprint expectations. So the first step to constructing indistinguishable distributions is to ensure that the corresponding vectors of fingerprint expectations are approximately equal. As we show, this is essentially the *only* step, though proving that the construction is this "easy" requires considerable work.

5.1 Technique Overview: Fourier Analysis, Hermite Polynomials, "Fattening", and the Laguerre construction

As introduced in Definition 4.2, generalized multinomial distributions capture the distribution of fingerprints induced by drawing a $Poi(k)$–sample from a given distribution. And thus the final step of the proof that p^+ and p^- are indistinguishable in $Poi(k)$-samples will be to apply the central limit theorem for generalized multinomial distributions (Theorem 4.2) to the distributions of fingerprints of p^+, p^- respectively, approximating each as a discretized Gaussian. This will be sufficient provided a) the Gaussians are sufficiently similar, and b) the total variational distance bound when Theorem 4.2 is applied is suitably small. We consider each part in turn, and conclude with the intuition behind the Laguerre construction of the lower bound distributions p^+, p^-. Throughout we admit the slight abuse of notation and use p^+, p^- to refer both to the distributions, as well as to the histograms of the distributions.

Similar Expectations Induce Similar Covariances

For two Gaussians to be statistically close, three things should hold: the Gaussians have similar expectations; the Gaussians have similar covariances; and the minimum eigenvalue of the covariance matrix must be large. We begin by describing the intuition for the somewhat surprising fact that, in the present setting, similar expectations induce similar covariances.

Recall the effect of a single element of probability x on the distribution of fingerprints: for each integer $i > 0$, with probability $poi(xk, i)$, that element will occur i times in the

sample and hence end up incrementing the ith fingerprint entry by 1. Thus the contribution of this element to the expectation of the ith fingerprint entry equals $poi(xk, i)$.

Similarly, since covariance adds for sums of independent distributions, we may compute the contribution of an element of probability x to the (i, j)th entry of the fingerprint covariance matrix, which we compute here for the case $i \neq j$. The covariance of random variables X, Y is $E[XY] - E[X]E[Y]$; since in our case X represents the indicator random variable for the event that the element is sampled i times, and Y represents the indicator for the event that it is sampled j times, $E[XY] = 0$ as they can never both occur. Thus the contribution to the covariance is

$$-poi(xk, i)poi(xk, j) = -\frac{(xk)^{i+j}}{e^{2xk}i!j!} = -\frac{\binom{i+j}{i}}{2^{(i+j)}}poi(2xk, i + j).$$

Our claim that similar expectations imply similar covariances may now be rephrased as: each "skinny Poisson" function $poi(2xk, \ell)$ can be approximated as a linear combination of "regular Poisson" functions $\sum_i \alpha_{i,\ell} poi(xk, i)$, with small coefficients. Specifically, the coefficients $\alpha_{i,\ell}$ allow one to approximate the fingerprint covariances as a linear function of the fingerprint expectations; thus if two distributions have similar fingerprint expectations, then they must also have similar fingerprint covariances, since the covariances can be expressed as functions of the expectations.

We show that one can approximate such a "skinny Poisson" to within ϵ as a sum of regular Poissons using coefficients of total magnitude (roughly) no more than $\frac{1}{\epsilon}$. Intuitively, this is true for exactly the same reasons that the analogous claim holds for Gaussians. However, as opposed to the relatively simple case of Gaussians, proving this claim is perhaps the most technical part of this chapter, making heavy use of Hermite polynomials in Fourier space.

CLT Performance

If we apply Theorem 4.2 to the distribution of the first m fingerprint entries, and the covariance matrix of the distribution of these fingerprint entries has minimum eigenvalue σ^2, then the resulting bound on the total variational distance is $\frac{m^{4/3}}{\sigma^{1/3}}$ times logarithmic factors. Since σ^2 is never going to exceed $O(k)$, we clearly cannot use $m = k$. That is, we must apply the central limit theorem to only a small subset of the fingerprints entries. Additionally, we must ensure that σ^2 is big for this portion—intuitively that the distribution of these fingerprints is "fat in every direction".

Set $m = \log k$. We assume that p^+ and p^- are constructed so as to be supported on probabilities at most $\frac{\log k}{8k}$, and have similar fingerprint expectations and covariances. This bound of $\frac{\log k}{8k}$ ensures that we will almost never see any element of p^+ or p^- more than $\log k$ times; that is, the portion of the fingerprint below m "captures the whole story". However, if we were to try to apply the central limit theorem at this stage, the bound would be horrendous because the variance in the higher fingerprints (say the mth), is tiny. Thus we "fatten" the distributions of fingerprints by smearing a small $(1/polylog(k))$ amount of the

probability mass in p^+ and p^- uniformly among probabilities, up to m/k. Because we fatten p^+ and p^- *identically*, their fingerprint expectations and covariances still closely match. Given the fattened pair of distributions, we can now obtain satisfactory bounds from our central limit theorem. To complete the argument, we make use of the natural coupling of the portions of the fingerprints above m, stemming from the identical fattened portions of the distributions p^+, p^-.

Thus the Hermite polynomial argument guarantees matching covariances; "fattening" in conjunction with our central limit theorem for generalized multinomial distributions guarantees all the rest. What remains is to construct p^+, p^- with matching fingerprint expectations.

The Laguerre Construction

We will construct the pair of histograms, p^+, p^- explicitly, via Laguerre polynomials. We begin by letting p^+, p^- be the uniform distributions over support n and $n/2$, respectively. We then modify p^+, p^- by transferring some of the probability mass to make elements with higher probabilities, so as to ensure that the fingerprint expectations of $Poi(k)$ sized samples from p^+ and p^- roughly agree.

The condition that the expected ith fingerprint entries of p^+ and p^- agree is simply that $\sum_{x:p^+(x)\neq 0} p^+(x) poi(kx, i) = \sum_{x:p^-(x)\neq 0} p^-(x) poi(kx, i)$. Equivalently, define the function $f(x) : [0, 1] \to \mathbb{R}$ by $f(x) = p^+(x) - p^-(x)$. The condition that p^+ and p^- have the same expected first j fingerprints can be expressed as $\sum_{x:f(x)\neq 0} f(x) poi(kx, i) = 0$, for all integers $i \leq j$. Since $poi(kx, i) := \frac{e^{-kx}k^i x^i}{i!}$, this condition is equivalent to the function $g(x) := f(x)e^{-kx}$ being *orthogonal* to polynomials of degree at most j. The following easy fact (proved in Section 5.3) outlines an approach to creating such a function.

Fact. *Given a polynomial P of degree $j + 2$ whose roots $\{x_i\}$ are real and distinct, letting P' be the derivative of P, then for any $\ell \leq j$ we have $\sum_{i=1}^{j+2} \frac{x_i^\ell}{P'(x_i)} = 0$.*

To construct $f(x)$, choose a polynomial $P(x) = (x - 1/n)(x - 2/n)\prod_{i=1}^{j}(x - r_i)$, for some set of j distinct values r_i, with $2/n < r_i < 1$, then let $g(x)$ be the function that is supported at the roots of P, and takes value $1/P'(x)$ for the $j + 2$ values of x for which $P(x) = 0$. To obtain $f(x)$, simply set $f(x) = g(x)e^{kx}$.

If we interpret the positive portion of $f(x)$ as p^+ and the negative portion as p^-, we will, by the above fact, have two histograms whose first j fingerprint expectations agree. Additionally, p^+ will have some probability mass at probability $1/n$, and p^- will have some probability mass at $2/n$.

The tricky part, however, is in picking the r_i so as to ensure that *most* of the probability mass of p^+ is on probability $1/n$, and most of the mass of p^- is on probability $2/n$. If this is not the case, then p^+ and p^- will not be close (in relative–earthmover distance) to the uniform distributions over n and $n/2$ elements, respectively and thus may have similar entropies, or support sizes, failing us as a lower bound. Further complicating this task, is that whatever weight is at $x > 2/n$ in $g(x)$, ends up being multiplied by e^{kx}. To offset this

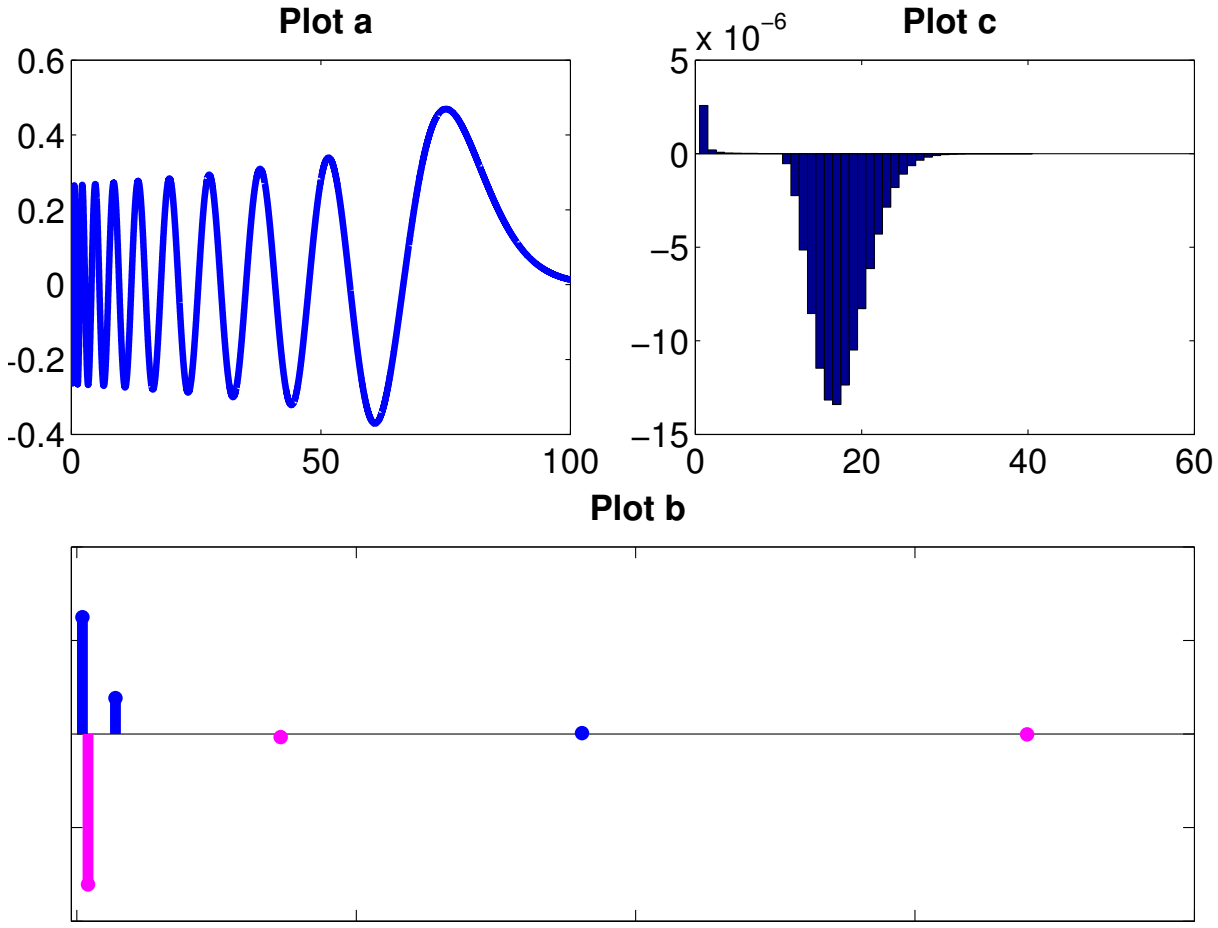

Figure 5.1: a) The 10th Laguerre polynomial, multiplied by $e^{-x/2}x^{1/4}$, illustrating that it behaves as $a \cdot e^{x/2}x^{-1/4} \cdot \sin(b \cdot \sqrt{x})$ for much of the relevant range.
b) $f(x)$, representing histograms $p^+(x), p^-(x)$ respectively above and below the x-axis.
c) The discrepancy between the first 40 fingerprint expectations of p^+, p^-; the first 10 expected fingerprint entries almost exactly match, while the discrepancy in higher fingerprint expectations is larger, though still bounded by $2 \cdot 10^{-5}$.

exponential increase, we should carefully choose the polynomial P so that the inverses of its derivatives, $1/P'(x)$, decay exponentially when evaluated at roots x of P. Such polynomials are hard to come by; fortunately, the Laguerre polynomials have precisely this property. See Figure 5.1 for a plot of our lower bound construction, and some insight into the structure of Laguerrre polynomials that may prove helpful.

In the remainder of this chapter, we make the above outline of the proof of Theorem 5.1 rigorous. We begin by showing that the covariance of the fingerprint entries can be expressed as a low-weight sum of the expectations of the fingerprint entries.

5.2 Linear Combinations of Poisson Functions

Our central limit theorem for generalized multinomial distributions is in terms of the first and second moments. Our lower bound construction will be a pair of distributions that have similar fingerprint expectations—i.e. similar first moments. In this section, we show the convenient fact that for fingerprints, "similar expectations" imply "similar covariances".

For a histogram h, the ith fingerprint expectation is $\sum_{x:h(x)\neq 0} h(x)\cdot\, poi(xk,i)$. Since, for random variables X, Y, their covariance equals $E[XY]-E[X]E[Y]$, and covariance sums for independent distributions, we have that the covariance of the ith and jth fingerprint entries, for $i \neq j$, equals $-\sum_{x:h(x)\neq 0} h(x) poi(xk,i) poi(xk,j)$. We simplify this product,

$$poi(xk,i)poi(xk,j) = \frac{(xk)^{i+j}e^{-2xk}}{i!j!} = 2^{-(i+j)}\binom{i+j}{i} poi(2xk, i+j),$$

to reveal a scaled version of a "squashed" version of the usual Poisson—that is, with $2xk$ instead of xk as the argument. The variance of the ith fingerprint entry may similarly be computed as $\sum_{x:h(x)\neq 0} h(x)\cdot\, (poi(xk,i) - poi(xk,i)^2)$, where $poi(xk,i)^2 = 2^{-2i}\binom{2i}{i} poi(2xk, 2i)$.

The point of the next result is that one may express "squashed" Poisson functions $poi(2xk,i)$ as linear combinations of Poisson functions $poi(xk,j)$; thus, since the expectations relative to (regular) Poisson functions $poi(xk,j)$ match for $p^{F+}_{\log k,\phi}$ and $p^{F-}_{\log k,\phi}$, the same will hold true (though with greater error) for the expectations relative to the "squashed" Poisson functions $poi(2xk,i)$, and hence the variances and covariances will also approximately match. We note that the Stone-Weierstrass theorem of Analysis trivially implies the convergence of this type of approximation; however, we require much stronger bounds on the relationship between the approximation factor and the coefficient sizes.

Lemma 5.3. *For any $\epsilon > 0$ and integer $i \geq 0$, one may approximate $poi(2x,i)$ as a linear combination $\sum_{j=0}^{\infty} \alpha(j) poi(x,j)$ such that*

1. *For all $x \geq 0$, $|poi(2x,i) - \sum_{j=0}^{\infty} \alpha(j) poi(x,j)| \leq \epsilon$; and*
2. *$\sum_{j=0}^{\infty} |\alpha(j)| \leq \frac{1}{\epsilon}\cdot 200 \max\{\sqrt[4]{i}, 24\log^{3/2}\frac{1}{\epsilon}\}$.*

We prove these strong bounds via a Fourier analysis approach relying on properties of Hermite polynomials.

To see the intuition both behind the result, and our approach, consider the above problem but with Poisson functions replaced by Gaussians, and all errors evaluated in the L_2 sense: for each $\epsilon > 0$ there exists a function K_ϵ of L_2 norm $\frac{1}{\epsilon}$ that when convolved with $\mathcal{N}(0,1)$ approximates $\mathcal{N}(0,\frac{1}{2})$ to within ϵ, in the L_2 sense. Let $\hat{K}_\epsilon$ be the ratio of the Fourier transforms of the pdfs of $\mathcal{N}(0,1)$ and $\mathcal{N}(0,\frac{1}{2})$ respectively, restricted to be 0 outside the interval $[-2\sqrt{\log\frac{1}{\epsilon}}, 2\sqrt{\log\frac{1}{\epsilon}}]$ and let K_ϵ be the inverse Fourier transform of $\hat{K}_\epsilon$. By Parseval's theorem, we may bound the L_2 norm of K_ϵ and the L_2 norm of the error $||\mathcal{N}(0,\frac{1}{2}), K_\epsilon * \mathcal{N}(0,1)||_2$, as the L_2 norms of their corresponding Fourier transforms. As the Fourier transform of K_ϵ is $\hat{K}_\epsilon$, which grows as $e^{x^2/4}$ but is zero outside the interval $[-2\sqrt{\log\frac{1}{\epsilon}}, 2\sqrt{\log\frac{1}{\epsilon}}]$, its L_2 norm is roughly $\frac{1}{\epsilon}$. Further, the Fourier transform of $K_\epsilon * \mathcal{N}(0,1)$ equals $\hat{K}_\epsilon \cdot \mathcal{N}(0,1)$, which by construction is exactly the Fourier transform of $\mathcal{N}(0,\frac{1}{2})$ within the interval $[-2\sqrt{\log\frac{1}{\epsilon}}, 2\sqrt{\log\frac{1}{\epsilon}}]$, and zero outside this interval. Since the Fourier transform of $\mathcal{N}(0,\frac{1}{2})$ decays as $e^{-x^2/4}$, the L_2 norm of the portion outside this interval is thus roughly ϵ, the desired bound.

Our proof of Lemma 5.3 relies on the substitution $x \to x^2$ to make the Poisson functions "look like" Gaussians, where the relationship between the transformed Poisson functions and Gaussians is controlled by properties of Hermite polynomials. Additionally, since we require an L_1 bound on the coefficients, as opposed to the L_2 bound that comes more naturally (via Parseval's theorem), instead of a sharp cutoff outside a designated interval (as we had done in the previous paragraph in our construction of K_ϵ), we must use a smooth cutoff function T, constructed as the convolution of the indicator function of an interval with a Gaussian of carefully chosen width.

Proof of Lemma 5.3. Let $g_k(x) := poi(x^2/2, k) = \frac{e^{-x^2/2}x^{2k}}{2^k k!}$. We consider the Fourier transform of $g_k(x)$, using the facts that the Fourier transform of $f(x) = e^{-x^2/2}$ is $\hat{f}(w) = e^{-w^2/2}$, and that if $f(x)$ is differentiable with Fourier transform $\hat{f}(w)$, then the Fourier transform of $\frac{d}{dx}f(x)$ is $-\mathbf{i}w\hat{f}(w)$:

$$\begin{aligned}\hat{g_k}(w) &= (-\mathbf{i})^{2k}\frac{d^{2k}}{dw^{2k}}\left(e^{-w^2/2}\right)\cdot\frac{1}{2^k k!} \\ &= \frac{(-1)^k e^{-w^2/2}H_{2k}(w)}{2^k k!},\end{aligned}$$

where $H_j(x) := (-1)^j e^{x^2/2}\frac{d^j}{dx^j}e^{-x^2/2}$, is the jth Hermite polynomial. Since Hermite polynomials form an orthogonal basis with respect to the Gaussian measure $e^{-w^2/2}$, and the even numbered Hermite polynomials are even functions while the odd numbered Hermite polynomials are odd functions, we have that the even numbered Hermite polynomials form an orthogonal basis with respect to the Gaussian measure $e^{-w^2/2}$ for the set of even functions. Incorporating the (square root) of the normalizing function $e^{-w^2/2}$ into the basis yields that the

set of functions $\hat{g}_k(w)e^{w^2/4}$ form an orthogonal basis for the set of even functions with respect to the *uniform* measure. In particular, since the set of functions $e^{-w^2/4}H_{2k}(w)/\sqrt{(2k)!\sqrt{2\pi}}$, sometimes known as the Hermite functions, are *orthonormal*, we define the orthonormal basis for even functions $\mathcal{G}_k(w) = \hat{g}_k(w)e^{w^2/4}\frac{2^k k!}{\sqrt{(2k)!\sqrt{2\pi}}}$.

Define $h_i(x) = g_i(x\sqrt{2})$. Recall our goal of approximating h_i as a linear combination of $\{g_j\}$. We work in Fourier space, and more specifically, to compute a linear combination of $\{\hat{g}_j\}$ which approximates $\hat{h}_i$, we multiply both sides by $e^{w^2/4}$ so that we may make use of the orthonormal basis $\{\mathcal{G}_j\}$. Explicitly, defining $T_{r,c}(w) = I_{[-r,r]}(w) * e^{-cw^2}\frac{\sqrt{c}}{\sqrt{\pi}}$, where $I_{[-r,r]}$ denotes the indicator function of the interval $[-r, r]$, for constants c and r to be specified later, and "$*$" denotes convolution, we use the basis $\{\mathcal{G}_j\}$ to express $T_{r,c}(w) \cdot e^{w^2/4} \cdot \hat{h}_i(w)$. Since $\{\mathcal{G}_j\}$ is orthonormal, the coefficient of $\mathcal{G}_j$ is exactly the inner product of $\mathcal{G}_j$ with this expression. That is, defining

$$\beta_{i,r,c}(j) := \int_{-\infty}^{\infty} T_{r,c}(w)\cdot e^{w^2/4}\cdot\hat{h}_i(w)\mathcal{G}_j(w)dw = \frac{2^j j!}{\sqrt{(2j)!\sqrt{2\pi}}}\int_{-\infty}^{\infty} T_{r,c}(w)\cdot e^{w^2/2}\cdot\hat{h}_i(w)\hat{g}_j(w)dw$$

we have expressed $T_{r,c}(w)\cdot e^{w^2/4}\cdot\hat{h}_i(w) = \sum_{j=0}^{\infty}\beta_{i,r,c}(j)\cdot\mathcal{G}_j(w)$. Invoking the definition of $\mathcal{G}_j$ and dividing both sides by $e^{w^2/4}$, we see that if we define

$$\alpha_{i,r,c}(j) := \frac{2^j j!}{\sqrt{(2j)!\sqrt{2\pi}}}\beta_{i,r,c}(j) = \frac{2^{2j}(j!)^2}{(2j)!\sqrt{2\pi}}\int_{-\infty}^{\infty} T_{r,c}(w)\cdot e^{w^2/2}\cdot\hat{h}_i(w)\hat{g}_j(w)dw, \tag{5.1}$$

then we have expressed

$$T_{r,c}(w)\cdot\hat{h}_i(w) = \sum_{j=0}^{\infty}\alpha_{i,r,c}(j)\cdot\hat{g}_j(w). \tag{5.2}$$

We bound $|\alpha_{i,r,c}(j)|$ in two ways from Equation 5.1.

We first note that since for a real number $a \neq 0$, the Fourier transform of a function $s(x) = f(a\cdot x)$ is $\hat{s}(w) = \frac{1}{a}\hat{f}(w/a)$, we have $\hat{h}_i(w) = \frac{1}{\sqrt{2}}\hat{g}_i(\frac{w}{\sqrt{2}})$. Further, we recall the basic fact that $|\mathcal{G}_j(w)|$ is maximized, over all j and w, when $j = w = 0$ (see [119] p. 190). Thus by definition of $\mathcal{G}_j(w)$, we bound $|e^{w^2/4}\hat{g}_j(w)| \leq \frac{\sqrt{(2j)!\sqrt{2\pi}}}{2^j j!}\mathcal{G}_0(0) = \frac{\sqrt{(2j)!}}{2^j j!}$, and thus since $\hat{h}_i(w) = \frac{1}{\sqrt{2}}\hat{g}_i(\frac{w}{\sqrt{2}})$, we have $|e^{w^2/8}\hat{h}_i(w)| \leq \frac{\sqrt{(2i)!}}{2^i i!\sqrt{2}}$. Thus we may bound

$$|\alpha_{i,r,c}(j)| \leq \frac{2^j j!}{\sqrt{(2j)!2\pi}}\frac{\sqrt{(2i)!}}{2^i i!\sqrt{2}}\int_{-\infty}^{\infty} T_{r,c}(w)\cdot e^{w^2/8}dw$$

To evaluate this integral, we use a trick which we will use twice more below: we "complete the square" and make the substitution (in this case) $s = t\frac{c}{c-1/8}$, yielding

$$\begin{aligned} T_{r,c}(w) \cdot e^{w^2/8} &:= \frac{\sqrt{c}}{\sqrt{\pi}} \int_{-r}^{r} e^{w^2/8} e^{-c(w-t)^2} dt \\ &= \frac{\sqrt{c}}{\sqrt{\pi}} \int_{-r}^{r} e^{-(w\sqrt{c-1/8} - tc/\sqrt{c-1/8})^2} e^{t^2 \frac{c}{8(c-1/8)}} dt \\ &= \frac{c-\frac{1}{8}}{\sqrt{c\pi}} \int_{-rc/(c-1/8)}^{rc/(c-1/8)} e^{-(w-s)^2 \cdot (c-1/8)} e^{s^2 \cdot \frac{c-1/8}{8c}} ds \\ &= \frac{c-\frac{1}{8}}{\sqrt{c\pi}} \left[I_{[-r\frac{c}{c-\frac{1}{8}}, r\frac{c}{c-\frac{1}{8}}]}(w) \cdot e^{\frac{c-1/8}{8c} \cdot w^2} \right] * e^{-(c-\frac{1}{8})w^2}. \end{aligned}$$

We may thus integrate this over $\mathbb{R}$ as the product of the integrals of the terms on each side of the convolution, that is: $\frac{c-\frac{1}{8}}{\sqrt{c\pi}} \cdot \frac{\sqrt{8\pi c}}{\sqrt{c-1/8}} \text{erfi}\left(r\sqrt{\frac{c}{8(c-\frac{1}{8})}}\right) \cdot \frac{\sqrt{\pi}}{\sqrt{c-1/8}} = \sqrt{8\pi}\text{erfi}\left(r\sqrt{\frac{c}{8(c-\frac{1}{8})}}\right)$, where erfi is the imaginary error function, defined as $\text{erfi}(x) := \frac{2}{\sqrt{pi}} \int_0^x e^{y^2} dy$. Noting the bound that $\text{erfi}(x) \leq \frac{3}{4}\frac{1}{x} e^{x^2}$ (which can be derived by differentiating), we have

$$|\alpha_{i,r,c}(j)| \leq \frac{2^j j!}{\sqrt{(2j)!2\pi}} \frac{\sqrt{(2i)!}}{2^i i!\sqrt{2}} \sqrt{8\pi} \frac{3}{4} \frac{\sqrt{8}}{r} \sqrt{\frac{c-\frac{1}{8}}{c}} e^{\frac{r^2}{8}\frac{c}{c-1/8}} = \frac{2^j j!}{\sqrt{(2j)!}} \frac{\sqrt{(2i)!}}{2^i i!} \frac{3}{r} \sqrt{\frac{c-\frac{1}{8}}{c}} e^{\frac{r^2}{8}\frac{c}{c-1/8}} \tag{5.3}$$

To bound $\alpha_{i,r,c}(j)$ a second way, we first note that a second application of "completing the square" allows us to reexpress part of Equation 5.1 as

$$T_{r,c}(w) \cdot e^{w^2/2} = \frac{c-\frac{1}{2}}{\sqrt{c\pi}} \left[I_{[-r\frac{c}{c-\frac{1}{2}}, r\frac{c}{c-\frac{1}{2}}]}(w) \cdot e^{\frac{c-1/2}{2c} \cdot w^2} \right] * e^{-(c-\frac{1}{2})w^2}.$$

Let $f_{r,c}(x)$ be the inverse Fourier transform of $I_{[-r\frac{c}{c-\frac{1}{2}}, r\frac{c}{c-\frac{1}{2}}]}(w) \cdot e^{\frac{c-1/2}{2c} \cdot w^2}$. We thus evaluate Equation 5.1 by noting that inner products are preserved under Fourier transform, that the (inverse) Fourier transform of $e^{-(c-\frac{1}{2})w^2}$ equals $\frac{1}{\sqrt{2(c-\frac{1}{2})}} e^{-\frac{1}{4(c-1/2)}x^2}$, and that multiplication and convolution swap roles under the Fourier transform, we have that

$$\alpha_{i,r,c}(j) = \frac{2^{2j}(j!)^2}{(2j)!\sqrt{2\pi}} \frac{\sqrt{c-\frac{1}{2}}}{\sqrt{2c\pi}} \int_{-\infty}^{\infty} \left[\left(f_{r,c}(x) \cdot e^{-\frac{1}{4(c-1/2)}x^2} \right) * h_i(x) \right] \cdot g_j(x) dx \tag{5.4}$$

By definition of the Fourier transform, for any function f, we have $||f||_\infty \leq \frac{1}{\sqrt{2\pi}} ||\hat{f}||_1$. Thus we may bound the maximum value of $|f_{r,c}|$ by $\frac{1}{\sqrt{2\pi}}$ times the L_1 norm of its Fourier transform, that is,

$$|f_{r,c}(x)| \leq \frac{1}{\sqrt{2\pi}} \int_{-r\frac{c}{c-\frac{1}{2}}}^{r\frac{c}{c-\frac{1}{2}}} e^{\frac{c-1/2}{2c}w^2} dw = \sqrt{\frac{c}{c-\frac{1}{2}}} \text{erfi}\left(\frac{r}{\sqrt{2}} \sqrt{\frac{c}{c-\frac{1}{2}}} \right)$$

We now bound $g_j(x) = \frac{e^{-x^2/2}x^{2j}}{2^j j!}$, the final term of Equation 5.4, by noting that, since $x \le e^{x-1}$ always, and replacing x by x/y yields $x \le e^{x/y-1}y$, we set $y = \sqrt{2j}$ and raise both sides to the power $2j$ to yield that, for positive x,

$$g_j(x) = \frac{e^{-x^2/2}x^{2j}}{2^j j!} \le \frac{e^{-x^2/2+x\sqrt{2j}-2j}j^j}{j!} = e^{-\frac{1}{2}(x-\sqrt{2j})^2}\frac{e^{-j}j^j}{j!}$$

Thus by definition of $h_i(x) = g_i(x\sqrt{2})$ we have $h_i(x) \le e^{-(x-\sqrt{i})^2}\frac{e^{-i}i^i}{i!}$ for positive x. Generally, we may see that $h_i(x) \le \left(e^{-(x-\sqrt{i})^2} + e^{-(x+\sqrt{i})^2}\right)\frac{e^{-i}i^i}{i!}$ for all x. We may thus bound Equation 5.4 as

$$|\alpha_{i,r,c}(j)| \le \frac{2^{2j}(j!)^2}{(2j)!2\pi}\text{erfi}\left(\frac{r}{\sqrt{2}}\sqrt{\frac{c}{c-\frac{1}{2}}}\right)\frac{e^{-i}i^i}{i!}\frac{e^{-j}j^j}{j!}\sum_{\pm,\pm}\int_{-\infty}^{\infty}\left[e^{-\frac{1}{4(c-1/2)}x^2} * e^{-(x\pm\sqrt{i})^2}\right]\cdot e^{-\frac{1}{2}(x\pm\sqrt{2j})^2}dx,$$

where the summation is over the four possible combinations of the two choices of "$\pm$". We note that the integral is equal to the convolution of the three terms inside of it, evaluated at $x = 0$, namely $\sqrt{\frac{8(c-\frac{1}{2})}{4c+1}}e^{-\frac{1}{4c+1}(x\pm\sqrt{i}\pm\sqrt{2j})^2}\Big|_{x=0}$, since the denominators in the exponents of Gaussians add under convolution. Thus we bound

$$|\alpha_{i,r,c}(j)| \le \frac{2^{2j}(j!)^2}{(2j)!2\pi}\text{erfi}\left(\frac{r}{\sqrt{2}}\sqrt{\frac{c}{c-\frac{1}{2}}}\right)\frac{e^{-i}i^i}{i!}\frac{e^{-j}j^j}{j!}\sqrt{\frac{8(c-\frac{1}{2})}{4c+1}}\cdot 4\cdot e^{-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2}$$

Since, as noted above, $\text{erfi}(x) \le \frac{3}{4}\frac{1}{x}e^{x^2}$, we have

$$|\alpha_{i,r,c}(j)| \le \frac{2^{2j}e^{-j}j^j j!}{(2j)!2\pi}\frac{e^{-i}i^i}{i!}\frac{4(c-\frac{1}{2})}{\sqrt{c(4c+1)}}\cdot\frac{3}{r}\cdot e^{\frac{r^2}{2}\frac{c}{c-1/2}-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2}$$

We bound $\frac{2^{2j}e^{-j}j^j j!}{(2j)!} \le 1$ as a combination of Stirling's formula, $e^{-j}j^j \le \frac{j!}{\sqrt{2\pi j}}$, and the bound on the middle binomial coefficient $\binom{2j}{j} \ge \frac{2^{2j}}{\sqrt{2\pi j}}$. A second application of Stirling's formula yields that $\frac{e^{-i}i^i}{i!} \le \frac{1}{\sqrt{2\pi i}}$, and we trivially bound $\frac{4(c-\frac{1}{2})}{\sqrt{c(4c+1)}} \le 2$ to yield

$$|\alpha_{i,r,c}(j)| \le \frac{3}{\pi r\sqrt{2\pi i}}\cdot e^{\frac{r^2}{2}\frac{c}{c-1/2}-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2} \tag{5.5}$$

Having thus derived two bounds on $|\alpha_{i,r,c}(j)|$, that of Equation 5.3 and that of Equation 5.5, we now aim to bound $\sum_{j\ge 0}|\alpha_{i,r,c}(j)|$ via a combination of these bounds: using Equation 5.3 when $2j$ is near i, and using Equation 5.5 otherwise.

Let $c = r^2$, and consider two cases.

Case 1: $i \le 2c^2$**.**

We first bound $\sum_{j\geq 4c^2}|\alpha_{i,r,c}(j)|$ from Equation 5.5. Specifically, consider $\sum_{j\geq 4c^2} e^{-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2}$. We note that the first term of the sum is at most $e^{-\frac{2c^2}{4c+1}} \leq e^{-\frac{c}{2}}e^{\frac{1}{8}}$. To bound the ratio between successive terms, we note that $\frac{d}{dj}(\sqrt{i}-\sqrt{2j})^2 = 2(1-\frac{\sqrt{i}}{\sqrt{2j}}) \geq 1$, which implies $\sum_{j\geq 4c^2} e^{-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2} \leq e^{-\frac{c}{2}}e^{\frac{1}{8}}\sum_{\ell\geq 0} e^{-\frac{1}{4c+1}\ell} = e^{-\frac{c}{2}}e^{\frac{1}{8}}\frac{1}{1-e^{-1/(4c+1)}}$. We note the general inequality $e^a \geq 1+a$, or equivalently, $e^{1/a} \geq 1+\frac{1}{a}$, which may be rearranged to $\frac{1}{1-e^{-1/a}} \leq a+1$, yielding a bound of $(4c+2)e^{-\frac{c}{2}}e^{\frac{1}{8}}$ on the sum. To bound the sum of Equation 5.5, we note that for $c \geq 1$, we have $\frac{r^2}{2}\frac{c}{c-1/2} \leq \frac{c}{2}+\frac{1}{2}$, leading to a bound of $\sum_{j\geq 4c^2}|\alpha_{i,r,c}(j)| \leq \frac{3}{\pi\sqrt{2\pi ic}}(4c+2)e^{5/8} < 5\sqrt{\frac{c}{i}}$

To bound $|\alpha_{i,r,c}(j)|$ for small j we instead use Equation 5.3. We note for $\ell \geq 1$ the bounds on the middle binomial coefficient of $\frac{1}{\sqrt{2\pi\ell}} \leq 2^{-2\ell}\binom{2\ell}{\ell} \leq 1$. Further, for $c \geq 1$ we have $\frac{r^2}{8}\frac{c}{c-1/8} \leq \frac{c}{8}+\frac{1}{56}$, yielding that $\sum_{j<4c^2}|\alpha_{i,r,c}(j)| \leq 4c^2\sqrt[4]{2\pi\cdot 4c^2}\frac{3}{r}e^{1/56}e^{c/8} < 28c^2e^{c/8}$. Combining this with the result of the previous paragraph yields $\sum_{j=0}^{\infty}|\alpha_{i,r,c}(j)| \leq 32c^2e^{c/8}$.

Case 2: $i > 2c^2$**.**

We use the bound of Equation 5.3 when $j \in (\frac{i}{2}-2c\sqrt{i}, \frac{i}{2}+3c\sqrt{i})$, and Equation 5.5 otherwise.

Consider $\sum_{j\geq\frac{i}{2}+3c\sqrt{i}}|\alpha_{i,r,c}(j)|$. Invoking Equation 5.5, we analyze $\sum_{j\geq\frac{i}{2}+3c\sqrt{i}} e^{-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2}$. We aim for $|\sqrt{i}-\sqrt{2j}| \geq \sqrt{2}c$, and show this by considering $(\sqrt{i}+\sqrt{2}c)^2 = i+2\sqrt{2}\sqrt{i}c+2c^2 < i+3\sqrt{2}\sqrt{i}c < 2j$, as desired. Thus the first term of this sum is at most $e^{-\frac{2c^2}{4c+1}} \leq e^{-\frac{c}{2}}e^{\frac{1}{8}}$. As above, we bound the ratio of successive terms by noting that $\frac{d}{dj}(\sqrt{i}-\sqrt{2j})^2 = 2(1-\frac{\sqrt{i}}{\sqrt{2j}}) \geq \frac{c\sqrt{2}}{\sqrt{i}}$, which implies that $\sum_{j\geq\frac{i}{2}+3c\sqrt{i}} e^{-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2} \leq e^{-\frac{c}{2}}e^{\frac{1}{8}}\sum_{\ell\geq 0} e^{-\frac{c\sqrt{2}}{(4c+1)\sqrt{i}}} = e^{-\frac{c}{2}}e^{\frac{1}{8}}\frac{1}{1-e^{-c\sqrt{2}/((4c+1)\sqrt{i})}}$, which, as analyzed in the previous case, yields a bound of $e^{-\frac{c}{2}}e^{\frac{1}{8}}(\frac{(4c+1)\sqrt{i}}{c\sqrt{2}}+1) \leq 4\sqrt{i}e^{-\frac{c}{2}}$ on $\sum_{j\geq\frac{i}{2}+3c\sqrt{i}} e^{-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2}$.

We now bound the small terms of the sum, $\sum_{j\leq\frac{i}{2}-2c\sqrt{i}} e^{-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2}$. As above, we show that $\sqrt{i}-\sqrt{2j} \geq \sqrt{2}c$ for such j by noting that $(\sqrt{i}-\sqrt{2}c)^2 = i-2\sqrt{2}\sqrt{i}+2c^2 > 2j$. Thus the last term in the sum is at most $e^{-\frac{2c^2}{4c+1}} \leq e^{-\frac{c}{2}}e^{\frac{1}{8}}$. As above, we bound the ratio of successive terms, this time as j decreases, by noting $\frac{d}{dj}(\sqrt{i}-\sqrt{2j})^2 = \frac{2}{\sqrt{2j}}(\sqrt{2j}-\sqrt{i})$, which since $2j < i$, has magnitude at least $\frac{2\sqrt{2}c}{\sqrt{i}}$. Thus the bound of the previous paragraph holds, yielding $\sum_{j\leq\frac{i}{2}-2c\sqrt{i}} e^{-\frac{1}{4c+1}|\sqrt{i}-\sqrt{2j}|^2} \leq 4\sqrt{i}e^{-\frac{c}{2}}$. As shown in Case 1, the remaining part of Equation 5.5 is bounded as $\frac{3}{\pi r\sqrt{2\pi i}}\cdot e^{\frac{r^2}{2}\frac{c}{c-1/2}} \leq \frac{3}{\pi r\sqrt{2\pi i}}e^{c/2}e^{1/2}$, yielding $\sum_{j\notin(\frac{i}{2}-2c\sqrt{i},\frac{i}{2}+3c\sqrt{i})}|\alpha_{i,r,c}(j)| \leq 8\sqrt{i}\frac{3}{\pi r\sqrt{2\pi i}}e^{1/2} < 6$.

For intermediate $j \in (\frac{i}{2}-2c\sqrt{i}, \frac{i}{2}+3c\sqrt{i})$ we bound $|\alpha_{i,r,c}(j)|$ from Equation 5.3. From the fact that $i!$ lies between its Stirling estimate and 1.1 times its Stirling estimate, we have that $\frac{2^j j!}{\sqrt{(2j)!}} \in (\sqrt[4]{\pi j}, 1.1\sqrt[4]{\pi j})$. Thus, since $j < 6i$, we have $\frac{2^j j!}{\sqrt{(2j)!}}\frac{\sqrt{(2i)!}}{2^i i!} \leq 1.1\sqrt[4]{6} < 2$, and we thus bound Equation 5.3 as $|\alpha_{i,r,c}(j)| \leq 2\frac{3}{r}e^{1/56}e^{c/8}$, and the sum of the $5c\sqrt{i}$ of these

terms as at most $31\sqrt{ci}e^{c/8}$. Combining this result with that of the previous paragraph yields $\sum_{j=0}^{\infty}|\alpha_{i,r,c}(j)| \leq 32\sqrt{ci}e^{c/8}$.

Having bounded $\sum_{j=0}^{\infty}|\alpha_{i,r,c}(j)|$, namely the second claim of the lemma, we now turn to bounding the first claim of the lemma—showing that the error of our approximation is small. As above, our expressions will involve the parameter c; as the final step of the proof, we choose c appropriately to obtain the claimed bounds.

Taking the inverse Fourier transform of both sides of Equation 5.2 yields that the difference between $h_i(w)$ and $\sum_{j=0}^{\infty}\alpha_{i,r,c}(j)\cdot g_j(w)$ equals the inverse Fourier transform of $(1-T_{r,c}(w))\hat{h}_i(w)$; we thus aim to bound the absolute value of this, pointwise. We note that from the definition of the Fourier transform, for a function f, $||f||_\infty \leq \frac{1}{\sqrt{2\pi}}||\hat{f}||_1$, so thus the maximum error of our approximation is bounded by $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}(1-T_{r,c}(w))|\hat{h}_i(w)|dw \leq \frac{\sqrt{(2i)!}}{2^i i!2\sqrt{\pi}}\int_{-\infty}^{\infty}(1-T_{r,c}(w))e^{-w^2/8}dw$. Again using the "completing the square" trick yields that this integral equals $\sqrt{8\pi}\text{erfc}\left(r\sqrt{\frac{c}{8(c+\frac{1}{8})}}\right) \leq \sqrt{8\pi}\text{erfc}(\frac{\sqrt{c}}{\sqrt{8}})$, where $\text{erfc} = 1-\text{erf}$ is the complementary error function. Noting the general bound that $\text{erfc}(x) \leq \frac{e^{-x^2}}{x\sqrt{\pi}}$, and from the above bound that $\frac{2^j j!}{\sqrt{(2j)!}} \geq \sqrt[4]{\pi j}$, the maximum error of our approximation is seen to be at most $\frac{8}{\sqrt[4]{\pi i}\sqrt{c}}e^{-c/8}$.

We have thus shown that $\sum_{j=0}^{\infty}\alpha_{i,r,c}(j)poi(x,j)$ approximates $poi(2x,i)$ to within $\frac{8}{\sqrt[4]{\pi i}\sqrt{c}}e^{-c/8}$, pointwise, while $\sum_{j=0}^{\infty}|\alpha_{i,r,c}(j)|$ is at most $32e^{c/8}\max\{c^2,\sqrt{ci}\}$, where c is arbitrary. Thus for desired error ϵ, we may choose $c \leq 8|\log\epsilon|$ so as to make $\frac{8}{\sqrt[4]{\pi i}\sqrt{c}}e^{-c/8} = \epsilon$, yielding that

$$\sum_{j=0}^{\infty}|\alpha_{i,r,c}(j)| \leq 32e^{c/8}\max\{c^2,\sqrt{ci}\} = \frac{1}{\epsilon}\cdot 200\max\{\sqrt[4]{i},\frac{c\sqrt{c}}{\sqrt[4]{i}}\} \leq \frac{1}{\epsilon}\cdot 200\max\{\sqrt[4]{i},24\log^{3/2}\frac{1}{\epsilon}\},$$

as desired. □

5.3 The Laguerre Lower Bound Construction

We will construct the p^+, p^- of Theorem 5.1 explicitly, via Laguerre polynomials. We now state the properties of these polynomials that we will use.

Let $L_j(x)$ denote the jth Laguerre polynomial, defined as $L_j(x) = \frac{e^x}{j!}\frac{d^j}{dx^j}(e^{-x}x^j)$.

Fact 5.4. *For each integer $j \geq 0$,*

1. *For $x \in [0,\frac{1}{j}]$, $L_j(x) \in [1-jx, 1]$;*
2. *L_j has j real roots, all lying in $[\frac{1}{j}, 4j]$;*
3. *Letting x_i denote the ith root of L_j, for $i \in \{1,\ldots,j\}$, we have $x_i \geq \frac{i^2}{3j}$;*

4. For $i < j/2$, $|\frac{dL_j(x)}{dx}(x_i)| \geq \frac{e^{x_i/2}j^{1/4}}{2x_i^{3/4}}$ and for any i, $|\frac{dL_j(x)}{dx}(x_i)| \geq \frac{e^{x_i/2}}{\sqrt{\pi}x_i^{3/4}}$.

Proof. Since L_j is a polynomial of degree j with j positive real roots, none of the inflection points lie below the smallest root. Since $L_j(0) = 1$, $L_j'(0) = -j$, and $L_j''(0) > 0$, we have that $L_j(x) \geq 1 - jx$ for x less than or equal to the smallest root of L_j. Thus the smallest root of L_j must be at most $\frac{1}{j}$, and $L_j(x) \geq 1 - jx$ for $x \leq \frac{1}{j}$. The fact that the largest root is at most $4j$ follows from [119], Theorem 6.32. The third fact appears in [119], p. 129, and the fourth fact follows from [119] p. 100. □

Definition 5.5. *Given real number $\phi \in (0, \frac{1}{4})$ and letting $j = \log k$, consider the degree $j+2$ polynomial $M_{j,\phi}(x) := -(x - \phi\frac{1}{j})(x - 2\phi\frac{1}{j})L_j(x)$. Let $v(x)$ be the function that takes value $1/M_{j,\phi}'(x)$ for every x where $M_{j,\phi}(x) = 0$, and is 0 otherwise, where M' is the derivative of M. Define the distributions $p^+_{j,\phi}, p^-_{j,\phi}$ such that for each x where $v(x) > 0$, the corresponding histogram $p^+_{j,\phi}$ contains $v(x)e^{x/32}$ probability mass at probability $\frac{1}{32k}x$, and for each x where $v(x) < 0$ the histogram $p^-_{j,\phi}$ contains $|v(x)|e^{x/32}$ probability mass at probability $\frac{1}{32k}x$, where each distribution is then normalized to have total probability mass 1.*

We note that since each element in the support of either $p^+_{\log k,\phi}$ or $p^-_{\log k,\phi}$ is defined to have probability at least $\frac{\phi}{32k\log k}$, both distributions have support at most $\frac{32}{\phi}k\log k$, which we take as n, in the context of both the entropy and the support size problems.

Lemma 5.6. *Distributions $p^+_{\log k,\phi}$ and $p^-_{\log k,\phi}$ are $O(\phi|\log\phi|)$-close, respectively, in the relative earthmover distance to the uniform distributions on $\frac{32}{\phi}k\log k$ and $\frac{16}{\phi}k\log k$ elements.*

Proof. Letting $j = \log k$, consider the values of $\frac{d}{dx}M_{j,\phi}(x)$ at its zeros. We first consider the two zeros at $\frac{\phi}{j}$ and $2\frac{\phi}{j}$. Note that $-\frac{d}{dx}(x - \phi\frac{1}{j})(x - 2\phi\frac{1}{j}) = -2x + 3\phi\frac{1}{j}$, having values $\pm\phi\frac{1}{j}$ respectively at these two points. By the product rule for differentiation and the first part of Fact 5.4, $|\frac{d}{dx}M_{j,\phi}(x)| \leq \phi\frac{1}{j}$ at these points.

Let x_i denote the ith zero of L_j. We note that since by definition, $\phi < \frac{1}{4}$, and from Fact 5.4, each $x_i \geq \frac{1}{j}$, we have $(x_i - \phi\frac{1}{j})(x_i - 2\phi\frac{1}{j}) \geq \frac{3}{8}x_i^2$. At each x_i, we may thus bound $|\frac{d}{dx}M_{j,\phi}(x)| = |(x - \phi\frac{1}{j})(x - 2\phi\frac{1}{j})\frac{d}{dx}L_j(x)| \geq \frac{3}{8}x^2\frac{e^{x/2}j^{1/4}}{2x^{3/4}}$ for $i < j/2$ and by $\frac{3}{8}x^2\frac{e^{x/2}}{\sqrt{\pi}x^{3/4}}$ otherwise, which we will denote as $\frac{3}{8}e^{x/2}x^{5/4}\left(\frac{j^{1/4}}{2}[i < j/2] + \frac{1}{\sqrt{\pi}}[i \geq j/2]\right)$.

Consider the *unnormalized* versions of $p^+_{j,\phi}, p^-_{j,\phi}$, that is, containing probability mass $|1/\frac{d}{dx}M_{j,\phi}(x)|e^{x/32}$ at each probability $\frac{1}{32k}x$ where $\frac{d}{dx}M_{j,\phi}(x)$ is positive or negative respectively (without scaling so as to make total probability mass be 1). Let c_1, c_2 respectively be the constants that $p^+_{j,\phi}, p^-_{j,\phi}$ respectively must be multiplied by to normalize them. Recall from above that $|\frac{d}{dx}M_{j,\phi}(x)| \leq \phi\frac{1}{j}$ for the point $x = \phi\frac{1}{j}$ in the support of $p^+_{j,\phi}$ and the point $x = 2\phi\frac{1}{j}$ in the support of $p^-_{j,\phi}$, which implies that the probability mass at each of these points is at least $e^{\phi\frac{1}{j}/32}\frac{j}{\phi} \geq \frac{j}{\phi}$. From these point masses alone we conclude $c_1, c_2 \leq \frac{\phi}{j}$.

We now consider the earthmover cost of moving all the weight of the unnormalized version of $p^+_{j,\phi}$ to $x = \phi\frac{1}{j}$ or all the weight of the unnormalized version of $p^-_{j,\phi}$ to $x = 2\phi\frac{1}{j}$, which

we will then multiply by c_1, c_2 respectively to yield upper bounds on the relative earthmover distances for the normalized distributions. The per-unit-weight relative earthmover cost of moving weight from an x_i to either $x = \phi\frac{1}{j}$ or $x = 2\phi\frac{1}{j}$ is at most $\log|\phi| + \log(jx_i)$. As we have bounded the weight at x_i (for either $p^+_{j,\phi}$ or $p^-_{j,\phi}$) as $\frac{8}{3}e^{-15x_i/32}x_i^{-5/4}\left(\frac{2}{j^{1/4}}[i < \frac{j}{2}] + \sqrt{\pi}[i \geq \frac{j}{2}]\right)$, and since, from Fact 5.4, $x_i \geq \frac{i^2}{3j}$, we may thus bound the relative earthmover distance by substituting this into the preceding expression, multiplying by the cost $|\log\phi| + \log(jx_i)$ and our bound $c_1, c_2 \leq \frac{\phi}{j}$, and summing over i:

$$\sum_{i=1}^{j} \frac{\phi}{j}\left(|\log\phi| + 2\log i\right)\frac{8}{3}e^{-\frac{5i^2}{32j}}\left(\frac{i^2}{3j}\right)^{-5/4}\left(\frac{2}{j^{1/4}}[i < j/2] + \sqrt{\pi}[i \geq j/2]\right) = O(\phi|\log\phi|)$$

as desired. □

We note the following general fact that we will use to bound the discrepancy in the fingerprint expectations of $p^+_{j,\phi}$ and $p^-_{j,\phi}$.

Fact 5.7. *Given a polynomial P of degree j whose roots $\{x_i\}$ are real and distinct, letting P' be the derivative of P, then for any $\ell \leq j-2$ we have $\sum_{i=1}^{j}\frac{x_i^\ell}{P'(x_i)} = 0$.*

Proof. We assume, without loss of generality, that P is monic.

To prove this, consider the general prescription for constructing a degree $j-1$ polynomial through j given points (x_i, y_i): $f(x) = \sum_{i=1}^{j} y_i\left(\prod_{m\neq i}(x - x_m)\right)\Big/\left(\prod_{m\neq i}(x_i - x_m)\right)$. We note that the coefficient of x^{j-1} in this polynomial is $f(x) = \sum_{i=1}^{j} y_i\left(\prod_{m\neq i}(x_i - x_m)\right)^{-1}$, where for each i, the expression $\left(\prod_{m\neq i}(x_i - x_m)\right)^{-1}$ is exactly $1/P'(x_i)$. Thus since polynomial interpolation is unique, $\sum_{i=1}^{j}\frac{x_i^\ell}{P'(x_i)}$ computes the x^{j-1} coefficient in the polynomial x^ℓ, which, for $\ell \leq j-2$ equals 0, as desired. □

The following lemma is the cornerstone of the proof of correctness of our lower bound construction, and guarantees that the expected fingerprints of the lower bound pair closely match.

Lemma 5.8. *For any i, the ith fingerprint expectations for distributions $p^+_{j,\phi}, p^-_{j,\phi}$ are equal to within $o(1)$.*

Proof. Consider, as in the proof of Lemma 5.6, the unnormalized versions of $p^+_{j,\phi}, p^-_{j,\phi}$, that is, containing weight $|1/\frac{d}{dx}M_{j,\phi}(x)|e^{x/32}$ at each probability $\frac{1}{32k}x$ where $\frac{d}{dx}M_{j,\phi}(x)$ is positive or negative respectively (without scaling so as to make total probability mass be 1), and let c_1, c_2 respectively be the constants that $p^+_{j,\phi}, p^-_{j,\phi}$ respectively must be multiplied by to normalize them.

Fact 5.7 directly implies that for any $i \leq j$, the ith fingerprint expectations for (unnormalized) $p^+_{j,\phi}$ and $p^-_{j,\phi}$ are identical. The proof of the lemma will follow from the following

two steps: first, we will show that the normalizing factors c_1, c_2 are nearly identical, and thus after normalizing, the ith fingerprint expectations for $i \leq j$ continue to be very close. In the second step, we show that by construction, and the second part of Fact 5.4, $p^+_{j,\phi}$ and $p^-_{j,\phi}$ consist of elements with probability at most $4j\frac{1}{32k} = \frac{\log k}{8k}$, and thus the discrepancy in the expected ith fingerprint entries for $i > j = \log k$ will be tiny, as the absolute values of these expectations are tiny. We begin by making the first point rigorous.

We note that

$$\sum_{i=1}^{\infty} i \cdot poi(xk, i) = \sum_{i=1}^{\infty} xk \frac{(xk)^{i-1}e^{-xk}}{(i-1)!} = xk \sum_{i=0}^{\infty} poi(xk, i) = xk,$$

and thus for any generalized histogram h,

$$\sum_{x:h(x)\neq 0} \sum_{i=1}^{\infty} h(x) i \cdot poi(kx, i) = k \sum_{x:h(x)\neq 0} h(x)x.$$

In particular, the sum over all i of i times the ith fingerprint expectations is exactly k times the probability mass of the unnormalized distribution we started with. We thus compare these sums for histograms $p^+_{j,\phi}, p^-_{j,\phi}$ to show that they have similar probability masses.

As noted above, by construction, $p^+_{j,\phi}$ and $p^-_{j,\phi}$ consist of elements with probability at most $\frac{\log k}{8k}$. Thus, for $x \leq \frac{\log k}{8k}$, we bound $\sum_{i=1+\log k}^{\infty} i \cdot poi(xk, i)$. We note that $i \cdot poi(xk, i) = xk \cdot poi(xk, i-1)$, yielding, from Fact A.19 a bound $xk \sum_{i=\log k}^{\infty} poi(\frac{1}{8}\log k, i) \leq \frac{7}{8} xk \cdot poi(\frac{1}{8}\log k, \log k)$. We compare this to $poi(\log k, \log k) \leq 1$ by noting that, in general, $\frac{poi(y/8,y)}{poi(y,y)} = \frac{e^{7y/8}}{8^y} \leq 3.3^{-y}$, yielding a bound of $\frac{7}{8} xk \cdot 3.3^{-\log k} < \frac{x}{k^{1/6}}$. That is, for an element of the distribution with probability x, its total contribution to the expected fingerprint entries with index greater than $\log k$ is at most $\frac{x}{k^{1/6}}$; summing over all x yields $\frac{1}{k^{1/6}}$ for the sum of these fingerprint expectations.

As noted above, the sum over *all* fingerprint entries equals k multiplied by the total mass of the corresponding generalized histogram. As noted in the proof of Lemma 5.6, $c_1, c_2 \leq \frac{\phi}{j} < 1$, and thus

$$\sum_{x:p^+_{j,\phi}(x)\neq 0} \sum_{i=1}^{\infty} p^+_{j,\phi}(x) i \cdot poi(kx, i) \geq k,$$

and similarly for the corresponding expression with $p^-_{j,\phi}$. Yet, as we just showed, the contribution towards this sum from fingerprints $i > j$ is at most $\frac{1}{k^{1/6}}$. Thus the unnormalized mass of $p^-_{j,\phi}$ and $p^-_{j,\phi}$ are at least 1, and the discrepancy in mass is at most $\frac{1}{k} \cdot \frac{1}{k^{1/6}}$, since the first j expected fingerprints agree exactly, and thus the normalizing constants c_1, c_2 differ by a factor of at most $1 \pm \frac{1}{k^{7/6}}$.

To conclude, since the unnormalized distributions had identical expected fingerprints for $i \leq j = \log k$, for the normalized distributions these expected fingerprints differing by a

factor of $1 \pm 1/k^{7/6}$, and thus differ in magnitude by at most $1/k^{1/6} = o(1)$ as desired. Further, as shown above, the expected sum of expected fingerprint entries above $\log k$ is bounded by $c_i/k^{1/6} \leq 1/k^{1/6} = o(1)$, yielding that the corresponding expectations for $p^+_{j,\phi}$ and $p^-_{j,\phi}$ match to within this bound. □

Our goal now is to mould $p^+_{j,\phi}$ and $p^-_{j,\phi}$ into distributions with the property that the distributions of their respective fingerprints are "close", respectively, to two very similar multivariate Gaussian distributions. As fingerprints are integer-valued vectors, while Gaussian distributions are continuous, we instead consider Gaussian distributions *rounded to the nearest lattice point*. Discreteness is still an obstacle, however, and the central limit theorem we use (Theorem 4.2) yields better bounds as the variance of the distributions in each direction increases. With this motivation in mind, we introduce the next construction which will modify $p^+_{j,\phi}$ and $p^-_{j,\phi}$ very little in the relative earthmover metric, while making the distributions of their histograms suitably "fat" so as to be amenable to applying our central limit theorem. Additionally, for the sake of rigor, in this step we also round so as to ensure that the histograms are integer-valued, and thus correspond to actual distributions. This rounding moves very little mass, and changes the fingerprint expectations little, though we must still round rather delicately.

Definition 5.9. *Define the* fattening *operator F that, given a histogram p, constructs a new histogram p^F as follows:*

- *Set $q := \sum_{i=1}^{\log k} \frac{i}{k} \lceil \frac{k}{\log^3 k} \rceil$.*
- *Provisionally set $p^F(x) = \lfloor (1-q) \cdot p(x) \rfloor$ for each x, let $x^* = \max(x : p^F(x) \geq 1)$ and decrement $p^F(x^*)$ by 1.*
- *Set $m := \sum_{x:p^F(x)\neq 0} x p^F(x)$, and increment $p^F\left(\frac{1-q-m}{\lceil 2+\log k \rceil}\right)$ by $\lceil 2 + \log k \rceil$.*
- *For each integer $i \in \{1, \ldots, \log k\}$, increment $p^F(\frac{i}{k}) \leftarrow p^F(\frac{i}{k}) + \lceil \frac{k}{\log^3 k} \rceil$*

Lemma 5.10. *Let $p^{F+}_{j,\phi}$ and $p^{F-}_{j,\phi}$ denote the results of applying Definition 5.9 to histograms $p^+_{j,\phi}, p^-_{j,\phi}$, respectively, as constructed in Definition 5.5. Then $p^{F+}_{j,\phi}$ and $p^{F-}_{j,\phi}$ satisfy the following conditions:*

- *The histograms take integral values and have total probability mass 1 (and thus correspond to actual distributions).*
- *There is no probability mass on probabilities below $\frac{\phi}{32k \log k}$.*
- *The discrepancy in fingerprint expectations of $Poi(k)$-sample fingerprints from $p^{F+}_{j,\phi}$ and $p^{F-}_{j,\phi}$ is at most $3 \log k$.*

- *The relative earthmover distances between the fattened and original version of $p^+_{j,\phi}$ and $p^-_{j,\phi}$ respectively are both $O(\frac{|\log\phi|+\log\log k}{\log k})$.*

Proof. First, observe that given a generalized histogram with total probability mass 1, fattening returns a distribution, since all histogram entries of the fattened histogram take integer values, and the total probability mass is 1, since the total mass after the second step is m, the third step adds $1-q-m$ mass, and q mass is added in the final step.

By Fact 5.4, $x^* \in [\frac{logk}{3\cdot 32k}, \frac{\log k}{8k}]$, and since these histograms will be supported on $j+2 = \log k + 2$ values which each can get rounded, and thus we crudly bound the amount of mass that was lost in the rounding and decrementing of $p^F(x^*)$ by $1-q-m \in [\frac{\log k}{3\cdot 32k}, \frac{\log k(2+\log k)}{8k}]$, and thus the location at which the extra $1-q-m$ units of mass are added in the third step is in the range $[\frac{1}{6\cdot 32k}, \frac{\log k}{8k}]$, provided k is sufficiently large. In particular, for the sake of our bounds on support size, no elements are added below probability $O(\frac{1}{k})$, so that $p^{F+}_{j,\phi}$ and $p^{F-}_{j,\phi}$ retain the lower bound of $\frac{\phi}{32k\log k}$ on the probability of each domain element.

We now argue that the fingerprint expectations of the pair $p^{F+}_{j,\phi}$ or $p^{F-}_{j,\phi}$ will still be close. If we had simply set $p^F(x) = (1-q)\cdot p(x)$, and then performed the fourth step, the discrepancy in fingerprint expectations could only decrease, as the modification of the fourth step is performed identically in each histogram, and the scaling factor $1-q<1$. Now, note that the actual histograms output by the procedure differ from this in only the second and third steps, that decreases at most $2+\log k$ histogram values by at most 2, and then adds at most $3+\log k$ histogram entries. These modification can alter the expected fingerprints by at most their sum, and thus from Lemma 5.8, the total discrepancy in the fingerprint expectations of $p^{F+}_{j,\phi}$ and $p^{F-}_{j,\phi}$ is at most $2+\log k+3+\log k+o(1) < 3\log k$..

All the probabilities of $p^+_{j,\phi}$ and $p^-_{j,\phi}$ and their fattened versions are between $\frac{\phi}{32k\log k}$ and $\frac{\log k}{k}$, incurring a per-unit-mass relative earthmover cost of at most $O(|\log\phi| + \log\log k)$. Since $q+m = O(\frac{1}{\log k})$, the relative earthmover cost is thus trivially bounded by the product of these terms:

$$\frac{1}{\log k}\cdot O(|\log\phi| + \log\log k) = O(\frac{|\log\phi| + \log\log k}{\log k}),$$

as claimed. □

We next show that for any fattened distribution, the variance of the distribution of the fingerprint is large in any direction. Specifically, for any unit vector $v \in \mathbb{R}^{\log k}$, we find an integer i such that elements of probability $\frac{i}{k}$—such as those added in Definition 5.9—have high-variance fingerprints along the direction v. Instead of proving this result only for $p^{F+}_{j,\phi}$ and $p^{F-}_{j,\phi}$, we prove it more generally, so that we may more easily invoke our central limit theorem.

Lemma 5.11. *For any vector unit vector $v \in \mathbb{R}^m$ with $v(0) = 0$, there exists an integer $i \in \{1,\ldots,m\}$ such that, drawing $\ell \leftarrow Poi(i)$ conditioned on $\ell \le m$, the variance of $v(\ell)$ is at least $\frac{1}{12m^{7/2}}$.*

Proof. We note the crucial stipulation that $v(0) = 0$, for otherwise, a uniform vector would trivially have zero variance.

Given a unit vector v, there exists $i \in \{1, \ldots, m\}$ such that $|v(i) - v(i-1)| \geq \frac{1}{m^{3/2}}$, since otherwise (since $v(0) = 0$) we would have $|v(i)| \leq \frac{i}{m^{3/2}}$, implying $\sum_{i=1}^{m} v(i)^2 < 1$. Consider such an i.

Since in general, $i! \leq \frac{i^i}{e^i} 3\sqrt{i}$, we have that $poi(i, i-1) = poi(i, i) = \frac{i^i e^{-i}}{i!} \geq \frac{1}{3\sqrt{i}}$, which implies that, just the two possibilities $Poi(i) = i$ or $Poi(i) = i-1$ alone are enough to induce variance in $v(\ell)$ of the product of our bound on their total probability mass, $\frac{1}{3\sqrt{i}} \geq \frac{1}{3\sqrt{m}}$ and the square of half $|v(i) - v(i-1)| \geq \frac{1}{m^{3/2}}$, yielding $\frac{1}{12m^{7/2}}$. □

5.4 Proof of Theorem 5.1

We now assemble the pieces of the main result of this chapter.

Proposition 5.12. *For a positive constant $\phi < 1/4$, the total variational distance (ℓ_1 distance) between the distribution of $Poi(k)$-sample fingerprints from $p^{F+}_{\log k, \phi}$ and $p^{F-}_{\log k, \phi}$ goes to 0 as k goes to infinity.*

Proof. Since both $p^{+}_{\log k, \phi}$ and $p^{-}_{\log k, \phi}$ consist of elements with probabilities at most $\frac{1}{8} \log k$, tail bounds (see the proof of Lemma 5.8 for the calculations) show that the probability that *any* such element occurs more than $\log k$ times is $o(1)$. We thus assume for the rest of this proof that all such domain elements are seen at most $\log k$ times.

Consider, for either fattened distribution, $p^{F+}_{\log k, \phi}$ or $p^{F-}_{\log k, \phi}$, the portion of the fingerprint above $\log k$, which we denote $\mathcal{F}_{>\log k}$. Since by assumption, only the "fattened" portion of either distribution contributes to $\mathcal{F}_{>\log k}$, and since these portions are identical, we have that the probability of a given $\mathcal{F}_{>\log k}$ occurring from $p^{F+}_{\log k, \phi}$ equals its probability of occurring from $p^{F-}_{\log k, \phi}$. We complete the proof by comparing, for each $\mathcal{F}_{>\log k}$, the conditional distributions of the fingerprints at or below $\log k$ *conditioned on the value $\mathcal{F}_{>\log k}$ and which elements of the distribution contributed to $\mathcal{F}_{>\log k}$.*

The fattening process introduces $\frac{k}{\log^3 k}$ elements to the distribution at probability $\frac{i}{k}$ for each $i \in \{1, \ldots, \log k\}$. Since the number of occurrences of one of these elements is distributed as $Poi(i)$, for $i \leq \log k$, in expectation no more than half of these elements will be sampled more than $\log k$ times. Since the number of times each element is sampled is independent (as we are taking a Poisson-distributed sample size), Chernoff bounds imply that the number of elements sampled more than $\log k$ times will be at most $\frac{3}{4} \frac{k}{\log^3 k}$ with probability $1 - e^{k^{\Theta(1)}}$, for each i. By a union bound over $i \leq \log k$, with probability at least $1 - o(1)$, conditioning on which elements contribute to $\mathcal{F}_{>\log k}$ will leave at least $\frac{1}{4} \frac{k}{\log^3 k}$ elements at each probability $\frac{i}{k}$ that are not fixed by the conditioning.

By Lemma 5.11, for each unit vector $v \in \mathbb{R}^{\log k}$, there is an index i such that each element of probability $\frac{i}{k}$ contributes $\frac{1}{12 \log^{7/2} k}$ to the (conditional) fingerprint variance in the direction

of v. As the previous paragraph showed that there are at least $\frac{1}{4}\frac{k}{\log^3 k}$ elements with this property that are disjoint from the elements comprising $\mathcal{F}_{>\log k}$. Thus the fingerprint variance is at least $\sigma^2 := \frac{k}{48\log^{13/2} k}$, in any direction v.

We thus apply our central limit theorem, Theorem 4.2, to the distributions of the first $\log k$ fingerprint entries corresponding to drawing $Poi(k)$-sized samples from $p^{F+}_{\log k,\phi}$ and $p^{F-}_{\log k,\phi}$, conditioned on $\mathcal{F}_{>\log k}$. Each such distribution is a generalized multinomial distribution (see Definition 4.2) with $\log k$ columns and at most $n = \frac{32}{\phi} k \log k$ rows. We invoke the central limit theorem, to conclude that each such distribution may be approximated by the multivariate Gaussian distribution of the same mean and covariance, rounded to the nearest lattice points, to within total variational distance $\frac{\log^{4/3} k}{\sigma^{1/3}} \cdot 2.2 \cdot (3.1 + 0.83 \log n)^{2/3}$, which is $o(1)$ since the k in the numerator of $\sigma^2 = \frac{k}{48\log^{13/2} k}$ dominates the logarithmic terms.

For a given $\mathcal{F}_{>\log k}$, let μ^+, μ^- denote respectively the vectors of conditional fingerprint expectations, for $p^{F+}_{\log k,\phi}$ and $p^{F-}_{\log k,\phi}$ respectively; let Σ^+, Σ^- denote respectively the corresponding covariance matrices. As we have just shown that the conditional distributions of fingerprints are statistically close to the multivariate Gaussian distributions $\mathcal{N}(\mu^+, \Sigma^+)$, $\mathcal{N}(\mu^-, \Sigma^-)$, respectively, each rounded to the nearest lattice point, it remains to compare the total variational distance of these distributions. We note immediately that rounding to the nearest lattice point can only decrease the total variational distance. We thus must bound $D_{tv}(\mathcal{N}(\mu^+, \Sigma^+), \mathcal{N}(\mu^-, \Sigma^-))$, which we will do with Proposition A.13 once we have analyzed the disparities between the means and covariances.

Lemma 5.10 showed that the fingerprint expectations of $p^{F+}_{\log k,\phi}$ and $p^{F-}_{\log k,\phi}$ match to within $O(\log k)$. Since the conditioning applies only to the identical fattened region, it remains true that $|\mu^+(i) - \mu^-(i)| = o(1)$ for each i.

As we noted in the discussion preceding this result, approximating Poisson functions $poi(2xk, i)$ as linear combinations of Poisson functions $poi(xk, j)$ means that we can approximate each entry of the covariance matrix Σ by a linear combination of entries of the expectation vector μ. We thus invoke Lemma 5.3 for $\epsilon = \frac{1}{\sqrt{k}}$ to see that, indeed, there exist constants $\alpha_i(j)$ with $\sum_{j=0}^{\infty} |\alpha_i(j)| \leq \sqrt{k} \cdot 200 \max\{\sqrt[4]{i}, 24\log^{3/2}\sqrt{k}\} = O(\sqrt{k}\log^{3/2} k)$ such that we may approximate entries $\Sigma(\ell, m)$ via coefficients $\alpha_{\ell+m}(j)$, where the error contributed by each domain element is at most ϵ. As there are at most $n = \frac{32}{\phi} k \log k$ domain elements, this approximation error is at most $\frac{32}{\phi}\sqrt{k}\log k$. Thus by the triangle inequality, the discrepancy $|\Sigma^+(\ell, m) - \Sigma^-(\ell, m)|$ for each element of the covariance matrix is bounded by twice this, plus the discrepancy due to $|\alpha_i(j)|$ times the difference $|\mu^+(i) - \mu^-(i)|$. We combine the bounds we have just derived to yield

$$|\Sigma^+(\ell, m) - \Sigma^-(\ell, m)| = O(\frac{\sqrt{k}}{\phi}\log^{5/2} k).$$

The two Gaussians $\mathcal{N}(\mu^+, \Sigma^+)$ and $\mathcal{N}(\mu^-, \Sigma^-)$ thus have means that are component-wise within $O(\log k)$ and thus within Euclidean distance $O(\log^{3/2} k)$, covariance matrices within $O(\frac{\sqrt{k}}{\phi}\log^{5/2} k)$, and variances at least $\sigma^2 = \frac{k}{48\log^{13/2} k}$ in each direction—which thus lower-

bounds the magnitude of the smallest eigenvalues of Σ^+, Σ^- respectively. For any positive constant ϕ, as k gets large, Proposition A.13 implies that $D_{tv}(\mathcal{N}(\mu^+, \Sigma^+), \mathcal{N}(\mu^-, \Sigma^-)) = o(1)$, as the k terms dominate the logarithmic terms, as claimed. □

We now prove our main theorem:

Proof of Theorem 5.1. Let k be such that $n = \frac{32}{\phi} k \log k$. Construct $p^+ = p^{F+}_{\log k, \phi}$ and $p^- = p^{F-}_{\log k, \phi}$ according to Definition 5.5 followed by Definition 5.9. Lemma 5.6 and Lemma 5.10 imply that p^+, p^- that are $O(\phi |\log \phi|)$-close in the relative earthmover metric, respectively, to the uniform distributions on n and $\frac{n}{2}$ elements. Proposition 5.12 shows that the distribution of fingerprints derived from $Poi(k)$ sized samples have statistical distance $o(1)$. Thus, if one were presented with a fingerprint derived from a $Poi(k)$-sized sample that was either drawn from p^+ or p^- (with probability 1/2 each), then no algorithm can distinguish the case that the fingerprint was drawn from p^+ versus p^- with any constant probability greater than 1/2. The same claim holds for fingerprints derived from a sample of size *exactly* k, since $Pr[Poi(k) \leq k] > 1/2$, and thus if such an algorithm existed achieving probability of success $1/2 + \alpha$, then it could be used to give correct answers with probability of success at least $1/2 + \alpha/2$ in the case of $k' \leftarrow Poi(k)$ sized samples by simply guessing randomly if $k' < k$, and otherwise running the algorithm of k-sized samples on a random subsample of size $k \leq k'$. □

5.5 A Lower Bound for the Distinct Elements Problem

In this section, we explain how extend our lower bound of Theorem 5.1 to show a lower bound for the distinct elements problem. For clarity, we briefly restate the distinct elements problem:

Definition 5.13. *Given access to n buckets, each of which contains one object that is not necessarily distinct from those in the other buckets, how many buckets must one inspect in order to estimate the total number of distinct objects to $\pm \epsilon n$, with high probability?*

Theorem 5.3. *For constant $\delta < 1/2$, any algorithm for the distinct elements problem with n buckets that estimates the number of distinct elements to within additive error $\frac{n}{8}$ with failure probability at most δ, must look in at least $\Omega(\frac{n}{\log n})$ buckets.*

There are two differences between the distinct elements setting, and the more general problem of estimating the support size of a distribution which could, conceivably, make the former problem easier. The first difference is simply that in the distinct elements setting, one is essentially sampling elements from the distribution described by the contents of the buckets *without replacement*, whereas in the general setting, a sample is drawn from the distribution (with replacement). Note that nothing can be gained in the distinct elements

setting by specifying which bucket to examine next, as an adversary can always re-shuffle the buckets, and thus each bucket that is revealed is equivalent to a randomly selected unexamined bucket. (See Lemma 3.1 of [107], for example.)

The second, and more significant difference between the distinct elements setting and the general setting is that the histogram corresponding to the distinct elements setting is supported at probabilities that are integer multiples of $1/n$, since an element either occurs in $0, 1, 2, \ldots$ of the buckets. The lower bound instance constructed in our proof of Theorem 5.1 did not have this property—its histogram is supported at the roots of a Laguerre polynomial. This construction was rather delicate, and we relied fundamentally on the properties of the Laguerre polynomials.[1]

Fortunately, we eschew having to deal with either of these issues directly, by showing that any algorithm for the distinct elements problem can be adapted to yield an algorithm with comparable (though slightly worse) performance on the general problem of estimating support size. The following simple lemma is adapted from Lemma 3.3 of [107], and implies Theorem 5.3 by plugging in $\alpha = 3$.

Lemma 5.14. *[Based on Lemma 3.3 of [107]] For any constant $\alpha \geq 1$, given an algorithm A for the distinct elements problem that looks in $k(m)$ buckets when presented with a total of m buckets, and estimates the number of distinct elements to within additive error ϵm with probability of failure δ, we show how to use it to estimate the support size of a general distribution D (all of whose elements occur with probability $\geq 1/n$) with probability of failure $\delta + o(1)$, using a sample of size $k(\alpha n)$, and achieving error $(\epsilon + 2e^{-\alpha})n$.*

Proof. The simple procedure is described below:

- Draw $k' \leftarrow Poi(\alpha n)$, and set $m := k'$.
- Select a sample of size k_m from D, and give them as input to algorithm A, and return the output of algorithm A.

We now prove the correctness of the above procedure. Given a distribution D all of whose domain elements occur with probability at least $1/n$, let D_m denote the empirical distribution of a sample of size m drawn from D. While D and D_m might be quite different distributions, we note that with high probability, their support sizes will be similar. Trivially, the support of D_m is at most that of D. Furthermore, the probability that a given domain element does not occur in D_m is at most $poi(\alpha, 0) = e^{-\alpha}$, and by the independence of the number of occurrences of different domain elements, with probability $1 - o(1)$, the discrepancy between the support sizes of D and D_m is at most $2ne^{-\alpha}$.

To conclude the proof of correctness, simply note that a set of $k_m \leq m$ draws *without* replacement from the distribution D_m is distributed identically to a set of k draws taken *with*

[1]The lower bound construction of Definition 5.5 is extremely delicate: while the roots of the Laguerre polynomials are roughly quadratically spaced, constructing a lower bound instance by using the polynomial whose roots are exactly quadratically spaced does not seem to allow our analysis to go through.

replacement from distribution D, and thus the input to the algorithm A is indistinguishable from a sample of size k_m drawn from the distinct elements instance defined by the empirical distribution D_m. □

5.6 Lower Bounds for Total Variational Distance

In this section, we leverage Theorem 5.1 to prove a lower bound for the task of estimating the total variational distance (ℓ_1 distance) between a pair of distributions. For convenience, we restate the theorem:

Theorem 5.2. *For any constants $0 < a < b < \frac{1}{2}$, and probability of failure $\delta < 1/2$, for sufficiently large n, given samples from a pair of distributions of support at most n, distinguishing whether their total variational distance (ℓ_1 distance) is less than a or greater than b with probability of success at least $1 - \delta$, requires samples of size $\Omega(\frac{n}{\log n})$.*

Proof. Theorem 5.1 shows that we can construct, for any sufficiently small constant α, a pair of distributions, $p^n, p^{n/2}$ on support n such that they are α-close in the relative earthmover—and hence also in ℓ_1 distance—to uniform distributions, respectively, on n and $\frac{n}{2}$ elements, yet whose fingerprints are indistinguishable given samples of size less than $O(\frac{n}{\log n})$. For our ℓ_1 lower bound, construct such distributions for $\alpha < \frac{\min\{a, \frac{1}{2} - b\}}{2}$. Consider now the task of distinguishing, for random permutations σ_1, σ_2, the *pair* of distributions $(\sigma_1(p^n), \sigma_2(p^n))$ from the pair $(\sigma_1(p^{n/2}), \sigma_2(p^n))$, where we consider that the application of a permutation relabels its elements.

Assume for the sake of contradiction that these pairs are distinguishable with probability $\delta > 1/2$ given $k = o(\frac{n}{\log n})$ sized samples from each. We could thus construct an algorithm that distinguishes k-sample fingerprints from p^n from those from $p^{n/2}$ with probability δ by simulating the application of this hypothetical algorithm on the two samples consisting of the given k-sample, and a k-sample constructed ad hoc from a random permutation of p^n. If the hypothetical algorithm returned that the ℓ_1 distance is smaller than a we could return "p^n", and if the hypothetical algorithm returned that the ℓ_1 distance is at least b, we could return "$p^{n/2}$".

Thus we have the desired contradiction and no such algorithm can exist. □

Chapter 6

The Power of Linear Estimators

Our algorithmic toolbox is large. Given independent draws from a distribution, one might imagine a wide gamut of algorithmic strategies for recovering information about the underlying distribution. When limited by data instead of computational resources, a brute-force search through hypotheses might be the best option. More specifically, one might be guided by a Bayesian heuristic, or otherwise try to optimize "likelihood". More firmly in the realm of polynomial-time algorithms, convex programming is a powerful tool for rapidly traversing a sufficiently structured search space. At the far extreme of simplicity, are *linear estimators*. Given the fingerprint vectors of the sample, a linear estimator multiplies each entry by a fixed, position-dependent constant and returns the sum.

For the broad class of "symmetric" distribution properties, despite the plethora of algorithmic options and a rich history of study by both the statistics and computer science communities, nearly all the proposed estimators are these algorithmically-hollow linear estimators.

Because of, or perhaps despite, their rather pedestrian nature, linear estimators have many features to recommend: they are easy to use, easy to describe, and, because of the especially transparent fashion in which they use the data, generally easy to analyze. These niceties, though, make it even more urgent to resolve the question: "*How good are linear estimators?*"

Despite much effort constructing linear estimators during the past century, and perhaps even more effort analyzing these estimators, for many symmetric distribution properties the best known linear estimators require much larger samples than necessary to achieve a desired accuracy of estimation. Specifically, to achieve constant additive error (with high probability) for any of the following properties: entropy, distinct elements, ℓ_1 distance and KL-divergence, previously proposed linear estimators require $\Theta(n)$ sized samples, where, as throughout, n is a bound on the support size of the distributions being sampled, and is a natural parametrization of the sample complexities of these estimation problems. Corresponding statements hold for estimating support size and distance to uniformity, for which the sample complexities are parameterized slightly differently.[1]

[1] As in the previous chapter, the problem of estimating support size is parameterized in terms of a lower

In Chapter 3, we applied the algorithmic power of linear programming to these estimation tasks, yielding estimators for entropy and support size that require only $O(n/\log n)$ sized samples. This intriguing state of affairs provokes the question:

> *What richness of algorithmic machinery is needed to effectively estimate these properties?*

Answers to this question could serve to guide future endeavors to construct and analyze estimators. Additionally, questions of this nature lie at the philosophical core of the theoretical approach to computing.

The main result of this chapter is the near-optimality of linear estimators for additively estimating a subclass of symmetric distribution properties that includes entropy, variants of distance to uniformity, and support size (which may be viewed as a version of the distinct elements problem). Our proof is constructive, in that we give a polynomial-time algorithm that is practically viable which, on input n, k, and the property in question, outputs a linear estimator which, on input k independent draws from a distribution of support at most n, will with high probability return an ϵ-accurate approximation of the property value; this estimator is near-optimal in the sense that there exist $k' = k(1-o(1))$, and $\epsilon' = \epsilon(1-o(1))$ and two sets of distributions of support at most n where the property values of the distributions in one set differ from those in the other set by ϵ', yet no algorithm when given a k'-sized sample from one of the distributions can distinguish whether the sample was drawn from a distribution in the first set from being drawn from a distribution in the second set, with any fixed probability greater than $1/2$.

While our result shows that linear estimators are near-optimal, the proof does not yield any bounds on the sample complexity. The proof is constructive, and gives an algorithm for constructing these estimators, yet it does not reveal the sample size necessary to achieve a desired estimation accuracy. In this sense, the results of this chapter complement the results of Chapters 3 and 5. In Chapter 7, inspired by numerical solutions to the algorithm of this chapter, we are able to describe and analyze the performance of some explicit linear estimators for several properties. This analysis reveals the surprising inverse linear rate of convergence of optimal estimators for entropy (as opposed to error decreasing as the inverse of the square root of the sample size, as one might expect), and allows us to establish the sample complexity of estimating the ℓ_1 distance to the closest uniform distribution over a specified number of elements.

---

bound, $1/n$ on the probability of any domain element. The problem of estimating the distance to the uniform distribution on m elements is parameterized by m.

6.1 A Duality of Estimators and Lower Bound Instances

The main result of this chapter—the near-optimality of linear estimators—hinges on a new connection between constructing "good" lower bounds, and "good" linear estimators.

The canonical approach to creating lower bounds for estimating symmetric properties consists of finding a pair of distributions, A^+, A^- with rather different property values, such that given only the fingerprint of a sample, one cannot distinguish whether the sample was drawn from A^+ or A^-.[2] This condition of indistinguishability is very stringent, and requires showing that the distribution of fingerprints derived from a sample drawn from A^+ is close in total variation (ℓ_1) distance to the corresponding distribution for a sample drawn from A^-. The central limit theorems of Chapter 4 and lower bound tools of Chapter 5 suggest and enable a *principled* approach to constructing lower bounds for property estimation. Here, we show the perhaps surprising result that despite the effort required to assemble the required tools, the condition of indistinguishability in this framework can be roughly expressed via an intuitive set of *linear* constraints.

Turning, for a moment, to the side of constructing linear estimators, a natural and popular approach is to represent the "characteristic function" of the property in question as a linear combination of "Poisson functions" $poi(x, i) := \frac{e^{-x}x^i}{i!}$; see [34, 90, 102, 101, 123, 136]. Indeed, in [102, 101], Paninski showed the existence of a sublinear-sample linear estimator for entropy via a simple nonconstructive proof that applies the Stone-Weierstrass theorem to approximate the logarithm function (the characteristic function of entropy) via the set of Poisson functions. We show that the task of constructing such a representation of a given accuracy can also be framed as a set of linear constraints.

Thus general techniques for proving property estimation upper and lower bounds can both be *characterized* by linear constraints. One may then ask how the performance of the best such lower bound compares to the performance of the best such upper bound. Optimizing each notion of performance relative to the corresponding linear constraints can be expressed as a linear program. Amazingly (though in retrospect not unexpectedly) these two linear programs—one for constructing good lower bound example pairs, and one for constructing good linear estimators, are *dual* to each other.

The fundamental complication, however, is that the range of parameters for which the lower bound program will be pertinent, and those for which the estimator program will be pertinent, are non-intersecting. Intuitively, it is clear that these parameter ranges *must* be disjoint, as one would not expect the *exact* correspondence between optimal lower bounds of this form, and optimal linear estimators, as would be implied if these programs were

[2]Specifically, as described in Chapter 5, distributions A^+, A^- will not themselves be indistinguishable, but rather, the ensembles that arise from considering a random permutation of the domain elements of A^+, A^- respectively will be indistinguishable. Because we are considering symmetric properties of distributions, such permutations do not affect the property value and thus are benign. The purpose of these permutations is to remove any useful information from the sample except the fingerprint.

dual for pertinent parameters. Thus the main technical challenge is relating optimal values of the lower bound program to optimal values of the estimator program corresponding to slightly different parameters. Establishing this relation traverses some nice math involving the exponentials of infinite "Poisson-matrices".

Definitions

We begin with some definitions that will be used in this chapter. We refer the reader to Chapter 3 for the basic definitions related to property estimation, including the definitions of *fingerprints* and *histograms*.

Definition 6.1. *A k-sample* linear *estimator α is defined by a set of at least k coefficients, $\alpha = (\alpha_1, \ldots, \alpha_k)$. The estimator is defined as the dot product between the fingerprint vector $\mathcal{F}$ of a sample of size k, and the vector α, namely $S_k(\mathcal{F}) := \sum_{i=1}^{k} \alpha_i \mathcal{F}_i$.*

Recall that any symmetric property is a function of only the histogram of a distribution. Additionally, a symmetric property is *linear*, if the property value is a linear function of the histogram:

Definition 6.2. *A symmetric property π is* linear *if there exists some function $f_\pi : (0, 1] \to \mathbb{R}$ which we term the* characteristic function *of π, such that for any distribution A with histogram h,*

$$\pi(A) = \sum_{x:h(x)\neq 0} h(x) f_\pi(x).$$

As the following examples illustrate, most of the properties that we have been dealing with are linear. (In the following, $\mathcal{D}^n$ denotes the set of distributions over support $[n] = \{1, \ldots, n\}$.)

Example 6.3. *The entropy of a discrete distribution $p \in \mathcal{D}^n$ with histogram h is given by $H(h) := \sum_{i=1}^{n} p(i) |\log p(i)| = \sum_{x:h(x)\neq 0} h(x) f(x)$, for the function $f(x) := x |\log x|$.*

Example 6.4. *The support size of a discrete distribution $p \in \mathcal{D}^n$ with histogram h is given by $\sum_{x:h(x)\neq 0} h(x) f(x)$, for the function $f(x) := 1$.*

Example 6.5. *The total variational distance between a discrete distribution $p \in \mathcal{D}^n$ with histogram h and the closest uniform distribution on s elements can be approximated to within a factor of 2 as $\sum_{x:h(x)\neq 0} h(x) f(x)$, for the function*

$$f(x) := \begin{cases} x & \text{for } x \leq \frac{1}{2s} \\ |x - \frac{1}{s}| & \text{for } x > \frac{1}{2s}. \end{cases}$$

Summary of Results

The main theorem of this chapter shows that linear estimators are near-optimal for additively estimating the class of linear symmetric distribution properties, provided that they satisfy a mild continuity condition:

Theorem 6.1. *Let π be a symmetric linear property that is $\delta(k)$-relative earthmover continuous on distributions of support $n(k)$. If for some constant $c > 0$ and parameter $\epsilon(k) = \delta/k^{o(1)}$, any distributions of support n whose π values differ by at least ϵ are distinguishable with probability at least $\frac{1}{2} + c$ using a sample of size k, then for each k there exists a linear estimator that estimates π on distributions of support n to within error $(1 + o(1))\epsilon$ using a sample of size $(1+o(1))k$, and which has probability of success $1-o(\frac{1}{poly(k)})$. Additionally, such a linear estimator is given as the solution to a polynomial-sized linear program.*

To clarify, the above theorem trivially implies the following corollary:

Corollary 6.6. *Given a symmetric linear property π that is 1-relative earthmover continuous (such as entropy), if there exists an estimator which on input k independent draws from any distribution A of support n outputs a value v such that $|v - \pi(A)| < \epsilon$ with probability .51, then there exists a linear estimator which, given a sample of size $1.01k$, outputs a value v' such that $|v' - \pi(A)| \leq 2.01\epsilon$, with probability $> .99$, provided $\epsilon \geq \frac{1}{\log^{100} k}$ and k is sufficiently large.*

Given that the proof of Theorem 6.1 is via duality, unsurprisingly, it does not give any bounds on the sample complexities of these estimation tasks. Nevertheless, as mentioned above, in Chapter 7 we leverage the insights provided by Theorem 6.1 to give explicit constructions of near-optimal linear estimators for entropy, distance to uniformity, and ℓ_1 distance between pairs of distributions, establishing new bounds on the sample complexities of these tasks that seem inaccessible to the machinery and analysis techniques of Chapter 3.

6.2 Constructing Lower Bound Instances

Given a property π, a sample size k, and an upper bound n on the support size of the distributions in question, we wish to construct lower-bound instances via a principled—and in some sense mechanical—approach. Specifically, we would like to find two distributions A^+, A^- (of support at most n) which are extremal in the sense that they maximize $\delta = \pi(A^+) - \pi(A^-)$ while having the property that the distributions over fingerprints derived from sets of k independent draws from A^+, A^- respectively are *indistinguishable* with high probability. Given such a pair of distributions, if one defines D to be the distribution over distributions that assigns probability $1/2n!$ to each of the $n!$ distributions obtained from A^+ via a permutation of the domain, and assigns probability $1/2n!$ to each of the $n!$ distributions obtained from A^- via a permutation of the domain, then *no* algorithm, on

input k independent draws from a distribution chosen according to D can estimate property π to within $\pm\delta/2$.

At least intuitively, the distribution in fingerprints derived from samples of size k drawn from A^+ and A^- will be difficult to distinguish if their fingerprint expectations are very similar (relative to the size of the covariance of the distribution of fingerprints). The machinery of Chapter 5 makes this intuition rigorous; specifically, Lemma 5.3 shows that similar expectations imply similar fingerprint covariances, and the central limit theorem for "generalized multinomial distributions", Theorem 4.2, implies that if the first and second moments match, then the distributions are similar in total variational distance (ℓ_1 distance).

Since these fingerprint expectations are simply *linear* functions of the histograms, this constraint that the fingerprints of A^+ and A^- should be indistinguishable can be characterized by a set of linear constraints on the histograms of A^+ and A^-. Additionally, the constraint that A^+ and A^- have support size at most n is a linear constraint on the histograms: $\sum_{x:h_A(x)\neq 0} h_A(x) \leq n$. Since we are concerned with a symmetric linear property, π, which is given as $\pi(A) := \sum_{x:h_A(x)\neq 0} h_A(x) f_\pi(x)$, for some function f_π, our aim of maximizing the discrepancy in property values, $\pi(A^+) - \pi(A^-)$, is just the task of optimizing a linear function of the histograms. Thus, at least intuitively, we can represent the task of constructing an optimal lower-bound instance (A^+, A^-), as a semi-infinite linear program whose variables are $h_{A^+}(x), h_{A^-}(x)$, for $x \in (0, 1]$.

Before writing the linear program, there are a few details we should specify. First, to avoid the messiness that comes with semi-infinite linear programs, we will restrict ourselves to a finite set of variables, corresponding to x values in some set $X \subset (0, \frac{k^{c_1}}{2k})$ that consists of a polynomially-fine mesh of points, the details of which are largely irrelevant. Rather than solving for histogram values $h_{A^+}(x)$, it will be more convenient to solve for variables y_x^+, which are related to histogram values by $y_x^+ := h_{A^+}(x) \cdot x$. Thus y_x^+ represents the amount of probability mass accounted for by $h_{A^+}(x)$. Thus $\sum_x y_x^+ = 1$ for any distribution A^+. Additionally, we have

$$\sum_x y_x poi(kx, i) = \sum_x h(x) x \frac{e^{-kx}(kx)^i}{i!} = \frac{k}{i+1} \sum_x h(x) poi(kx, i+1).$$

Thus $\sum_x y_x poi(kx, i)$ is the expected $i+1$st fingerprint scaled by $k/(i+1)$, given a sample of size $Poi(k)$ drawn from a distribution with histogram given by $h(x) = \frac{y_x}{x}$.

Finally, we will restrict ourselves to the "infrequently-occurring" portion of the histogram: namely, we will only be concerned with fingerprint indices up to k^{c_1}, for a parameter $c_1 \in (0, 1)$, and will only solve for histogram entries corresponding to probabilities $x \leq \frac{1}{2}\frac{k^{c_1}}{k}$.

Linear Program 6.7. LOWER BOUND LP

The Lower Bound LP *corresponding to parameters* k, c_1, c_2, X, *and property* π *satisfying* $\pi(A) := \sum_{x:h(x)\neq 0} h_A(x) f_\pi(x)$, *is the following:*

$$\textit{Maximize:} \sum_{x\in X} \frac{f_\pi(x)}{x}\left(y_x^+ - y_x^-\right)$$

Subject to:

$$\forall i \in [k^{c_1}] \cup \{0\}, \qquad \sum_x \left(y_x^+ - y_x^-\right) \cdot poi(xk, i) \leq k^{-c_2}$$

$$\forall i \in [k^{c_1}] \cup \{0\}, \qquad \sum_x \left(y_x^+ - y_x^-\right) \cdot poi(xk, i) \geq -k^{-c_2}$$

$$\sum_{x\in X} y_x^+ + y_x^- \leq 2$$

$$\sum_{x\in X} \frac{y_x^+}{x} \leq n \quad \textit{and} \quad \sum_{x\in X} \frac{y_x^-}{x} \leq n$$

$$\forall x \in X, \qquad y_x^+ \geq 0, y_x^- \geq 0$$

In words, this linear program maximizes the discrepancy in property values of the distributions corresponding to y^+ and y^- subject to the following conditions: the first two constraints ensure that the fingerprint expectations of the two distributions are similar, the third condition ensures that y^+ and y^- together represent at most 2 units of probability mass, the fourth condition ensures that the two distributions have support at most n, and the last condition ensures that all elements of the support are assigned nonnegative probability values.

We now argue that the intuition for the above linear program is well founded. For any reasonably well-behaved property π, given a solution to the above linear program y^+, y^- that has objective function value v, we will construct distributions A^+, A^- whose fingerprints derived from samples of size k are indistinguishable, and A^+, A^- satisfy $\pi(A^+) - \pi(A^-) \geq v - \epsilon$ for some tiny ϵ. As shifting a property by a constant, $\pi \to \pi + C$ does not affect the property estimation problem, for the sake of convenience we assume that the property takes value 0 on the trivial distribution with support 1, though the following proposition remains true for rather extreme (though not unbounded) shifts away from this.

Proposition 6.8. *Let* π *be a* δ*-relative earthmover continuous property that takes value 0 on the trivial distribution. Given any feasible point* y^+, y^- *to the* Lower Bound LP *of Linear Program 6.7 that has objective function value* v*, then, provided* $k^{c_1} \in [\log^2 k, k^{1/32}]$ *and* $c_2 \geq \frac{1}{2} + 6c_1$*, there exists a pair of distributions* A^+, A^- *of support at most* n *such that:*

- $\pi(A^+) - \pi(A^-) > v \cdot (1 - o(1)) - O(\delta \cdot k^{-c_1} \log k)$,
- no *algorithm, when given a fingerprint derived from a sample of size* $Poi(k)$ *can distinguish whether the sample was drawn from* A^+ *versus from* A^- *with probability* $1 - \Theta(1)$.

To construct A^+, A^- from the solution y^+, y^-, there are three hurdles. First, y_x^+, y_x^- must be rounded so as to be integer multiples of $1/x$, since the corresponding histograms must be integral. Next, we must ensure that A^+, A^- have total probability mass 1. Most importantly, we must ensure that the fingerprints derived from A^+, A^- are actually indistinguishable—i.e. that we can successfully apply the central limit theorem, Theorem 4.2—a more stringent condition than simply having similar fingerprint expectations. These three tasks must be accomplished in a delicate fashion so as to ensure that $\pi(A^+) - \pi(A^-) \approx v$.

These three hurdles exactly correspond to the modifications that needed to be made to the Laguerre polynomial construction of Chapter 5 in order to yield a lower bound instance. In fact, the Lower Bound LP (Linear Program 6.7) can be seen as mechanizing the approach of Chapter 5. Whereas in that chapter there was a single explicit pair of distributions for which the goal was an indistinguishability result, the analysis there, while quite involved, essentially relied on nothing beyond the conditions of our Lower Bound LP. We refer the reader to Section 5.1 for a discussion of the intuition behind these modifications.

Proof of Proposition 6.8

In this section we show that a solution y^+, y^- to the Lower Bound LP (Linear Program 6.7), for appropriate parameters, corresponds to a pair of distributions p^+, p^- of support n whose property values differ by roughly the objective value of the linear program and which are indistinguishable given $Poi(k)$ sized samples.

The essential approach is: 1) round y^+ and y^- to distributions p^+, p^- where the constraints of the linear program imply that p^+ and p^- will have almost identical expected fingerprints; 2) invoke Lemma 5.3 as summarized in Corollary 6.9 below to argue that the fingerprint distributions will thus also have almost identical covariances; 3) invoke the central limit theorem (Theorem 4.2) to conclude that the distributions of fingerprints are essentially multivariate Gaussian distributions of almost matching expectation and covariance, and hence indistinguishable.

Corollary 6.9. *Given two distributions p^+, p^- such that when taking $Poi(k)$ sized samples from p^+, p^- respectively the expectations of the fingerprints match to within $k\epsilon$, element-by-element, for some $\epsilon > 0$, then the i, jth entry of the covariance matrices of the fingerprints match to within $O(k\sqrt{\epsilon}|\log\epsilon|(i+j)^{1/4})$.*

Proof of Proposition 6.8. We prove the lemma for the case $\delta = 1$, as otherwise, we may divide the property by δ, and only the objective of the linear program will be affected, and thus both sides of the first claim of the proposition are proportional to δ, and nothing else is affected.

We note that 1-relative earthmover continuity implies that $|\frac{f_\pi(x)}{x}| \leq |\log x|$ for any x. Further, for the range under consideration, $x \in X = (0, \frac{k^{c_1}}{2k})$, this implies $|f_\pi(x)| \leq x|\log x| < \frac{k^{c_1}}{2k}\log k$. For the case when $n < k^{1-2c_1}$, we thus have the LP constraint $\sum_{x\in X}\frac{y_x^+}{x} \leq n$ implies

that the corresponding portion of the objective function is bounded as $\left|\sum_{x\in X}\frac{f_\pi(x)}{x}y_x^+\right| \leq n\frac{k^{c_1}}{2k}\log k \leq \frac{1}{2}k^{-c_1}\log k$, implying that the objective value of the LP is at most twice this, and thus that the proposition may be trivially satisfied by the pair consisting of any distribution and itself.

The other trivial case is when (for $n \geq k^{\log k}$) there exists some $x \geq \frac{1}{n}$ for which $|\frac{f_\pi(x)}{x}| \geq \log^2 k$. Let x^+ be the number in the interval $[\frac{1}{n}, \frac{1}{k^3}]$ that maximizes $\frac{f_\pi(x)}{x}$, and let x^- be the number that minimizes this. It is straightforward to see that relative earthmover continuity implies that, for the optimum (y_x^+, y_x^-) of the linear program, $\sum_{x\in X}\frac{f_\pi(x)}{x}y_x^+ \leq \frac{f_\pi(x^+)}{x^+} + 3\log k$ and $\sum_{x\in X}\frac{f_\pi(x)}{x}y_x^- \geq \frac{f_\pi(x^-)}{x^-} - 3\log k$, implying that $\frac{f_\pi(x^+)}{x^+} - \frac{f_\pi(x^-)}{x^-} \geq v\cdot(1-o(1))$. Thus the uniform distributions on, respectively, $1/x^+$ and $1/x^-$ elements will have property values that differ by $v\cdot(1-o(1))$, and further, will have indistinguishable fingerprint distributions (statistical distance $O(1/k)$ from each other), as in either case, no element will be seen more than once in a sample of size $Poi(k)$, except with $O(1/k)$ probability.

Otherwise, if neither of the above two cases apply, then we derive the distributions p^+, p^- directly from the linear program solution (y^+, y^-), via "fattening and rounding", applying Corollary 6.9 and then the central limit theorem, Theorem 4.2 to prove indistinguishability.

We first analyze what corresponds to "total probability mass" in each of y^+, y^-. Note that for any positive λ, $\sum_{i=0}^{\infty} poi(\lambda, i) = 1$. Consider combining the first two LP constraints into, for each $i \in \{0, \ldots k^{c_1}\}$, $\left|\sum_{x\in X}(y_x^+ - y_x^-)\cdot poi(xk, i)\right| \leq k^{-c_2}$, and then summing over $i < k^{c_1}$ to yield $\left|\sum_{x\in X}(y_x^+ - y_x^-)\left(\sum_{i<k^{c_1}} poi(xk, i)\right)\right| \leq k^{c_1}k^{-c_2}$. Since X consists only of elements less than $\frac{k^{c_1}}{2k}$, and by assumption, $k^{c_1} \geq \log^2 k$, Poisson tail inequalities yield that for any such x, we have $1 > \sum_{i<k^{c_1}} poi(xk, i) > 1 - o(\frac{1}{poly(k)})$. Thus $\sum_{x\in X} y_x^+$ and $\sum_{x\in X} y_x^-$ differ by at most $2k^{c_1}k^{-c_2} + o(\frac{1}{poly(k)})$. Our first modification to y^+, y^- is to take whichever one has the higher sum and decrease its entries arbitrarily until the two sums are equal. Since $poi(xk, i) \leq 1$ in general, this will affect each constraint by at most $2k^{c_1}k^{-c_2} + o(\frac{1}{poly(k)})$, and will affect the objective function by at most $O(k^{c_1}k^{-c_2}\log^2 k)$. Next, multiply each of the entries in y^+, y^- by the largest number less than 1 that would make $\sum_{x\in X} y_x^+ \leq 1 - k^{-2c_1}$ and $\sum_{x\in X}\frac{y_x^+}{x} \leq n - k^{1-3c_1} - 1$, along with the corresponding statements for y^-. We note that the LP constraints imply this scaling is by $1 - o(1)$. Since before this scaling we had for each $i \leq k^{c_1}$ that $|\sum_x (y_x^+ - y_x^-)\cdot poi(xk, i)| \leq 3k^{c_1}k^{-c_2}$, after scaling both y^+, y^- by the same number less than 1, this will remain true.

The final steps of the transformation are to round each of y^+, y^- into histograms h^+, h^- with integral entries, though which will not have total probability mass 1; fatten: for each $i \in [k^{c_1}]$ increment $h_{i/k}^+$ and $h_{i/k}^-$ by $\phi = k^{1-4c_1}$; to make each histogram have total probability mass 1, let m be the probability mass that must be added to each (which will be the same for each, by construction), and increment both h_m^+ and h_m^- by 1. (There are some details involved in rounding appropriately, but the analysis is straightforward, and neither the objective value term nor the constraint terms corresponding to the difference in expected fingerprints will be affected by more than $o(k^{c_1}k^{-c_2})$.)

Thus h^+, h^- are now histograms of distributions, each having support at most n. Since

$poi(xk, i) = \frac{i+1}{xk} \cdot poi(xk, i+1)$, we have, since h_x^+, h_x^- correspond to rounded versions of $\frac{y_x^+}{x}, \frac{y_x^-}{x}$, that the LP constraints for a certain i yield bounds on the $i+1$st fingerprint entries, specifically, the fact that $|\sum_x (y_x^+ - y_x^-) \cdot poi(xk, i)| \leq 3k^{c_1}k^{-c_2}$ implies that the expected fingerprint entries up to k^{c_1} must match to within $3k^{1+c_1-c_2}$. Corollary 6.9 yields that the fingerprint covariances must thus match to within $O(k^{1-\frac{c_2}{2}+\frac{3c_1}{4}} \log k)$.

Further, since there are at least $\phi = k^{1-4c_1}$ elements in each distribution at each probability $\frac{i}{k}$ for $i < k^{c_1}$, Lemma 5.11 implies that each such element contributes at least $\frac{1}{12k^{7c_1/2}}$ towards the minimum covariance of either fingerprint distribution, in any direction, and thus this minimum covariance in any direction is at least $\Omega(k^{1-15c_1/2})$. Thus Theorem 4.2 yields that the total variational distance of each fingerprint distribution from the Gaussian of corresponding mean and covariance is $O(\frac{k^{4c_1/3}}{k^{(1-17c_1/2)/6}} \log n) < O(\frac{k^{3c_1}}{\sqrt[6]{k}} \log n)$. While we cannot bound n directly, distribution h^+ is indistinguishable (statistical distance $O(\frac{1}{k})$ from a distribution obtained by modifying it so that no probabilities lie below $\frac{1}{k^3}$. Thus if we modify both h^+, h^- in this fashion before applying the central limit theorem, effectively making $n \leq k^3$, and thus, for $c_1 \leq \frac{1}{20}$ we have $O(\frac{k^{3c_1}}{\sqrt[6]{k}} \log k^3) = o(1)$.

We have thus shown that h^+, h^- are indistinguishable from Gaussians of corresponding mean and covariance. Comparing multivariate Gaussians is straightforward—Proposition A.13 shows that two Gaussians are indistinguishable when the smallest covariance in any direction is $\omega(1)$ times larger than both the square of the distance between their means, and the product of the dimension (k^{c_1}) and the largest pairwise discrepancy between any entries of the two covariance matrices. The smallest covariance has been bounded by $\Omega(k^{1-15c_1/2})$; the element-wise difference between the means is at most $O(k^{1+c_1-c_2})$ implying that the square of their Euclidean distances is at most $O(k^{2+3c_1-2c_2})$. To ensure that the squared distance between the means is $o(1)$ times the smallest covariance, it is enough to let $c_2 \geq \frac{1}{2} + 6c_1$. Finally, the pairwise discrepancy between the two covariance matrices was bounded above by $O(k^{1-c_2/2+3c_1/4} \log k)$, which, plugging in our bound for c_2 yields $O(k^{3/4-5c_1/2} \log k$; the condition that this times the dimension (k^{c_1}) is $o(1)$ times the minimum covariance in any direction yields that we may set $c_1 < \frac{1}{25}$ (since by assumption $k^{c_1} \geq \log^2 k$), yielding the desired indistinguishability. □

6.3 Constructing Linear Estimators

Perhaps the most natural approach to constructing estimators for linear properties, dating back at least to the 1950's, [90] and, implicitly, far longer, is to approximate the characteristic function of the desired linear property as a linear combination of Poisson functions. To see the intuition for this, consider a property π such that $\pi(A) := \sum_{x:h_A(x)\neq 0} h_A(x) f_\pi(x)$, and assume that there exist coefficients $\beta = \beta_1, \beta_2, \ldots$ such that, for all $x \in (0, 1]$, $\sum_{i=1}^{\infty} \beta_i poi(xk, i) =$

$f_\pi(x)$. Thus for a distribution with histogram h, we have

$$\begin{aligned}\sum_{x:h(x)\neq 0} h(x)f_\pi(x) &= \sum_{x:h(x)\neq 0} h(x)\sum_{i\geq 1}\beta_i poi(kx,i)\\ &= \sum_{i\geq 1}\beta_i \sum_{x:h(x)\neq 0} h(x)poi(kx,i)\\ &= \sum_{i\geq 1}\beta_i E[\mathcal{F}(i)],\end{aligned}$$

where $E[\mathcal{F}(i)]$ is the expected ith fingerprint entry derived from $Poi(k)$ independent draws. By linearity of expectation, this quantity is precisely the expected value of the linear estimator given by the coefficients β, and thus such an estimator would have *zero* bias. Additionally, since we expect the fingerprint entries to be closely concentrated about their expectations, such an estimator would also have relatively small variance, provided that the magnitudes of the coefficients $|\beta_i|$ are small relative to $1/\sqrt{k}$. (Roughly, the contribution to the variance of the estimator from the ith fingerprint entry is the product of $|\beta_i|^2$ and the variance of the ith fingerprint entry which can be as high as k.)

For several reasons which will become apparent, instead of approximating the function $f_\pi(x)$ as $\sum_{i=1}^{\infty}\beta_i poi(kx,i)$, we instead approximate the function $\frac{f_\pi(x)}{x}$ as the 0-indexed sum $\sum_{i=0}^{\infty} z_i poi(kx,i)$. These two approaches are formally identical by setting $\beta_i = \frac{i}{k}\cdot z_{i-1}$, since $x\cdot poi(kx,i) = poi(kx,i+1)\frac{i+1}{k}$.

The following proposition formalizes this intuition, establishing the requisite relationship between the magnitudes of the coefficients, error in approximating the function $\frac{f_\pi(x)}{x}$, and the performance of the derived estimator.

Proposition 6.10. *Let π be a linear symmetric property with characteristic function f_π, and define the function $r:(0,1]\to\mathbb{R}$ by $r(x) := \frac{f_\pi(x)}{x}$. Given integers k, n, and a set of coefficients $z_0, z_1, \ldots$ define the function $err:(0,1]\to\mathbb{R}$ by*

$$r(x) = err(x) + \sum_{i\geq 0} z_i poi(xk,i).$$

If, for positive real numbers a, b, c, the following conditions hold,

1. *$|err(x)| < a + \frac{b}{x}$,*
2. *for all $j\geq 1$ let $\beta_j = \frac{j}{k}\cdot z_{j-1}$ with $\beta_0 = 0$, then for any j,ℓ such that $|j-\ell|\leq \sqrt{j}\log k$ we have $|\beta_j - \beta_\ell| \leq c\frac{\sqrt{j}}{\sqrt{k}}$*

Then the linear estimator given by coefficients $\beta_1,\ldots,\beta_k$, when given a fingerprint derived from a set of k independent draws chosen from a distribution of support at most n will estimate the property value with error at most $a + bn + c\log k$, with probability of failure $o(1/poly(k))$.

The condition on the magnitude of the error of approximation: $|err(x)| < a + \frac{b}{x}$, is designed to take into account the inevitable increase in this error as $x \to 0$. Intuitively, this increase in error is offset by the bound on support size: for a distribution of support at most n, the amount of probability mass at probability x is bounded by nx, and thus provided that the error at x is bounded by $\frac{b}{x}$, the error of the derived estimator will be at most $nx\frac{b}{x} = nb$.

Before proving Proposition 6.10, we first show that the task of finding these coefficients z_i, can be expressed as a linear program:

Linear Program 6.11. LINEAR ESTIMATOR LP
The Linear Estimator LP *corresponding to parameters* k, c_1, c_2, X, *and property* π *with characteristic function* f_π *is the following:*

$$\textit{Minimize:} \qquad 2z^a + n \cdot (z^{b+} + z^{b-}) + k^{-c_2} \sum_{i=0}^{k^{c_1}} (z_i^+ + z_i^-)$$

Subject to:

$$\forall x \in X, \qquad \sum_{i=0}^{k^{c_1}} poi(xk, i)(z_i^+ - z_i^-) \geq \frac{f_\pi(x)}{x} - (z^a + \frac{z^{b-}}{x})$$

$$\forall x \in X, \qquad \sum_{i=0}^{k^{c_1}} poi(xk, i)(z_i^+ - z_i^-) \leq \frac{f_\pi(x)}{x} + z^a + \frac{z^{b+}}{x}$$

$$\forall i \in [k^{c_1}], \qquad z_i^+ \geq 0, z_i^- \geq 0,$$

$$z^a \geq 0, z^{b+} \geq 0, z^{b-} \geq 0\,.$$

To see the relation between the above definition and Proposition 6.10, we let the coefficients $z_i = z_i^+ - z_i^-$. The parameter a in the proposition corresponds to z^a in the LP, and the parameter b in the proposition corresponds to $\max(z^{b+}, z^{b-})$. The first two sets of constraints ensure that z^a, z^{b+}, z^{b-} capture the bias of the estimator. The objective function then minimizes this bias, while also penalizing unduly large coefficients.

Proof of Proposition 6.10

Proof of Proposition 6.10. To start, consider that instead of a sample of size k, we are given $k' \leftarrow Poi(k)$ draws from the distribution. Trivially, if we prove the proposition in this setting, then, because $k' = k$ with probability at least $\frac{1}{O(\sqrt{k})}$, and our probability of failure is $o(1/poly(k))$, the conditional probability of failure given exactly k draws must also be $o(1/poly(k))$. Thus, for the remainder of the proof, assume we are given a sample of size $k' \leftarrow Poi(k)$.

The proof consists of two parts, we first argue that the first condition of the proposition guarantees that the expected value of the estimator is within $a + bn$ of the true property value—thus the resulting estimator has small *bias*. We then argue that the second condition

of the proposition implies, via basic tail bounds, that the value of the estimator will be closely concentrated about its expectation.

For a histogram h corresponding to a distribution of support at most n, we have the following:

$$\begin{aligned} r(h) &= \sum_{x:h(x)\neq 0} h(x)x \cdot r(x) \\ &= \sum_{x:h(x)\neq 0} h(x)x \left(err(x) + \sum_{i\geq 0} z_i poi(xk, i) \right) \\ &= \left(\sum_{i\geq 0} \beta_{i+1} \frac{k}{i+1} \sum_{x:h(x)\neq 0} h(x)x \cdot poi(xk, i) \right) + \sum_{x:h(x)\neq 0} h(x) \cdot x \cdot err(x) \ . \quad (6.1) \end{aligned}$$

We start by bounding the magnitude of the second term (the error term). Since $\sum_x h(x) \leq n$, and $\sum_x h(x)x = 1$, we have

$$\sum_{x:h(x)\neq 0} h(x)x \cdot err(x) \ \leq \ \sum_{x:h(x)\neq 0} h(x)x \cdot a + \sum_{x:h(x)\neq 0} h(x)x \cdot \frac{b}{x} \leq a + nb.$$

We now turn to the first term in Equation 6.1. Observe that

$$x \cdot poi(xk, i) = x\frac{(xk)^i e^{-xk}}{i!} = \frac{(xk)^{i+1} e^{-xk}}{(i+1)!} \frac{i+1}{k} = poi(xk, i+1)\frac{i+1}{k}.$$

Additionally, $\sum_{x:h(x)\neq 0} h(x)poi(xk, j)$ is simply $E[\mathcal{F}_j]$, the expected jth fingerprint entry given $Poi(k)$ draws from h. Thus the first term in Equation 6.1 becomes:

$$\sum_{i\geq 0} \beta_{i+1} \frac{k}{i+1} \sum_{x:h(x)\neq 0} h(x)x \cdot poi(xk, i) \ = \ \sum_{i\geq 0} \beta_{i+1} \sum_{x:h(x)\neq 0} h(x)poi(xk, i+1) = \sum_{i\geq 1} \beta_i E[\mathcal{F}_i],$$

which is the expected value of our estimator. Thus the bias of the estimator is at most $a + bn$, as desired.

We now argue that with high probability the error will be tightly concentrated about this bias. Tail bounds for Poisson distributions (Fact A.19) shows that for $\lambda \geq 1$, the probability of a Poisson distribution $Poi(\lambda)$ taking a value outside the range $\lambda \pm \sqrt{\lambda} \log k$ decays super-polynomially fast with k. Thus letting $j = \lfloor \lambda \rfloor$, we thus also have that $Poi(\lambda)$ will lie outside $j \pm \sqrt{j} \log k$ with $o(1/poly(k))$ probability. Thus, with all but $o(1/poly(k))$ probability, each element in the support of the distribution that occurs with probability $p(i) \geq 1/k$ will be sampled a number of times that lies in the interval $j \pm \sqrt{j} \log k$, for $j = \lfloor k \cdot p(i) \rfloor$. Thus from the second condition of the proposition, each such element will contribute to the property estimate a number in an interval of radius $c\frac{\sqrt{j}}{\sqrt{k}} \leq c\frac{\sqrt{k \cdot p(i)}}{\sqrt{k}} = c\sqrt{p(i)}$ and

hence diameter at most $2c\sqrt{p(i)}$. With a view towards applying Hoeffding's inequality, we bound the sum of the squares of the diameters of these intervals: $\sum_{i:p(i)\geq 1/k} 4c^2 \cdot p(i) \leq 4c^2$. Thus Hoeffding's inequality yields that the contribution of the elements of probability at least $1/k$ to the estimate will be within $\sqrt{4c^2}\frac{\log k}{4} = \frac{c\log k}{2}$ of its expectation, except with $2 \cdot e^{-\frac{\log^2 k}{8}} = o(1/poly(k))$ probability.

Next we consider those elements for which $p(i) < \frac{1}{k}$. We note that for $\lambda < 1$ and $\ell \geq 1$ we have $poi(\lambda, \ell) = \frac{\lambda^\ell e^{-\lambda}}{\ell!} \leq \frac{\lambda}{\ell!}$. Thus the total probability that any element of probability less than $1/k$ appears more than $\log k$ times is at most $\left(\sum_{\ell > \log k} \frac{1}{\ell!}\right) \sum_i k \cdot p(i)$. The first term is $o(1/poly(k))$, and the second term equals k, leading to a total bound of $o(1/poly(k))$. Similar to above, we may use the bound from the second condition of the proposition, for $j = 1$ to say that, except with this negligible probability, each such element with $p(i) < \frac{1}{k}$ contributes to the property estimate a value in an interval of radius $\frac{c}{\sqrt{k}}$. We further bound the variance of each such contribution: since an element of probability $p(i) < \frac{1}{k}$ will likely be seen 0 times, and in fact will be seen a nonzero number of times only with probability less than $k \cdot p(i)$, the variance of each such contribution will be at most $k \cdot p(i) \cdot (2\frac{c}{\sqrt{k}})^2 = 4c^2 \cdot p(i)$, which must thus sum to at most $4c^2$. Thus we have a sum of independent random variables each in an interval of diameter $\frac{2c}{\sqrt{k}}$ and having total variance at most $4c^2$. Bennett's inequality says that in such a case, with a sum of independent random variables of total variance σ^2, each bounded to be within m of its mean, then the probability that the sum is more than t away from its mean is at most $2\exp(-\frac{\sigma^2}{m^2} \cdot \phi(\frac{mt}{\sigma^2}))$ where the function ϕ is defined as $\phi(x) = (1+x)\log(1+x) - x$. In our present case, we consider the probability that the contribution to the estimate from the small distribution elements deviates from its mean by more than $\frac{c\log k}{2}$, yielding a bound of $2\exp(-k \cdot \phi(\frac{\log k}{4\sqrt{k}})$. Since for $x \leq 1$, $\phi(x) > \frac{x^2}{3}$, our bound becomes $2\exp(-\frac{\log^2 k}{48})$, which is negligible.

Thus in either case, the probability of deviating from the expectation by more than $\frac{c\log k}{2}$ is negligible in k, so thus the total estimate will never deviate by more than $c\log k$ from its expectation, except with probability $o(1/poly(k))$. Thus the error of our estimator is at most $a + bn + c\log k$, with probability $1 - o(1/poly(k))$. □

6.4 Duality, and Matrix Exponentials

The impetus for our main result is the observation that the Lower Bound LP (Linear Program 6.7) and the Linear Estimator LP (Linear Program 6.11) are dual linear programs. Complications arise, however, when one considers the allowable settings of the parameters. Intuitively, the Lower Bound LP only begins to make sense when $c_2 > 1/2$—namely, when the discrepancy in fingerprint expectations of the implicitly described pair of distributions is less than $k^{1/2}$, since the standard deviation in fingerprint entries can never exceed this value. Conversely, the Linear Estimator LP yields reasonable estimators only when $c_2 < 1/2$, since this corresponds to coefficients at most $1/k^{1/2}$, which, coupled with the variance in fingerprint

entries of up to k, would lead to an estimator having constant variance.

As our goal is to find a linear estimator of near–optimal performance, we start with a solution to the Lower Bound LP with objective value v, which, provided $c_2 > \frac{1}{2}$ is suitably chosen, yields a lower bound of $\approx \frac{v}{2}$, on the accuracy of estimating (via *any* algorithm) the desired property given a sample of size k. We invoke duality to yield a k-sample linear estimator with coefficients described by the vector z, and with objective value also v in the Linear Estimator LP, with parameter $c_2 > \frac{1}{2}$ as above. The issue is that the entries of z may be unsuitably large, as the only bound we have on them is that of the objective function of the Linear Estimator LP, which yields that their sum is at most $v \cdot k^{c_2}$. Since $c_2 > \frac{1}{2}$, the entries may be bigger than $\sqrt{k}$, which corresponds to an estimator with inadmissibly super-constant variance.

We now show how to transform a solution to the Linear Estimator LP with $c_2 > 1/2$ into a related estimator that: 1) has smaller coefficients; 2) takes a slightly larger sample; and 3) has almost unchanged bias. Intuitively, we have a vector of Poisson coefficients, z, whose magnitudes exceed $\sqrt{k}$, yet whose linear combination, the function $g : [0, \infty) \to \mathbb{R}$ defined as $g(x) = \sum_{i=0}^{\infty} z(i) \cdot poi(xk, i)$ closely approximates $\frac{f_\pi(x)}{x}$, and thus, despite its huge coefficients, the resulting function is small and well-behaved. The task is to transform this into a different linear combination that has smaller coefficients and is almost equally well-behaved. The principal tool we may leverage is the increased sample size. While $poi(xk, i)$ captures the Poisson functions corresponding to taking samples of size k, if we instead take samples of size $\frac{k}{\alpha}$ for $\alpha < 1$, then the corresponding functions are $poi(\frac{xk}{\alpha}, i)$, which are "thinner" than the original Poisson functions. To phrase the intuition differently, if the target function $\frac{f_\pi(x)}{x}$ is so finely structured that approximating it with "fat" Poisson functions requires coefficients exceeding $\sqrt{k}$, we might hope that using "thinner" Poisson functions will lower the required coefficients.

It is straightforward to re-express a linear combination of Poisson functions in terms of "thinner" Poisson functions. Intuitively, this is the process of simulating a $Poi(k)$-sample estimator using $Poi(\frac{k}{\alpha})$-sized samples, and corresponds to subsampling. We let z_α denote the vector of coefficients induced from subsampling by α—that is, $z_\alpha(\ell) = \sum_{i=0}^{\ell} z(i) Pr[Bin(\ell, \alpha) = i]$, where $Bin(\ell, \alpha)$ represents the binomial distribution taking ℓ trials each with success probability α. The question becomes: how does the magnitude of z_α decrease with α?

We show that the square of the L_2 norm of the vector z_α is a quadratic form in z, defined by an infinite matrix M_α. We are able to analyze these norms because of the fortuitous form of its *matrix logarithm*: there exists an infinite tri-diagonal matrix A such that for all $\alpha \in (0, 1)$, $M_\alpha = \frac{1}{\alpha} e^{(1-\alpha)A}$. We show this via the Gauss relations for contiguous hypergeometric functions. Our main result, Theorem 6.1, then follows from the fact that the quadratic form $||z_\alpha||_2^2 = z e^{\alpha X} z^\intercal$ is a *log-convex* function of α, for arbitrary z and X, and thus we can bound the size of the entries of the coefficient vector z_α, for α in the interval $(0, 1)$, by interpolating between the values of its L_2 norm at the endpoints. We now make this high-level approach is rigorous.

Matrix Exponentials of Poisson Matrices

Given a vector of Poisson coefficients, z, indexed from 0 through ∞, we may associate it with the real function $g : [0, \infty) \to \mathbb{R}$ defined as $g(x) = \sum_{i=0}^{\infty} z(i) \cdot poi(x, i)$. The input of the function g will typically by scaled by the sample size, as in $g(xk)$. Consider the task we call "resampling", that is, given coefficients z and a constant α, finding a vector z_α that yields a corresponding g_α such that $g(\alpha x) = g_\alpha(x)$ for all $x \geq 0$. That is, if z is the vector of coefficients for a k-sample estimator, z_α will be a vector of coefficients for a $\frac{k}{\alpha}$ sample estimator that has *identical* expected estimates. (See Proposition 6.10.) Constructing such an estimator for $\alpha < 1$ is straightforward—intuitively, taking a larger sample can never hurt. More specifically, given a Poisson process $Poi(\frac{x}{\alpha})$ that returns an integer ℓ, namely, "ℓ Poisson events have occurred", we may simulate a Poisson process $Poi(x)$ by, for each "event", accepting it with probability α and otherwise ignoring it; that is, when the Poisson process $Poi(\frac{x}{\alpha})$ returns ℓ, our simulation of $Poi(x)$ returns $i \leq \ell$ with probability $\alpha^i(1-\alpha)^{\ell-i}\binom{\ell}{i}$, that is, the probability that a binomial distribution with parameter α returns i heads out of ℓ draws. Symbolically, $poi(x, i) = \sum_{\ell=i}^{\infty} poi(\frac{x}{\alpha}, \ell)\alpha^i(1-\alpha)^{\ell-i}\binom{\ell}{i}$. To ensure $\sum_{i=0}^{\infty} z(i) \cdot poi(x, i) = \sum_{\ell=0}^{\infty} z_\alpha(\ell) \cdot poi(\frac{x}{\alpha}, \ell)$ for all x, we expand and then change the order of summation:

$$\begin{aligned}\sum_{i=0}^{\infty} z(i) \cdot poi(x, i) &= \sum_{i=0}^{\infty}\sum_{\ell=i}^{\infty} z(i) poi(\frac{x}{\alpha}, \ell)\alpha^i(1-\alpha)^{\ell-i}\binom{\ell}{i} \\ &= \sum_{\ell=0}^{\infty} poi(\frac{x}{\alpha}, \ell)\sum_{i=0}^{\ell} z(i)\alpha^i(1-\alpha)^{\ell-i}\binom{\ell}{i}\end{aligned}$$

which implies that we should set $z_\alpha(\ell) = \sum_{i=0}^{\ell} z(i)\alpha^i(1-\alpha)^{\ell-i}\binom{\ell}{i}$, as we do in the following construction.

Construction 6.12 (Resampling)**.** *Given a vector z, indexed from 0 through ∞, let z_α be the* resampled *version of z, defined as $z_\alpha(\ell) = \sum_{i=0}^{\ell} z(i)\alpha^i(1-\alpha)^{\ell-i}\binom{\ell}{i}$. We define $z_1 := z$.*

Lemma 6.13. *Resampling a vector z by factor α to yield z_α satisfies $\sum_{i=0}^{\infty} z(i) \cdot poi(x, i) = \sum_{\ell=0}^{\infty} z_\alpha(\ell) \cdot poi(\frac{x}{\alpha}, \ell)$ for all $x \geq 0$.*

To bound the size of the coefficients as α decreases, we prove the following general structural result, which is central to this section.

Proposition 6.14. *For an arbitrary vector z of finite support and for $\alpha \in (0, 1]$, let z_α be the α-resampled version of z, and let $||\cdot||_2$ denote the L_2 norm. Then $\sqrt{\alpha}||z_\alpha||_2$ is log-convex in α. Further, letting g denote the function represented by z, that is, $g(x) = \sum_{i=0}^{\infty} z(i) \cdot poi(x, i)$, then the limit as α approaches 0 of $\sqrt{\alpha}||z_\alpha||_2$ equals the L_2 norm of g.*

We first set up some preliminaries that will help us characterizes the behavior of $||z_\alpha||_2$.

Definition 6.15. *Define the matrix M_α for $\alpha \in (0,1)$ by $M_\alpha(i,j) = \sum_{\ell=0}^{\infty} \binom{\ell}{i}\binom{\ell}{j}\alpha^{i+j}(1-\alpha)^{2\ell-i-j}$, and the matrix A such that $A(i,i) = 1-2i, A(i,i+1) = A(i+1,i) = i+1$ for all $i \geq 0$, and with all other entries set to zero, where both matrices are indexed by the nonnegative integers.*

The matrix M_α is chosen so that, trivially, $||z_\alpha||_2^2 = zM_\alpha z^\intercal$. We relate M_α to the much simpler matrix A by the following lemma, in terms of *matrix exponentiation*.

Lemma 6.16. $M_\alpha = \frac{1}{\alpha}e^{(1-\alpha)A}$.

Proof. Note that $\frac{d}{d\alpha}\frac{1}{\alpha}e^{(1-\alpha)A} = -A\frac{1}{\alpha}e^{(1-\alpha)A} - \frac{1}{\alpha^2}e^{(1-\alpha)A}$, so we prove the result by showing that $\frac{d}{d\alpha}M_\alpha = -AM_\alpha - \frac{1}{\alpha}M_\alpha$, and noting that when $\alpha = 1$ we have that $\frac{1}{\alpha}e^{(1-\alpha)A}$ equals the identity matrix, which is easily seen to equal $\lim_{\alpha\to 1} M_\alpha$. We treat this as our initial condition.

We first evaluate $M_\alpha(i,j)$. Assume for the moment that $i \leq j$. Thus the sum that defines $M_\alpha(i,j)$ only has nonzero terms for $\ell \geq j$, so we may substitute $m = \ell - j$ and sum over m going from 0 to infinity instead. We aim to represent the terms using *rising factorial* notation, namely, for a number x, let $(x)_m$ denote $x(x+1)(x+2)\cdot\ldots\cdot(x+m-1)$. Further, aiming to use only an argument of m in the rising factorial notation for the mth component of the sum, we note that $\binom{\ell}{j} = \binom{m+j}{j} = \frac{(j+1)_m}{m!}$ and $\binom{\ell}{i} = \binom{m+j}{i} = \frac{(i+1)_{m+j-i}}{(m+j-i)!} = \frac{(j+1)_m}{(j-i+1)_m}\binom{j}{i}$. Thus $M_\alpha(i,j) = \alpha^{i+j}(1-\alpha)^{j-i}\binom{j}{i}\sum_{m=0}^{\infty}\frac{(j+1)_m(j+1)_m}{(j-i+1)_m m!}(1-\alpha)^{2m}$, where we may immediately read off the sum as the hypergeometric function ${}_2F_1(j+1,j+1,j-i+1;(1-\alpha)^2)$. Thus for $i \leq j$,

$$M_\alpha(i,j) = \alpha^{i+j}(1-\alpha)^{j-i}\binom{j}{i}{}_2F_1(j+1,j+1,j-i+1;(1-\alpha)^2).$$

We now turn to the claim, that $AM_\alpha + \frac{1}{\alpha}M_\alpha + \frac{d}{d\alpha}M_\alpha = 0$. Because of the structure of A, the (i,j)th entry of AM_α equals $iM_\alpha(i-1,j) + (1-2i)M_\alpha(i,j) + (i+1)M_\alpha(i+1,j)$. Further, to evaluate the derivative of M_α, we note that in general, we have the Gauss relation $\frac{d}{dt}{}_2F_1(x,y,z;t) = \frac{z-1}{t}\left({}_2F_1(x,y,z-1;t) - {}_2F_1(x,y,z;t)\right)$. Combining everything yields a linear combination of the hypergeometric functions ${}_2F_1(j+1,j+1,j-i;(1-\alpha)^2)$, ${}_2F_1(j+1,j+1,j-i+1;(1-\alpha)^2)$, and ${}_2F_1(j+1,j+1,j-i+2;(1-\alpha)^2)$ which equals zero because of the corresponding Gauss relation between these three contiguous hypergeometric functions. (A slightly different linear combination arises for the border case where $i = j$, but again, the Gauss relations are sufficient.) □

We now prove our main proposition.

Proof of Proposition 6.14. Since by construction, $||z_\alpha||_2^2 = zM_\alpha z^\intercal$, and by Lemma 6.16 $M_\alpha = \frac{1}{\alpha}e^{(1-\alpha)A}$, we have that $(\sqrt{\alpha}||z_\alpha||_2)^2 = ze^{(1-\alpha)A}z^\intercal$. Substituting $1-\alpha \to \alpha$ yields that this is a log-convex function of α provided $ze^{\alpha A}z^\intercal$ is. Denote $f(\alpha) = ze^{\alpha A}z^\intercal$. We note that since the second derivative of the logarithm of a positive function f equals $\frac{f''\cdot f - f'^2}{f^2}$, we

have that f is log-convex provided $f \cdot f'' \geq f'^2$. Since the vectors z are constant, we may differentiate $e^{\alpha A}$ and post- and pre-multiply by z. By definition, $\frac{d}{d\alpha}e^{\alpha A} = Ae^{\alpha A}$, and thus further $\frac{d^2}{d\alpha^2}e^{\alpha A} = A^2e^{\alpha A}$. We note that the power series representation $e^X := \sum_{i=0}^{\infty} \frac{X^i}{i!}$ implies, since A is symmetric, that A commutes with $e^{\alpha A}$. Since the square of $e^{\frac{1}{2}\alpha A}$ equals $e^{\alpha A}$, we may thus reexpress the first derivative of f as $ze^{\frac{1}{2}\alpha A}Ae^{\frac{1}{2}\alpha A}z^\intercal$, and the second derivative as $ze^{\frac{1}{2}\alpha A}A^2e^{\frac{1}{2}\alpha A}z^\intercal$. Letting $v_\alpha := ze^{\frac{1}{2}\alpha A}$, since all the matrices are symmetric, we thus have that $f(\alpha) = vv^\intercal$, $f'(\alpha) = vAv^\intercal$, and $f''(\alpha) = vA^2v^\intercal$, and the desired relation $f \cdot f'' \geq f'^2$ is simply the Cauchy-Schwarz inequality: $f'(\alpha)^2 = (vAv^\intercal)^2 \leq |vA|^2|v|^2 = (vAAv^\intercal)(vv^\intercal) = f''(\alpha) \cdot f(\alpha)$.

Finally, we show that for $g(x) = \sum_{i=0}^{\infty} z(i) \cdot poi(x, i)$, we have $\lim_{\alpha\to 0} \sqrt{\alpha}||z_\alpha||_2 = ||g||_2$. Note that $z_\alpha(\ell) = \sum_{i=0}^{\ell} z(i)Bin(\ell, \alpha, i)$, where $Bin(\ell, \alpha, i)$ denotes the probability that a binomial distribution with parameter α will draw i heads from ℓ trials. Recall that as α approaches 0, the binomial distribution becomes very well approximated by the Poisson process of parameter $\alpha\ell$, yielding $z_\alpha(\ell) \approx \sum_i z(i)poi(\alpha\ell, i) = g(\alpha\ell)$. Thus $\lim_{\alpha\to 0} \alpha \cdot \sum_\ell z_\alpha(\ell)^2 = \int g(x)^2 dx = ||g||_2^2$, yielding the claim. □

We have thus shown that $\sqrt{\alpha}||z_\alpha||_2$ varies log-concavely with α; to complete the analysis of its behavior for $\alpha \in (0, 1)$ we need to understand its behavior at the endpoints. The Linear Estimator LP provides us, in rough form, with bounds on both the size of the coefficients $z(i)$, and the size of the function the coefficients represent, $g(x) = \sum_{i=0}^{\infty} z(i) \cdot poi(x, i)$—that is, intuitively, bounds for the $\alpha = 1$ and $\alpha = 0$ cases respectively. However, we must eliminate one odd possibility before proceeding: for very small x, the linear program essentially bounds the linear combination of poisson functions as a multiple of $1/x$. The function $1/x$, however, has *infinite* L_2 norm, so a $1/x$ blowup would in fact be unworkable. Fortunately, this kind of blowup is in fact overly pessimistic: a linear combination of Poisson functions with bounded coefficients cannot "blowup" like $1/x$ at the origin; the following lemma characterizes this.

Lemma 6.17. *Given a vector z of coefficients that induces a function $g(x) = \sum_{i=0}^{\infty} z(i) \cdot poi(x, i)$, where for each i, $|z(i)|$ is at most some bound b, and $|g(x)| \leq \frac{1}{x}$, then the L_2 norm of g is $O(\log b)$.*

Proof. We note that $\int_1^\infty g(x)^2 dx \leq 1$, so we need only bound the blowup as x approaches 0. We reexpress $g(x)^2$ as a sum of "thin" Poisson functions, $g(x)^2 = \sum_{\ell=0}^{\infty} \omega(\ell)poi(2x, \ell)$ via $poi(x, i) \cdot poi(x, j) = poi(2x, i + j)2^{-(i+j)}\binom{i+j}{i}$, and note that the new coefficients are bounded by b^2 since for any index ℓ, we have $\omega(\ell) = \sum_{i=0}^{\ell} 2^{-\ell}\binom{\ell}{i}z(i)z(\ell - i)$, and $\sum_{i=0}^{\ell} 2^{-\ell}\binom{\ell}{i} = 1$.

We may further alter $g(x)^2$ so that it is still expressible by Poisson functions as: $g(x)^2e^{-2x} = \sum_{\ell=0}^{\infty} \omega(\ell)poi(2x, \ell)e^{-2x} = \sum_{\ell=0}^{\infty} \omega(\ell)2^{-\ell}poi(4x, \ell)$. Since $|\omega(\ell)| \leq b^2$, we may cut off this sum at $\ell = 2\log_2 b$ without altering its value by more than 1. Define $h(x) = \sum_{\ell=0}^{2\log_2 b} \omega(\ell)2^{-\ell}poi(4x, \ell)$. We note that the integral of h differs from the integral of $g(x)^2e^{-2x}$ by less than 1, since $\int_0^\infty poi(4x, \ell)dx = \frac{1}{4}$, and thus the integral of the ℓth term of the sum is bounded by $\frac{1}{4}b^2 2^{-\ell}$, so the terms beyond $2\log_2 b$ will contribute at most $\frac{1}{4}$ to the integral.

We express h as $e^{-4x}P(x)$ where P is some polynomial of degree $2\log_2 b$. We may thus approximate $\int_0^1 h(x)^2$ to within factor e^4 by $\int_0^1 P(x)dx$. Gauss-Legendre quadrature trivially implies that if a polynomial of degree d is bounded on the interval $[\frac{1}{d^2}, 1]$, then its integral over $[0, 1]$ is bounded identically. Since by assumption, $|h(x)| \leq e^{-2x}\frac{1}{x^2} + 1$, where the final 1 captures the error from truncating at $2\log_2 b$, setting $d = 2\log_2 b$ yields the desired result. □

Finally, we assemble the pieces to transform a solution to the linear program into a near-optimal estimator, using Proposition 6.10 for the final step. The following construction will yield a vector of "Poisson coefficients," in terms of a parameter α and a solution to the linear estimator LP, that will yield, under Proposition 6.10, a $\frac{k}{\alpha}$-sample estimator whose performance—when α converges to 1 suitably—will be good enough to yield Theorem 6.1.

Construction 6.18. *Given a solution $z = z^+ - z^-$ to the linear estimator LP for a property represented as f_π, letting $\epsilon = \frac{2\log k}{k^{c_1}}$, for parameter $\alpha \in (0, 1)$, construct the* α-scaled estimator *as follows: Attenuate the coefficients, defining $\tilde{z}(i) := z(i) \cdot (1-\epsilon)^i$. Resample $\tilde{z}$ by α to yield $\tilde{z}_\alpha$, as in Construction 6.12. Finally, construct the Poisson coefficients $z_E(i) := \tilde{z}_\alpha(i) + (1 - e^{-\epsilon\alpha i})f_\pi(\frac{(i+1)\alpha}{k})\frac{k}{(i+1)\alpha}$ for $i \leq k$.*

For the next proposition, it will simplify the analysis to scale the property π under consideration so that it is 1-relative earthmover continuous, and shift it so that it takes value 0 on the trivial distribution with support 1: $\pi(\text{“1”}) = 0$. Clearly such a transform will not affect the behavior of linear estimators that are correspondingly transformed.

Proposition 6.19. *Let $z = z^+ - z^-$ be a solution to the linear estimator LP that has objective value v for a property π that is 1-relative earthmover continuous and takes value 0 on the trivial distribution, where $k^{c_1} \in [\log^2 k, k^{1/4}]$ and $c_2 < 1$. Then Proposition 6.10 when applied to the results of Construction 6.18 for $\alpha \in (\frac{1}{2}, 1)$ will yield a $\frac{k}{\alpha}$-sample estimator with error $v \cdot (1 + o(1)) + O(k^{\alpha c_2 + (3/2-\alpha)c_1 - 1/2}\log^4 k + k^{-c_1/2}\log^2 k)$ and probability of failure $o(\frac{1}{poly(k)})$ provided $v \leq \log^2 k$; if $v > \log^2 k$ then the "estimator" that returns 0 always will have error at most $v \cdot (1 + o(1))$.*

Proof. Defining the linear combination of Poissons $g(x) := \sum_{i=0}^{k^{c_1}} poi(xk, i)z(i)$, we first note that if we attenuate the coefficients, as in the construction, letting $\tilde{z}(i) := z(i) \cdot (1-\epsilon)^i$ and consider the corresponding linear combination of Poissons, $\tilde{g}(x)$, then g and $\tilde{g}$ are related as $\tilde{g}(x) := \sum_{i=0}^{k^{c_1}} z(i)(1-\epsilon)^i \frac{e^{-xk}(xk)^i}{i!} = g(x \cdot (1-\epsilon))e^{-\epsilon kx}$. We then resample this vector by α to yield $\tilde{z}_\alpha$. Our first task is to bound the coefficients here. We do this using the log-convexity of the resampling operation, as shown by Proposition 6.14. Explicitly, Proposition 6.14 implies $\sqrt{\alpha}||\tilde{z}_\alpha||_2 \leq ||\tilde{z}||_2^\alpha \cdot ||\tilde{g}||_2^{1-\alpha}$. We must bound each term on the right hand side. For the first term, we note that because each term in the objective function of the linear program is non-negative, the objective value v thus bounds the portion of the objective function $k^{-c_2}\sum_{i=0}^{k^{c_1}} |z(i)|$. Thus the ℓ_1 norm of z is at most $v \cdot k^{c_2}$, which hence also bounds the ℓ_1

norm of the attenuated coefficients, $\tilde{z}$; further, the ℓ_1 norm of a vector bounds its L_2 norm, so we have $||\tilde{z}||_2 \leq v \cdot k^{c_2}$.

Bounding the second term, $||\tilde{g}||_2$ takes a bit more work. Consider the characteristic function of the property, f_π. By assumption, $f(1) = 0$. Further, relative-earthmover continuity imposes the condition $|f_\pi(x)/x - f_\pi(y)/y| \leq |\log \frac{x}{y}|$; letting $y = 1$ yields $|f_\pi(x)|/x \leq |\log x|$. We note that for the range of x considered in the linear program, $x \in (0, \frac{k^{c_1}}{2k})$, we may crudely bound $|\log x| < \frac{k^{c_1}}{kx} \log k$. For each such x, the linear program bounds the positive error of the Poisson approximation by $z^a + \frac{z^{b+}}{x}$ and the negative error by $z^a + \frac{z^{b-}}{x}$, where the objective function penalizes large z^a, z^{b+}, z^{b-} via the term $2z^a + n \cdot (z^{b+} + z^{b-})$. We consider two cases. For $n < k^{1-c_1}$ we note that if we replace the triple (z^a, z^{b+}, z^{b-}) by $(0, z^{b+} + \frac{z^a}{n}, z^{b-} + \frac{z^a}{n})$ then the objective function remains unchanged, and further, each of the linear program constraints becomes looser, as, since $x < \frac{1}{n}$, we have $z^a + \frac{z^{b+}}{x} \leq 0 + \frac{z^{b+} + z^a/n}{x}$ with the corresponding statement for z^{b-}. Thus at optimum, we may assume $z^a = 0$. Since as noted above, $|\frac{f_\pi(x)}{x}| < \frac{k^{c_1}}{kx} \log k$, we have that letting $z^{b+} = z^{b-} = \frac{k^{c_1}}{k} \log k$ and all the other variables being 0 is a feasible point of the linear program with objective value $n(z^{b+} + z^{b-})$ and thus the since all variables of the linear program are restricted to be nonnegative, the sum $z^{b+} + z^{b-} = 2\frac{k^{c_1}}{k} \log k$ bounds both z^{b+} and z^{b-} at the optimum of the linear program. Thus at optimum, the bound in each constraint of the linear program may be bounded as $z^a + \frac{z^{b\pm}}{x} \leq 2\frac{k^{c_1}}{xk} \log k$. We analyze this in a moment.

For the other case, when $n \geq k^{1-c_1}$, we note that the bound in each constraint of the linear program may be bounded as $z^a + \frac{z^{b\pm}}{x} \leq 2z^a + \frac{z^{b+} + z^{b-}}{x} \frac{n}{k^{1-c_1}} \leq \frac{2z^a + n \cdot (z^{b+} + z^{b-})}{xk^{1-c_1}} \leq \frac{v}{xk^{1-c_1}}$. Thus for both cases we have the bound $z^a + \frac{z^{b\pm}}{x} \leq \frac{k^{c_1}}{xk} \max\{2 \log k, v\}$. Adding this to the above bound $|\frac{f_\pi}{x}| \leq \frac{k_1^c}{xk} \log k$ yields a bound on the right hand sides of each constraint in the linear program, namely a bound on g, the left hand side of the linear program constraints, of $|g(x)| \leq \frac{k^{c_1}}{xk}(v + 3 \log k)$ for $x \in (0, \frac{k^{c_1}}{2k})$. To bound $|g(x)|$ for $x \geq \frac{k^{c_1}}{2k}$ we note that g is a linear combination of Poissons with coefficients as high as $v \cdot k^{c_2}$, and may thus reach as high as $v \cdot k^{c_2}$. We note, however, that we are dealing with the *attenuated* version of g, namely, as derived above, $\tilde{g}(x) = g(x \cdot (1 - \epsilon))e^{-\epsilon k x}$ where $\epsilon = \frac{2 \log k}{k_1^c}$. Thus at $x = \frac{k^{c_1}}{2k}$ the attenuation is already $e^{-\log k} = \frac{1}{k}$, and will clearly decay at least as fast as $\frac{1}{x}$ beyond this. Thus, for all x, we have $\tilde{g}(x) \leq 2\frac{k^{c_1}}{xk}(v + 3 \log k)$, where the 2 is a crude bound on $\frac{1}{1-\epsilon}$. Thus if we scale $\tilde{g}$ by $\frac{1}{2k^{c_1}(v+3 \log k)}$ so that it is bounded by $\frac{1}{kx}$ and apply Lemma 6.17 to $\frac{\tilde{g}(xk)}{2k^{c_1}(v+3 \log k)}$, we thus have a bound on the L_2 norm of $\tilde{g}$ of $||\tilde{g}||_2 = O(2k^{c_1}(v + 3 \log k) \log(v \cdot k^{c_2})) = O(k^{c_1} \log^3 k)$ for $v < \log^2 k$.

Thus, as discussed at the beginning of the proof, we may combine this bound and the bound $||\tilde{z}||_2 \leq v \cdot k^{c_2}$ via log-convexity to yield a bound on the L_2 norm of the resampled coefficients: $\sqrt{\alpha}||\tilde{z}_\alpha||_2 = O(k^{\alpha c_2 + (1-\alpha)c_1} \cdot \log^3 k)$. We will consider cases where $\alpha \in (\frac{1}{2}, 1)$, so we may drop the $\sqrt{\alpha}$ term from the left hand side while preserving the asymptotic expression.

As each element of $\tilde{z}_\alpha$ must be at most the L_2 norm of the whole, we have the element-by element bound of $|\tilde{z}_\alpha(i)| = O(k^{\alpha c_2 + (1-\alpha)c_1} \log^3 k)$. We are now in a position to analyze the application of Proposition 6.10 to the coefficients $z_E(i) = \tilde{z}_\alpha(i) + (1 - e^{\alpha \epsilon i}) f_\pi(\frac{(i+1)\alpha}{k}) \frac{k}{(i+1)\alpha}$,

where for $i > k$ we extend this definition by letting $z_E(i) = z_E(k)$.

We first analyze the second condition of Proposition 6.10, where we separately bound the contributions from $\tilde{z}_\alpha$ and from the remaining term. We have just derived the bound $|\tilde{z}_\alpha(i)| = O(k^{\alpha c_2+(1-\alpha)c_1}\log^3 k)$, and we use this for $i \le 2k^{c_1}$. Our aim is to find a bound c such that for all j, ℓ between 1 and $2k^{c_1}$ such that $|j-\ell| \le \sqrt{j}\log k$ we have $c > \frac{\sqrt{k}}{\sqrt{j}}|\frac{j}{k}\tilde{z}_\alpha(j-1) - \frac{\ell}{k}\tilde{z}_\alpha(\ell-1)|$. We note that $\frac{j}{\sqrt{j}} = O(k^{c_1/2})$, and that $\frac{\ell}{\sqrt{j}} \le \frac{j+\sqrt{j}\log k}{\sqrt{j}} = O(k^{c_1/2})$, which implies that we may set c to be $O(k^{c_1/2})$ times our just-derived bound on $|\tilde{z}_\alpha(i)|$, namely, $c = O(k^{\alpha c_2+(3/2-\alpha)c_1-1/2}\log^3 k)$.

For the case where one of j, ℓ is greater than $2k^{c_1}$ we now derive a bound on how $\tilde{z}_\alpha(i)$ decays for large i. As each original coefficient $z(i)$ is bounded by $v \cdot k^{c_2}$, each attenuated coefficient is bounded as $|\tilde{z}(i)| \le v \cdot k^{c_2}(1-\epsilon)^i$. Assume for the moment that each coefficient equals exactly this. The corresponding linear combination of Poissons is hence $\tilde{g}(x) = v \cdot k^{c_2}e^{-\epsilon k x}$; resampling by α factor replaces x with αx, which has the effect of replacing ϵ by $\alpha\epsilon$, yielding coefficients $v \cdot k^{c_2}(1-\alpha\epsilon)^i$. Since resampling involves a *positive* linear combination of the coefficients, we thus have the bound $|\tilde{z}_\alpha(i)| \le v \cdot k^{c_2}(1-\alpha\epsilon)^i$. As $v < \log^2 k$ and $(1-\alpha\epsilon)^i < e^{-\alpha\epsilon i} = e^{-\frac{2\alpha i \log k}{k^{c_1}}}$, then for $\alpha \ge \frac{1}{2}$, $c_2 < 1$, and $i > k^{c_1}$ we have $|\tilde{z}_\alpha(i)| < \log^2 k$ and decaying by another factor of k for each addition of k^{c_1} to i. Thus, trivially, the c from above applies to this region.

We now examine the contribution to the second condition of Proposition 6.10 from the remaining term of z_E, namely $(1-e^{-\alpha\epsilon i})f_\pi(\frac{(i+1)\alpha}{k})\frac{k}{(i+1)\alpha}$. As above, we desire a bound $c' > \frac{\sqrt{k}}{\alpha\sqrt{j}}|(1-e^{-\alpha\epsilon(j-1)})f_\pi(\frac{j\alpha}{k}) - (1-e^{-\alpha\epsilon(\ell-1)})f_\pi(\frac{\ell\alpha}{k})|$ for pairs $j, \ell \ge 1$ such that $|j-\ell| \le \sqrt{j}\log k$. For the case that $j \le \sqrt{k}$, we use the bound $f_\pi(x) \le x|\log x|$, the trivial bound $(1-e^y) < 1$ for any y, and the triangle inequality to yield a bound of $c' = O(k^{-1/4}\log k)$. For $j > \sqrt{k}$, we note that $e^{-\alpha\epsilon(j-1)}$ and $e^{-\alpha\epsilon(\ell-1)}$ are both negligible in k, and thus it is sufficient to bound $\frac{\sqrt{k}}{\alpha\sqrt{j}}|f_\pi(\frac{j\alpha}{k}) - f_\pi(\frac{\ell\alpha}{k})|$. To bound this change in f_π, recall that for general x, y we have $|f_\pi(x)/x - f_\pi(y)/y| \le |\log\frac{x}{y}|$, yielding $|f_\pi(\frac{j\alpha}{k}) - \frac{j}{\ell}f_\pi(\frac{\ell\alpha}{k})| = O(\frac{j\alpha}{k}|\log\frac{\ell}{j}| = O(\frac{\sqrt{j}\log k}{k})$. We add this to the bound $|\frac{\ell-j}{\ell}f_\pi(\frac{\ell\alpha}{k})| = O(\frac{\sqrt{j}\log^2 k}{k})$. Combining, yields a bound of $c' = \frac{\sqrt{k}}{\alpha\sqrt{j}}O(\frac{\sqrt{j}\log^2 k}{k}) = O(\frac{\log^2 k}{\sqrt{k}})$. We note that, since $\alpha \in (\frac{1}{2}, 1)$ and $c_2 > \frac{1}{2}$, the bound derived earlier of $c = O(k^{\alpha c_2+(3/2-\alpha)c_1-1/2}\log^3 k)$ is at least $O(k^{-1/4}\log^3 k)$, which thus subsumes the two just derived bounds of respectively $O(k^{-1/4}\log k)$ and $O(\frac{\log^2 k}{\sqrt{k}})$. Thus we take $c = O(k^{\alpha c_2+(3/2-\alpha)c_1-1/2}\log^3 k)$ for the bound on the second condition of Proposition 6.10.

We now turn to the first condition of Proposition 6.10, essentially examining the bias of the estimator. We must compare $\frac{f_\pi(x)}{x}$ to the linear combination of Poissons $\sum_{i\ge 0} z_E(i) \cdot poi(\frac{xk}{\alpha}, i)$. We consider each of the two terms of z_E separately, and start by comparing the fraction of our target $(1-e^{-\epsilon k x})\frac{f_\pi(x)}{x}$ to the combination of Poissons corresponding to the second term of z_E, namely $\sum_{i\ge 0}(1-e^{-\alpha\epsilon i})f_\pi(\frac{(i+1)\alpha}{k})\frac{k}{(i+1)\alpha} \cdot poi(\frac{xk}{\alpha}, i)$. Since $\sum_{i\ge 0} poi(\frac{xk}{\alpha}, i) =$

1, we may thus bound

$$\sum_{i\geq 0}\left|(1-e^{-\alpha\epsilon i})f_\pi\left(\frac{(i+1)\alpha}{k}\right)\frac{k}{(i+1)\alpha}-(1-e^{-\epsilon kx})\frac{f_\pi(x)}{x}\right|\cdot poi(\frac{xk}{\alpha},i)$$
$$\leq\left(\sum_{i\geq 0}\left|e^{-\alpha\epsilon i}-e^{-\epsilon kx}\right|\frac{f_\pi(x)}{x}\cdot poi(\frac{xk}{\alpha},i)\right)$$
$$+\left(\sum_{i\geq 0}(1-e^{-\alpha\epsilon i})poi(\frac{xk}{\alpha},i)\left|f_\pi\left(\frac{(i+1)\alpha}{k}\right)\frac{k}{(i+1)\alpha}-\frac{f_\pi(x)}{x}\right|\right)$$

We bound each of the sums separately, noting throughout that $\alpha\in(\frac{1}{2},1)$. Recalling that $\epsilon=\frac{2\log k}{k^{c_1}}$, we bound the first sum for $x\leq\frac{1}{\epsilon k}$ by noting that since e^{-y} has derivative at most 1 for positive inputs, we have $|e^{-\alpha\epsilon i}-e^{-\epsilon kx}|\leq\alpha\epsilon|i-\frac{kx}{\alpha}|$. Since $\left|\frac{f_\pi(x)}{x}\right|\leq|\log x|$, the first sum is thus bounded by $\alpha\epsilon|\log x|$ times the expected distance of $Poi(\frac{xk}{\alpha})$ from its mean, which is bounded by the square root of its variance, namely $\sqrt{\frac{kx}{\alpha}}$, yielding a bound on the first sum of $O(\epsilon\sqrt{kx}|\log x|)$. We apply this bound for $x\leq\frac{1}{\epsilon k}$; since $|x\log x|$ is an increasing function of x for $x<e^{-1}$, we evaluate this bound by plugging in $x=\frac{1}{\epsilon k}$ to yield $O(k^{-c_1/2}\log^{3/2}k)$. For $x>\frac{1}{\epsilon k}$, we note that $poi(\frac{xk}{\alpha},i)$ is negligible unless i is within a factor of 2 of $\frac{xk}{\alpha}$. Thus for $\epsilon i\geq\frac{\epsilon kx}{2a}$ we bound $|e^{-\alpha\epsilon i}-e^{-\epsilon kx}|\leq\alpha\epsilon|i-\frac{kx}{\alpha}|e^{-\epsilon kx/2}$, and thus, corresponding to the above bound on the first sum, we now have a bound of $O(\epsilon\sqrt{kx}|\log x|e^{-\epsilon kx/2})$. Because of the exponential term, this expression is maximized for $x=O(\frac{1}{\epsilon k})$, and as above we may bound the first sum as $O(k^{-c_1/2}\log^{3/2}k)$.

For the second sum, consider $x>\frac{1}{k\sqrt{\epsilon}}$. We note that $|\frac{f_\pi(y)}{y}-\frac{f_\pi(x)}{x}|\leq|\log\frac{x}{y}|$, which, when y is within a factor of two of x is bounded as $2\frac{|y-x|}{x}$. Since with all but negligible probability, when i is drawn from $Poi(\frac{xk}{\alpha})$ we will have $\frac{(i+1)\alpha}{k}$ within a factor of 2 of x, we have a bound for this case of $\left|f_\pi\left(\frac{(i+1)\alpha}{k}\right)\frac{k}{(i+1)\alpha}-\frac{f_\pi(x)}{x}\right|\leq 2\frac{|(i+1)\alpha/k-x|}{x}=2\frac{|(i+1)-xk/\alpha|}{xk/\alpha}$. Further, $1-e^{-\alpha\epsilon i}\leq\alpha\epsilon i\leq\frac{2xk}{\alpha}$, and is also at most 1. Thus we can bound the second term by $O(\frac{\min\{1,\epsilon kx\}}{xk}$ times the expected distance of $Poi(\frac{xk}{\alpha})$ from its mean; this latter quantity is bounded by $O(\sqrt{xk})$, yielding a bound on the second sum of $O(\frac{\min\{1,\epsilon kx\}}{\sqrt{xk}})$. The expression inside the asymptotic notation is maximized when $x=\frac{1}{\epsilon k}$, yielding a bound on the second sum of $O(\sqrt{\epsilon})=O(k^{-c_1/2}\log^{1/2}k)$ for $x>\frac{1}{k\sqrt{\epsilon}}$. Otherwise, for $x\leq\frac{1}{k\sqrt{\epsilon}}$ we analyze the second sum in two parts, noting that, since $i\geq 0$, we have $\frac{(i+1)\alpha}{k}\geq\frac{\alpha}{k}$, yielding that $\left|f_\pi\left(\frac{(i+1)\alpha}{k}\right)\frac{k}{(i+1)\alpha}\right|\leq|\log\frac{\alpha}{k}|<1+\log k$. Since $(1-e^{-\alpha\epsilon i})\leq\alpha\epsilon i$, we have $\sum_{i\geq 0}(1-e^{-\alpha\epsilon i})\left|f_\pi\left(\frac{(i+1)\alpha}{k}\right)\frac{k}{(i+1)\alpha}\right|\cdot poi(\frac{xk}{\alpha},i)\leq(1+\log k)\alpha\epsilon\cdot\mathrm{E}[Poi(\frac{xk}{\alpha})]=\epsilon xk(1+\log k)$. For $x<\frac{1}{k\sqrt{\epsilon}}$ this is $O(\sqrt{\epsilon}\log k)=O(k^{-c_1/2}\log^{3/2}k)$. The remaining part of the second sum we easily bound as $\sum_{i\geq 0}(1-e^{-\alpha\epsilon i})\left|\frac{f_\pi(x)}{x}\right|\cdot poi(\frac{xk}{\alpha},i)\leq\alpha\epsilon\left|\frac{f_\pi(x)}{x}\right|\sum_{i\geq 0}i\cdot poi(\frac{xk}{\alpha},i)=\epsilon xk\left|\frac{f_\pi(x)}{x}\right|\leq\epsilon xk|\log x|$. This last

expression is increasing in x, and hence we have a bound for $x \leq \frac{1}{k\sqrt{\epsilon}}$ of $O(\sqrt{\epsilon}\log k) = O(k^{-c_1/2}\log^{3/2} k)$. Thus we have shown that the portion of z_E other than $\tilde{z}_\alpha$ contributes to the linear combination of Poissons a function that is within $O(k^{-c_1/2}\log^{3/2} k)$ of $(1-e^{-\epsilon kx})\frac{f_\pi(x)}{x}$.

It remains to compare the remaining portion of z_E with the remaining fraction of $\frac{f_\pi(x)}{x}$, namely, compare $\sum_{i\geq 0}\tilde{z}_\alpha(i)\cdot poi(\frac{xk}{\alpha}, i)$ to $e^{-\epsilon kx}\frac{f_\pi(x)}{x}$. We start the analysis by considering the vector z returned by the linear program, which, for positive numbers a, b satisfies $\left|\frac{f_\pi(x)}{x} - \sum_{i\geq 0} z(i)\cdot poi(xk, i)\right| \leq a + \frac{b}{x}$, for $x \in [0, \frac{k^{c_1}}{2k}]$, where the objective value of the linear program, v, is guaranteed by the linear program to be at least as large as $a + bn$.

As argued above, attenuating z to form $\tilde{z}(i) := z(i)\cdot(1-\epsilon)^i$ transforms the linear combination of Poissons $g(x) := \sum_{i=0}^{k^{c_1}} poi(xk, i)z(i)$ into $\tilde{g}(x) = g(x\cdot(1-\epsilon))e^{-\epsilon kx}$. Thus $\tilde{g}(x)$ is within $a + \frac{b}{x(1-\epsilon)}$ of $e^{-\epsilon kx}\frac{f_\pi(x(1-\epsilon))}{x(1-\epsilon)}$, where $\frac{f_\pi(x(1-\epsilon))}{x(1-\epsilon)}$ is within $|\log(1-\epsilon)|$ of $\frac{f_\pi(x)}{x}$. By the triangle inequality, $\tilde{g}(x)$ is thus within $a + O(\epsilon) + \frac{b\cdot(1+O(\epsilon))}{x}$ of $e^{-\epsilon kx}\frac{f_\pi(x)}{x}$, provided $x(1-\epsilon) \in [0, \frac{k^{c_1}}{2k}]$. Otherwise, we have $x > \frac{k^{c_1}}{2k}$, implying $e^{-\epsilon kx} \leq e^{-\log k} = \frac{1}{k}$, which is small enough to wipe out any discrepancy that may occur in this region. Specifically: since the Poisson coefficients sum to at most $k^{c_2}v \leq k\log^2 k$, and since any Poisson distribution of parameter λ has each probability bounded by $O(\frac{1}{\sqrt{\lambda}})$, we have that for $x > \frac{k^{c_1}}{2k}$, the linear combination of Poissons $g(x)$ must be at most $O(k^{1-c_1/2}\log^2 k)$, implying $\tilde{g}(x) = O(k^{-c_1/2}\log^2 k)$ in this range. Trivially, $e^{-\epsilon kx}\frac{f_\pi(x)}{x} = O(\frac{\log k}{x})$. Thus for arbitrary positive x we have that $\tilde{g}(x)$ is within $a + O(k^{-c_1/2}\log^2 k) + \frac{b\cdot(1+O(k^{-c_1}\log k))}{x}$ of $e^{-\epsilon kx}\frac{f_\pi(x)}{x}$. Resampling $\tilde{z}$ to $\tilde{z}_\alpha$ is exact, with $\tilde{g}(x) = \sum_{i\geq 0}\tilde{z}_\alpha(i)\cdot poi(\frac{xk}{\alpha}, i)$, so thus these bounds apply to $\sum_{i\geq 0}\tilde{z}_\alpha(i)\cdot poi(\frac{xk}{\alpha}, i)$ as well, as desired.

We thus invoke Proposition 6.10. For the first condition, we have shown that $\sum_{i\geq 0} z_E(i)\cdot poi(\frac{xk}{\alpha}, i)$ approximates $\frac{f_\pi(x)}{x}$ to within $a + O(k^{-c_1/2}\log^2 k) + \frac{b\cdot(1+O(k^{-c_1}\log k))}{x}$, where $a + bn \leq v$. We have shown that the second condition applies for $c = O(k^{\alpha c_2 + (3/2-\alpha)c_1 - 1/2}\log^3 k)$. Thus Proposition 6.10 yields that: the linear estimator z_E estimates the property π to within error $v\cdot(1+o(1)) + O(k^{\alpha c_2+(3/2-\alpha)c_1-1/2}\log^4 k + k^{-c_1/2}\log^2 k)$ using $\frac{k}{\alpha}$-sized samples, with probability of failure $o(\frac{1}{poly(k)})$, provided $v \leq \log^2 k$.

The proof will be complete upon analyzing the unusual but essentially trivial case of $v > \log^2 k$. Note that any distribution of support at most n must have relative earthmover distance from the trivial distribution (support on 1 element) at most $\log n$, and thus property value between $\pm\log n$. Thus if $n < v\cdot k^3$ then the "estimator" that always returns 0 will always have error at most $\log v + 3\log k = v\cdot(1+o(1))$. We consider the case when $n \geq v\cdot k^3$. Let π^+, π^- denote respectively the maximum and minimum value of $\frac{f_\pi(x)}{x}$ for $x \in [\frac{k}{n}, \frac{1}{vk^2}]$, with x^+, x^- denoting respectively the values at which π^+, π^- are attained. For this range of x, the Poisson functions take very limited values: $poi(xk, 0) = e^{-xk} \in [1-\frac{1}{vk}, 1]$, and thus the remaining Poissons sum up to at most $\frac{1}{vk}$. Thus since the coefficients of the vector z are at most $v\cdot k^{c_2} \leq vk$, we may use the triangle inequality to bound the difference between the expected estimates returned in the "+" and "−" case: $\sum_{i\geq 0} z(i)\cdot|poi(x^+k, i) - poi(x^+k, i)| \leq 4$. Letting e^+ be the expected estimate returned in the "+" case, we consider the constraints

corresponding to x^+ and x^- from the linear program: $|\pi^+ - e^+| \leq z^a + \frac{\max\{z^{b+}, z^{b-}\}}{x^+}$ and $|\pi^- - e^+| \leq 4 + z^a + \frac{\max\{z^{b+}, z^{b-}\}}{x^-}$. Since $v = 2z^a + n \cdot (z^{b+} + z^{b-})$, we note that $x \geq \frac{k}{n}$ implies $\frac{z^{b+}+z^{b-}}{x^\pm} \leq \frac{v}{k}$, and we have, letting $\pi^\pm$ denote either π^+ or π^- that $|\pi^\pm - e^+| \leq \frac{v}{2} + \frac{v}{k} + 4$. Thus by the triangle inequality we have $|\pi^+ - \pi^-| \leq v + \frac{2v}{k} + 8$. Consider the relative earthmover cost of taking an arbitrary distribution of support at most n, and making all its probabilities lie in the interval $[\frac{k}{n}, \frac{1}{vk^2}]$. We note that trivially, this is at most $\max\{\log k, \log vk^2\} = \log v + 2\log k$. Thus the interval encompassing *all* possible values π might take has diameter at most $v + \frac{2v}{k} + 8 + 2(\log v + 2\log k)$ and contains 0. Hence the "estimator" that always returns 0, without looking at the sample, will be accurate to within $v \cdot (1 + o(1))$ for $v = \omega(\log k)$, as desired. □

Theorem 6.1 *Let π be a symmetric linear property that is $\delta(k)$-relative earthmover continuous on distributions of support $n(k)$. If for some constant $c > 0$ and parameter $\epsilon(k) = \delta/k^{o(1)}$, any distributions of support n whose π values differ by at least ϵ are distinguishable with probability at least $\frac{1}{2} + c$ given samples of size k, then for each k there exists a linear estimator that estimates π on distributions of support n to within error $(1 + o(1))\epsilon$ using a sample of size $(1 + o(1))k$, and which has probability of failure $o(\frac{1}{poly(k)})$.*

Proof of Theorem 6.1. Without loss of generality, we assume $\delta = 1$, as we may replace π, ϵ, δ by $\frac{\pi}{\delta}, \frac{\epsilon}{\delta}, 1$ respectively, and scaling the property by $\frac{1}{\delta}$ simply scales the estimation error correspondingly. Further, without loss of generality, we assume that the property has value 0 on the trivial distribution of support 1, as the property estimation problem is unaffected by constant shifts.

Let c_1, as a function of k, be such that it converges to 0 as k increases, yet large enough that $k^{-c_1/2}\log^2 k = o(\min\{\epsilon, 1\})$. Let $c_2 = \frac{1}{2} + 6c_1$. Consider k large enough so that $c_1 \leq \frac{1}{25}$. Proposition 6.8 implies that, for these parameters, any solution to the Lower Bound LP with objective value v induces a pair of indistinguishable distributions whose property values differ by at least $v \cdot (1 - o(1)) - O(k^{-c_1}\log k)$, which must thus be smaller than than ϵ, as defined by the theorem. Thus $v \leq \epsilon \cdot (1 + o(1))$.

We then apply Proposition 6.19 to conclude that, for any $\alpha \in (\frac{1}{2}, 1)$ there exists a $\frac{k}{\alpha}$-sample estimator that has $o(\frac{1}{poly(k)})$ probability of failure, and error at most $v \cdot (1 + o(1)) + O(k^{\alpha c_2 + (3/2-\alpha)c_1 - 1/2}\log^4 k + k^{-c_1/2}\log^2 k)$. As already noted, $v \leq \epsilon \cdot (1 + o(1))$, and by assumption, $k^{-c_1/2}\log^2 k = o(\epsilon)$. For the remaining (middle) term, we note that since $c_2 = \frac{1}{2} + 6c_1$ we have $\alpha c_2 + (3/2 - \alpha)c_1 - 1/2 \leq \frac{1}{2}(\alpha - 1) + \frac{13}{2}c_1$. Setting $\alpha = 1 - 15c_1$ yields that this expression is at most $-c_1$, yielding that $k^{\alpha c_2 + (3/2-\alpha)c_1 - 1/2}\log^4 k \leq k^{-c_1}\log^4 k$. By assumption, this is $o(\min\{\epsilon, 1\}^2) = o(\epsilon)$. Thus, the estimator guaranteed by Proposition 6.19 has total error at most $\epsilon \cdot (1 + o(1))$, as desired. Since $\alpha = 1 - o(1)$, the estimator uses a sample of size $k \cdot (1 + o(1))$. □

Chapter 7

Explicit Linear Estimators

In this chapter we describe machinery for constructing and analyzing the performance of explicit linear estimators for symmetric linear properties, such as entropy. The main result of Chapter 6, Theorem 6.1, shows that for such properties, there exist linear estimators whose performance is essentially optimal among the class of all possible estimators; additionally, the proof of Theorem 6.1 is constructive in that it gives an algorithm for generating such near-optimal linear estimators. Unfortunately, the proof does not allow one to extract any bounds on the performance of these estimators—the guarantee is that they are essentially optimal, yet the proof does not reveal what sample size is required to yield an estimator with a desired accuracy.

The tools we develop in this chapter are robust and general, and allow us to both construct and analyze linear estimators for several properties, revealing tight bounds on the sample complexities of the corresponding estimation tasks that were not accessible using the approach of *estimating the unseen* of Chapter 3. The approach of Chapter 3 was to first recover an approximation of the histogram of the distribution from which the sample was drawn, and then simply return the property value of the returned histogram.

The canonical nature of that approach (in that the reconstruction of the histogram does not depend on the specific property in question), has benefits and downsides. The obvious benefit is that the returned histogram gives more information than a single property value. Nevertheless, some properties might be easy to estimate, requiring far smaller sample sizes than would be necessary to accurately estimate the histogram of the true distribution. The machinery of this chapter allows us to create estimators that are tailor-made to a specific property.

The main results of this chapter are summarized below:

Theorem 7.1. *For any $\epsilon > \frac{1}{n^{0.02}}$, the estimator described in Construction 7.4, when given a sample of size $\Omega(\frac{n}{\epsilon \log n})$ (consisting of independent draws) from a distribution of support at most n will compute an estimate of the entropy of the distribution, accurate to within ϵ, with probability of failure $o(1/poly(n))$.*

The performance of this estimator, up to constant factors, matches the lower bounds of Corollary 3.6, both in terms of the dependence on n and the dependence on ϵ. In particular, this resolves the question as to whether the sample complexity increases linearly with $1/\epsilon$, or the much slower quadratic sample complexity shown in Theorem 3.1.

This inverse linear rate of convergence is rather surprising; given a coin with probability of landing heads p, estimating p to within $\pm\epsilon$ requires $O(1/\epsilon^2)$ coin tosses, and thus estimating the entropy of this distribution of support $\{heads, tails\}$ to within $\pm\epsilon$ also requires $O(1/\epsilon^2)$ draws. Indeed, this example shows that the asymptotic rate of convergence of any entropy estimator is $O(1/\sqrt{k})$, where k is the sample size. Our theorem, however, specifies that for the critical range of parameters when $O(1) \leq \epsilon > 1/n^{0.02}$, our estimator converges at the much faster inverse linear rate.

This theorem largely completes the picture of the sample complexity of estimating entropy: given a sample of size k drawn from a distribution of support size at most n, if $k = o(\frac{n}{\log n})$, accurate estimation is impossible. For $k = \theta(\frac{n}{\log n})$, the optimal estimator can achieve constant error, and as k increases, the error decreases very quickly—inverse linearly with the sample size. Then, when $k = \Omega(n^\alpha)$, for some $\alpha \geq 1.03$, the rate of convergence slows, eventually tending towards the asymptotic rate of $\Theta(1/\sqrt{k})$.

Closely following the form of our linear estimator for entropy, we construct a linear estimator for the *distance to uniformity*:

Definition 7.1. *Given a distribution p, the distance to the uniform distribution of support m, $D(p, Unif(m))$, is the ℓ_1 distance between p, and the "closest" uniform distribution of support m. Formally, for all $i \geq 1$, letting $x_i \geq 0$ denote the probability with which the ith most frequently occurring element arises, we have*

$$D\left(p, Unif(m)\right) := \sum_{i=1}^{m} |x_i - \frac{1}{m}| + \sum_{i>m} x_i.$$

The following theorem describes the performance of our estimator for estimating distance to uniformity:

Theorem 7.2. *For any $\epsilon > \frac{1}{4\log m}$, there is an explicit linear estimator that, when given a sample of size $\Omega\left(\frac{1}{\epsilon^2} \cdot \frac{m}{\log m}\right)$ drawn from a distribution of* any *support, will compute the ℓ_1 distance to $Unif(m)$ to within accuracy ϵ, with probability of failure $o(1/poly(m))$.*

This is the first $o(m)$-sized sample estimator for distance to uniformity, and the lower bounds of Theorem 5.1 proved in Chapter 5 imply that for any constant error ϵ, this estimator is optimal, to constant factor. This tight bound of $\Theta(m/\log m)$ on the sample size required to yield constant error contrasts with the tight bound of $\Theta(m^{1/2})$ shown in [19, 57] for the related problem of distinguishing a uniform distribution on m samples from one that has constant distance from such a distribution.

It is worth stressing that the sample complexity expressed in the above theorem is in terms of m, and is independent of the support size of the distribution from which the sample was drawn. This makes intuitive sense, and also explains why such a result would be difficult to obtain via the approach of Chapter 3 which relies crucially on an upper bound on the support size of the distribution from which the sample was drawn.

7.1 Constructing Estimators with "Skinny Bumps"

We consider a symmetric linear property π with characteristic function f_π—that is, the property value of a distribution with histogram h is given by

$$\pi(h) = \sum_{x:h(x)\neq 0} f_\pi(x)h(x).$$

As discussed in Section 6.3 of Chapter 6, the challenge of constructing a good linear estimator for π is the task of accurately approximating $f_\pi(x)$ in the basis of Poisson functions, $poi(kx, i)$.

The key technical tool that we use to construct good linear estimators is the "Chebyshev bump" construction (Definition 3.17) that we defined in Chapter 3 and used as a component of an earthmoving scheme. Here, we use this Chebyshev bump construction to turn the basis of Poisson functions into a more adroit basis of "skinny" bumps, which are, in a very rough sense, like the Poisson functions compressed by a factor of $\log k$ towards the origin. Intuitively, this super-constant factor is what allows us to construct sublinear-sample estimators.

Perhaps the most simplistic attempt to represent the characteristic function f_π as a sum of Poisson functions is to simply set the coefficient of $poi(xk, i)$ equal to $f_\pi(\frac{i}{k})$. This estimator is the "naive" or "plug-in" estimator, and simply returns the property value of the empirical distribution of the sample. For most properties, such as entropy, this estimator should be good for the high-probability region. Note that rather than approximating $f_\pi(x)$ as $\sum_{i=1}^{\infty} poi(kx, i)$, we will instead approximate $\frac{f_\pi(x)}{x}$ by the zero-indexed sum $\sum_{i=0}^{\infty} poi(kx, i)$. These two tasks are formally equivalent, as $x \cdot poi(kx, i) = \frac{i+1}{k} poik(x, i+1)$. The following lemma, which we will use later, characterizes the performance of any "plug-in" estimator. The relatively straight-froward proof of this lemma is deferred to Section 7.3.

Lemma 7.2. *Given a function $f : \mathbb{R} \to \mathbb{R}$ whose fourth derivative at x is bounded in magnitude by $\frac{\alpha}{x^4}$ for $x \geq 1$ and by α for $x \leq 1$, and whose third derivative at x is bounded by $\frac{\alpha}{x^3}$, then for any real x, $\sum_{i=0}^{\infty} f(i) \cdot poi(x, i)$ is within $O(\frac{\alpha}{x^2})$ of $f(x) + \frac{1}{2}xf''(x)$.*

To interpret this lemma, consider the case of entropy estimation; this lemma implies that

$$\log x - \sum_{i=0}^{\infty} \log(i/k) poi(kx, i) = \frac{1}{2kx} + O(\frac{1}{k^2x^2}).$$

In some regimes this error is satisfactorily small (and this estimator is in fact widely used in practice). However, for $x = 1/k$ the error is constant, and for smaller x the error increases. Given this analysis, it is clear why the naive estimator performs poorly on, say, a uniform distribution of support larger than k.

How can one improve this estimator? The obvious correction is to account for the second-derivative term of Lemma 7.2, corresponding to the term $\frac{1}{2kx}$ in the above expression for the bias for the naive estimator for entropy. This yields the "Miller-Madow Corrected Estimator" for entropy. Nevertheless, the error term is still constant for $x = 1/k$, making sublinear-sample estimation impossible. Such error is, in some sense, to be expected: the first few Poisson functions $poi(kx, i)$ have "width" $O(1/k)$.

A "plug-in" estimator in terms of a "skinnier" basis than the Poisson functions would make the estimate correspondingly more accurate. The crux of our estimator is to employ the skinny Chebyshev bumps of Definition 3.17 in place of the fat Poisson functions to get correspondingly better estimators. As we proved in Chapter 3, each of these skinny bumps can be expressed a low-weight linear combinations of Poisson functions, and thus, ultimately, we will still end up with an approximation of the desired characteristic function in the basis of Poisson functions.

For convenience, we restate the definition of the Chebyshev bumps, and the key lemmas proved in Chapter 3 that we will reuse. Recall that the jth Chebyshev polynomial T_j is defined so as to satisfy $T_j(\cos(y)) = \cos(j \cdot y)$.

Definition 3.17. *The* Chebyshev bumps *are defined in terms of k as follows. Let $s = 0.2 \log k$. Define $g_1(y) = \sum_{j=-s}^{s-1} \cos(jy)$. Define*

$$g_2(y) = \frac{1}{16s}\left(g_1(y - \frac{3\pi}{2s}) + 3g_1(y - \frac{\pi}{2s}) + 3g_1(y + \frac{\pi}{2s}) + g_1(y + \frac{3\pi}{2s})\right),$$

and, for $i \in \{1, \ldots, s-1\}$ define $g_3^i(y) := g_2(y - \frac{i\pi}{s}) + g_2(y + \frac{i\pi}{s})$, and $g_3^0 = g_2(y)$, and $g_3^s = g_2(y + \pi)$. Let $t_i(x)$ be the linear combination of Chebyshev polynomials so that $t_i(\cos(y)) = g_3^i(y)$. We thus define $s + 1$ functions, the "skinny bumps", to be $B_i(x) = t_i(1 - \frac{xk}{2s}) \sum_{j=0}^{s-1} poi(xk, j)$, for $i \in \{0, \ldots, s\}$. That is, $B_i(x)$ is related to $g_3^i(y)$ by the coordinate transformation $x = \frac{2s}{k}(1 - \cos(y))$, and scaling by $\sum_{j=0}^{s-1} poi(xk, j)$.

The following lemma showed that each of the Chebyshev bumps defined above can be expressed as a low-weight linear combination of the Poisson functions.

Lemma 3.19. *Each $B_i(x)$ may be expressed as $\sum_{j=0}^{\infty} a_{ij} poi(kx, j)$ for a_{ij} satisfying $\sum_{j=0}^{\infty} |a_{ij}| \leq k^{0.3}$*

The above lemma will allow us to attempt to approximate the characteristic function of the property in question with Poisson functions by directly approximating the characteristic function via these conveniently-skinny bumps. The bound $k^{0.3}$ on the coefficients is crucial, as the coefficients of our estimator must be somewhat less than $\sqrt{k}$ in order for our k-sample estimator to have sub-constant variance. As the coefficients of Chebyshev polynomials grow

exponentially in their degree, this is what limits us to the first $s = O(\log k)$ Chebyshev polynomials. Thus our approximation of the characteristic function via the Chebyshev bumps will only apply to the very low-probability region—but this is acceptable, since above this region, the more crude approximation via the Poisson functions, using Lemma 7.2, will suffice.

The quality of the estimators we construct in this chapter rests on the following lemma, which is the Chebyshev bump analog of Lemma 7.2, and shows that if one constructs the naive "plug-in" approximation using the skinny Chebyshev bumps, instead of the Poisson functions, the approximation is very good. While the proof of this lemma is somewhat laborious, the guiding intuition is simply that the Chebyshev bumps are reasonably symmetric and skinny. The proof of this lemma is deferred to Section 7.3.

Lemma 7.3. *Given $\alpha \leq \beta$ and a twice-differentiable function $f(x) : [0, \frac{s}{2k}] \to \mathbb{R}$ satisfying $|f(x)| \leq \gamma$, $|f'(x)| \leq \frac{\alpha}{x}$, and $|f''(x)| \leq \frac{\beta}{x^2}$, then $f(x)$ can be approximated as $\sum_i w_i B_i(x)$ for weights $w_i = f(c_i)$ for $c_i = \frac{2s}{k}(1 - \cos\frac{i\pi}{s})$, with error of approximation at x bounded in magnitude by*

$$O(\frac{\gamma}{(xks)^{3/2}}) + O(\frac{\beta}{xks}) + O(\frac{\alpha}{(xks)^{3/2}}) + e^{-s/7}.$$

7.2 Linear Estimators for Entropy and Distance to Uniformity

We now define explicit linear estimators for entropy, and "distance to uniformity". The approach will be to use the plug-in estimator with the Poisson functions in the high-probability part of the distribution $x > s/k$, and will use the skinny Chebyshev bumps in the small-probability regime ($x < s/k$). Additionally, we will need a smooth cutoff between these regimes, as the bounds of Lemmas 7.2 and 7.3 depend on the derivates of the function we are trying to approximate in the Poisson basis, and hence an abrupt cutoff would yield a poor bound. As a final step in our proofs of the performance guarantees of our estimators, we will apply Proposition 6.10, which argues that if the characteristic function is sufficiently accurately approximated, and the coefficients of our linear estimator are sufficiently small, then the estimator performs well.

As we are hoping to achieve an inverse linear convergence rate for our entropy estimator, our construction and analysis for entropy will be significantly more delicate than that of our estimator of distance to uniformity.

Entropy

The following construction defines a set of coefficients $\{z_i\}$ such that $\sum_{i \geq 0} z_i \cdot poi(xk, i) \approx \log x$.

Construction 7.4. *As in Definition 3.17, let* $s := (0.2)\log k$. *Define the interpolation function* $I : \mathbb{R} \to \mathbb{R}$ *such that* $I(y) = 0$ *for* $y \leq \frac{s}{4}$, $I(y) = 1$ *for* $y \geq \frac{s}{2}$, *and* $I(y)$ *is continuous, and four-times differentiable, where for* $i \in 1, \ldots, 4$, *the magnitude of the* i*th derivative is at most* c/s^i, *for some fixed constant* c. *(Such a function* I *can trivially be constructed.)*

Consider the function $f(y) := I(y)\left[\frac{1}{2y} + \log y - \log k\right]$, *and provisionally set* $z_i := f(i)$. *Note that by Lemma 7.2 we have accurately represented the logarithm function via the Poisson bumps in the interval* $[\frac{s}{2k}, 1]$.

We will now use the skinny Chebyshev bumps to approximate the function $v(x)$ *defined as*

$$v(x) := \begin{cases} \log x - I(2kx) \cdot \sum_{i=0}^{\infty} poi(xk, i) f(i) & \text{for } x \geq \frac{1}{ks} \\ \log(\frac{1}{ks}) - 1 + xsk & \text{for } x \leq \frac{1}{ks} \end{cases}$$

Thus $v(x)$ *is twice differentiable for* $x > 0$, $v(x) \approx 0$ *for* $x > \frac{s}{2k}$, $v(x) = \log x$ *for* $x \in (1/ks, \frac{s}{8k})$, *and* $v(x)$ *is a linear approximation to* $\log x$ *for* $x < 1/ks$.

Define the coefficient b_i *of the* i*th Chebyshev bump* B_i, *with "center"* $c_i = \frac{2s}{k}\left(1 - cos\left(\frac{i\pi}{s}\right)\right)$, *to be* $v(c_i)$. *To conclude the construction, letting the* i*th Chebyshev bump* B_i *be represented as a sum of Poisson functions, as guaranteed by Lemma 3.19:* $B_i(x) = \sum_j a_{i,j} poi(xk, j)$, *for each* $i \in \{0, \ldots, s\}$, *increment* z_j *by* $\sum_i a_{i,j} v(c_i)$.

Define the linear estimator given by coefficients $\beta_1, \ldots, \beta_k$, *where* $\beta_i := z_{i-1} \cdot \frac{i}{k}$.

Theorem 7.1, which we restate here for convenience, describes the performance of the above estimator. The proof is given in Section 7.3.

Theorem 7.1. *For any* $\epsilon > \frac{1}{n^{0.02}}$, *the estimator described in Construction 7.4, when given a sample of size* $\Omega(\frac{n}{\epsilon \log n})$ *(consisting of independent draws) from a distribution of support at most* n *will compute an estimate of the entropy of the distribution, accurate to within* ϵ, *with probability of failure* $o(1/poly(n))$.

The bound that $\epsilon > \frac{1}{n^{0.02}}$ can be relaxed slightly, though we prove this with the exponent 0.02 for ease of exposition.

Distance to Uniformity

We now describe our linear estimation for distance to uniformity. While distance to uniformity is not a linear property, it can be 2-approximated by a linear property:

Fact 7.5. *The total variational distance between a discrete distribution* $p \in \mathcal{D}^n$ *with histogram* h *and a uniform distribution on* m *elements, denoted by* $D(h, Unif(m))$, *can be approximated to within a factor of 2 as* $\sum_{x:h(x)\neq 0} h(x) f_u(x)$, *for the function*

$$f_u(x) := \begin{cases} x & \text{for } x \leq \frac{1}{2m} \\ |x - \frac{1}{m}| & \text{for } x > \frac{1}{2m}. \end{cases}$$

Proof. From Definition 7.1, to compute the distance between a distribution p and the uniform distribution on m elements, one takes the m elements p_i of h that have the highest probability, and computes the cost of changing each of their probability masses to $\frac{1}{m}$, namely $|p_i - \frac{1}{m}|$, and then adds to this the cost of changing every other mass p_i to 0, namely $|p_i|$. This is lower-bounded and 2-approximated by the cost of sending every element that is below $\frac{1}{2m}$ down to 0, and sending every element above $\frac{1}{2m}$ to $\frac{1}{m}$, as defined in Fact 7.5. □

As for estimating entropy, we will use the Chebyshev bump construction of Definition 3.17 to help approximate the function $\frac{f_u(x)}{x}$ via a sum of Poisson functions, then apply Proposition 6.10.

Construction 7.6. *Let $s = (0.3)\log k$. Define the interpolation function $I : \mathbb{R} \to \mathbb{R}$ such that $I(x) = 0$ for $x \leq \frac{s}{4}$, $I(x) = 1$ for $x \geq \frac{s}{2}$, and $I(x)$ is continuous, and four-times differentiable, where for $i \in 1, \ldots, 4$, the magnitude of the ith derivative is at most c/s^i, for some fixed constant c. Such a function I can be easily constructed.*

Consider the function $g(x) := I(kx)\frac{f_u(x)}{x}$, and provisionally set $z_i := g(\frac{i}{k})$. We will now use the skinny Chebyshev bumps to approximate the function $v(x) = (1 - I(kx))\frac{f_u(x)}{x}$.

Define the coefficient of the ith Chebyshev bump B_i, with "center" $c_i = \frac{2s}{k}\left(1 - cos\left(\frac{i\pi}{s}\right)\right)$, to be $v(c_i)$. To conclude the construction, letting the ith Chebyshev bump B_i be represented as a sum of Poisson functions, as guaranteed by Lemma 3.19: $B_i(x) = \sum_j a_{i,j} poi(xk, j)$, for each $i \in \{0, \ldots, s\}$, increment z_j by $\sum_i a_{i,j} v(c_i)$.

Define the linear estimator given by coefficients $\beta_1, \ldots, \beta_k$, where $\beta_i := z_{i-1} \cdot \frac{i}{k}$.

The following theorem asserts the quality of our estimator:

Theorem 7.2. *For any $\epsilon > \frac{1}{4\log m}$, there is an explicit linear estimator that, when given $\Omega\left(\frac{1}{\epsilon^2} \cdot \frac{m}{\log m}\right)$ independent draws from a distribution of* any *support, will compute the ℓ_1 distance to $Unif(m)$ to within accuracy ϵ, with probability of failure $o(1/poly(m))$.*

The lower-bound construction of Theorem 5.1 of Chapter 5 shows that this is tight for constant ϵ; in particular, for any constant $\epsilon > 0$, there is a constant c such that for sufficiently large m, there exist two distributions A, A' such that $D(A, Unif(m)) < \epsilon$, $D(A', Unif(m)) > 0.49$, but distributions A, A' are indistinguishable (with any constant probability greater than $1/2$) given $c\frac{m}{\log m}$-sized samples.

In contrast to the estimator for entropy, the estimator for distance to uniformity does not need any assumption on the support size of the distribution being sampled. Additionally, the convergence rate is as the inverse of the square root of the sample size, as opposed to the much faster inverse linear relationship of the estimator for entropy. Intuitively, this is because the function $f_u(y)$ has a kink at probability $y = 1/m$, as opposed to the smooth logarithm function.

The proof of Theorem 7.2 is considerably easier than for our estimator of entropy:

Proof of Theorem 7.2. Consider setting $m = \epsilon^2 k \log k$, for some $\epsilon > \frac{4}{\log m}$, and thus the portion of $f_u(x)/x$ approximated exclusively by the Poisson bumps (i.e. $x > \frac{s}{2k}$) corresponds

to $x > \frac{1}{m}$, and in this range $f_u(x)/x = 1 - \frac{1}{xm}$. In particular, the function $\frac{f_u(x/k)}{x/k}$ has jth derivative bounded in magnitude by $O(\frac{k}{mx^{j+1}})$, for constant j, and thus satisfies the conditions of Lemma 7.2 with $\alpha = O(\frac{k}{ms})$, and thus the approximation in this regime is accurate to $O(\frac{k}{msx^2}) + O(\frac{x}{2}\frac{k}{mx^3}) = O(\frac{k}{mx^2})$, which is maximized by minimizing x, in which case the error is $O(\frac{k}{ms^2}) = O(\frac{1}{\epsilon^2 \log^3 k})$, which is at most $O(\epsilon)$, as in the case that $\epsilon = 1/\log k$.

We now consider the error in approximation from the skinny bumps (i.e. for $x < \frac{s}{2k}$). In this regime, the function $f_u(x)/x$ is $O(\frac{1}{mx^2})$-Lipschitz for $x > 1/2m$. By Lemma 3.21 (arguing that the functions g_3^i decay super quadratically), Lemma 7.11, and the change of coordinates, the width of the Chebyshev bumps centered at x are $O(\frac{\sqrt{xk \log k}}{k \log k})$; thus the error of approximation is the product of this width and the Lipschitz constant, yielding $O(\frac{1}{\epsilon^2 (xk \log k)^{3/2}})$. This is maximized by minimizing x, and thus taking $x = O(1/m)$ yields error $O(\epsilon)$, as desired. Since $f_u(x)/x = 1$ is constant for $x < 1/2m$, the error in this small regime is $o(\epsilon)$. Thus the error of approximating the function $f_u(x)/x$ is $O(\epsilon)$. To conclude, since the coefficients of the approximation are sufficiently small (at most $k^{0.3}$, by Lemma 3.19), we may now apply Proposition 6.10 to yield the claim. □

7.3 Missing Proofs

Proof of Lemma 7.2. Consider the Taylor expansion of f to third order around x, $f(i) \approx a + b \cdot (i - x) + c \cdot (i - x)^2 + d \cdot (i - x)^3 + e(i)$, for $a = f(x), b = f'(x), c = \frac{1}{2}f''(x)$, and $d = \frac{1}{6}f'''(x)$, where the error, e, is a function which we will analyze later. By assumption, $d \leq \frac{\alpha}{6x^3}$. We bound $\sum_{i=\beta}^{\infty} f(i) \cdot poi(x, i)$ by thus decomposing $f(i)$. We note that we may take the lower limit of the sum to be 0, since $f(i)$ equals zero for $i < \beta$. We evaluate the first four terms by noting, respectively, that the Poisson distribution of parameter x has total probability mass 1, has mean x, has variance x, and has third moment about its mean x, leading to $\sum_{i=0}^{\infty} a \cdot poi(x, i) = a = f(x)$, $\sum_{i=0}^{\infty} b(i-x) \cdot poi(x, i) = 0$, $\sum_{i=0}^{\infty} c(i-x)^2 \cdot poi(x, i) = cx = \frac{1}{2}xf''(x)$, and $\sum_{i=0}^{\infty} d(i - x)^3 \cdot poi(x, i) = dx \leq \frac{\alpha}{6x^2}$.

We now analyze the error function $e(i)$. By construction, it and its first three derivatives are 0 at $i = x$, while its fourth derivative is everywhere equal to the fourth derivative of f, which by assumption is bounded by $\frac{\alpha}{i^4}$. Thus for $i \geq x$, the fourth derivative of $e(i)$ is bounded by $\frac{\alpha}{x^4}$ implying a bound of $|e(i)| \leq \frac{\alpha}{24x^4}(i - x)^4$ for $i \geq x$. Similarly, for $i \in [\frac{x}{2}, x]$ we have that the fourth derivative of f is bounded by $\frac{16\alpha}{x^4}$, yielding a bound of $|e(i)| \leq \frac{2\alpha}{3x^4}(i-x)^4$ for $i \in [\frac{x}{2}, x]$. For general $i < x$, we bound e by repeated integration. Since $|e''''(i)| \leq \frac{\alpha}{i^4}$ and $e'''(x) = 0$ we may integrate from i to x to yield $|e'''(i)| \leq \frac{1}{4}\alpha(\frac{1}{i^3} - \frac{1}{x^3})$, which we crudely bound by $\frac{1}{4}\frac{\alpha}{i^3}$. We repeat this process, since $e''(x) = e'(x) = 0$, to yield, successively, $|e''(i)| \leq \frac{1}{12}\frac{\alpha}{i^2}$, and $|e'(i)| \leq \frac{1}{24}\frac{\alpha}{i}$. We integrate once more, though without discarding the constant term, to yield $|e(i)| \leq \frac{1}{24}\alpha(\log x - \log i)$, again, valid for $i \leq x$. Instead of using this bound directly,

we sum from 1 to x:

$$\sum_{i=1}^{x} |e(i)| \leq \frac{1}{24}\alpha \sum_{i=1}^{x} (\log x - \log i) \leq \frac{\alpha}{24} \int_0^x |\log x - \log i| \, di = \frac{\alpha}{24} x.$$

We now bound $e(0)$. If $x < 1$ then, directly, since $e'''' \leq \alpha$, we have $|e(0)| \leq \frac{\alpha}{24}x^4 \leq \frac{\alpha}{24}x$. Otherwise if $x \geq 1$, note from above that $|e(1)| \leq \frac{\alpha \log x}{24}$, $|e'(1)| \leq \frac{\alpha}{24}$, $|e''(1)| \leq \frac{\alpha}{12}$, $|e'''(1)| \leq \frac{\alpha}{4}$, and for all $i \in [0,1]$, $e''''(i) \leq \alpha$. This immediately yields a bound that $|e(0)| \leq \alpha \left[\frac{\log x}{24} + \frac{1}{24} + \frac{1}{24} + \frac{1}{24}\right]$. Since $3 + \log x \leq 2 + x \leq 3x$ for $x \geq 1$, we have that $\sum_{i=0}^{x} |e(i)| \leq \frac{\alpha}{6}x$.

Trivially, we use this bound to bound the sum over half the domain: $\sum_{i=0}^{x/2} |e(i)| \leq \frac{\alpha}{6}x$. In sum, we will use the bound $|e(i)| \leq \frac{2\alpha}{3x^4}(i-x)^4$ for $i \geq \frac{x}{2}$, and $\sum_{i=0}^{x/2} |e(i)| \leq \frac{\alpha}{6}x$ otherwise.

To complete the proof, we note the basic fact that the Poisson distribution dies off super-polynomially fast away from its mean, relative to its standard deviation. That is, for any positive integer—we choose 6 here—there is a constant γ such that for all i, x, we have $poi(x,i) \leq \frac{\gamma}{\sqrt{x}} \left|\frac{i-x}{\sqrt{x}}\right|^{-6}$.

We thus bound $\sum_{i=0}^{\infty} e(i)poi(x,i)$ piecewise. For $i \in [x - \sqrt{x}, x + \sqrt{x}]$, we have that since $poi(x,i)$ is a distribution over i, it sums to at most 1 here; since we have the bound here that $|e(i)| \leq \frac{2\alpha}{3x^4}(i-x)^4$, we note that when $|i - x| \leq \sqrt{x}$ we have $|e(i)| \leq \frac{2\alpha}{3x^2}$, which is thus also a bound on $\left|\sum_{i=x-\sqrt{x}}^{x+\sqrt{x}} e(i)poi(x,i)\right|$. For $i > x + \sqrt{x}$ we use the bound $poi(x,i) \leq \frac{\gamma}{\sqrt{x}} \left|\frac{i-x}{\sqrt{x}}\right|^{-6}$ to see that

$$\left|\sum_{i>x+\sqrt{x}} e(i)poi(x,i)\right| \leq \sum_{i>x+\sqrt{x}} \frac{2\alpha}{3x^4}(i-x)^4 \cdot \frac{\gamma}{\sqrt{x}} \left|\frac{i-x}{\sqrt{x}}\right|^{-6} = \frac{2\alpha\gamma}{3x^{3/2}} \sum_{i>x+\sqrt{x}} \frac{1}{(i-x)^2} = O(\frac{\alpha}{x^2}).$$

The same argument yields the same bound for the sum over $i \in [\frac{x}{2}, x - \sqrt{x}]$. To bound the remaining region, when $i \leq \frac{x}{2}$, we note that for this region $poi(x,i) \leq \frac{64\gamma}{x^{7/2}}$, and since, as noted, $\sum_{i=0}^{x/2} |e(i)| \leq \frac{\alpha}{6}x$ we have that $\sum_{i=1}^{x/2} |e(i)|poi(x,i) = o(\frac{\alpha}{x^2})$. Combining all the bounds yields that $|\sum_{i=1}^{\infty} e(i)poi(x,i)| = O(\frac{\alpha}{x^2})$, and combining this with the bounds from the power series expansion of f yields $\sum_{i=0}^{\infty} f(i)poi(x,i)$ equals $f(x) + \frac{1}{2}xf''(x)$ to within $O(\frac{\alpha}{x^2})$, as desired. □

Proof of Lemma 7.3

Before tackling the proof of Lemma 7.3, we prove several helpful lemmas that establish properties of the Chebyshev bump basis.

Lemma 7.7. *For $y \in [-\pi/s, \pi/s]$, sufficiently large s, and positive integers $a, b \leq s$,*

$$|\sum_{i=-a}^{b} (y + \pi i/s) \cdot g_2(y + \pi i/s)| \leq \frac{12}{s}\left(\frac{1}{a^2} + \frac{1}{b^2}\right).$$

Proof. We will first show that

$$|\sum_{i=-s}^{s-1}\left(\sin(y+\pi i/s)+\frac{\sin^3(y+\pi i/s)}{6}\right)\cdot g_2(y+\pi i/s)|=0,$$

and then will use the fact that $\sin(x)+\sin^3(x)/6\approx x$ near zero, and that $g_2(x)$ decays quickly away from zero to yield the claim. To begin, note that $g_2(x)$ is an even function, and can be written as a weighted sum of $\cos(jx)$, for integers j at most $s-1$. Since $\cos(jx)\sin(x)=\frac{1}{2}\sin((j+1)x)-\frac{1}{2}\sin((j-1)x)$, and $\sum_{i=-s}^{s-1}\sin(j(x+\frac{i\pi}{s}))=0$, for any integer $j\leq 2s-1$, we have

$$\sum_{i=-s}^{s-1}\sin(y+\pi i/s)\cdot g_2(y+\pi i/s)=0.$$

Additionally, $\sin^3(x)=\frac{3\sin(x)-\sin(3x)}{4}$, and by the above, $\cos(jx)\sin(3x)=\frac{1}{2}\sin((j+3)x)-\frac{1}{2}\sin((j-3)x)$, and thus for $s>3$, by the above,

$$\sum_{i=-s}^{s-1}\sin^3(y+\pi i/s)\cdot g_2(y+\pi i/s)=0.$$

Next, note that $|x-\sin(x)-\sin^3(x)/6|\leq 3x^5/40$, and thus from the above,

$$|\sum_{i=-s}^{s-1}(y+\pi i/s)\cdot g_2(y+\pi i/s)|\leq\sum_{i=-s}^{s-1}|g_2(y+\pi i/s)|\cdot|3(y+\pi i/s)^5/40|.$$

We now leverage the bounds on $|g_2(y)|$ from Lemma 3.21. For the at most 5 terms in the above sum for which $y+\pi i/s\in(-3\pi/s,3\pi/s)$, since $g_2(y)\leq 1/2$, we get a contribution of at most $\frac{5}{2}\frac{3^5\pi^5}{40s^5}\leq\frac{4700}{s^5}$. For the remaining terms, we have $|g_2(x+\pi i/s)|\leq\frac{285}{(x+\pi i/s)^4s^4}$, and thus the contribution of the remaining terms, since $|y|<\pi/s$, is at most $2\sum_{i=2}^{s}\frac{855(\pi i/s)}{40s^4}\leq\frac{43\log s}{s^5}$. Thus for sufficiently large s,

$$|\sum_{i=-s}^{s-1}(y+\pi i/s)\cdot g_2(y+\pi i/s)|\leq\frac{1}{s^4}.$$

To conclude, the claim clearly holds for $a=1$ or 2, and for $a\geq 3$ we have

$$\begin{aligned}\sum_{i=a}^{s}|(y+\pi i/s)\cdot g_2(y+\pi i/s)| &\leq \sum_{i=a}^{s}(y+\pi i/s)\frac{285}{(y+\pi i/s)^4s^4}\\ &\leq \frac{285}{\pi^3 s}\sum_{i=a-1}^{s}\frac{1}{i^3}<\frac{23}{2a^2s}.\end{aligned}$$

□

Lemma 7.8. *For $y \in [-\pi/s, \pi/s]$, sufficiently large s, and positive integer $a \leq s$,*

$$|\sum_{i=0}^{a}(y+\pi i/s)^2 \cdot g_2(y+\pi i/s)| = O(\frac{1}{s^2}).$$

Proof. From our bounds on g_2 given in Lemma 3.21, have the following:

$$\begin{aligned}|\sum_{i=0}^{a}(y+\pi i/s)^2 \cdot g_2(y+\pi i/s)| &\leq 3 \cdot \frac{1}{2} \cdot \frac{3^2\pi^2}{s^2} + \sum_{i=3}^{a}(y+i\pi/s)^2 \frac{285}{(y+i\pi/s)^4 s^4} \\ &\leq O(\frac{1}{s^2}) + \frac{285}{\pi^2 s^2}\sum_{i=3}^{\infty}\frac{1}{(i-1)^2} \leq O(\frac{1}{s^2}).\end{aligned}$$

□

Lemma 7.9. *For $y \in [-\pi/s, \pi/s]$, sufficiently large s, and positive integers a, b, c, d such that $c \leq a \leq s$ and $d \leq b \leq s$, and a twice-differentiable function $f : [-\frac{a\pi}{s}, \frac{b\pi}{s}] \to \mathbf{R}$ satisfying $|f'(0)| \leq \alpha$, $\max_{y\in[-c\pi/s, d\pi/s]} |f''(y)| \leq \beta$, and $\max_y |f(y)| \leq \gamma$,*

$$|\sum_{i=-a}^{b} g_2(y+\frac{i\pi}{s})f(y+\frac{i\pi}{s}) - f(0)| \leq O\left(\gamma\left(\frac{1}{c^3}+\frac{1}{d^3}\right) + \frac{\alpha}{s}\left(\frac{1}{c^2}+\frac{1}{d^2}\right) + \frac{\beta}{s^2}\right).$$

Proof. We first bound the contribution of the terms with $i \in -a, \ldots, -c, d, \ldots, b$. Using the bounds on $|g_2|$ from Lemma 3.21, we have

$$|\sum_{i\in -a,\ldots,-c,d,\ldots,b} g_2(x+\frac{i\pi}{s})f(x+\frac{i\pi}{s})| \leq \gamma\frac{285}{\pi^4}\left(\sum_{i=c}^{\infty}\frac{1}{(i-1)^4} + \sum_{i=d}^{\infty}\frac{1}{(i-1)^4}\right) \leq O(\gamma(1/c^3+1/d^3)).$$

We now consider $\sum_{i=-c}^{d} g_2(y+\frac{i\pi}{s})f(y+\frac{i\pi}{s})$. We express each $f(y+\frac{i\pi}{s})$ in terms of the first order Taylor expansion about 0, and note that $|f(y+\frac{i\pi}{s}) - (f(0)+(y+\frac{i\pi}{s})f'(0))| \leq (y+\frac{i\pi}{s})^2\beta$. Thus we have the following:

$$\begin{aligned}|\sum_{i=-c}^{d} g_2(y+\frac{i\pi}{s})f(y+\frac{i\pi}{s}) - \sum_{i=-c}^{d} g_2(y+\frac{i\pi}{s})\left(f(0)+(y+\frac{i\pi}{s})f'(0)\right)| & \\ \leq \beta\sum_{i=-c}^{d} g_2(y+\frac{i\pi}{s})(y+\frac{i\pi}{s})^2 & \\ = O(\beta/s^2) \qquad \text{from Lemma 7.8.} &\end{aligned}$$

We now turn to analyzing the term involving the Taylor approximation:

$$\sum_{i=-c}^{d} g_2(y+\frac{i\pi}{s})\left(f(0)+(y+\frac{i\pi}{s})f'(0)\right) = \sum_{i=-c}^{d} g_2(y+\frac{i\pi}{s})f(0) + f'(0)\sum_{i=-c}^{d} g_2(y+\frac{i\pi}{s})(y+\frac{i\pi}{s}).$$

To analyze the first term above, by Lemma 3.20, $\sum_{i=-s}^{s} g_s(y+\frac{i\pi}{s})f(0) = f(0)$. Additionally, by Lemma 3.21, $\sum_{d+1}^{s-1} g_s(y+\frac{i\pi}{s}) \leq \sum_{d}^{\infty} \frac{285}{\pi^4 i^4} \leq \frac{2}{d^3}$, and analogously, $\sum_{-s}^{-c-1} g_s(x+\frac{i\pi}{s}) \leq \frac{2}{c^3}$. Thus

$$|f(0) - \sum_{i=-c}^{d} g_2(y+\frac{i\pi}{s})f(0)| \leq 2f(0)(\frac{1}{c^3}+\frac{1}{d^3}).$$

To analyze the second term, by Lemma 7.7,

$$|\sum_{i=-c}^{d}(y+\frac{i\pi}{s})f'(0)g_2(y+\frac{i\pi}{s})| \leq O\left(\frac{1}{s}f'(0)\left(\frac{1}{c^2}+\frac{1}{d^2}\right)\right).$$

The desired statement now follows from adding up the above bounds. □

Lemma 7.10. *For $y \in [0,\pi/2]$, sufficiently large s, and twice-differentiable function f satisfying $|f(y)| \leq \gamma$, $|f'(y)| \leq \frac{\alpha}{y}$ and $|f''(y)| \leq \frac{\beta}{y^2}$,*

$$|f(y) - \sum_{i=0}^{s} g_3^i(y) f(\frac{i\pi}{s})| \leq O\left(\frac{\gamma}{y^3 s^3} + \frac{\beta}{y^2 s^2} + \frac{\alpha}{y^3 s^3}\right).$$

Proof. From Lemma 3.21, we have $g_3^0(y)f(0) + g_3^s(y)f(\pi) \leq O(\frac{\gamma}{y^4 s^4})$.

Next, define $i_y := \lfloor \frac{ys}{\pi} \rfloor$, and let $\delta_y := y - \frac{i_y \pi}{s}$. Thus $\delta_y \in [0, \pi/s]$. For any $j \in \{-i_y+1, \ldots, s-i_y-1\}$, we have

$$\begin{aligned} g_3^{i_y+j}(y) &= g_2(y - \frac{(i_y+j)\pi}{s}) + g_2(y + \frac{(i_y+j)\pi}{s}) \\ &= g_2(\delta_y - \frac{j\pi}{s}) + g_2(\delta_y + \frac{(2i_y+j)\pi}{s}). \end{aligned}$$

Defining the function $r_y(w) = f(y-w)$, we have the following:

$$\begin{aligned} \sum_{i=1}^{s-1} g_3^i(y) f(\frac{i\pi}{s}) &= \sum_{i=1}^{s-1}\left(g_2(y-\frac{i\pi}{s}) + g_2(y+\frac{i\pi}{s})\right) r_y\left(-(\frac{i\pi}{s} - y)\right) \\ &= \sum_{i=1}^{s-1}\left(g_2(\delta_y + \frac{(i_y-i)\pi}{s}) + g_2(y+\frac{i\pi}{s})\right) r_y\left(\delta_y + \frac{(i_y-i)\pi}{s}\right) \\ &= \sum_{j=-s+i_y+1}^{i_y-1} g_2(\delta_y + \frac{j\pi}{s}) r_y(\delta_y + \frac{j\pi}{s}) + \sum_{i=1}^{s-1} g_2(y+\frac{i\pi}{s}) r_y\left(\delta_y + \frac{(i_y-i)\pi}{s}\right). \end{aligned}$$

The idea now is that Lemma 7.9 guarantees that the first term above is roughly $r_y(0) = f(y)$, and it is easy to show that the second term above will be very small. We start by bounding the magnitude of the second term, using the bound on g_2 given in Lemma 3.21:

$$\sum_{i=1}^{s-1} g_2(y+\frac{i\pi}{s}) r_y\left(\delta + \frac{(i_y-i)\pi}{s}\right) \leq \gamma / i_y^3.$$

We now consider the first term above, and apply Lemma 7.9 with $a = c = |-s+i_y+1|$, $d = \lfloor i_y/2 \rfloor$, and $b = i_y - 1$ to yield:

$$|f(y) - \sum_{j=-s+i_y+1}^{i_y-1} g_2(\delta_y + \frac{j\pi}{s}) r_y(\delta_y + \frac{j\pi}{s})| \leq O\left(\tfrac{\gamma}{i_y^3} + \tfrac{1}{s^2}\tfrac{\beta}{y^2} + \tfrac{1}{si_y^2}\tfrac{\alpha}{y}\right),$$

from which the desired claim follows. □

Lemma 7.11. *For $x \leq \frac{s}{2k}$, $1 - \sum_{i=0}^{s-1} poi(xk, i) \leq e^{-s/6}$.*

Proof. This discrepancy is maximized at $x = \frac{s}{2k}$, and by tail bounds of Poissons and Fact A.19, this is at most

$$\sum_{i=s}^{\infty} poi(s/2, i) \leq 2poi(s/2, s) \leq e^{-s/6}.$$

□

Proof of Lemma 7.3. Recall from Definition 3.17 that $B_i(x)$ is related to $g_3^i(y)$ by the coordinate transformation $x = \frac{2s}{k}(1 - \cos(y))$, and scaling by $\sum_{j=0}^{s-1} poi(xk, j)$. By Lemma 7.11 we can ignore the scaling factor for $x \leq \frac{s}{2k}$ and lose only $s \cdot (1 + \log ks) \cdot e^{-s/6} < e^{-s/7}$ in approximation, since there are s skinny bumps, and in Construction 7.4 each skinny bump has a coefficient of magnitude at most $\max_x |v(x)| = 1 + \log ks + e^{-s/7}$. To represent $f(x)$ as a linear combination of $B_i(x)$'s, we will represent $r(y)$ as a linear combination of $g_3^i(y)$'s, where r is chosen so that $r(y) = f(\frac{2s}{k}(1 - \cos(y)))$. Note that

$$\begin{aligned}
|r'(y)| &\leq |f'(\frac{2s}{k}(1-\cos(y)))\frac{2s}{k}\sin(y)| \\
&\leq \frac{\alpha k}{2s(y^2/3)}\frac{2s}{k}y \qquad \text{since for } y \in [0, \pi/2], 1-\cos(y) \geq y^2/3, \text{ and } \sin(y) \leq y. \\
&= \frac{3\alpha}{y}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
|r''(y)| &\leq |f''(\frac{2s}{k}(1-\cos(y)))(\frac{2s}{k}\sin(y))^2 + f'(\frac{2s}{k}(1-\cos(y)))(\frac{2s}{k}\cos(y))| \\
&\leq \frac{9\beta}{y^2} + \alpha \leq \frac{30\beta}{y^2}.
\end{aligned}$$

Thus by Lemma 7.10, we can approximate $r(y)$ as a linear combination of $g_3^i(y)$ to within error $O(\frac{\gamma}{y^3 s^3}) + O(\frac{\beta}{y^2 s^2}) + O(\frac{\alpha}{y^3 s^3}) + e^{-s/7}$. For $y \in [0, \pi/2]$, note that $(1-\cos(y)) \in [y^2/3, y^2/2]$

and thus the error in the corresponding approximation of $f(x)$ via the linear combination of $B_i(x)$'s will have error at most

$$O(\frac{\gamma}{(\sqrt{\frac{3xk}{2s}})^3 s^3}) + O(\frac{\beta}{(\sqrt{\frac{3xk}{2s}})^2 s^2}) + O(\frac{\alpha}{(\sqrt{\frac{3xk}{2s}})^3 s^3}) + e^{-s/7},$$

$$= \quad O(\frac{\gamma}{(xks)^{3/2}} + O(\frac{\beta}{xks}) + O(\frac{\alpha}{(xks)^{3/2}}) + e^{-s/7},$$

as desired.

We now turn to bounding the approximation of $f(x)$ for small $x \leq 1/ks$, which thus equates to bounding the approximation of $r(y)$ via the $g_3^i(y)$'s for $y < 2/s$. The desired lemma now follows from noting that the approximation of $r(y)$ for such values of y is a convex combination of $r(i\pi/s)$ for $i \in 0, 1, 2, \ldots$, where the weight on $r(0)$ is trivially seen to be at least .1, and the contribution to the approximation from g_3^j for $j \geq 100$ is bounded by $\sum_{j \geq 101} g_3^j(y) r(i\pi/s) \leq .1$, from Lemma 3.21 and the assumption that $|f(x)| \leq 1/x^{.5}$. □

Proof of Theorem 7.1

Proof of Theorem 7.1. Consider the function $f(x) := I(x)\left[\log x - \log k + \frac{1}{2x}\right]$, and note that it satisfies the conditions of Lemma 7.2, with $\alpha = O(1)$, and thus

$$|\sum_{i=0}^{\infty} f(i) \cdot poi(x, i) - \left(f(x) + \frac{1}{2} x f''(x)\right)| \leq O(1/x^2).$$

For $x > s/2$, we have $I(x) = 1$ and thus for such x

$$f(x) + \frac{1}{2} x f''(x) = \log x - \log k + O(\frac{1}{x^2}).$$

Thus via the change of variables $y = \frac{x}{k}$, we have that for $y \in [\frac{s}{2k}, \infty]$,

$$|\log y - \sum_{i=0}^{\infty} poi(yk, i) f(i)| \leq O(\frac{1}{k^2 y^2}).$$

Thus we have accurately represented the logarithm function via the Poisson bumps in the interval $[\frac{s}{2k}, 1]$.

We now consider the Chebyshev-bump approximation of the function $v(y)$ defined in Construction 7.4 as

$$v(y) := \begin{cases} \log y - I(2ky) \cdot \sum_{i=0}^{\infty} poi(yk, i) f(i) & \text{for } y \geq \frac{1}{ks} \\ \log(\frac{1}{ks}) - 1 + ysk & \text{for } y \leq \frac{1}{ks} \end{cases}$$

Note that $v(y)$ satisfies the conditions of Lemma 7.10 with $\gamma < \log(sk) + 2$ and $\alpha, \beta = O(1)$. Thus $v(y)$ can be accurately represented by $\sum_i B_i(y)v(c_i)$, yielding that for sufficiently large k,

$$|\sum_{i=1}^{s} B_i(y)v(c_i) + \sum_{i=1}^{\infty} poi(yk,i)f(i) - \log(y)| \leq \begin{cases} O(\frac{1}{yks}) + e^{-s/7} & \text{for } y \in (\frac{1}{ks}, \frac{s}{2k}) \\ O(\frac{1}{k^2y^2}) & \text{for } y \geq \frac{s}{2k}. \end{cases}$$

Finally, we have the trivial crude bound for the extremely small region:

$$|\sum_{i=1}^{s} B_i(y)v(c_i) + \sum_{i=1}^{\infty} poi(yk,i)f(i) - \log(y)| \leq \log(y) + \log(ks) + O(1) \quad \text{for } y \leq \frac{1}{ks}.$$

We will now apply Proposition 6.10 with $a = O(\epsilon), b = O(\epsilon/n)$, and $c = k^{-0.1}$. Note that by Lemma 3.19, the coefficients are sufficiently small and vary sufficiently slowly, satisfying the second condition of Proposition 6.10. For the first condition of Proposition 6.10, it suffices to show that $err(y) \leq \epsilon$ for $y \geq \frac{1}{n}$, and $err(y) \leq \frac{\epsilon}{yn}$ for $y \leq \frac{1}{n}$. To show this, consider setting $n = \epsilon ks$. For $y \leq 1/ks$, since $y < 1/n$, we have

$$\begin{aligned} (ny)\left(\log(y) + \log(ks) + O(1)\right) &\leq (\epsilon ksy)\left(\log(ksy) + O(1)\right) \\ &\leq \epsilon\left(\log(ksy)ksy + O(1)ksy\right) \\ &\leq \epsilon O(1), \end{aligned}$$

and thus the error in this region is good enough to yield an $O(\epsilon)$ estimator. For $y \in (\frac{1}{ks}, \frac{s}{2k})$, $e^{-s/7} = O(k^{-\frac{0.2}{7}}) = o(\epsilon)$, and for $y > 1/n$, we have error of approximation of the logarithm function at most $O(n/ks) = O(\epsilon)$, and if $y < 1/n = 1/\epsilon ks$, we have $ny \cdot O(1/yks) = O(\epsilon)$, which is sufficient to yield an $O(\epsilon)$ estimator. Finally, in the region $y \geq \frac{s}{2k}$, if $y > 1/n$, which implies that $\epsilon > 1/yks$, we have error $O(1/k^2y^2) = O(1/yks) \cdot \frac{s}{yk}$. Because of our bound on y, $s/yk \leq 2$, and thus this error is $O(1/yks) = O(\epsilon)$. In the case that $y \leq 1/n$, we have $ny \cdot O(1/k^2y^2) \leq \epsilon ksO(1/k^2y) = O(\epsilon s/ky) \leq O(\epsilon)$, again because of our bound on y. Thus the above approximation scheme of the logarithm function is sufficiently accurate to yield $O(\epsilon)$-error estimators of entropy for distributions of support at most $O(\epsilon k \log k)$. □

Chapter 8

Estimating Properties in Practice

In Chapters 3, 6, and 7, we described three approaches to estimating symmetric properties that yield sublinear sample estimators achieving near-optimal performance on worst-case instances. In Chapters 4 and 5, we developed techniques for proving information theoretic lower bounds on the sample size required to obtain accurate estimates of symmetric properties; in the process, we developed insights into the mapping between distributions, and the distribution of fingerprints obtained from a sample consisting of independent draws from the distribution. Much of the analysis in the previous four chapters is asymptotic, and some of the constants involved are daunting: a direct implementation of the estimators to which our theorems apply would be impractical, and would likely yield disappointing results on all but truly enormous datasets. Nevertheless, in this chapter we propose a heuristic adaptation of the "unseen" approach of Chapter 3, and demonstrate via simulations that it performs exceptionally well for a variety of estimation tasks (estimating entropy, the number of distinct elements, and ℓ_1 distance), on a variety of natural distributions, for a wide range of parameters. We compare this estimator with previously proposed estimators from the literature: for all settings considered, our estimator performs at least as well as the best previously proposed estimator that we consider, and significantly outperforms all these estimators in some settings.

While our experiments do not form a comprehensive evaluation of our estimators, they provide strong evidence that this theoretically principled approach yields robust and general estimators that seem to perform very well in practice. We expect (and hope) that the estimator described in this section may be fruitfully used in practice, both directly, and as a component within larger machine learning and data analysis systems.

In Section 8.1 we describe the practical adaption of the estimator of Chapter 3. In Section 8.2 we describe our experimental setup for evaluating the quality of this approach for estimating entropy; our experimental setup is based on that used in [137] to compare various entropy estimators. In Section 8.3 we apply this approach to estimate the total variational distance (ℓ_1 distance) between pairs of distribution. Finally, we demonstrate the versatility of our estimator, and show that it can be adapted to estimate the total number of distinct words that appear in Shakespeare's *Hamlet* based on the word counts of surprisingly

short passages.

8.1 A Practical Algorithm for Estimating the "Unseen"

In Chapters 3, 6, and 7, we described three approaches to estimating symmetric properties. While these three approaches all yield provably constant-factor optimal estimators, from a practical viewpoint, they are not equivalent. The estimators of Chapters 6 and 7 are, in a rigorous sense, tailored specifically towards optimizing the performance on worst-case instances—the estimators of Chapter 6 are described as the dual to a linear program that explicitly searches for optimal worst-case (lower bound) instances, and the tradeoff between the magnitude of the coefficients and bias of the estimators of Chapter 7 was chosen to optimize the performance on worst-case instances. In contrast, the estimators of Chapter 3 achieve the same worst-case performance, yet are not tailored to worst-case instances in any sense; thus one might suspect that their performance on "typical" instances may be superior. For the remainder of this section on experimental performance, we will focus on these estimators.

Given the fingerprint $\mathcal{F}$ of a sample of size k, consisting of independent draws from a distribution with histogram h, the high-level approach of the estimator of Chapter 3 is to find a histogram $\hat{h}$ that has the property that if one were to draw a sample of size k from a distribution with histogram h', the fingerprint of the resulting sample would be similar to the observed fingerprint $\mathcal{F}$. The hope is then that h and $\hat{h}$ will be similar, and, in particular, have similar entropies, support sizes, etc.

For general fingerprints, how does one obtain a "plausible" histogram from a fingerprint in a principled fashion? The approach of Chapter 3 is based on the observation that, given a distribution p, and some domain element α occurring with probability $x = p(\alpha)$, the probability that it will be drawn exactly i times in a sample of size k drawn from p is $Pr[Binomial(k, x) = i] \approx poi(kx, i)$. By linearity of expectation, the expected ith fingerprint entry will roughly satisfy

$$E[\mathcal{F}_i] \approx \sum_{x:h_p(x)\neq 0} h(x)poi(kx, i). \tag{8.1}$$

This mapping between histograms and expected fingerprints is linear in the histogram, with coefficients given by the Poisson probabilities. Additionally, $Var[\mathcal{F}_i] \leq E[\mathcal{F}_i]$, and thus the fingerprint is tightly concentrated about its expected value. This motivated the "first moment" approach of Chapter 3. The new central limit theorem for generalized multinomial distributions of Chapter 4 (Theorem 4.2) and Lemma 5.3 of Chapter 5 showing that the fingerprint expectations actually determine the fingerprint covariance, together imply that the distribution of fingerprint entries is robustly determined by its vector of first moments—thus there would be little gained by attempting to match higher-order moments.

The most significant difference between our practical estimators and those of Chapter 3 is that the practical ones do not need an upper bound on the true support size of the distribution. Without such an upper bound on the support size, the space of "plausible" histograms can be very large. To illustrate, suppose we obtain fingerprint $\mathcal{F} = (10, 0, 0, 0, \ldots)$, and consider the two histograms given by the uniform distributions with respective support sizes 10,000, and 100,000. Given either distribution, the probability of obtaining the observed fingerprint from a sample of size 10 is $> .99$, yet these distributions are quite different and have very different entropy values and support sizes. They are both very plausible–which distribution should we return if we are not provided with an upper bound on the support size?

To resolve this question in a principled fashion, we strengthen the initial goal of our linear program of finding a histogram that could have plausibly generated the observed fingerprint: we instead return the *simplest* histogram that could have plausibly generated the observed fingerprint of the sample. There are many potential implementations of this Occam's razor: we return the plausible distribution of minimal support size.

Thus we pose this problem of finding the simplest plausible histogram as a pair of linear programs. The first linear program is based on Linear Program 3.9 of Chapter 3 and returns a histogram $\hat{h}$ that minimizes the distance between its expected fingerprint and the observed fingerprint, where we penalize the discrepancy between $\mathcal{F}_i$ and $E[\mathcal{F}_i^{\hat{h}}] = \sum_{x:\hat{h}(x)\neq 0} \hat{h}(x) poi(kx, i)$ in proportion to the inverse of the standard deviation of $\mathcal{F}_i$, which we estimate as $1/\sqrt{1+\mathcal{F}_i}$. The second linear program will then find the histogram $\hat{g}$ of minimal support size, subject to the constraint that the distance between its expected fingerprint, and the observed fingerprint, is not much worse than that of the histogram found by the first linear program.

As with the estimators of Chapter 3, we will only apply the linear programs to the "rare events" regime, and will simply use the trivial empirical estimate for the regime in which the fingerprint entries are sparse, and represent a few domain elements that are seen often. Algorithm 8.1 describes a simple and crude procedure for separating these two regimes, which performs well in practice.[1]

[1] A unified approach is possible, using an earthmover distance metric as part of the linear programs to cleanly circumvent these issues. However, the experimental results this yielded were indistinguishable from those presented here, and thus do not seem to justify the additional computational expense.

Algorithm 8.1. ESTIMATE UNSEEN

Input: *Fingerprint* $\mathcal{F} = \mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_m$, *derived from a sample of size* k, *vector* $x = x_1, \ldots, x_\ell$ *with* $0 < x_i \leq 1$, *and error parameter* $\alpha > 0$.

Output: *List of pairs* $(y_1, \hat{h}_{y_1}), (y_2, \hat{h}_{y_2}), \ldots,$ *with* $y_i \in (0, 1]$, *and* $\hat{h}_{y_i} \geq 0$.

- *Initialize the output list of pairs to be empty, and initialize a vector* $\mathcal{F}'$ *to be equal to* $\mathcal{F}$.
- *For* $i = 1$ *to* k,
  - *If* $\sum_{j \in \{i - \lceil \sqrt{i} \rceil, \ldots, i + \lceil \sqrt{i} \rceil\}} \mathcal{F}_j \leq 2\sqrt{i}$ *[i.e. if the fingerprint is ''sparse'' at index* i*]*
    Set $\mathcal{F}'_i = 0$, *and append the pair* $(i/k, \mathcal{F}_i)$ *to the output list.*
- *Let* v_{opt} *be the objective function value returned by running Linear Program 8.2 on input* $\mathcal{F}', x$.
- *Let* $v = (v_{x_1}, \ldots)$ *be the vector returned by running Linear Program 8.3 on input* $\mathcal{F}', x, v_{opt}, \alpha$.
- *For all* i *s.t.* $v_{x_i} > 0$, *append the pair* (x_i, v_{x_i}) *to the output list.*

Linear Program 8.2. FIND PLAUSIBLE HISTOGRAM

Input: *Fingerprint* $\mathcal{F} = \mathcal{F}_1, \ldots, \mathcal{F}_m$, k, *and vector* $x = x_1, \ldots, x_\ell$ *consisting of a fine mesh of points in the interval* $(0, 1]$.

Output: *vector* $v = (v_{x_1}, \ldots, v_{x_\ell})$, *and objective value* $v_{opt} \in \mathbb{R}$.

Let $v_{x_1}, \ldots, v_{x_\ell}$ *and* v_{opt} *be, respectively, the solution assignment, and corresponding objective function value of the solution of the following linear program, with variables* $v_{x_1}, \ldots, v_{x_\ell}$*:*

$$\textit{Minimize:} \quad \sum_{i=1}^{m} \frac{1}{\sqrt{1 + \mathcal{F}_i}} \left| \mathcal{F}_i - \sum_{j=1}^{\ell} v_{x_j} \cdot poi(k x_j, i) \right|$$

$$\textit{Subject to:} \quad \sum_{j=1}^{\ell} x_j v_{x_j} = \frac{\sum_{i \geq 1} i \mathcal{F}_i}{k}, \textit{ and } \forall j, \quad v_{x_j} \geq 0.$$

Linear Program 8.3. FIND SIMPLEST PLAUSIBLE HISTOGRAM
Input: Fingerprint $\mathcal{F} = \mathcal{F}_1, \ldots, \mathcal{F}_m$*,* k*, and vector* $x = x_1, \ldots, x_\ell$ *consisting of a fine mesh of points in the interval* $(0, 1]$*, optimal objective function value* v_{opt} *from Linear Program 8.2, and error parameter* $\alpha > 0$*.*
Output: vector $v = (v_{x_1}, \ldots, v_{x_\ell})$*.*
Let $v_{x_1}, \ldots, v_{x_\ell}$ *be the solution assignment of the following linear program, with variables* $v_{x_1}, \ldots, v_{x_\ell}$*:*

$$\begin{aligned} &\textit{Minimize:} && \textstyle\sum_{j=1}^{\ell} v_{x_j} \\ &\textit{Subject to:} \\ &&& \textstyle\sum_{i=1}^{m} \frac{1}{\sqrt{1+\mathcal{F}_i}} \left| \mathcal{F}_i - \sum_{j=1}^{\ell} v_{x_j} \cdot poi(kx_j, i) \right| \leq v_{opt} + \alpha, \\ &&& \textstyle\sum_{j=1}^{\ell} x_j v_{x_j} = \frac{\sum_{i \geq 1} i\mathcal{F}_i}{k}\textit{, and } \forall j, \quad v_{x_j} \geq 0. \end{aligned}$$

8.2 Estimating Entropy

In this section we demonstrate that Algorithm 8.1 can be used to very accurately estimate the entropy of a distribution. Before describing our experimental setup and results, we list the five estimators for entropy that we use for comparison. The first three are standard, and are, perhaps, the most commonly used estimators [102]. The final two estimators were proposed more recently, and have been shown to perform well in some practical settings [137]. For a more detailed discussion of these estimators, see Section 2.3.

The "naive" estimator: The entropy of the empirical distribution, namely, given a fingerprint $\mathcal{F}$ derived from a sample of size k, $H^{naive}(\mathcal{F}) := \sum_i \mathcal{F}_i \frac{i}{k} |\log \frac{i}{k}|$.

The Miller-Madow corrected estimator [90]: The naive estimator H^{naive} corrected to try to account for the second derivative of the logarithm function, namely $H^{MM}(\mathcal{F}) := H^{naive}(\mathcal{F}) + \frac{(\sum_i \mathcal{F}_i)-1}{2k}$.

The jackknifed naive estimator [144, 51]:

$$H^{JK}(\mathcal{F}) := k \cdot H^{naive}(\mathcal{F}) - \frac{k-1}{k} \sum_{j=1}^{k} H^{naive}(\mathcal{F}^{-j}),$$

where $\mathcal{F}^{-j}$ is the fingerprint given by removing the contribution of the jth sample point.

The coverage adjusted estimator (CAE) [36]: Chao and Shen proposed the CAE, which is specifically designed to apply to settings in which there is a significant component of the distribution that is unseen, and was shown to perform well in practice in [137].[2] Given

[2]One curious weakness of the CAE, is that its performance is exceptionally poor on some simple large instances. For example, given a sample of size k drawn from a uniform distribution over k elements, it is not hard to show that the bias of the CAE is $O(\log k)$. This error is not even bounded! For comparison, even the naive estimator has error bounded by a constant in the limit as $k \to \infty$ in this setting. This bias of the CAE is easily observed in our experiments as the "hump" in the left plot of Figure 8.1.

a fingerprint $\mathcal{F}$ derived from a sample of size k , let $P_s := 1 - \mathcal{F}_1/k$, representing the Good–Turing estimate of the probability mass of the "seen" portion of the distribution [59]. The CAE adjust the empirical probabilities according to P_s, then applies the Horvitz–Thompson estimator for population totals [68] to take into account the probability that the elements were seen. This yields:

$$H^{CAE}(\mathcal{F}) := -\sum_i \mathcal{F}_i \frac{(i/k)P_s \log\left((i/k)P_s\right)}{1-(1-(i/k)P_s)^k}.$$

The *Best Upper Bound* estimator [102]: The final estimator to which we compare ours, is the *Best Upper Bound* (BUB) estimator of Paninski. This estimator is obtained by searching for a minimax linear estimator, with respect to a certain error metric. The linear estimators of Chapter 6 can be viewed as a variant of this estimator with provable performance bounds.[3] The difficulty in using the BUB estimator is that it requires, as input, an upper bound on the support size of the distribution from which the sample was drawn. In many settings, such an upper bound is either unknown, or nonexistent; and if the bound provided to the estimator is inaccurate, the performance degrades considerably, as was also remarked in [137]). In our experiments, for the distributions with finite support, we gave the true support size as input, and thus we are arguably comparing our estimator to the best–case performance of the BUB estimator.

Experimental setup

We performed a comparison of our estimator with the above five estimators for a variety of sample sizes and distributions. We considered three classes of distributions, the uniform distribution, $Unif[n]$ that assigns probability $p_i = 1/n$ for $i = 1, 2, \ldots, n$; the Zipf distribution $Zipf[n]$ that assigns probability $p_i = \frac{i/n}{\sum_{j=1}^{n} j/n}$ for $i = 1, 2, \ldots, n$ and is commonly used to model naturally occurring "power law" distributions, particularly in natural language processing; and the geometric distribution $Geom[n]$ which has infinite support, and assigns probability $p_i = (1/n)(1-1/n)^i$, for $i = 1, 2 \ldots$. For each distribution, we considered three settings of the parameter n, $1000, 10000$, and 100000, and for each setting, we considered drawing a sample of size k, for k ranging from $n^{0.6}$ to $n^{1.25}$. For each setting of the parameters, 500 trials were run: each trial consisted of drawing a sample of size k, then evaluating each of the estimators on the resulting sample.

All experiments were run in Matlab. The error parameter α in Algorithm 8.1 was set to be 0.1 for all trials, and the vector $x = x_1, x_2, \ldots$ used as the support of the returned histogram was chosen to be a coarse geometric mesh, with $x_1 = 1/(200n)$, and $x_i = 1.1x_{i-1}$. The experimental results are essentially unchanged if the error parameter is varied within the range $[0.01, 0.5]$, or if x_1 is decreased, or if the mesh is made more fine. We used

[3] We also implemented the linear estimators of Chapter 6 though found that the BUB estimator performed better.

Paninski's implementation of the BUB estimator (publicly available on his website), with default parameters. This estimator requires a parameter upper-bounding the support size of the distribution, which we set to be n for the three cases considered here ($Unif[n], Zipf[n]$, and $Geom[n]$).

Results and Discussion

The performance of our Algorithm 8.1 for estimating entropy is evaluated and compared with that of the five other estimators in Figure 8.1. We plot the *root-mean-square* error of each estimator for each of the three distributions ($Unif[n]$, $Zipf[n]$, and $Geom[n]$), parameterized by each of the three values $n = 1,000$, $n = 10,000$, and $n = 100,000$. Our "unseen" estimator performs well in both the super-linear and sub-linear regime, matching or exceeding the performance of all the other estimators in all cases. Further, each of the other estimators seems to have quirks that drastically limit their performance in certain regimes, while our estimator performs robustly throughout. Each of the standard three entropy estimators—the naive estimator, the Miller-Madow corrected estimator, and the jackknifed estimator—performs well only in the super-linear regime, where the sample size is larger than n, though their performance improves smoothly, with the jackknifed estimator performing better than the Miller-Madow estimator, which performs better than the naive estimator. The Coverage-Adjusted estimator performs virtually identically to the unseen estimator on the uniform distribution in the sublinear regime, suggesting that perhaps the CAE estimator was specifically designed for this case. In other cases, however, the CAE estimator performs disappointingly; even in the super-linear regime for the uniform distributions, when all the other estimators are converging rapidly, the CAE has a "hump", which in fact grows with n. The BUB estimator, as noted above, has a free parameter representing an upper bound on the support size, which we set to be equal to n to yield a "best-possible" performance for the uniform and Zipf distributions; this performance may not be representative of the BUB estimator in general.

8.3 Estimating ℓ_1 Distance, and the Number of Words in *Hamlet*

The other two properties that we consider do not have such widely-accepted estimators as entropy, and thus our evaluation of the unseen estimator will be more qualitative. We include these two examples here because they are of a substantially different flavor from entropy estimation, and highlight the robustness and versatility of our approach.

In Figure 8.2 we show the results of estimating the total variation distance (ℓ_1 distance) between two uniform distributions A, B on $n = 10,000$ points, in three cases: the two distributions are identical (left plot, $d = 0$), the two distribution supports overlap on *half* their elements (center plot, $d = 0.5$), and the two distributions have disjoint supports (right plot, $d = 1$). The estimate of the statistical distance is plotted, along with error bars at

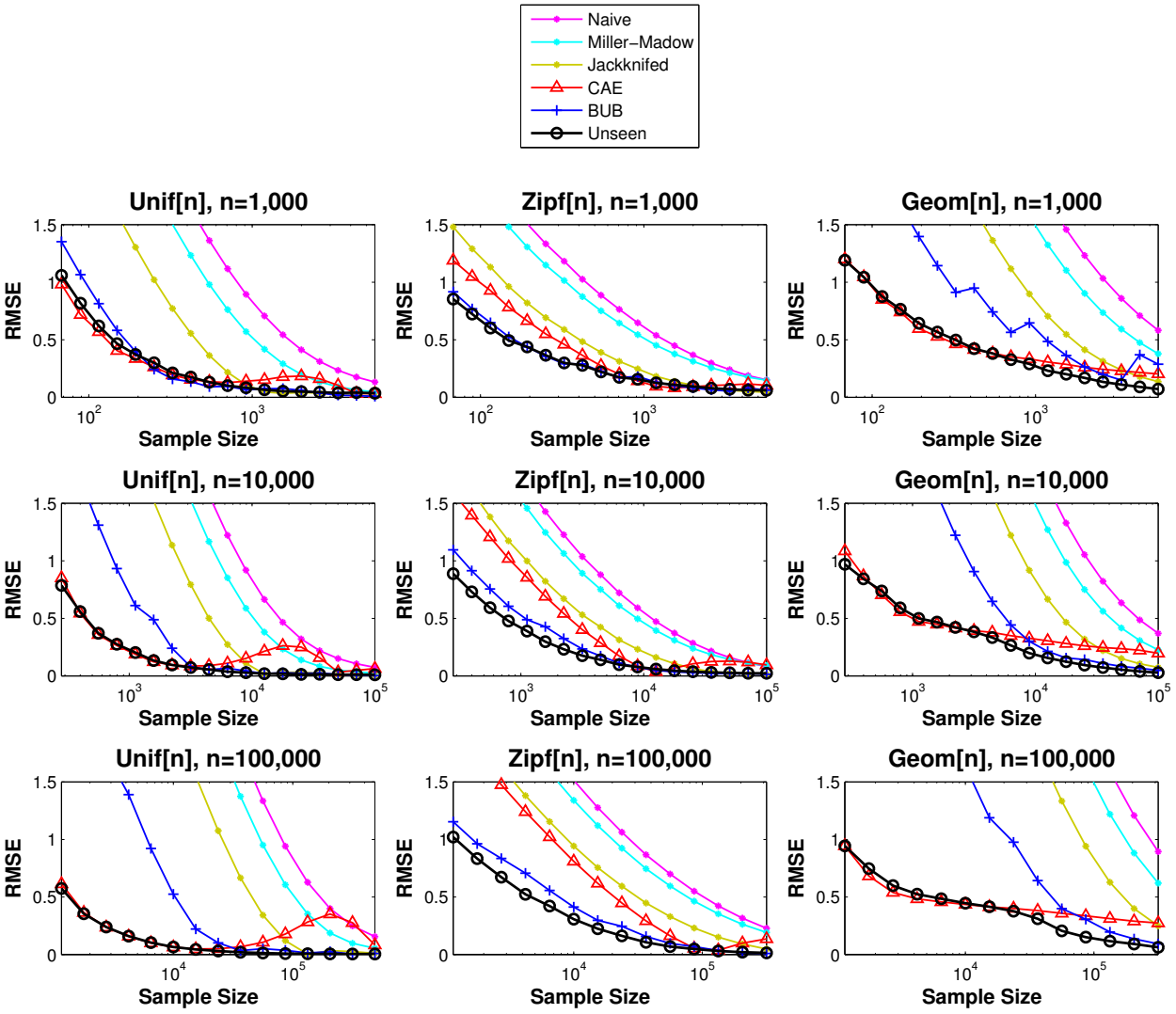

Figure 8.1: Plots depicting the square root of the mean squared error (RMSE) of each entropy estimator over 500 trials, plotted as a function of the sample size; note the logarithmic scaling of the x-axis. The samples are drawn from a uniform distribution $Unif[n]$ (left column), a Zipf distribution $Zipf[n]$ (center column), and a geometric distribution $Geom[n]$ (right column), for $n = 1,000$ (top row), $n = 10,000$ (middle row), and $n = 100,000$ (bottom row).

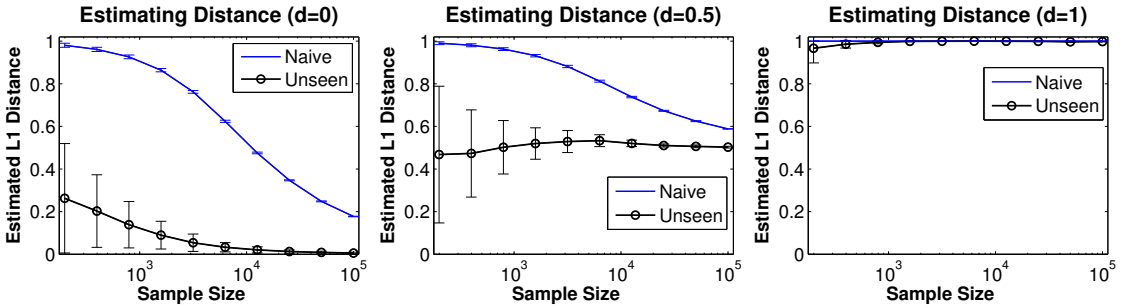

Figure 8.2: Plots depicting the estimated ℓ_1 distance (total variational distance) along with error bars showing one standard deviation, for samples from two uniform distributions of support 10,000 having distance 0 (left plot), distance 0.5 (center plot), and distance 1 (right plot) as a function of the sample size.

plus and minus one standard deviation; our results are compared with those for the naive estimator. The naive estimator, of course, returns the total variation distance between the empirical distributions observed, and thus in the third plot, when the distributions are disjoint, the naive estimator will perform *perfectly*; in the other cases, when the sample size is rather small, the two empirical distributions will be essentially disjoint, so the distance estimate of the naive estimate starts near 1 in all cases, and only gradually converges for super-linear sample sizes. Meanwhile, our unseen estimator can be seen to reliably distinguish between the $d = 0$, $d = \frac{1}{2}$, and $d = 1$ cases even for samples of size as few as several hundred.

Because total variation distance is a property of two distributions instead of one, fingerprints and histograms are two-dimensional objects in this setting (see Section 3.3 of Chapter 3), and Algorithm 8.1 and the linear programs are extended accordingly, replacing single indices by pairs of indices, and Poisson coefficients by corresponding products of Poisson coefficients.

Estimating the number of distinct words in *Hamlet*

Finally, in contrast to the synthetic tests above, we also evaluated our estimator on a real-data problem which may be seen as emblematic of the challenges in a wide gamut of natural language processing problems: *given a (contiguous) fragment of Shakespeare's* Hamlet*, estimate the number of distinct words in the whole play.* We use this example to showcase the flexibility of our linear programming approach—our estimator can be customized to particular domains in powerful and principled ways by adding or modifying the constraints of the linear program. To estimate the histogram of word frequencies in *Hamlet*, since the play is of length $\approx 25,000$, the minimum probability with which any word can occur is $\frac{1}{25,000}$. Thus in contrast to our previous approach of using Linear Program 8.3 to bound the support of the returned histogram, we instead simply modify the input vector x of Linear Program 8.2 to

contain only values $\geq \frac{1}{25{,}000}$, and forgo running Linear Program 2.[4] The results are plotted in Figure 8.3. The estimates converge towards the true value of 4268 distinct word forms extremely rapidly, and are slightly negatively biased, perhaps reflecting the fact that words appearing close together are correlated.

In contrast to Hamlet's charge that "there are more things in heaven and earth...than are dreamt of in your philosophy," we can say that there are almost exactly as many things in *Hamlet* as can be dreamt of from 10% of *Hamlet*.

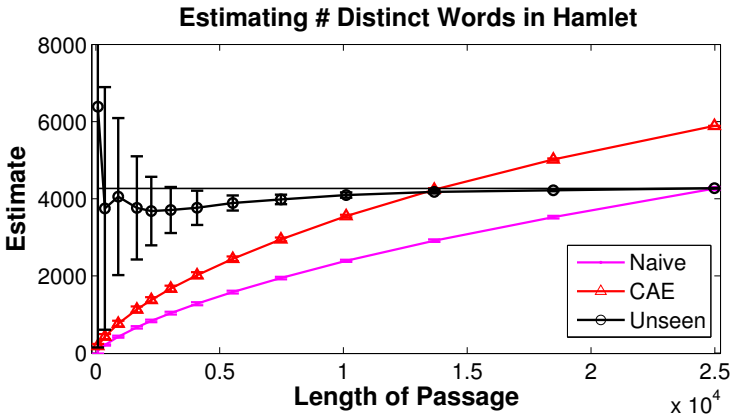

Figure 8.3: Estimates of the total number of distinct word forms in Shakespeare's *Hamlet* (excluding stage directions and proper nouns) as a function of the length of the passage from which the estimate is inferred. The error bars depict one standard deviation in the estimate over the random choice of each contiguous passage of the given length. The true number of distinct word forms, 4268, is shown as the horizontal line.

[4] A passage of text should be regarded as a sample *without replacement* from the set of words in the work; thus rather than applying our estimator to the fingerprint of the word counts of a passage, we first apply the natural transformation from a sample taken without replacement, to a sample taken with replacement, and then apply our estimator to the fingerprint of the resulting modified set of word counts.

Part II

Correlations, Parities, and Juntas

Chapter 9

Finding Correlations and the Closest Pair Problem

One of the most basic statistical tasks is the problem of finding correlations in data. In some settings, the data is gathered with the specific goal of ascertaining the set of correlated variables; in many other settings, the identification of correlated features is used repeatedly as a key data analysis primitive within the context of more complex algorithms.

From a theoretical perspective, this problem of finding correlations is extremely rich. Most of the theoretical work has focussed on the closely related problem of finding a pair of vectors with minimal distance from among some set of vectors, also known as the "closest pair problem". To see the relation between these problems, recall that the *Pearson–correlation* of two vectors with mean zero and unit length is defined to be their inner product; and the pair of unit-length vectors with maximal inner product will also be the pair with minimal Euclidean distance.

9.1 Discussion of Previous Work

Historically, the first line of work on finding close pairs of vectors focussed on the "nearest neighbor search": given a set of vectors how can one preprocess them such that given a new vector, one can efficiently find the vector in the set that is closest of the new vector (with respect to some metric–typically Euclidean distance, or Hamming distance in the Boolean setting)? For such problems, there are typically two parameters of interest: the amount of space required to store the preprocessed set of vectors, and the amount of time required to perform a single query.

The earliest work on this question considered the case in which the n points lie in very low dimensional space, $d = 1, 2, 3, \ldots$. In the case that $d = 1$, each point is a real number, and one can simply sort the list of numbers, and store the sorted list. Given a new number, one can perform a binary search over the sorted list to find the closest number. Thus the storage space is linear in the size of the input, and each query requires $O(\log n)$ comparisons. For

$d \geq 2$, the analogous scheme corresponds to partitioning the space into n regions, indexed by the points in the set, where the region corresponding to the ith point x_i consists of those points that are closer to x_i than to any of the other $n-1$ points in the set.

In the case of $d = 2$, such a partition of the plane is known as the *Voronoi diagram* of the set of n points, and yields space bounds and query time analogous to the $d = 1$ setting. Unfortunately, such schemes suffer a curse of dimensionality and do not generalize well to higher dimensions; while the query time remains polynomial in $d \log n$, the space required to store such partitions scales as $O(n^{\lceil d/2 \rceil})$ [41], and quickly cease to be preferable to performing the brute-force comparisons (see, for example, [89, 140]). On the practical side, there has been considerable work in developing reasonable data structures to partition slightly higher dimensional Euclidean spaces ($d < 20$), starting with the notion of k-dimensional trees (*kd–trees*), introduced by Bentley [24] (see [116] for a summary including more recent developments).

In an effort to overcome some of the difficulties of returning the exact closest point, starting in the late 1990's, significant effort was spent considering the c–approximate nearest neighbor problem in which the goal of returning the closest point is relaxed to the more modest goal of returning a point whose distance is at most a multiplicative factor of $c = 1 + \epsilon$ larger than that of the closest point. Additionally, a small probability of failure is allowed. In many practical settings, such a relaxation is essentially equivalent to the exact nearest neighbor problem. Starting with the results of Kushilevitz et al. [81] and Indyk and Motwani [71], algorithms requiring space that is polynomial in n and d, with query time polynomial in $d \log n$ were given (for constant $\epsilon > 0$).

Introduced in work of Indyk and Motwani [71], the concept of *locality sensitive hashing* offered both sublinear query time, as well as subquadratic space, thereby yielding nontrivial algorithms for the approximate closest pair problem. Specifically, they gave an algorithm with query time $O(n^{\frac{1}{1+\epsilon}})$ and space $O(n^{1+\frac{1}{1+\epsilon}})$. (Throughout, we ignore $\log n$ factors, and the additive dn term in the space.) The basic idea was to use a series of hashing functions that all have the property that close points have a higher probability of hashing to the same bucket. To perform a given query, one simply hashes the query point, and then checks the subset of the n data points that have also been hashed to those buckets. Since the original paper, there have been a number of improvements in the parameters, and generalizations from Hamming and Euclidean distance to other ℓ_p metrics [49, 100, 10]. The current state of the art for the $1 + \epsilon$ nearest neighbor problem under the Euclidean metric is given in Andoni and Indyk [10], achieving query time and space $O(dn^{\alpha}), O(n^{1+\alpha})$, respectively, for $\alpha = \frac{1}{(1+\epsilon)^2} + o(1)$. These results were recently shown to be essentially tight in the sense that for any scheme based on locality sensitive hashing, the exponent $\alpha \geq \frac{1}{(1+\epsilon)^2} - o(1)$ [94]. (See [9] for a survey on locality sensitive hashing.)

For the problem of finding a pair of points whose distance is at most a factor of $1 + \epsilon$ further than that of the closest pair of points, by simply running the nearest–neighbor search n times—once for each vector—one obtains algorithms with runtimes $O(n^{1+\frac{1}{1+\epsilon}})$, and $O(n^{1+\frac{1}{(1+\epsilon)^2}})$, respectively in the Hamming and Euclidean settings which are the best

previously known algorithms for these problems. For small ϵ, these runtimes are roughly $O(n^{2-\epsilon})$, and $O(n^{2-2\epsilon})$, respectively.

The Light Bulb Problem

For completeness, we restate the definition of the light bulb problem:

Definition 9.1. *Given a set of n vectors in $\{-1,+1\}^d$, with the promise that the vectors are chosen uniformly at random with the exception of two vectors that have Pearson–correlation ρ (Hamming distance $d \cdot \frac{1-\rho}{2}$), the* light bulb problem *with parameters n, d, ρ is the problem of recovering the true correlated pair of vectors.*

The light bulb problem is easily seen to be a special case of the approximate closest pair problem. To see the correspondence in parameters, for sufficiently large dimension, d, the Hamming distance between the correlated vectors will be at most $\frac{1-\rho}{2}d$ whereas, with high probability, the Hamming distance between any other pair of vectors will be close to $\frac{d}{2}$. Thus solving the $1+\epsilon$ approximate closest pair problem for $1+\epsilon \approx \frac{1}{1-\rho}$ will, with high probability, return the correlated pair of vectors.

The light bulb problem has received much less attention than the nearest-neighbor problem; the early work on locality sensitive hashing seemed unaware that somewhat similar ideas had appeared nearly a decade earlier in the work of Paturi et al. [104], which gave an algorithm for the light bulb problem with runtime $O(n^{1+\frac{\log\frac{\rho+1}{2}}{\log 1/2}})$, which is slightly better than that given by the application of locality sensitive hashing for the Hamming cube given in [71]. More recently, Dubiner introduced the "Bucketing Codes" approach [50], which is similar in spirit to the approach of Andoni and Indyk [10], and yields an algorithm for the light bulb problem with a runtime of $O(n^{\frac{2}{\rho+1}})$.

For small values of ρ, all these approaches yield runtimes of $n^{2-O(\rho)}$, with [50] achieving the best asymptotic runtime of $O(n^{2-2\rho})$, in the limit as $\rho \to 0$.

For our results on the light bulb problem, and closest pair problem, we will perform some metric embeddings: the hope is to construct some embedding $f : \mathbb{R}^d \to \mathbb{R}^m$ with the property that if $\langle u, v\rangle$ is large, then the inner product of the images of u and v, $\langle f(u), f(v)\rangle$ will also be large, and if the inner product is small, the inner product of the images is tiny. For the light bulb problem, we will be able to choose the embedding f to be a simple "XOR"/"tensor" embedding, which sets each coordinate of $f(u)$ to be the product of entries of u. Such an embedding has appeared previously in several contexts, and was used by Lyubashevsky [83] to show that given few examples from an instance of learning parity with noise, one can generate new "simulated" examples, that can be used in place of actual examples.

Our results for the approximate closest pair problem will require a more sophisticated embedding. In fact, it seems unlikely that we would be able to construct a single embedding f with the desired properties (see the comments in Section 9.3 on Schoenberg's characterization of what is achievable via a single embedding). Instead of using a single embedding, we will construct a pair of embeddings, $f, g : \mathbb{R}^d \to \mathbb{R}^m$ with the property that $\langle f(u), g(v)\rangle$ is large

[small] if $\langle u, v \rangle$ is large [small]. This observation that a pair of embeddings can be more versatile than a single embedding was first fruitfully leveraged by Alon and Naor in their work on approximating the cut norm of a matrix [6].

9.2 A New Algorithm for the Light Bulb Problem

We begin by presenting our new approach to the light bulb problem; our improved algorithm for finding the closest pair of vectors in the general setting will build upon this approach. There are two essential features of the light bulb problem which we exploit:

- With the exception of the correlated pair of vectors, the Hamming distance between pairs of vectors is tightly concentrated around a single value, $d/2$.
- One need not iteratively solve n nearest neighbor search problems, one for each vector; instead, one can effectively perform many such searches *simultaneously*.

We now provide an intuitive overview of how we exploit these features. We begin by assuming that we can choose the dimension of the vectors, d.

Given a $d \times n$ matrix X with entries in $\{-1, +1\}$ whose columns are uniformly random, with the exception of two ρ-correlated columns, one naive approach to finding the correlated columns is to simply compute $W = X^t X$, the matrix whose i, jth entry is the inner product between the ith and jth columns of matrix X. With overwhelming probability, the largest off-diagonal entry of W will correspond to the correlated columns, as that entry will have value roughly $d\rho$, whereas all the other off-diagonal entries have expected value 0, and will be tightly concentrated around 0, with standard deviation $O(\sqrt{d})$. The obvious issue with this approach is that W has n^2 entries, precluding a sub–quadratic runtime. This remains an issue even if the number of rows, d is taken to be near the information theoretic limit of $O(\frac{\log n}{\rho^2})$.

Our approach is motivated by the simple observation that if two columns of X are highly correlated, then we can compress X, by simply aggregating sets of columns. If one randomly partitions the n columns into, say, $n^{2/3}$ sets, each of size $n^{1/3}$, and then replaces each set of columns by a single vector, each of whose entries is given by the sum (over the real numbers) of the corresponding entries of the columns in the set, then we have shrunk the size of the matrix from $m \times n$, to an $m \times n^{2/3}$ matrix, Z (that now has integer entries in the range $[-n^{1/3}, n^{1/3}]$). It is still the case that most pairs of columns of Z will be uncorrelated. If, in the likely event that the two original correlated columns are assigned to distinct sets, the two columns of Z to which the two correlated columns contribute, will be *slightly* correlated. Trivially, the expected inner product of these two columns of Z is $O(\rho m)$, whereas the inner product between any two other columns of Z has expected value 0, and variance $O(n^{2/3} m)$. Thus provided $\rho m >> \sqrt{n^{2/3} m}$, and hence $m >> n^{2/3}/\rho^2$, there should be enough data to pick out the correlated columns of matrix Z, by computing $W' = Z^t Z$, and then finding the largest off-diagonal element. This computation of the product of an $n^{2/3} \times n^{2/3}/\rho^2$

matrix with its transpose, via fast matrix multiplication, is relatively cheap, taking time $n^{2\omega/3}poly(1/\rho) < n^{1.6} \cdot poly(1/\rho)$, where ω is the exponent of matrix multiplication.

Once one knows which two columns of Z contain the original correlated columns of W, one can simply brute-force check the inner products between all pairs of columns of W that contribute to the two correlated columns of Z, which takes time $dn^{2/3}$. (One could also recursively apply the algorithm on the two relevant sets of $n^{1/3}$ columns, though this would not improve the asymptotic running time.) The computation, now, is dominated by the size of the initial $n^{2/3}/\rho^2 \times n > n^{1.66}$ matrix! It is also clear that the runtime of this algorithm will depend only inverse polynomially on the correlation–in particular, ρ will not appear in the exponent of n. Optimizing the tradeoff between the size of the initial matrix, and the time spent computing the product, yields an exponent of $(5-\omega)/(4-\omega) < 1.62$.

In the rest of this section we formally describe the algorithm, and give the proof of correctness.

Algorithm 9.2. VECTOR AGGREGATION
Input: An $m \times n$ matrix X with entries $x_{i,j} \in \{-1, +1\}$, and constant $\alpha \in (0, 1]$.
Output: A pair of indices, $c_1, c_2 \in [n]$.

- Randomly partition $[n]$ into $n^{1-\alpha}$ disjoint subsets, each of size n^α, denoting the sets $S_1, \ldots, S_{n^{1-\alpha}}$, and form the $m \times n^{1-\alpha}$ matrix Z with entries $z_{i,j} = \sum_{k \in S_j} x_{i,k}$, where the sum is taken over the reals.
- Let $W = Z^t Z$, and denote the largest off-diagonal entry by $w_{i,j}$.
- Using a brute-force search, taking time $O(mn^{2\alpha})$, find and output the pair

$$(c_1, c_2) := argmax_{c_1 \in S_i, c_2 \in S_j} \sum_{k=1}^{m} x_{k,c_1} x_{k,c_2}$$

.

The following proposition describes the performance of the above algorithm:

Proposition 9.3. *For any constant $\epsilon > 0$, setting $\alpha = \frac{1}{2(4-\omega)}$ the algorithm* VECTOR-AGGREGATION*, when given as input the value α as above and the matrix X whose columns consist of the vectors given in an n, d, ρ instance of the light bulb problem with $d = \frac{n^{2\alpha+\epsilon}}{\rho^2}$, will output the true pair of correlated columns with probability $1 - o(1)$, and will run in time*

$$O\left(\frac{n^{\frac{5-\omega}{4-\omega}+\epsilon}}{\rho^{2\omega}}\right) < n^{1.62} \cdot \text{poly}(1/\rho),$$

where $\omega < 2.38$ is the exponent of matrix multiplication.

Proof. We first verify the runtime of the algorithm. The input matrix X has size $dn = n^{1+2\alpha+\epsilon}/\rho^2$, and the creation of the matrix Z takes time linear in this size. The only remaining

bottleneck is the computation of $W = Z^t Z$, which is the product of a $n^{1-\alpha} \times d$ matrix with its transpose, and hence can be computed in time $\max(d^\omega, (n^{1-\alpha}/d)^2 d^\omega) < n^{\frac{5-\omega}{4-\omega}}/\rho^{2\omega}$, where the second argument to the max operation is the case that $d < n^{1-\alpha}$.

We now verify the correctness of the algorithm. Assume without loss of generality that the true correlated columns are the first and second columns, and that they contribute to distinct sets (which happens with probability $> 1 - 1/n^\alpha$), and denote those sets S_1, S_2, respectively. By a union bound over Chernoff bounds, with probability $1 - o(1)$, for all i, j, $|z_{i,j}| \leq n^{\alpha/2} \log^2 n$. Additionally, aside from the first and second columns of Z, which correspond to the sets S_1, S_2, all columns are independent, with each value $z_{i,j}$ having expectation 0, and thus by another union bound over Chernoff bounds, with probability $1 - o(1)$ each off-diagonal entry of W aside from $w_{1,2}$ and $w_{2,1}$ will have magnitude at most $|z_{i,j}|^2 \cdot \sqrt{d} \log^2 n \leq \frac{n^{2\alpha+\epsilon/2}}{\rho}$polylog n.

We now argue that $w_{1,2}$ will be significantly larger than this value. Indeed, $w_{1,2} = \sum_{i \in S_1, j \in S_2} \langle X_i, X_j \rangle$, where X_i denotes the ith column of matrix X, and by the above calculation, the magnitude of the contribution to this sum from all terms other than $\langle X_1, X_2 \rangle$ will be at most $\frac{n^{2\alpha+\epsilon/2}}{\rho}$polylog n, with probability $1 - o(1)$. To conclude, the expected value of $\langle X_1, X_2 \rangle = \rho d$, and with probability $1 - o(1)$, $\langle X_1, X_2 \rangle \geq \frac{\rho d}{2} = \frac{n^{2\alpha+\epsilon}}{2\rho}$, which dominates $\frac{n^{2\alpha+\epsilon/2}}{\rho}$polylog n for any constant $\epsilon > 0$ and sufficiently large n, and thus $w_{1,2}$ or $w_{2,1}$ will be the largest off-diagonal entries of W with probability $1 - o(1)$, in which case the algorithm outputs the correct indices. □

Projecting Up

The algorithm described above shows how to solve the light bulb problem *provided that the points have dimension* $d \geq n^{\frac{1}{4-\omega}+\epsilon}/\rho^2 \approx n^{0.62}/\rho^2$. What happens if d is quite small? Information theoretically, one should still be able to recover the correlated pair even for $d = O(\log n/\rho^2)$. How can one adapt the VECTOR-AGGREGATION approach to the case when d is near this information theoretic boundary? We shall carefully project the vectors into a larger space in such a way so as to guarantee that the projected vectors act like vectors corresponding to an n, d', ρ' instance of the light bulb problem for some $d' > d$, and $\rho' < \rho$, with the property that d' is sufficiently large so as ensure that the approach of the previous section succeeds. We rely crucially on the fact that ρ does not appear in the exponent of n in the runtime, since this transformation will yield $\rho' << \rho$.

Consider randomly selecting a small set of the rows of the matrix, and producing a new row by component–wise multiplication. We term such a process of generating new examples (rows) as "XORing together a set of rows", and we claim that the new row of data thus produced is reasonably faithful. In particular, if two columns are completely correlated, then the result of XORing together a number of rows will produce a row for which the values in the two correlated columns will still be identical. If the correlation is not 1, but instead ρ, after combining q rows, the corresponding columns will only be ρ^q correlated, as XORing degrades the correlation. Recall, however, that the algorithm of the previous section was

extremely noise robust, and thus we can afford to degrade the correlation considerably. For constant ρ, we can certainly take $q = o(\log n)$ without increasing the exponent of n in the runtime.

Note that as we are viewing the vectors as having entries in $\{-1, 1\}$, this XORing of sets of rows is simply component-wise multiplication of the rows. Equivalently, it can be seen as replacing each column with a sample of the entries of the qth tensor power of the column.

In the context of learning parity with noise, this expansion approach was used by Lyubashevsky [83] to show that given few examples from an instance of learning parity with noise, one can generate new "simulated" examples, that can be used in place of actual examples. In contrast to the current setting, the challenge in that work was arguing that the new instances are actually information theoretically *indistinguishable* from new examples (with higher noise rate). To prove this strong indistinguishability, Lyubashevsky employed the "Leftover Hash Lemma" of Impagliazzo and Zuckerman [70].

In our setting, we do not need any such strong information theoretic guarantees, and hence our results will apply more generally than the random setting of the Light Bulb problem. Our approach only requires some guarantee on the inner products of pairs of columns, which can be given by inductively applying the following trivial lemma:

Lemma 9.4. *Given vectors $u, v, w, z \in \mathbb{R}^d$ with $\langle u, v\rangle = \rho_1 d$ and $\langle w, z\rangle = \rho_2 d$, for i, j chosen uniformly at random from $[d]$,*

$$E[(u_i w_j) \cdot (v_i z_j)] = \rho_1 \rho_2.$$

Phrased differently, letting $x \in \mathbb{R}^{d^2}$ be the vector whose entries are given by the d^2 entries of the outer-product uw^t, and y is given by the entries of vz^t, then $\langle x, y\rangle = \rho_1 \rho_2 d^2$. Elementary concentration bounds show that provided one samples sufficiently many indices of this outer product, the inner product between the sampled vectors will be close to this expected value (normalized by the dimension).

Proof. The proof follows from the independence of i, j, the facts that $E[u_i v_i] = \rho_1, E[w_j z_j] = \rho_2$, and the basic fact that the expectation of the product of independent random variables is the product of their expectations. □

We now state our theorem, which applies more generally than the Light Bulb problem, and can alternately be viewed as a result on the complexity of approximating the product of two matrices, under the assumption that the product has only a moderate number of large entries.

Theorem 9.1. *Consider a set of n vectors in $\{-1, 1\}^d$ and constants $\rho, \tau \in [0, 1]$ such that the following condition holds:*

- *For all but at most $n^{\frac{3}{4}\omega - \frac{1}{2}} \approx n^{1.3}$ pairs u, v of distinct vectors, $|\langle u, v\rangle| \leq \tau d$.*

There is an algorithm that, with probability $1 - o(1)$*, will output* all *pairs of vectors whose inner product is least* ρd*. Additionally, the runtime of the algorithm is*

$$dn^{\frac{3\omega}{4}} \cdot n^{4\frac{\log \rho}{\log \tau}} \operatorname{polylog} n \leq O(dn^{1.79+4\frac{\log \rho}{\log \tau}}),$$

where $\omega < 2.38$ *is the exponent of matrix multiplication.*

Algorithm 9.5. EXPAND AND AGGREGATE
Input: An $m \times n$ *matrix* X *with entries* $x_{i,j} \in \{-1, +1\}$*,* $\epsilon \in (0, 1)$*, and* $\rho, \tau \in (0, 1)$*, with* $\rho > \tau$
Output: Two indices $c_1, c_2 \in [n]$.

- *Let* $m' = n^{\frac{3}{4}+2\frac{\log \rho}{\log \tau}} \log^5 n$*, and* $q = \frac{\log n}{-\log \tau}$.
- *If* $m' \geq n$*, do the* $O(mn^2)$ *time brute-force search.*
- *Otherwise, we create an* $m' \times n$ *matrix* Y *with entries in* $\{-1, +1\}$*:*
  - *For each of the* m' *rows of* Y*, select a list* $t_1, \ldots, t_q$ *with each* t_i *selected uniformly at random from* $[m]$*, and set the* j*th component of the corresponding row to be* $\prod_{i=1}^{q} x_{t_i,j}$.
- *Let* c_1, c_2 *be the output of algorithm* VECTOR-AGGREGATION *on input* Y *with the parameter* $\alpha = \frac{1}{4}$*, where the algorithm is modified to brute-force-search for each of the top* $n^{\frac{3}{4}\omega - \frac{1}{2}}$ *entries of* W.

The intuition of the above algorithm is that the matrix Y resulting from the XOR expansion step has the property that the expected inner product between any two "bad" columns is bounded in magnitude by $m'\tau^q = m'\frac{1}{n}$, and the expected inner product of a "good" pair of vectors will be $m'\rho^q = m'n^{-\frac{\log \rho}{\log \tau}} >> m'\frac{1}{n}$. We now hope to argue that the inner products of the "bad" vectors are closely concentrated about their expectations, in which case the Vector Aggregation step of the algorithm will find a "good" pair of vectors. The minor technical issue is that the entries of matrix Y resulting from the XOR expansion step are not independent. Even if we start with an instance of the Light Bulb problem—while the expansion step, intuitively, can be thought of allowing us to pretend that we were given much more data than we were, the added dimensions are far from independent. This lack of independence harms the exponent slightly—we leverage the randomness of the partitioning of the Vector Aggregation algorithm to obtain slightly worse concentration than we would obtain in the truly random setting. This results in a worse exponent of ≈ 1.79 instead of ≈ 1.62 as in Proposition 9.3, though this discrepancy can, perhaps, be removed via a tighter analysis.

Before proving Theorem 9.1, we state a simple corollary that applies to the light bulb problem in the setting in which the dimension, d, is near the information theoretic limit of $O(\frac{\log n}{\rho^2})$. This corollary results immediately from applying Theorem 9.1 together with elementary Chernoff bounds.

Corollary 9.6. *For any constant $\rho, \epsilon > 0$, there exists a constant c_ϵ dependent on ϵ such that for sufficiently large n, given an instance of the Light Bulb problem with parameters n, d, ρ with $d \geq c_\epsilon \frac{\log n}{\rho^2}$, with probability $1 - o(1)$, the pair of correlated vectors can be found in time $O(dn^{\frac{3\omega}{4}+\epsilon}) \leq O(dn^{1.79})$.*

Before proving Theorem 9.1, we establish a simple concentration result for the sum of the entries of a random submatrix of a specified size:

Lemma 9.7. *Given an $s \times s$ matrix X, with entries bounded in magnitude by b, let $S_1, S_2 \subset [s]$ be two sets of size h chosen uniformly at random. Define the random variable $y := \sum_{i \in S_1, j \in S_2} X_{i,j}$. Then*

$$Pr\left[|y - E[y]| > b \cdot h^{3/2} \log h\right] = o(1/poly(h)).$$

Proof. First consider selecting set S_1, and then selecting set S_2. Let $z_{S_1} := E[y|S_1]$ denote the expected value of y given the choice of S_1. We now argue that $Pr[|z_{S_1} - E[z_{S_1}]| \geq b \cdot h^{3/2} \log h] = o(1/poly(h))$. To see this, let $p_i = \frac{\sum_{j=1}^{s} x_{j,i}}{s}$ denote the average weight of the ith column, and thus $z_{S_1} = h \sum_{i \in S_1} p_i$. The probability of z_{S_1} deviating from its expectation by more than some value is easily seen to be dominated by the process of choosing the h contributing p_i's with replacement from the set $\{p_1, \ldots, p_s\}$, in which case a standard Chernoff bound applies, yielding that $Pr[|z_{S_1} - E[z_{S_1}]| \geq b \cdot h^{3/2} \log h] < e^{-\Theta(\log^2 h)} = o(1/poly(h))$.

We now argue that, with high probability, the value of y will be closely concentrated around z_{S_1}. In analogy with the above, fixing a set S_1 fixes the average weight of each row of the restriction of matrix X to the columns indexed by elements of S_1. An identical analysis to the above (over the randomness in the choice of S_2 rather than S_1) yields the desired lemma. □

We now prove Theorem 9.1.

Proof of Theorem 9.1. The runtime of Expand and Aggregate is dominated by the multiplication of an $n^{3/4} \times m'$ matrix with its transpose, and thus trivially, takes time at most $O\left(n^{\frac{3\omega}{4}} \cdot (\frac{m'}{n^{3/4}})^2\right)$.

To verify the correctness of the algorithm, we first proceed under the assumption that the only pair of columns with inner product greater than τd is the pair with inner product at least ρd. First observe that for a pair of vectors with inner product bounded in magnitude by τd, after the degree q XORing expansion, by Lemma 9.4, the magnitude of the expected inner product is at most $m' \cdot \tau^q \leq m'/n \leq 1$, which is negligible in comparison to the variance of this quantity. By a union bound over Chernoff bounds, with probability $1 - o(1)$ all such inner products will have magnitude at most $\sqrt{m'} \log n < n^{\frac{3}{8} + \frac{\log \rho}{\log \tau}} \log^{7/2} n := \beta$. Since the inner product of two sums of sets of vectors is simply the sum of the pairwise inner products between the elements of the sets, by Lemma 9.7, the contribution from these uncorrelated

columns to each entry of the product of the aggregated matrix and its transpose—matrix W—calculated in the VECTOR-AGGREGATION stage of the algorithm will be bounded by

$$(n^{1/4})^{3/2} \log n \cdot \beta = n^{\frac{3}{4}+\frac{\log \rho}{\log \tau}} \log^{9/2} n$$

On the other hand, the inner product of the expanded pair of correlated vectors will, with probability $1 - o(1)$, be at least

$$\frac{1}{2} m' \rho^q = \frac{1}{2} m' n^{-\frac{\log \rho}{\log \tau}} = \frac{1}{2} n^{\frac{3}{4}+\frac{\log \rho}{\log \tau}} \log^5 n,$$

which dominates the contribution of the uncorrelated vectors for sufficiently large n.

To conclude, we consider the case that there might be at most $n^{\frac{3}{4}\omega-\frac{1}{2}}$ pairs of columns with inner product $> \tau d$. With probability $1-o(1)$, all the pairwise correlations between the sets of vectors to which the most correlated pair get grouped will be at most τd. Additionally, as there are at most $n^{\frac{3}{4}\omega-\frac{1}{2}}$ pairs of vectors whose inner products have magnitude greater than τd, there will be at most this many entries of W that are larger than $n^{\frac{3}{4}+\frac{\log \rho}{\log \tau}} \log^{9/2} n$, with probability $1 - o(1)$. Thus one could modify the VECTOR-AGGREGATION algorithm by performing the $O(dn^{1/2})$ time brute-force search for each of the $n^{\frac{3}{4}\omega-\frac{1}{2}}$ largest entries of the matrix W of the VECTOR-AGGREGATION algorithm, taking total time $O(dn^{\frac{3}{4}\omega})$. The probability of success can trivially be boosted by repeating the entire algorithm. □

9.3 The Chebyshev Embedding, and Closest-Pair Problem

We now abstract and refine the main intuitions behind the EXPAND AND AGGREGATE algorithm, to yield our algorithm for the general approximate closest pair problem, which will work in both the Boolean and Euclidean settings. The VECTOR-AGGREGATION algorithm of the previous section relies, crucially, on the tight concentration around 0 of the inner products of the uncorrelated vectors. In the case of Proposition 9.3, this concentration came "for free", because we assumed that the dimensionality of the data was large $\approx n^{0.6}$. To obtain Theorem 9.1, we needed to work to obtain sufficiently tight concentration. In particular, we performed a metric embedding $f : \{-1, +1\}^d \to \{-1, +1\}^m$, with the crucial property that for an appropriately chosen integer q, for $u, v \in \{-1, +1\}^d$,

$$\frac{\langle f(u), f(v) \rangle}{m} \approx \left(\frac{\langle u, v \rangle}{d} \right)^q.$$

The key property of this mapping $x \to x^q$ is that if one pair of vectors has an inner product that is a factor of $(1 + \epsilon)$ larger than than that of any other pair, after performing this mapping, the inner product of the image of the close pair will now be a factor of $(1 + \epsilon)^q >> 1 + \epsilon$ larger than that of the images of any other pair of vectors; thus the "gap" has been significantly expanded. Of course, we can not take q to be arbitrarily large, as we

would like to maintain a subquadratic amount of data and thus $m << n$, and the variance in the inner products that arises from the subsampling process (choosing which subsets of the rows to XOR) will be $O(m)$. Thus if q is so large that the $O(\sqrt{m})$ standard deviation in the inner product dominates the $m\rho^q$ inner product of the images of the closest pair, all the signal in the data will be lost and the algorithm will fail. (Phrased more generally, if q is too large, the distortion caused by projecting to a lower dimensional space will swamp the signal.)

A simple calculation shows that if we try to obtain an algorithm for the $(1+\epsilon)$ approximate closest pair problem via this Expand and Aggregate approach, we would end up with an algorithm with runtime $n^{2-O(\epsilon)}$. Can we do any better? To simplify the exposition, assume that we are told that there is a "good" pair of vectors with inner product at least $(1+\epsilon)d/2$, and that all other pairs of vectors are "bad" and have inner product in the range $[-d/2, d/2]$. In order to improve upon this runtime of $n^{2-O(\epsilon)}$, we need an improved embedding—one that damps the magnitudes of the "bad" pairs of vectors as much as possible, while preserving the inner product between the closest pair. Specifically, we seek a mapping $f_c : \mathbb{R}^d \to \mathbb{R}^m$ with the following properties:

- For all $u, v \in \mathbb{R}^d$, if $\langle u, v\rangle \geq (1+\epsilon)d/2$, then $\langle f_c(u), f_c(v)\rangle \geq c$.
- For all $u, v \in \mathbb{R}^d$, if $\langle u, v\rangle \in [-d/2, d/2]$, then $\langle f_c(u), f_c(v)\rangle$ is as small as possible.
- For all $u \in \mathbb{R}^d$, $f_c(u)$ can be computed reasonably efficiently.

The dimension of the image, m, is not especially important, as we could always simply choose a random subset of the dimensions to project onto while roughly preserving the inner products (provided this can all be computed efficiently). In general, it is not clear what the optimal such embedding will be, or how extreme a "gap amplification" we can achieve. In the following section, we show how to construct one family of natural embeddings which allow us to give an algorithm for the $(1+\epsilon)$ approximate closest pairs problem with runtime $n^{2-\Omega(\sqrt{\epsilon})}$.

Embedding via Monic Polynomials

For the remainder of the chapter, it will prove convenient to consider vectors with unit Euclidean norm; hence the boolean vectors will be scaled by a factor of $1/\sqrt{d}$. Given a monic degree q polynomial P, with q real roots $r_1, \ldots, r_q \in (-1, 1)$ we wish to give a mapping $f : \mathbb{R}^d \to \mathbb{R}^m$ such that $\langle f(u), f(v)\rangle \approx P\left(\langle u, v\rangle\right)$.

Constructing such a mapping in general is not possible: a classical result of Schoenberg from the 1940s [117] characterizes the set of functions $g : \mathbb{R} \to \mathbb{R}$ which have the property that for any d, there exists $f : S^{d-1} \to \mathbb{R}^m$ such that $\langle f(u), f(v)\rangle = g\left(\langle u, v\rangle\right)$ for all $u, v \in S^{d-1}$, where S^{d-1} denotes the d-dimensional spherical shell. In particular, he showed that a necessary and sufficient condition for such functions g is that their Taylor expansion about 0 has exclusively nonnegative coefficients (and converges uniformly).

Given that realizing a general polynomial P via a single embedding is impossible, we now briefly describe how to construct *two* mappings, $f, g : \mathbb{R}^d \to \mathbb{R}^m$ with the property that

$$\langle f(u), g(v) \rangle \approx P\left(\langle u, v \rangle\right) \cdot \frac{1}{2^q}.$$

Note that for the purposes of the closest pair problem, such a pair of embeddings are just as good as a single embedding.

Lemma 9.4 shows that if we can construct such embeddings for the polynomials Q_1 and Q_2, then by simply taking the component-wise products of pairs of rows (as in the "XOR" embedding), we can obtain an embedding for the polynomial $Q(x) = Q_1(x)Q_2(x)$. Thus all that remains is showing that we can construct embeddings for the degree-1 polynomials $Q(x) = \frac{x - r_i}{2}$ for each root r_i of the desired polynomial P.

The mappings for $Q(x) = \frac{x+1}{2}$ is obtained by scaling the vector by $1/\sqrt{2}$, and then adding d additional dimensions to each vector, populated with $1/\sqrt{2d}$'s, thus sending an inner product of c to an inner product (in $2d$-dimensional space) of $\frac{c}{2} + \frac{1}{2}$. Generally, for $Q(x) = \frac{x - r_i}{2}$, the mapping f will scale the entries by $1/\sqrt{2}$, and then simply add d additional dimensions populated by $1/\sqrt{2}$. The mapping g will scale the entries by $1/\sqrt{2}$, and then add d additional dimensions where the first $\frac{1-r_i}{2}d$ dimensions are populated with $1/\sqrt{2d}$, and the remaining $\frac{1+r_i}{2}d$ dimensions are populated with $-1/\sqrt{2d}$. Given these mappings, an inner product of c will yield an inner product of $\frac{c}{2} + \frac{1-r_i}{4} - \frac{1+r_i}{4} = \frac{c-r_i}{2}$, as desired. Given this tool, the question is now *which polynomials should we use?* The following fact suggests an embedding which, at least among a certain class of embeddings, will be optimal.[1]

Fact 9.8. *(e.g. Thm. 2.37 of [112]) For any $x \notin [-1, 1]$,*

$$T_q(x) = \max\left\{|p(x)| : p \in \mathcal{P}_q \text{ and } sup_{y \in [-1,1]}|p(y)| \leq 1\right\},$$

where T_q is the degree q Chebyshev polynomial (of the first kind), and $\mathcal{P}$ denotes the set of all degree q polynomials with real coefficients.

Perhaps the most surprising aspect of this fact is that a single polynomial, T_q captures this extremal behavior for all x.

To illustrate the general approach, for the example above in which all the "bad" inner products of the boolean setting are in the range $[-d/2, d/2]$, which becomes $[-1/2, 1/2]$ after scaling, we will construct an embedding corresponding to the monic polynomial $P(x) = T_q(2x)/2^{2q-1}$, where $T_q(x)$ is the qth Chebyshev polynomial (of the first kind). Note that since $T_q(x)$ has q roots, all in the interval $[-1, 1]$, the polynomial $P(x)$ will also have q real roots in the interval $[-1/2, 1/2]$. The corresponding mappings f, g, constructed as described

[1] We note that better constants would be obtained by replacing Chebyshev polynomials of the first kind, with Chebyshev polynomials of the second kind, as we want to minimize the ℓ_1 norm of the inner products of the images of the "bad" vectors, rather than the ℓ_∞ norm, though the difference is a small constant, and the analysis is easier in the case of Chebyshev polynomials of the first kind.

above, will have the property that $\langle f(u), g(v)\rangle = P\left(\langle u, v\rangle\right)/2^q$. In particular, we will have the following two properties, the second of which will be the source of the $\sqrt{\epsilon}$ term in the $n^{2-\Omega(\sqrt{\epsilon})}$ runtime of our approximate closest pair algorithm:

- For u, v with $\langle u, v\rangle \in [-1/2, 1/2]$, $\langle f(u), f(v)\rangle \leq \frac{1}{2^{3q-1}}$.
- For u, v with $\langle u, v\rangle \geq \frac{1}{2} + \frac{\epsilon}{2}$, $\langle f(u), f(v)\rangle \geq \frac{e^{q\sqrt{\epsilon}}}{2^{3q-1}}$.

Our general algorithm for the approximate closest pair problem, which uses this Chebyshev embedding, together with the Vector Aggregation algorithm and the fast *rectangular* matrix multiplication of Coppersmith [42], yields the theorem given below. Roughly, we will choosing $q = O(\log n)$, and hence the above multiplicative gap of $e^{q\sqrt{\epsilon}} = n^{O\sqrt{\epsilon}}$, hence we will be able to aggregate sets of $n^{O(\sqrt{\epsilon})}$ vectors. We will ensure that the image of the original vectors have dimension $m < n^{0.29}$, hence the most computationally expensive step of our algorithm will be the computation of the product of an $n^{1-O(\sqrt{\epsilon})} \times m$ matrix and an $m \times n^{1-O\sqrt{\epsilon}}$ matrix, using fast rectangular matrix multiplication, which will have runtime $O(n^{1-O(\sqrt{\epsilon})})$.

The algorithm and proof of correctness are described in the following two sections. In Section 9.4 we show how to obtain a pair with an additive guarantee on the inner product. In Section 9.5 we show how to translate this algorithm for obtaining a pair with an additive guarantee on the inner product into an algorithm with a multiplicative guarantee on the distance.

Theorem 9.2. *Given n vectors in $\mathbb{R}^d$, an approximation parameter $\epsilon > 0$, and a probability of failure $\delta > 0$, our algorithm returns a pair of vectors u, v such that with probability at least $1 - \delta$, the Euclidean distance $||u - v|| \leq (1 + \epsilon)d^*$, where d^* is the Euclidean distance between the closest pair of vectors. Additionally, the algorithm runs in time*

$$\left(n^{2-\Omega(\sqrt{\epsilon})} + nd\right) poly(\log \frac{1}{\delta}, \log n).$$

9.4 Finding Vectors with Maximal Inner Product

We start by describing an algorithm that takes as input a set of vectors of unit length, and recovers a pair whose inner product is *additively* within ϵ from that of the pair with largest inner product. In the following section, we describe how to use such an algorithm to yield an algorithm with the desired $1 + \epsilon$ multiplicative distance guarantee.

The first step of our algorithm will reduce the dimensionality of the input vectors to $d' = O(\log n/\epsilon^2)$, and will make all entries have value $\pm\frac{1}{\sqrt{d}}$. Ensuring that all entries have comparable magnitudes is critical, as we will require Chernoff–type concentration bounds in the results of expanding the vectors; even if the entries of the vectors were chosen to be independent Gaussians (as would be yielded by applying a standard Johnson–Lindenstrauss transformation) after taking the qth degree XOR (tensor) of the vectors, we would not have

the desired concentration. Additionally, there is the added convenience that the transformed vectors will all have identical norm.

Algorithm 9.9. MAKE UNIFORM
Input: An $m \times n$ matrix X with entries $x_{i,j} \in \mathbb{R}$ whose columns have unit Euclidean norm, and $\delta \in (0,1)$.
Output: An $m' \times n$ matrix Y with entries $y_{i,j} \in \{\pm\frac{1}{\sqrt{m'}}\}$, where $m' = \frac{100 \log n}{\delta^2}$.

- For each $i = 1, \ldots, m'$, select a random unit vector $v \in \mathbb{R}^m$, and let $w = v^t X$. For all $j = 1, \ldots, n$, set $y_{i,j} = sign(w_j) \cdot \frac{1}{\sqrt{m'}}$.

The following basic lemma characterizes the performance of the above algorithm:

Lemma 9.10. *Letting Y denote the output of running Algorithm 9.9 on input X, δ, where X is a matrix whose columns have unit norm, with probability $1-o(1)$, for all pairs $i, j \in [n]$,*

$$\left| \langle Y_i, Y_j \rangle - \left(1 - 2\frac{\cos^{-1}(\langle X_i, X_j \rangle)}{\pi} \right) \right| \leq \delta,$$

where X_i, Y_i denote the ith columns of X and Y, respectively. And thus if $\langle Y_i, Y_j \rangle \geq \max_{k \neq \ell} \langle Y_k, Y_\ell \rangle - \delta$, then with probability $1 - o(1)$, $\langle X_i, X_j \rangle \geq \max_{k \neq \ell} \langle X_k, X_\ell \rangle - 2\pi\delta$.

Proof. Letting α denote the angle between X_i and X_j, hence $\langle X_i, X_j \rangle = \cos(\alpha)$, for any $k \in [m']$,

$$\Pr[y_{k,i} y_{k,j} = -\frac{1}{m'}] = \Pr[r \in [0, \alpha]] = \frac{\alpha}{\pi},$$

where $r \leftarrow Unif[0, \pi]$, is selected uniformly at random from the interval $[0, \pi]$. Hence

$$E[\langle Y_i, Y_j \rangle] = 1 - 2\frac{\alpha}{\pi}.$$

Since all entries $y_{i,j} \in \pm\frac{1}{\sqrt{m'}}$, and the different dimensions are chosen independently, a union bound over n^2 Chernoff bounds yields that with probability $1 - o(1)$, all pairwise inner products will be within $\pm\delta$ of their expectations.

The second claim follows from noting that the above guarantees that if $\langle Y_i, Y_j \rangle \geq \max_{k \neq \ell} \langle Y_k, Y_\ell \rangle - \delta$, then with the claimed probability the expected inner product of Y_i and Y_j is at most 2δ smaller than that of the maximal expected inner product, and hence the angle between the corresponding columns of X is at most $2\pi\delta$ smaller than that of the optimal pair, and hence the inner products of the corresponding columns of X are at most $2\pi\delta$ smaller than the optimal inner product, since the magnitude of the derivative of the cosine function is at most 1. □

Algorithm 9.11. EXPAND AND AGGREGATE (GENERAL)
Input: An $m \times n$ matrix X with entries $x_{i,j} \in \mathbb{R}$ whose columns have unit Euclidean norm, and $\epsilon \in (0,1)$.
Output: Two indices $c_1, c_2 \in [n]$, s.t. with probability > 0.4, $\langle X_{c_1}, X_{c_2} \rangle \geq \max_{i,j \in n} \langle X_i, X_j \rangle - \epsilon$, where X_c denotes the cth column of matrix X.

- Let X' denote the $m' \times n$ matrix with $m' = O(\frac{\log n}{\epsilon^2})$ resulting from applying Algorithm 9.9 to matrix X with input parameter $\delta = \epsilon/4\pi$.
- Choose n random pairs of distinct columns of X', for each pair compute their inner product, let v_{max} be the maximum such inner product, and let $b := 2 + \log_2 \frac{2}{1+v_{max}}$, and $t = \frac{0.27 \log_2 n}{2b}$.
- Randomly partition the columns of X' into two sets of size $n/2$, forming $m' \times \frac{n}{2}$ matrices X_1 and X_2.
- We now iteratively populate the $m'' \times \frac{n}{2}$ expanded matrices Y_1, Y_2, where $m'' = n^{0.28}$:
- For $i = 1$ to m'':
  - Construct the list $S = (s_1, \ldots, s_t)$, with $s_j \in [m'] \cup \{-, +\}$ by, independently for each $j \in [t]$, setting s_j as follows:

$$\begin{aligned} &\text{with prob. } \frac{1}{2} && : s_j \leftarrow Unif([m']) \\ &\text{with prob. } \frac{1}{2} \cdot \frac{1 - r_j}{2} && : s_j = \text{`+'} \\ &\text{with prob. } \frac{1}{2} \cdot \frac{1 + r_j}{2} && : s_j = \text{`-'} \end{aligned}$$

    where r_j is defined to be $(1 + q_j)^{\frac{v_{max}+1}{2}} - 1$, where q_j is the jth root of the tth Chebyshev polynomial of the first kind, and thus $r_j \in (-1, v_{max})$.
  - We populate the ith row of Y_1 as follows: for each column, $k \in [n/2]$, the element at position i,k of Y_1 is set to be $\prod_{j \in [t]} X_1(s_j, k)$, where $X_1(+,k)$ and $X_1(-,k)$ are interpreted as the value $\frac{1}{\sqrt{m'}}$.
  - We populate the ith row of Y_2 as follows: for each column, $k \in [n/2]$, the element at position i,k of Y_2 is set to be $\prod_{j \in [t]} X_2(s_j, k)$, where $X_2(+,k)$ is interpreted as the value $\frac{1}{\sqrt{m'}}$, and $X_2(-,k)$ is interpreted as the value $\frac{-1}{\sqrt{m'}}$.
- Rescale each entry by $\sqrt{m'}^t / \sqrt{m''}$. Let Z_1, Z_2 be the results of applying AGGREGATE-VECTORS to the respective matrices Y_1, Y_2, with the compression parameter $\alpha := 0.01\sqrt{\epsilon}$.
- Compute the product $W := Z_1^\dagger Z_2$, (using fast rectangular matrix multiplication) and for each of the $n^{1.1}$ largest entries of W, compute the pairwise inner products of all the original columns of X that contributed to that entry (via the aggregation partition into $n^{1-\alpha}$ sets). If the maximum inner product discovered in this phase is more than $v_{max} m$, output the corresponding indices, otherwise output the indices corresponding to the pair of columns whose inner product was v_{max}.

The runtime of the above algorithm is dominated by the computation $W = Z_1^\dagger Z_2$, where Z_i are $n^{0.28} \times n^{1-\Theta(\sqrt{\epsilon})}$ matrices. For any constant ϵ, this computation can be done in time $O(n^{2-\Theta(\sqrt{\epsilon})})$, using Coppersmith's fast rectangular matrix multiplication, whose performance is summarized below:

Fact 9.12 (Coppersmith [42])**.** *For any positive $\delta > 0$, provided $\alpha < .29$, the product of an $n \times n^\alpha$ with an $n^\alpha \times n$ matrix can be computed in time $O(n^{2+\delta})$.*

The following facts concerning Chebyshev polynomials will be helpful in proving the correctness of the above algorithm:

Fact 9.13. *Letting $T_q(x) := \frac{(x-\sqrt{x^2-1})^q + (x+\sqrt{x^2-1})^q}{2}$ denote the qth Chebyshev polynomial (of the first kind), the following hold:*

- *$T_q(x)$ has leading coefficient 2^{q-1}.*
- *$T_q(x)$ has q distinct real roots, all lying within the interval $[-1, 1]$.*
- *For $x \in [-1, 1]$, $|T_q(x)| \leq 1$.*
- *For $\delta \in (0, 1/2]$, $T_q(1+\delta) \geq \frac{1}{2} e^{q\sqrt{\delta}}$.*

Proof. The first 3 properties are standard facts about Chebyshev polynomials (see, e.g. [119]). To verify the fourth fact, note that for δ in the prescribed range, $\sqrt{(1+\delta)^2 - 1} \geq \sqrt{2\delta}$, and we have the following:

$$\begin{aligned} T_q(1+\delta) &\geq \frac{1}{2}(1 + \delta + \sqrt{2\delta})^q \geq \frac{1}{2}(1 + \sqrt{2\delta})^q \\ &\geq \frac{1}{2} e^{q\sqrt{\delta}} \end{aligned}$$

□

Proposition 9.14. *Algorithm 9.11, when given as input n unit vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ and constants $\epsilon, \delta > 0$ will output a pair of indices, i, j, such that with probability at least $1 - \delta$, $\langle v_i, v_j \rangle \geq \max_{k,\ell} \langle v_k, v_\ell \rangle - \epsilon$. Additionally the runtime of the algorithm is bounded by $O(n^{2-\Omega\sqrt{\epsilon}} \log \frac{1}{\delta} + nd \log n)$.*

Proof of Proposition 9.14. We first prove the correctness of the algorithm, and then verify the runtime. If v_{max} is within ϵ from the inner product of the pair with largest inner product, then, trivially, the algorithm succeeds. If v_{max} is not within ϵ from the maximal pairwise inner product, then we claim that with probability at least .4, the algorithm will find the pair of columns of X' with *maximal* inner product (henceforth referred to as the *maximal pair*. By repeating the algorithm $\log \frac{1}{\delta}$ times, the probability of success can be increased to $1 - \delta$. By Lemma 9.10, such a pair of columns of X' will correspond to a pair of columns

in the original matrix X with inner product within $\epsilon/2$ from the maximal inner product, as desired.

For the remainder of the proof, assume that v_{max} is at least ϵ smaller than the maximal inner product. Additionally, as there can not be more than $1/\epsilon^2$ vectors that are all more than ϵ anti-correlated, we may assume with probability $1 - o(1)$ that $v_{max} > -.5$. Additionally, assume that one of the members of the maximal pair is assigned to matrix X_1, and the other is assigned to matrix X_2, which will occur with probability 1/2. Note that with probability at least $1 - o(1)$, there are at most $n^{1.1}$ pairwise inner products that are larger than $v_{\max}$, otherwise we would have seen at least one such example in our initial sample of n out of the n^2 pairwise inner products. Hence, during the VECTOR-AGGREGATION stage of the algorithm, letting $R_1 \subset [n]$ denote the sets of vectors to which the first member of the maximal pair is assigned, and R_2 is the set of vectors to which the second member of the maximal pair is assigned, with probability at least $1 - o(1)$ the only pairwise inner product between vectors in R_1 and R_2 which does not lie within the range $(-1, v_{\max})$ is that of the maximal pair.

All that remains is to verify that with probability $1 - o(1)$, the value of matrix W corresponding to the inner product between the aggregated sets R_1 and R_2 is greater than the largest value of the matrix W that corresponds to an inner product between the aggregated sets R_1', R_2' for which no pairwise inner product between vectors in R_1' and R_2' lie in the range $[-1, v_{max}]$.

Each entry of W, corresponding to the inner product between aggregated sets R_1', R_2', is simply the sum of all pairwise inner products of the transformed vectors that are assigned to those sets. By Lemma 9.4, and Fact 9.13, the expected value of the inner product between a pair of transformed vectors whose original inner product lies within $[-m, v_{max}m]$ will be at most $\frac{2}{2^{bq}}$, where q is the degree of the Chebyshev polynomial that is being used. Thus $q = t$, as defined in the algorithm description, and hence the expected inner product will be at most $\frac{2}{2^{bt}} = \frac{2}{n^{0.27/2}}$. Since $m'' = n^{0.28}$, by a union bound over standard tail bounds, with probability $1 - o(1)$, all such inner products will be at most $\frac{2}{n^{0.27/2}} + \sqrt{\frac{1}{m''}}\text{polylog } n < \frac{3}{n^{0.27/2}}$. Since sets of $n^{0.01\sqrt{\epsilon}}$ are being aggregated, the total magnitude of the sum of all these $n^{0.02\sqrt{\epsilon}}$ such inner products contributing to each entry of W will be at most $\frac{3n^{0.02\sqrt{\epsilon}}}{n^{0.27/2}}$. We now consider the contribution of the inner product corresponding to the maximal pair. By Fact 9.13, and a Chernoff bound, the expected value of this inner product will be at least

$$\frac{2e^{t\sqrt{\epsilon}}}{2^{bt}} \geq \frac{n^{.03\sqrt{\epsilon}}}{n^{0.27/2}},$$

in which case it will dominate the contribution of magnitude at most $|\frac{3n^{0.02\sqrt{\epsilon}}}{n^{0.27/2}}|$ from the other inner products that contribute to that entry of W, and thus will, with probability $1 - o(1)$, be in the top $n^{1.1}$ entries of W, in which case the algorithm will find the maximal pair in the brute-force search phase of the algorithm. □

Remark 9.15. *The above algorithm scales the roots of the Chebyshev polynomial so as to occupy the range* $[-1, v_{max}]$. *Alternatively, in the second step of the algorithm when we sampled* n *pairwise inner products, we could have also recorded the minimal pairwise inner product,* $v_{min}m$, *and then scaled things so that the roots of the Chebyshev polynomial occupied the range* $[v_{min}, v_{max}]$. *At the expense of a slightly more tedious analysis, the* $\sqrt{\epsilon}$ *in the runtime would be replaced by* $\sqrt{\frac{\epsilon}{(v_{max}-v_{min})\log\frac{1}{v_{max}-v_{min}}}}$, *solidifying the intuition that the runtime of the algorithm improves as the bulk of the pairwise inner products become concentrated about some value.*

9.5 The Approximate Closest Pair

We now return to the issue of a multiplicative $(1+\epsilon)$ guarantee on the distance, versus an additive ϵ guarantee on the inner product. We first prove the claim in the Boolean setting, and then consider the Euclidean setting. In both settings, we may assume that we know the distance between the closest pair, up to a factor of 2, by, for example, running the locality sensitive hashing algorithm of [71], which returns a $(1+\epsilon')$ factor approximate closest pair, for $\epsilon' = 1$.

Algorithm 9.16. $(1+\epsilon)$ Closest Pair: Boolean Setting
Input: An $m \times n$ matrix X with entries $x_{i,j} \in \{-1, +1\}$, $\epsilon, \delta \in (0,1)$.
Output: Two indices $c_1, c_2 \in [n]$.

- Use locality sensitive hashing to obtain $\alpha \in (0,1]$, s.t. with probability $1 - o(1)$, the Hamming distance between the closest pair of columns of X is in the interval $[\alpha m, 2\alpha m]$.
- If $\alpha \geq 1/8$, then run Algorithm 9.11 on matrix X with target accuracy $\epsilon/4$.
- Otherwise, we create the $m' \times n$ matrix X', with $m' = \frac{100\log n}{\epsilon^2}$ as follows:
  - Set $q := \frac{1}{2\alpha}$, and for each $i \in [m']$, pick $s \subset [m]$ to be a randomly selected subset of $[m]$ of size q, and for each $j \in [n]$, set $X'_{i,j} = \prod_{k \in s} X_{s,j}$.
- Run Algorithm 9.11 on matrix X' with target accuracy 0.03ϵ.
- To boost the probability of success to $1 - \delta$, repeat the above $\log\frac{1}{\delta}$ times, and output the closest pair that was found in the $\log\frac{1}{\delta}$ runs.

The following proposition asserts the correctness of the above algorithm.

Proposition 9.17. *For any constant* $\epsilon > 0$, *Algorithm 9.16, when given as input a set of* n *vectors in* $\{-1,1\}^d$, *will output a pair with Hamming distance at most a factor of* $(1+\epsilon)$ *larger than that of the minimal pair, with probability of success at least* $1 - \delta$. *Additionally, the runtime is* $O(n^{2-\Omega(\sqrt{\epsilon})}\log\frac{1}{\delta} + dn\log n)$.

Proof. In the case that $\alpha > 1/8$, the correctness follows immediately from the fact that for vectors $u, v \in \{-1, 1\}^m$, the Hamming distance $D_H(u, v) = \frac{m - \langle u, v \rangle}{2}$, and hence if the closest pair has distance at least $\alpha m = \frac{m}{8}$, an $\epsilon/4$ additive approximation of the pair with maximal inner product will yield a multiplicative approximation of the closest pair.

In the case that $\alpha < 1/8$, consider the matrix X', and assume that the first two columns are the closest pair of matrix X, and have distance βm for some $\beta \in [\alpha, 2\alpha]$. By Lemma 9.4, for each index $i \in m'$,

$$\begin{aligned} \mathrm{E}[X'_{i,1} X'_{i,2}] &= \left(\frac{\langle X_1, X2 \rangle}{m} \right)^q \\ &= (1 - 2\beta)^q = (1 - 2\beta)^{\frac{1}{2\alpha}} \in [0.05, 0.35]. \end{aligned}$$

Additionally, for any pair of columns X_j, X_k for which $D_H(X_i, X_j) \geq (1 + \epsilon)\beta m$, we have that

$$\mathrm{E}[X'_{i,j} X'_{i,k}] \leq (1 - 2(1 + \epsilon)\beta)^{\frac{1}{2\alpha}} \leq (1 - \epsilon)(1 - 2\beta)^{\frac{1}{2\alpha}}.$$

Since we chose $m' = \frac{100 \log n}{\epsilon^2}$, by a union bound over Chernoff bounds, with probability $1 - o(1)$, all pairs of columns of X' will have their expected inner products to within $\pm 0.01 \epsilon m$, and hence if a pair of columns of X' has the maximal inner product to with an additive $0.03 \epsilon m'$ they will also have the minimal distance to within a multiplicative factor of $(1 + \epsilon)$.

To conclude, note that the formation of matrix X' takes time $O(qn \log n)$, and $q \leq d$, as otherwise, the minimum distance would be 0, and the closest pair could simply be found in near-linear time. □

$(1 + \epsilon)$ Multiplicative Closest Pair: Euclidean Setting

We now show how to achieve the analog of Proposition 9.17 in the Euclidean setting. In the Euclidean setting, the "law of cosines" gives $||v - w||^2 = ||v||^2 + ||w||^2 - 2\langle v, w \rangle$, relating the distance to the inner products. This will allow us to very easily translate between multiplicative bounds on the distance and additive bounds on the inner product provided that the vectors have roughly unit length, and provided that the closest distance is not too small. Showing that we may reduce the problem to the case where all vectors lie on the unit sphere is relatively straightforward, and we accomplish it in two steps (Lemma 9.18 and Algorithm 9.19).

The second obstacle—the issue of lower bounding the minimum distance—is more tedious. In contrast to the Hamming distance setting above, we cannot assume any lower bound on the minimum possible distance. In particular, in the previous section, we "XORed" sets of size $q = \frac{1}{\alpha}$, where α lower bounded the minimal distance. If we attempted to do the same XORing trick here, in the case that $q > n$, we would end up with n^2 runtime simply computing the new matrix that will be input to the additive approximation algorithm. Thus we first show how to solve the problem for $\alpha > \frac{1}{n^{0.9}}$. If $\alpha \leq \frac{1}{n^{0.9}}$, then the closest pair is extremely close, and, intuitively, we should be able to find it with a divide-and-conquer

approach, since either there are only a few pairs of points that are very close in which case we can easily pick them off, or there is a large set of points that are all extremely close in which case we will be able to subtract off their mean, and rescale that cluster of points; such a transformation has the property that it effectively increases the minimal distance by a factor of $1/z$, where z is the diameter of the cluster of points.

We begin by showing how to reduce the problem to the case in which all input vectors have lengths that lie in the interval $[1, 2]$.

Lemma 9.18. *For any constant $c > 1$, given an algorithm with runtime $O(n^c)$ for computing the $(1 + \epsilon)$ approximate closest pair for sets of n vectors $v_1, \ldots, v_n$ having the property that for all i, $||v_i|| \in [1, 2]$, one can obtain an algorithm for the general setting with runtime $O(n^c \log n)$.*

Proof. Assume without loss of generality that the n vectors $v_1, \ldots, v_n$ have lengths $r_1 \leq \ldots \leq r_n$. We now iteratively define several sets of vectors, $S_1, S_2, \ldots$: let $S_1 := \{i : r_i < r_2 + 2r_1\}$. Given the set $S_i \subset [n]$, we define S_{i+1} as follows: let $j = \min(S_i)$, and let $k := \min\{\ell : r_\ell > r_j + r_{j+1}\}$, and define $S_{i+1} := \{\ell : r_\ell \in [r_k, r_k + 2r_{k+1}]\}$. By the triangle inequality, the pair of vectors of minimal distance lie within one set, S_i; additionally, each vector v_j is contained in at most two sets, S_i, S_{i+1}, for some i. Thus given any algorithm for solving the approximate closest pair problem on a given set of vectors $w_1, \ldots, w_m$, with $||w_i|| \leq ||w_{i+1}||$, such that $||w_m|| \leq ||w_1|| + 2||w_2||$, one can obtain an algorithm for the $(1 + \epsilon)$ approximate pair on an arbitrary set of vectors with comparable runtime by computing the above sets S_i, and running the algorithm on each set, and then comparing the outputs of the various runs (using the fact that we can boost the probability of failure of a given run to $1 - 1/n^2$ at the expense of a multiplicative factor of $O(\log n)$ in the runtime).

Finally, for any set S_i, for all $j, k \in S_i$, $\frac{||r_j||}{||r_k||} \leq 2$, except when $j = \min(S_i)$. Given an algorithm for solving the approximate closest pair problem on a set of vectors $w_1, \ldots, w_m$, with $||w_i||/||w_j|| < 2$ for all i, j, we can obtain an algorithm with comparable runtime for the setting in which $||w_i||/||w_j|| < 2$ with the exception of a single value of i by simply explicitly computing the distances $||w_i = w_j||$ for all values of j, and then using the algorithm on the set of vectors with w_i removed. □

We now describing a procedure for splitting the problem into a series of problem that each only involve vectors of unit norm.

Algorithm 9.19. STANDARDIZE
Input: An $m \times n$ matrix X with entries $x_{i,j} \in \mathbb{R}$, with all columns having norm in the interval $[1, 2]$, constant $\epsilon \in (0, 1)$.
Output: A series of matrices $Y_1, \ldots, Y_t$, with each column of Y_i having unit norm. Y_i is has dimensions $m' \times n_i$ for $m' = \frac{100 \log n}{\epsilon^2}$, and $\sum_i n_i \leq 2n$.

- *Use locality sensitive hashing to obtain $\alpha \in (0, 1]$, s.t. with probability $1 - o(1)$, the Euclidean distance between the closest pair of columns of X is in the interval $[\alpha, 2\alpha]$.*
- *Sort the columns of X in terms of increasing norm. Form sets of columns of X s.t. the following properties hold:*
  - *For each set, all columns contained in that set have norm that falls within some interval of length 4α.*
  - *For each column, i, there is some set containing i and all columns whose norm differ from that of i by at most 2α.*
  - *Each column occurs in at most 2 sets.*

  (Note that such a collection of sets can be computed in linear time, and that the closest pair of vectors is contained in one of the sets.)
- *We iterate over the sets: For the ith set set, containing n_i columns, let X_i represent the $m \times n_i$ matrix of columns of one of the sets.*
- *Choose a random random Gaussian vector of covariance $\frac{100}{\epsilon^2}$, and add it to all vectors in X_i, then perform a Johnson--Lindenstrauss transformation of them into dimension $\frac{100 \log n}{\epsilon^2}$, and normalize them so as to have unit norm, to yield matrix Y_i.*

The following lemma characterizes the performance of the above algorithm:

Lemma 9.20. *Given an algorithm for finding the $1 + \epsilon/2$ approximate closest pair for unit vectors, with runtime $O(n^c)$, for some $c > 1$, there exists an algorithm for finding the $1 + \epsilon$ approximate closest pair for arbitrary vectors with runtime $O(n^c)$.*

Proof. Given an algorithm for finding the $1 + \epsilon$ approximate closest pair for unit vectors, we take our set of arbitrary vectors, run Algorithm 9.19 to yield k sets of unit vectors, and then iterate over all sets: for the ith set, containing n_i vectors, if $n_i < n^{0.9}$, then we simply do a brute-force-search for the closest pair among that set. If $n_i \geq n^{0.9}$, we run the hypothetical algorithm for closest pair on these n_i unit vectors. We will then output the closest pair from among the k sets. Since the total number of columns of the sets is at most $2n$, the runtime guarantee follows.

We now guarantee that the $(1 + \epsilon)$ approximate closest pair of a set X_i corresponds to a $(1 + \epsilon)$ approximate closest pair of the corresponding set of vectors of X. We now explain the motivation for adding a random Gaussian vector to all vectors, performing a Johnson–Lindenstrauss transformation, then normalizing the vectors so as to have unit norm. We

would like to simply normalize the vectors; the difficulty, however, is that there might be several pairs of vectors that are nearly collinear; the difference in lengths can be as much as 4α, and we are trying to obtain an additive $\alpha\epsilon$ approximation of the closest pair, and thus we cannot afford to simply renormalize the vectors and risk losing an additive α factor. Instead, we add a random Gaussian of large covariance, $O(1/\epsilon^2)$, which has the effect of preserving distances, while ensuring that with high probability, for every pair of vectors, their difference is nearly orthogonal to the vectors. Thus when we normalize the vectors, the small distances are not distorted by much. Specifically, with probability $1 - o(1)$, the normalizing distorts the distances between vectors by at most a multiplicative factor of $(1 + \epsilon/2)$. □

After reducing the problem to the setting where all vectors have unit norm, we now describe two algorithms that will apply to two different regimes of the distance of the closest pair. The first algorithm applies when the minimum distance is at at most $1/n^{0.9}$. The second algorithm applies when the minimum distance can be arbitrarily small.

The first algorithm applies the same XORing approach of Algorithms 9.16. The slight subtlety is that we need to ensure that the inner products between the columns of the matrix that results from the XORing will be concentrated about their expectations, hence we essentially switch to the Boolean setting before XORing. As throughout, we favor clarity of exposition over optimizing the constants.

Algorithm 9.21. $(1+\epsilon)$ Closest Pair: Euclidean Setting (large α)
Input: An $m \times n$ matrix X with entries $x_{i,j} \in \mathbb{R}$, where all columns have unit norm.
Output: Two indices $c_1, c_2 \in [n]$.

- Use locality sensitive hashing to obtain $\alpha \in (0, 1]$, s.t. with probability $1 - o(1)$, the Euclidean distance between the closest pair of columns of X is in the interval $[\alpha, 2\alpha]$.
- If $\alpha < 1/n^{0.9}$, run Algorithm 9.23. If $\alpha > 0.1$, apply Algorithm 9.11.
- Otherwise, perform a Johnson--Lindenstrauss transformation to X to yield a $d \times n$ matrix Y, with $d = \frac{1000 \log n}{\epsilon^2}$.
- Define the $d \times n$ matrix Z as follows: for each $i \in [d]$, select a set of $q = \lfloor \frac{\pi}{2\alpha} \rfloor$ uniformly random unit vectors $v_1, \ldots, v_q$ and for all $j \in [n]$, set

$$z_{i,j} = sign\left(\prod_{k=1}^{q} Y_j^t v_k\right).$$

- Apply Algorithm 9.11 with error parameter $\epsilon/1000$ to the matrix Z after scaling all entries by $1/\sqrt{d}$ so as to make them have unit norm.

The following proposition characterizes the performance of the above algorithm.

Proposition 9.22. *For any constant $\epsilon > 0$, Algorithm 9.21, when given input n unit vectors in $\mathbb{R}^m$ whose closest pair have distance at least ϵ, will output a pair with Euclidean distance*

at most a factor of $(1+\epsilon)$ *larger than that of the minimal pair, with probability of success* $1-o(1)$. *Additionally, the runtime is* $O(n^{2-\Omega(\sqrt{\epsilon})} + mn\log n)$.

Proof. With probability $1-o(1)$, the initial Johnson–Lindenstrauss step preserves all distance to within multiplicative factors of $1+\epsilon/100$ (see, e.g. [44] for an elementary proof). Assuming this holds, consider the formation of matrix Z, which takes time at most $O(qn\log^2 n) < O(n^{2-\Omega(\sqrt{\epsilon})})$. Consider two columns Y_i, Y_j that form an angle of β (and hence have distance $\sqrt{(1-\cos\beta)^2+\sin^2\beta}$). For each random vector v, we have that $\mathrm{E}[sign(Y_i^t v \cdot Y_j^t v)] = 1-2\frac{\beta}{\pi}$, and since expectations of independent random variables multiply, we have that for each k,

$$\mathrm{E}[z_{k,i}z_{k,j}] = \left(1-\frac{2\beta}{\pi}\right)^q.$$

Consider the pair of columns of X with minimal distance $\delta^* \in [\alpha, 2\alpha]$, and hence form an angle $\beta^* \in [\alpha/2, 2\alpha]$, in which case the expected Hamming distance between the corresponding columns of Z is at most $d\left(1-\frac{\alpha}{\pi}\right)^{\pi/2\alpha} \leq 0.65d$, and is at least $d\left(1-\frac{4\alpha}{\pi}\right)^{\pi/2\alpha} \geq 0.1d$. By the same reasoning, the image of any pair of columns of X whose distance was a multiplicative factor of at least $(1+\epsilon)$ larger than δ^* will have expected distance at least a multiplicative factor of $(1+\epsilon)$ larger than that of the image of the pair with distance δ^*. By a union bound over Chernoff bounds, with probability $1-o(1)$ the distance between any two columns of Z differs from its expectation by at most $d\epsilon/100$. Hence with probability $1-o(1)$, very crudely, any pair of columns of Z whose inner product is within an additive $d\epsilon/1000$ from that of the maximal inner product, will correspond to a pair of columns of the original matrix X whose distance is within a multiplicative factor of $1+\epsilon$ from the minimal distance δ^*, and hence the result of applying Algorithm 9.11 to the scaled matrix Z will be satisfactory. □

Finally, we address the setting in which the closest pair might be extremely close, having distance $< \frac{1}{n^{0.9}}$. Here, the difficulty is that we cannot XOR sufficiently large sets without spending super-quadratic time on the XORing step. The idea here is that if the minimum distance is so small, then we can recursively divide the set of points into small clusters, that are all far apart. If all clusters are small, then we can trivially find the closest pair by brute force search with each cluster. If we have a large cluster (with tiny diameter), then we can can simply subtract off the mean of the cluster; after re-normalizing via the Standardization algorithm, Algorithm 9.19, and the procedure of Lemma 9.18, the resulting points will have unit norm, and the smallest distance will have increased to at least $1/n^{0.8}$, and we can apply the above algorithm.

Algorithm 9.23. $(1+\epsilon)$ Closest Pair: Euclidean Setting (small α)
Input: An $m \times n$ matrix X with $m = \frac{100 \log n}{\epsilon^2}$, and entries $x_{i,j} \in \mathbb{R}$, with all columns having unit norm. Constant $\epsilon \in (0,1)$.
Output: Two indices $c_1, c_2 \in [n]$.

- *Let v_i denote the ith column of X. Use locality sensitive hashing to obtain $\alpha \in (0,1]$, s.t. with probability $1 - o(1)$, the Euclidean distance between the closest pair of columns of X is in the interval $[\alpha, 2\alpha]$.*
- *If $\alpha > 1/n^{0.9}$, run Algorithm 9.21.*
- *We now recursively split up the vectors:*
  - *Project all vectors onto an m dimensional Gaussian of unit covariance, sort the resulting projections, $x_1 \leq \ldots \leq x_n$, wlog x_i is the proj. of v_i.*
  - *We now traverse the list: we ''pause'' at some x_i, if there are fewer than $n^{0.6}$ points with projected value in the interval $[x_i - 2\alpha, x_i]$. If we have ''paused'' at x_i we do one of two prodecures:*
  - *if $|\{j : x_j < x_i\}| \leq n^{0.9}$:*
    - *Brute force search for the closest pair of points from the set $\{v_j : x_j < x_i\}$. Store the closest pair and their distance, and remove all points v_j (and their projections x_j) for which $x_j \leq x_i - 2\alpha$ from all further computations. Continue traversing the list (with those points removed).*
  - *if $|\{j : x_j < x_i\}| > n^{0.9}$:*
    - *Save set $S_i := \{v_j : x_j \leq x_i\}$, and continue traversing the list with all points v_j s.t. $x_j \leq x_i - 2\alpha$ removed.*
  - *Having finished traversing the list, if we have not stored any sets S_i, then we can simply compare the stored closest pair distances, and output the minimum. Otherwise, let $S_1, \ldots, S_k$ denote the sets that are stored. For each set $S = S_i$:*
    - *Points in S had projections x_i in sets of contiguous intervals of width 2α; each interval had $\geq n^{.6}$ points, hence all x_i are within $2\alpha n^{.4}$.*
    - *Choose $\sqrt{n}$ random pairs of vectors from S, and compute their distances. Let μ be the median of these $\sqrt{n}$ distances. If $\mu > \alpha n^{0.6}$, then the fact that these vectors were grouped together is a ''fluke'', and recurse the partitioning procedure on this set.*
    - *Otherwise, randomly select $v \in S$, sample $n^{0.1}$ distances between v and randomly chosen $v' \in S$; repeat until one has found a v that has distance at most $2\alpha n^{0.6}$ from at least $1/4$ of the points in S.*
    - *Let $0 = d_1 \leq \ldots \leq d_{|S|}$ be the distances beween v and all points in S. Find $c \in [2\alpha n^{0.6}, 4\alpha n^{0.6}]$ s.t. $|\{i : d_i \in [c, c+2\alpha]\}| < n^{0.1}$, and construct the sets $T := \{v_i : d_i < c + 2\alpha\}$, and $T' := \{v_i : d_i > c\}$. Since $|T'| < \frac{3}{4}|S|$, we recurse on set T'.*
    - *Subtract v from all vectors in T (and remove v) and run the standardization procedure of Lemma 9.18 and Algorithm 9.19 on T. (Since all points in T were distance at most $4\alpha n^{0.6}$ from v, after subtracting off v, and re-standardizing so as to be unit vectors the distance of the closest pair will have increased by a factor of $\Omega(\frac{1}{\alpha n^{0.6}})$, and hence will be at least $\Omega(\frac{1}{n^{0.6}}) >> \frac{1}{n^{0.8}}$.)*
    - *Run Algorithm 9.21 on the standardized version of T.*
- *Aggregate the closest pairs returned by all branches of the algorithm, and return the closest.*

Proposition 9.24. *In each run of Algorithm 9.23, either the closest pair is output, or the algorithm recurses, and with probability at least* 0.2 *the number of points in each set considered decreases by a factor of at least* 1.1.

Proof. Each time that the algorithm "pauses" at a projected value x_i, if no set is saved during that pause, then a brute-force-search is performed on at most $n^{0.9}$ vectors, which are then removed from all subsequent computation. If the closest pair involves one of those points, then we will find it, since our Gaussian projection typically shrinks distances (by a factor of $1/\sqrt{m}$), and only inflates distances with probability $1 - o(1/n)$.

If a set S is "saved", then the vectors correspond to a set of vectors that ended up unusually close together in the projection. In particular, we expect $O(\alpha n\sqrt{m}) < n^{0.2}$ projections in each interval of length 2α, yet each interval that contributed to S contained at least $n^{0.6}$ projections.

If this is deemed a "fluke": i.e. a random sample of $\sqrt{n}$ pairwise distances have median at least $\alpha n^{0.6}$, then with probability $1 - o(1/n)$, at least 1/3 of the $|S|^2$ pairwise distances are larger than $\Omega(\alpha n^{0.6}\sqrt{m})$, in which case in the random Gaussian projection performed in the next round of the algorithm, with probability at least 1/5, more than 1/6 of these projected pairwise distances will be at least $\Omega(\alpha n^{0.5})$, and each such pair of points must end up in disjoint sets in the partitioning algorithm, as there will be one index "paused" on in an interval between their projections (otherwise there would be $> n^{1.1}$ total points). Hence with probability at least 1/5, the set will end up being partitioned into smaller sets, as claimed.

If the set S is not deemed a fluke: i.e. a random sample of $\sqrt{n}$ pairwise distances have median at most $\alpha n^{0.6}$, then with probability $1 - o(1/n)$, the attempt to find a vector v that has distance at most $2\alpha n^{0.6}$ from at least a quarter of the points in S will be successful, and will take time $O(|S|)$. Since all points in T were distance at most $4\alpha n^{0.6}$ from v, after subtracting off v, and re-standardizing so as to be unit vectors the distance of the closest pair will have increased by a factor of $\Omega(\frac{1}{\alpha n^{0.6}})$, and hence will be at least least $\frac{1}{n^{0.9}}$, and hence we can apply Algorithm 9.21 to yield a $(1+\epsilon)$ multiplicative factor approximate closest pair. □

9.6 Further Directions: Beyond Fast Matrix Multiplication

Beyond the more obvious open questions posed by these improved algorithms, one very relevant direction for future work is to give algorithms with subquadratic asymptotic runtimes that improve over the brute-force search in practice for modest-sized datasets. For instance:

> *Does there exist an algorithm for finding a pair of* 0.05*-correlated Boolean vectors from among* $n = 100,000$ *uniformly random Boolean vectors that significantly beats brute-force-search,* in practice*?*

There are two natural approaches to this question. The first is to try to improve practical fast matrix multiplication implementations. While the algorithms of this chapter (as well as

the next chapter) rely on fast matrix multiplication, they do not require an especially accurate multiplication. In particular, our algorithms would still succeed if they used a noisy matrix multiplication, or even an algorithm that "misplaced" a constant fraction of the cross-terms. (For example, for $n \times n$ matrices A, B, C, in computing $AB = C$, the entry $c_{i,j}$ should be the sum of n cross terms $a_{i,k} \cdot b_{k,j}$; our algorithms would be fine if only, say, half of these cross terms ended up contributing to $c_{i,j}$.) Tolerating such "sloppiness" seems unlikely to allow for faster asymptotic bounds on the runtime (at least within the Coppersmith–Winograd framework), though it may significantly reduce the overhead on some of the more practically expensive components of the Coppersmith-Winograd framework.

The second approach to yielding a practical algorithm would be to avoid fast matrix multiplication entirely. Our Expand and Aggregate algorithm seems natural (if many pairwise inner products are extremely small, we should "bucket" them in such a way that we can process them in bulk, yet still be able to detect which bucket contains the large inner product). Nevertheless, if one replaces the fast matrix multiplication step with the naive quadratic-time multiplication, one gets no improvement over the quadratic brute-force search. It seems that no clever bucketing schemes (in the "aggregation" step, one need not simply add the vectors over the reals...), or fancy embeddings can remove the need for fast matrix multiplication.

One intuitive explanation for the difficulty of avoiding fast matrix multiplication is via the connection between finding correlations, and learning parity with noise. The statistical query (SQ) lower bound of Blum et al. [28], informally, implies that any algorithm that will beat brute-force-search must be highly non-SQ; in particular, it must perform nontrivial operations that intertwine at least $\log n$ rows of the matrix whose columns are the given vectors. Fast matrix multiplication is clearly such an algorithm.

Given this intuitive need for a non-SQ algorithm, perhaps the most likely candidate for an off-the-shelf algorithm that might replace fast matrix multiplication, is the Fast Fourier Transform. In a recent paper, Pagh gives an extremely clean (and practically viable) algorithm for computing [or approximating] the product of two matrices given the promise that their product is *sparse* [or has small Frobenius norm after one removes a small number of large entries] [99]. The algorithmic core of Pagh's approach is the computation of a Fourier transform. Perplexingly, despite the fact that Pagh's results specifically apply to the type of matrix products that we require for our algorithms that find correlations and parities, it does not seem possible to improve on the trivial brute-force search runtimes by using Pagh's matrix multiplication algorithm.

Chapter 10

Learning Parities and Juntas

An *example* (x, y) from an (n, k, η)-*instance* of parity with noise, consists of $x \in \{-1, +1\}^n$, chosen uniformly at random, together with a *label* $y \in \{-1, +1\}$ defined by $y = z \cdot \prod_{i \in S} x_i$, where $z \in \{-1, +1\}$ is chosen independently of x to be -1 with probability $\eta \in [0, 1/2)$, for some fixed set $S \subset [n]$ with $|S| = k$. The problem of *learning parity with noise* is the problem of recovering the set S from a set of examples. One reason for considering this specific noisy recovery problem is that it seems to be the most difficult, in the following sense: Given any set $S' \not\supset S$, the distribution of the values indexed by elements of S' and the label will be uniformly random elements of $\{-1, 1\}^{|S'|+1}$. Additionally, given any set $S' \neq S$, the corresponding Fourier coefficient—the expected correlation between the label and $\prod_{i \in S'} x_i$, will be zero. Thus this problem of recovering S is the epitome of a needle-in-a-haystack problem: if one finds the needle, it is obvious that one has found the needle, but it is not clear whether one can glean any sense of whether a guess is "close" to the right answer. While this problem of finding a needle in a seemingly structureless haystack should be reminiscent of many of the classical "hard" problems of computer science, it seems difficult to show that this problem is NP-hard, at least in part because the randomness of the examples render every instance equally difficult.

In the case that the *noise rate* $\eta = 0$, by translating the entries of the examples from being in $\{-1, 1\}$ to being elements of $\mathbb{F}_2$, this problem of recovering the set S is simply the task of solving a linear system of equations over $\mathbb{F}_2$, since the dot product (over $\mathbb{F}_2$) of each example with the indicator of S will yield the label. Such a linear system can trivially be solved in time $O(n^3)$ via Gaussian elimination, irrespective of $k = |S|$.

In contrast to the setting of solving systems of noisy linear equations over the real numbers, there is no easy least squares regression algorithm over finite fields. For even a small positive noise rate, $\eta > 0$, the complexity of this problem seems to change drastically; algorithms such as Gaussian elimination will no longer work, as they proceed by adding and subtracting examples from other examples, and the noise in the labels of the corrupted examples will thereby be spread throughout the set of examples until there is essentially no signal left in the final output of the algorithm.

It is worth stressing that the difficulty of this problem is strictly computational. From an

information theoretic standpoint, the addition of a small amount of noise does not change the problem significantly—given $O(n)$ examples, Chernoff bounds yield that with overwhelming probability, the true parity set S will be the only set for which the product of the corresponding indices correlates significantly with the labels.

10.1 The History of Parity with Noise

Interest in the problem of learning parity with noise was sparked by the results of Blum et al. [27], who first showed that there exists a class of functions that can be learned in polynomial-time with a constant amount of random classification noise, but which, provably, cannot be learned in the *statistical query* (SQ) learning model. The SQ learning framework, introduced by Kearns in 1993 [79], sought to abstract and formalize the restricted manner in which many types of learning algorithms interact with data. Specifically, given a distribution over labelled examples, an SQ algorithm interacts with the data via the following protocol: it describes a function, f_1 from an example/label pair to $\{0, 1\}$, and then receives the average value of that function over the examples, with the addition of a small amount of (potentially adversarial) noise. The algorithm then produces a second query, f_2, and receives a perturbed expectation of that function, and so on. This framework captures many learning algorithms: stochastic gradient descent, the perceptron algorithm, etc. The salient feature of all SQ algorithms, is that because they only interact with the data via receiving noisy expectations, they are robust to modest amounts of random classification noise. Intuitively, the main limitation of SQ algorithms is that they can not interact directly with the data, precluding algorithms such as Gaussian elimination which seem to require access to the actual data points.

Blum et al. [27] showed that parity functions on $O(\log n \log\log n)$ bit strings with constant noise rate can be learned in polynomial time, whereas the earlier results of Blum et al. [28] imply that any SQ algorithm provably requires a super-polynomial number of queries (provided the noise rate of each query is at least inverse polynomial). Phrased differently, they presented an algorithm for learning parity with noise over n bit strings, with runtime $2^{O\left(\frac{n}{\log n}\right)}$, whereas any SQ algorithm provably required runtime $2^{\Omega(n)}$.

Their algorithm proceeds by obtaining a huge number of examples, $2^{O\left(\frac{n}{\log n}\right)}$, and then performs a sort of "block" Gaussian elimination in which the vast number of examples is leveraged to ensure that sets of no more than $O(\sqrt{n})$ examples are added together, as opposed to $O(n)$ that would occur in typical Gaussian elimination. This reduction in the number of examples that are added together implies that the level of noise in the output (which increases geometrically with every additional addition of an example), is significantly reduced, allowing for a slightly sub-exponential algorithm.

This algorithm prompted several other works, including work by Lyubashevsky [83], who showed that a similar approach could be applied to a much smaller set of examples ($n^{1+\epsilon}$) and still obtain a sub-exponential, though slightly larger, runtime of $2^{O\left(\frac{n}{\log\log n}\right)}$. The algorithm

of Blum et al. was also shown to have applications to various lattice problems, including the shortest lattice vector [4].

The assumption that the noise in each example's label is determined independently seems crucial for the hardness of learning parity with noise. In the case in which noise is added in a structured manner—for example, if examples arrive in sets of three, with the promise that *exactly* one out of each set of three examples has an incorrect label, the recovery problem can be solved in polynomial time, as was shown by Arora and Ge [11].

More recently, with the surge of attention on lattice problems prompted by the development of lattice-based cryptosystems, there has been much attention on the related problem of *learning with errors* (LWE). The LWE problem, introduced by Regev in 2005 [109], corresponds to the problem of learning parity with noise with two modifications: instead of working over $\mathbb{F}_2$ the LWE is over a larger finite field, and every example is perturbed by adding a small amount of (discrete) Gaussian noise. One of the attractions of basing cryptosystems on the LWE problem is that it has been shown to be as hard as the *worst–case* hardness of lattice problems such as GapSVP (the decision variant of the shortest lattice vector problem), and SIVP (the shortest independent vectors problem) [109, 106]. See [110], for a relatively recent survey on LWE. There are no known such hardness reductions for learning parity with noise.

Sparse Parities and Juntas

The results of this thesis will be concerned with the problem of learning *sparse* parities with noise. Specifically, this is the problem of learning parities with noise in the special case when the size of the parity set $k = |S|$ is known to be very small. Such a restriction clearly makes the problem easier, as one could simply perform a brute-force search over all $\binom{n}{k} \approx n^k$ sets of k indices. In light of the subexponential algorithm of Blum et al. [27] for learning large parities, it is tempting to hope that analogous savings over the brute-force approach can be achieved in the sparse setting, perhaps yielding an $n^{o(k)}$ algorithm, though no such algorithm is known, and adapting the approach of Blum et al. to the sparse setting seems problematic.

This problem of learning sparse parities is especially relevant to learning theory, as several other basic problems in learning theory have been reduced to it. In 2006, Feldman et al. [54], showed that algorithms for learning k-sparse parities with noise can be used to learn k-juntas—functions from $\{0,1\}^n \to \{0,1\}$ which only depend on the values of $k << n$ of the indices (see definition 10.1)—and learning 2^k-term DNF, from uniformly random examples.

The reductions of Feldman et al. transform instances of k-juntas or 2^k-term DNF into instances of parity with noise, with a parity of size $\leq k$, by adding some specially designed extra noise, which zeros out nearly all the heavy Fourier coefficients of the juntas or DNF. With some reasonable probability, however, exactly one heavy Fourier coefficient will remain, in which case this process has created an instance of parity with noise. It is worth stressing that such a transformation adds a large amount of noise—corresponding to noise rate $\eta = \frac{1}{2} - \frac{1}{2^k}$, thus motivating the development of algorithms for sparse parity with noise that are

very noise robust; for example, algorithms whose runtimes depend only as $poly(\frac{1}{1/2-\eta})$, as opposed to having the noise rate in the exponent of n.

For completeness, in Appendix B.1 we include formal statements of the reductions of Feldman et al. [54], which we use to obtain improved algorithms for learning k-juntas and DNF from our algorithm for learning parities. We now briefly summarize the previous algorithmic work on learning sparse parities and k-juntas.

For the problem of learning k-sparse parities with noise, in a recent paper, Grigorescu et al. [62] adapt the approach of Hopper and Blum [67] to the noisy setting to give an algorithm that runs in time $poly(\frac{1}{1-2\eta})n^{(1+2\eta)^2+o(1))k/2}$. In particular, as the noise rate goes to 0, the performance of this algorithm tends to $O(n^{k/2})$, and as the noise rate tends towards $\frac{1}{2}$, the dependency on n tends towards $O(n^k)$.

For the problem of learning juntas over the uniform distribution, Mossel et al. [92] show that size k juntas can be learned *in the absence of noise*, in time $n^{\frac{\omega k}{\omega+1}}poly(2^k) \approx n^{0.70k}poly(2^k)$. This result leverages a powerful characterization of k-juntas: in particular, they show that any k-junta either has a nonzero Fourier coefficient of degree at most d, or, when regarded as a polynomial over $\mathbb{F}_2$, the k-junta has degree at most $k-d$. Their result follows from balancing a brute-force search for low-degree Fourier coefficients, with solving a large system of linear equation (using fast matrix multiplication) to find the low-degree representation over $\mathbb{F}_2$ in the case that the brute-force search did not find any heavy Fourier coefficients. As this approach involves solving a large system of linear equations, the assumption that there is no noise is necessary. In particular, for constant noise η, prior to the results of this dissertation, no algorithm for learning k-juntas with noise $\eta > 0$ running in time $O(n^{ck})$ for any constant $c < 1$ was previously known.

For the problem of (ϵ, δ) PAC-learning s-term DNF under the uniform distribution, the results of Grigorescu et al. [62] imply a runtime of

$$poly(\log \frac{1}{\delta}, \frac{1}{\epsilon}, s)n^{(1-\tilde{O}(\epsilon/s)+o(1))\log \frac{s}{\epsilon}},$$

which improves upon the $O(n^{\log \frac{s}{\epsilon}})$ of Verbeurgt [135] from 1990.

10.2 Summary of Approach and Results

The problem of finding a ρ-correlated pair of Boolean vectors from among n random vectors is easily seen to be equivalent to solving the parity with noise problem, in the special case that the size of the true parity set is $k = 2$; the correspondence between the correlation ρ and noise rate η is given by $\eta = 1/2 - \rho/2$. To see one direction of the equivalence, note that given an instance of such a parity with noise problem, if one removes all examples that have label 1, one will be left with a set of examples in which the two true parity indices are correlated. One could thus use the algorithm of Proposition 9.3 to find the pair of parity indices in time $n^{\frac{5-\omega}{2(4-\omega)}k}poly(\frac{1}{1/2-\eta}) \approx n^{1.62}poly(\frac{1}{1/2-\eta})$, where $\omega < 2.38$ is the exponent of matrix multiplication.

In general, given an algorithm for solving the parity with noise problem for parities of some fixed size c in time $O(n^\alpha)$, one may attempt to adapt it to obtain an algorithm for the parity with noise problem for parities of any value $k > c$ that runs in time $O(n^{k\frac{\alpha}{c}})$ by performing the following transformation: for each length n example with label ℓ, transform it into a length $N = \binom{n}{k/c} \approx n^{k/c}$ example with label ℓ, where each index represents the XOR (or product in the ± 1 setting) of some set of k/c of the indices of the original example. If the original set of examples contained a set of k indices whose XOR is correlated with the labels, then the transformed examples will contain (several) sets of c indices whose XOR is correlated with the labels. One can now simply apply the original algorithm for finding parities of size c to the transformed set of examples, to yield a runtime of $O\left((n^{k/c})^\alpha\right)$. The minor difficulty, of course, is that the transformed examples are no longer uniformly random bit strings, though most algorithms should be robust to the type of dependencies that are introduced by this transformation.

The above transformation motivates the search for improved algorithms for finding small constant–sized parities ($k = 2, 3, 4, \ldots$). Given the existence of a subquadratic time algorithm for the case $k = 2$, a natural hope is that one can design better and better algorithms for larger k, perhaps with the eventual hope of yielding an $n^{o(k)}$ algorithm.

While an extension of the algorithm of Proposition 9.3 (corresponding to $k = 2$) would yield an algorithm for learning k-sparse parities with runtime

$$n^{\frac{5-\omega}{2(4-\omega)}k} poly(\frac{1}{1/2-\eta}) \approx n^{0.81k} poly(\frac{1}{1/2-\eta}),$$

we instead consider the $k = 3$ case, and obtain an exponent of $< 0.80k$. While the constant in the exponent is only 0.02 better than what is yielded from leveraging the $k = 2$ case implied by the results of Chapter 9, this alternate approach may be of independent interest.

Theorem 10.1. *For any fixed $\epsilon > 0$, for sufficiently large n and k, given examples from an (n, k, η) instance of parity with noise, with probability $1 - o(1)$, our algorithm will correctly return the true set of k parity bits. Additionally, the algorithm will run in time*

$$n^{\frac{\omega+\epsilon}{3}k} poly(\frac{1}{1-2\eta}) < n^{0.80k} poly(\frac{1}{1-2\eta}).$$

The above theorem has immediate implications for the problems of learning juntas and DNF:

Definition 10.1. *An* example (x, y) *from a* (n, η)-instance *of a noisy k-junta consists of $x \in \{-1, +1\}^n$, chosen uniformly at random, together with a* label $y \in \{-1, +1\}$ *defined by $y = z \cdot f(x_S)$, where $z \in \{-1, +1\}$ is chosen independently of x to be -1 with probability η, f is a fixed though unknown function $f : \{-1, +1\}^k \to \{-1, +1\}$, and x_S denotes the indices of x occurring in a fixed (though unknown) set $S \subset [n]$ with $|S| = k$.*

The above theorem together with Theorem B.1 immediately imply the following corollary:

Corollary 10.2. *For sufficiently large n and k given access to examples from an (n, η) instance of a noisy k-junta, with constant probability our algorithm will correctly return the true set of $k' \leq k$ relevant indices, and truth table for the function. Additionally, the algorithm has runtime, and sample complexity bounded by*

$$n^{\frac{\omega+\epsilon}{3}k} poly(\frac{1}{1-2\eta}) < n^{0.80k} poly(\frac{1}{1-2\eta}).$$

The above theorem together with Corollary B.1 yields the following corollary for learning juntas without noise, where the exponent is obtained by setting $\alpha = \frac{3}{4}$ in the statement of Corollary B.1 so as to equate the two arguments of the max operation:

Corollary 10.3. *For sufficiently large n and k given access to examples from an (n, η) instance of a noisy k-junta with $\eta = 0$, with constant probability our algorithm will correctly return the true set of $k' \leq k$ relevant indices, and truth table for the function. Additionally, the algorithm has runtime, and sample complexity bounded by*

$$n^{\frac{\omega+\epsilon}{4}k} poly(n) < n^{0.60k} poly(n).$$

Definition 10.4. *An* example (x, y) *from a r-term DNF over n bits under the uniform distribution consists of $x \in \{-1, +1\}^n$, chosen uniformly at random, together with a* label $y \in \{-1, +1\}$ *given by a fixed (though unknown) r-term DNF applied to x.*

The following corollary follows from first arguing that an analog of Theorem 10.1 holds (Theorem 10.3) in which the sample complexity has been reduced, and then applying Theorem B.2.

Corollary 10.5. *For sufficiently large n and k, there exists an algorithm that (ϵ, δ)–PAC learns r-term DNF formulae over n bits from uniformly random examples that runs in time*

$$poly\left(\frac{1}{\delta}, \frac{r}{\epsilon}\right) n^{0.80 \log_2 \frac{r}{\epsilon}}.$$

A Little Bias Goes a Long Way

As with the results in Chapter 9, our $k = 3$ algorithm uses fast matrix multiplication to find a pair of correlated vectors. The crux of the approach is that a parity function has reasonably heavy low-degree Fourier coefficients if one changes from the uniform distribution over the Boolean hypercube to a slightly biased product distribution. The required bias is very small, thereby allowing one to efficiently subsample a set of uniformly random examples so as to produce a set of examples with the desired bias. In the remainder of this section we describe the main idea behind the algorithm.

Given an example x, y, for $x = (x_1, \ldots, x_n)$ from an $(n, 3, \eta)$-instance of parity with noise (with three parity bits), for any i, $\Pr[x_i = 1 | y = 1] = 1/2$. Similarly, $\Pr[x_i x_j = 1 | y = 1] = 1/2$ for distinct $i, j \in [n]$. The improved algorithm for finding parities rests on the following

observation about parity sets of size $k = 3$: if the bits of x are not chosen uniformly at random, but instead are chosen independently to be 1 with probability $\frac{1}{2} + \alpha$, for some small bias α, then the above situation no longer holds. In such a setting, it is still the case that $\Pr[x_i = 1|y = 1] \approx \frac{1}{2} + \alpha$, however

$$\Pr[x_i x_j = 1|y = 1] = \begin{cases} \frac{1}{2} + \Theta(\alpha) & \text{if } i \text{ and } j \text{ are both in the true parity set,} \\ \frac{1}{2} + \Theta(\alpha^2) & \text{if } i \text{ or } j \text{ is not in the set of parity bits.} \end{cases}$$

The punchline of the above discrepancy is that very small biases—even a bias of $\alpha = 1/\sqrt{n}$ can be quite helpful. Given such a bias, for any pair $i, j \in [n]$, for sufficiently large n, even $n^{1.01}$ examples will be sufficient to determine whether i and j are both in the parity set by simply measuring the correlation between the ith and jth indices for examples with odd label, namely estimating $\Pr[x_i x_j = 1|y = 1]$ based on the examples. How does one compute these $\binom{n}{2}$ correlations in time $o(n^3)$? By (fast) matrix multiplication. It is worth stressing that, provided this argument is sound, the resulting algorithm will be extremely noise-robust, since the discrepancy between $\Pr[x_i x_j = 1|y = 1]$ in the cases that i, j are both parity bits and the case that they are not will degrade linearly as $\eta \to 1/2$.

It should now be intuitively clear how to extend this approach from the small-biased setting to the setting in which the examples are generated uniformly at random, since a bias of $1/\sqrt{n}$ is quite modest. In particular, with constant probability, a random length–n example will have at least $\frac{n}{2} + \sqrt{n}$ positive indices, thus simply filtering the examples by removing those with fewer than $n/2$ positive indices should be sufficient to instill the necessary bias (at the minor expense of independence).

10.3 Learning Parity by Adding Bias

As in the case of learning a parity of size $k = 3$, outlined in the previous section, for the general case of parities of size k a bias of $1/\sqrt{n}$ in the examples will be sufficient. There are many approaches to achieving this bias; algorithmically, the most simple approach is to take examples, and reject those which have fewer than $\frac{n}{2} + \sqrt{n}$ positive indices. While this approach can be made to work, the conditioning on the total weight being large greatly complicates the analysis. Thus we instead argue that one can filter the examples in such a way that the distribution of the examples that remain is very close in total variational distance (ℓ_1 distance) to the distribution in which the examples are actually generated by independently choosing the value of each index with probability $\frac{1}{2} + \frac{1}{\sqrt{n}}$. Thus the result of applying our algorithm to the filtered examples will, with high probability be identical to the result of applying the algorithm to a set of examples generated according to the idealized process which selects the value of each index of each example independently, to be 1 with probability $1/2 + 1/\sqrt{n}$, and thus it suffices to perform the analysis of the simpler setting in which indices are chosen independently.

We first state the simple filtering process, and then prove that the resulting distribution of examples has the desired property. Throughout, we let $Bin[r, p]$ denote the binomial

random variable representing the number of heads that occur after flipping r i.i.d. coins that each land heads with probability p.

Algorithm 10.6. MAKE BIASED EXAMPLES
Input: An $m \times n$ matrix X with entries $x_{i,j} \in \{-1, +1\}$, a desired bias $\alpha \in (0, \frac{1}{2})$, and $t \in [n]$.
Output: an $m' \times n$ matrix Y, for some $m' \leq m$, consisting of a subset of the rows of X.

- Define $r = \frac{\Pr[Bin[n,\frac{1}{2}]>t]}{\Pr[Bin[n,\frac{1}{2}+\alpha]>t]}$.
- For each row $x_i = x_{i,1}, \ldots, x_{i,n}$ of X:
  - let s_i be the number of 1's in x_i.
  - If $s_i \geq t$ discard row x_i.
  - Otherwise, if $s_i < t$, then include row x_i in matrix Y with probability

$$r \cdot \frac{\Pr[Bin[n, \frac{1}{2}+\alpha] = s_i]}{\Pr[Bin[n, \frac{1}{2}] = s_i]}$$

(Note that this quantity is always bounded by 1.)

Proposition 10.7. *The algorithm* MAKE BIASED EXAMPLES *when given as input an $m \times n$ matrix X chosen with each entry being 1 independently with probability $1/2$, $\alpha = o(1)$, and $t > \frac{n}{2} + \sqrt{n} + \alpha n$, will output matrix Y satisfying the two following properties:*

- *The total variation distance between the distribution from which each row of Y is chosen and the distribution on rows defined by the process of picking each of the n elements independently to be 1 with probability $\frac{1}{2} + \alpha$, is at most $2e^{-\frac{(n(\frac{1}{2}+\alpha)-t)^2}{(1-2\alpha)n}}$.*
- *With probability at least $1 - e^{-\frac{mr^2}{32}}$, Y has at least $\frac{mr}{4}$ rows, where $r := \frac{1}{(1-2\alpha)^{n-t}(1+2\alpha)^t\sqrt{n}}$.*

The following lemma will be useful in the proof of the above proposition.

Lemma 10.8. *For $Bin[n,p]$ denoting a binomially distributed random variable, for $\alpha > 0$ with $\alpha = o(1)$ and $s > \sqrt{n} + \alpha n$,*

$$\frac{\Pr[Bin[n, \frac{1}{2}] > \frac{n}{2} + s]}{\Pr[Bin[n, \frac{1}{2} + \alpha] > \frac{n}{2} + s]} \geq \frac{(1-2\alpha)^{s-\frac{n}{2}}(1+2\alpha)^{-\frac{n}{2}-s}}{\sqrt{n}},$$

for sufficiently large n.

Proof. We first lowerbound the numerator; trivially, $\Pr[Bin[n, \frac{1}{2}] > \frac{n}{2} + s] > \Pr[Bin[n, \frac{1}{2}] = \frac{n}{2} + s] = \binom{n}{\frac{n}{2}+s}\frac{1}{2^n}$. We now upper bound the denominator. To this order, note that for any

$s' \geq s$, we have

$$\begin{aligned}
\frac{\Pr[Bin[n, \frac{1}{2}+\alpha] = \frac{n}{2}+s'+1]}{\Pr[Bin[n, \frac{1}{2}+\alpha] = \frac{n}{2}+s']} &= \frac{\binom{n}{\frac{n}{2}+s'+1}(\frac{1}{2}+\alpha)}{\binom{n}{\frac{n}{2}+s'}(\frac{1}{2}-\alpha)} \\
&= \frac{n-2s'}{2+n+2s'} \cdot \frac{\frac{1}{2}+\alpha}{\frac{1}{2}-\alpha} \\
&\leq \frac{n-2(\sqrt{n}+\alpha n)}{2+n+2(\sqrt{n}+\alpha n)} \cdot \frac{\frac{1}{2}+\alpha}{\frac{1}{2}-\alpha} \\
&= 1 - \frac{4\sqrt{n}-4\alpha+2}{(n+2\sqrt{n}+2\alpha n+2)(1-2\alpha)} \leq 1 - \frac{1}{\sqrt{n}},
\end{aligned}$$

for sufficiently large n. This shows that we may bound $\sum_{i=\frac{n}{2}+s}^{n} \binom{n}{i}(1/2-\alpha)^i(1/2+\alpha)^{n-i}$ by the geometric series

$$\binom{n}{\frac{n}{2}+s}(1/2-\alpha)^{\frac{n}{2}-s}(1/2+\alpha)^{\frac{n}{2}+s}\sum_{i=0}^{\infty}(1-\frac{1}{\sqrt{n}})^i = \binom{n}{\frac{n}{2}+s}(1/2-\alpha)^{\frac{n}{2}-s}(1/2+\alpha)^{\frac{n}{2}+s}\sqrt{n}.$$

Thus the desired ratio is at least

$$\frac{\binom{n}{\frac{n}{2}+s}\frac{1}{2^n}}{\binom{n}{\frac{n}{2}+s}(1/2-\alpha)^{\frac{n}{2}-s}(1/2+\alpha)^{\frac{n}{2}+s}\sqrt{n}} = \frac{(1-2\alpha)^{s-\frac{n}{2}}(1+2\alpha)^{-\frac{n}{2}-s}}{\sqrt{n}}.$$

□

Proof of Proposition 10.7. Each row of Y is distributed as a length-n string with each bit equaling 1 independently with probability α, conditioned on the total number of 1's to be at most t. This distribution has variation distance at most $2\Pr[Bin[n, \frac{1}{2}+\alpha] > t]$ from the corresponding distribution in which no conditioning occurs. By standard Chernoff bounds, $\Pr[Bin[n, \frac{1}{2}+\alpha] > t] \leq e^{-\frac{(n(\frac{1}{2}+\alpha)-t)^2}{(1-2\alpha)n}}$.

The expected number of rows of Y will be at least $m \cdot r \cdot (1-q)$, where $q = \Pr[Bin[n, \frac{1}{2}+\alpha] > t]$, and thus this expectation is trivially at least $\frac{mr}{2}$. Since each row of the input X is inserted into Y independently, with probability at least $\frac{r}{2}$, by a Chernoff bound, $\Pr[|Y| < \frac{mr}{4}] < e^{-\frac{mr^2}{32}}$. Using the lower bound on r of Lemma 10.8 yields the claim. □

We now state the general algorithm for learning parities of size k. Note that throughout, we assume that we know the size of the true parity set. This is without loss of generality, as we can always simply try $k = 1, 2, 3, \ldots$, and lose at most a factor of k in our runtime. Additionally, we aim to find the parity with some constant probability. Since we can always verify whether the returned parity set is correct (with all but inverse exponential probability), by simply repeating the algorithm many times this constant probability of success can become

probability $1-\delta$ at an extra multiplicative expense of $\log \frac{1}{\delta}$. Finally, we assume that k is divisible by 3. This is without loss of generality, as we can always insert up to two extra bits in each example and multiply the label by their values so as to yield examples from an instance of size at most $n+2$ where the size of the parity is actually divisible by 3.

Algorithm 10.9. LEARN PARITY WITH NOISE
Input: An $m \times n$ matrix X with entries $x_{i,j} \in \{-1,+1\}$, a length m vector $v \in \{-1,+1\}^m$ of labels, a parameter k that is divisible by 3.
Output: a set of k indices $S \subset [n]$.

- *Let Y be the result of running* MAKE BIASED EXAMPLES *on matrix X, with $\alpha = \frac{1}{\sqrt{n}}$ and $t = \frac{n}{2} + \frac{k \log n}{12}\sqrt{n}$.*
- *Remove all rows from Y whose corresponding label (in vector v) is -1, and denote the resulting smaller $m' \times n$ matrix Y'.*
- *Generate the $m' \times \binom{n}{\frac{k}{3}}$ matrix Z by taking each row of Y', and generating a row of Z of length $\binom{n}{k/3}$, with each position $z_{i,S}$ corresponding to a set $S \subset [n]$ of $k/3$ distinct indices, and setting $z_{i,S} = \prod_{j \in S} y'_{i,j}$.*
- *Compute the $\binom{n}{k/3} \times \binom{n}{k/3}$ matrix $C = Z^t Z$. For convenient, we regard the elements of C to be indexed by a pair of sets $S, S' \subset n$ with $|S| = k/3$, thus $c_{S,S'}$ is the entry corresponding to the product of the two columns of Z corresponding to the sets S and S'.*
- *For every pair of subsets $S, S' \subset [n]$ with S and S' each consisting of $k/3$ distinct elements, if $S \cap S' \neq \emptyset$, set $c_{S,S'} = 0$.*
- *Let c_{S_1,S'_1}, be the largest elements of matrix C. For all sets $S \subset [n]$ with $|S| = k/3$ satisfying $S \cap (S_1 \cup S'_1) \neq \emptyset$, zero out the row and column of C corresponding to set S. Let c_{S_2,S'_2} be the largest element of the resulting matrix, and return $S_1 \cup S'_1 \cup S_2$.*

Theorem 10.2. *For any fixed $\epsilon > 0$, for sufficiently large n and k given $m = \frac{n^{\frac{2k}{3}(1+\epsilon)}}{(1-2\eta)^{2+\epsilon}}$ examples from an (n,k,η) instance of parity with noise, with probability $1 - o(1/n)$, the algorithm* LEARN PARITY WITH NOISE*, when given as input the $m \times n$ matrix of examples, and length m vector of labels, will correctly return the true set of k parity bits. Additionally, the algorithm will run in time $O\left(\left(\frac{n^{\frac{k}{3}(1+\epsilon)}}{(1-2\eta)^{2+\epsilon}}\right)^{\omega}\right)$.*

Given that applying MAKE BIASED EXAMPLES to matrix X yields a matrix Y with suitably biased elements, we must make sure that the matrix Z inherits some bias from Y. In particular, the fact that each entry of Z is given as the product of $k/3$ entries of X should not completely erase the bias. While the bias will decrease, note that the length of the rows of Z are correspondingly larger, and we are only hoping that the bias of each element of Z is roughly $1/\sqrt{|Z|}$. The following basic lemma guarantees this.

Lemma 10.10. *Let $z = \prod_{i=1}^{s} w_i$, where each $w_i \in \{-1, +1\}$ is chosen independently to be 1 with probability $\frac{1}{2} + \alpha$. Then $\Pr[z = 1] = \frac{1}{2} + 2^{s-1}\alpha^s$.*

Proof. Letting $p = \frac{1}{2} - \alpha$, we have the following:

$$\begin{aligned} \Pr[z = 1] - \Pr[z = -1] &= \sum_{i=0}^{s} (-1)^i p^i (1-p)^{s-i} \binom{s}{i} \\ &= ((1-p) - p)^s \\ &= (1-2p)^s = 2^s \alpha^s, \end{aligned}$$

□

Proof of Theorem 10.2. Proposition 10.7 guarantees that with probability at least $1 - o(1/n)$ the matrix Y has at least the following number of rows:

$$\frac{m}{4(1-\frac{2}{\sqrt{n}})^{n-t}(1+\frac{2}{\sqrt{n}})^t\sqrt{n}} \geq \frac{m}{4(1+\frac{2}{\sqrt{n}})^{2(t-\frac{n}{2})}\sqrt{n}} = \frac{m}{4(1+\frac{2}{\sqrt{n}})^{\frac{\sqrt{n}}{2}(k\log n)/3}\sqrt{n}} \geq \frac{m}{4n^{k/3+\frac{1}{2}}}.$$

For m as specified in the theorem statement, any constant $\epsilon' < \epsilon$, for sufficiently large n and k, this is at least $\frac{n^{\frac{k}{3}(1+\epsilon')}}{(1-2\eta)^{2+\epsilon}}$. Additionally, Proposition 10.7 guarantees that the rows of matrix Y are τ-far in variation distance from the distribution defined by choosing each element independently to be 1 with probability $\frac{1}{2} + \frac{1}{\sqrt{n}}$, where

$$\tau \leq e^{-\frac{(\frac{n}{2}+\sqrt{n}-t)^2}{2n(\frac{1}{2}-\frac{1}{\sqrt{n}})}} \leq 2e^{-(\frac{k\log n}{12}-1)^2} \leq n^{-\frac{k^2\log n}{200}},$$

for sufficiently large k. This variation distance is super constantly smaller than $1/(mn)$, and thus with probability $1 - o(1/n)$, the algorithm will perform identically as in the case that the elements of matrix Y were actually generated independent with probability of being 1 equal to $\frac{1}{2} + \frac{1}{\sqrt{n}}$. For the remainder of the proof, we argue as if matrix Y is generated in that fashion.

We now consider the matrix Z. Let $z_S, z_{S'}$ be two element of the row z of Z corresponding to disjoint sets $S, S' \subset [n]$, and let ℓ denote the label corresponding to row z. Let $w = \prod_{j \in S \cup S'} y_j$, denote the random variable representing $z_S z_{S'}$. For notational convenience, define

$$F(\beta, h) = \sum_{i=0}^{\lfloor h/2 \rfloor} (\frac{1}{2} - \beta)^{2i} (\frac{1}{2} + \beta)^{h-2i} \binom{h}{2i} = \frac{1}{2}\left(1 + 2^h \beta^h\right),$$

which is the probability that when h identical independent coins that land heads with probability $\frac{1}{2} + \beta$ are tossed, an even number of heads occurs. Letting s denotes the number of

parity bits in $S \cup S'$, we have the following, where $\alpha = \frac{1}{\sqrt{n}}$:

$$\begin{aligned}\Pr[w = 1|\ell = 1] &= \frac{F(\alpha, s)F(\alpha, \frac{2k}{3} - s)F(\alpha, k - s)}{F(\alpha, k)} \\ &\quad + \frac{(1 - F(\alpha, s))(1 - F(\alpha, \frac{2k}{3} - s))(1 - F(\alpha, k - s))}{F(\alpha, k)} \\ &= \frac{1 + (2\alpha)^{2k/3} + (2\alpha)^k + (2\alpha)^{5k/3-2s}}{2(1 + (2\alpha)^k)},\end{aligned}$$

where the numerator of first line is computing the probability that $w = 1$ and $\ell = 1$.

In the case that $s = 2k/3$, which occurs when both S and S' are subsets of the set of parity bits, then we can lowerbound the above as

$$\Pr[w = 1|\ell = 1] \geq \frac{1 + (2\alpha)^{k/3}}{2(1 + (2\alpha)^k)} \geq \frac{1}{2} + \frac{(2\alpha)^{k/3}}{2} - (2\alpha)^{2k/3} \geq \frac{1}{2} + \frac{(2\alpha)^{k/3}}{3},$$

since $\alpha = o(1)$. In the case that $s \leq 2k/3 - 1$, we upperbound the quantity as follows:

$$\Pr[w = 1|\ell = 1] \leq \frac{1 + 2(2\alpha)^{k/3+2}}{2} \leq \frac{1}{2} + \frac{(2\alpha)^{k/3}}{100},$$

since $\alpha = o(1)$.

Putting the pieces together, letting m' denote the number of rows of matrix Z, which we showed is at least $\frac{n^{\frac{k}{3}(1+\epsilon')}}{(1-2\eta)^{2+\epsilon}}$, in the case that $\eta = 0$ (there is no noise in the labels), we have that for any entry $c_{S,S'}$ of matrix C corresponding to two disjoint sets S, S', where S and S' are not *both* subsets of the parity bits, $E[c_{S,S'}] \leq 2\frac{m'\left(2/\sqrt{n}\right)^{k/3}}{100}$. On the other hand, if S, S' are both subsets of the parity bits, then $E[c_{S,S'}] \geq 2\frac{m'\left(2/\sqrt{n}\right)^{k/3}}{3}$, and since these quantities have variance at most m', for any constant ϵ, via a union bound over Chernoff bounds, taking n large yields that with probability $1 - o(1/n)$, all the entries of C corresponding to pairs of disjoint sets that are not both subsets of the true parity bits will be smaller than all the entries that correspond to pairs of subsets of the true parity bits. In the case of $\eta > 0$, an identical argument holds. □

Reducing the Sample Complexity

In order to obtain our desired corollary for learning DNF (Corollary 10.5), we must reduce the number of examples used in our Learn Parity with Noise algorithm. Intuitively, provided one has a very noise–robust algorithms, such reduction in sample complexity is very easy; one simply takes a very small number of examples—in fact, $n^{1+\epsilon}$ will suffice—and then "manufactures" many examples by XORing together small sets of the actual examples. Provided the initial noise in the labels is η, if we XOR together q examples, then the XOR of the labels will be the correct label with probability at least $\frac{1}{2} + \frac{(1-2\eta)^q}{2}$.

Algorithm 10.11. MAKE MORE EXAMPLES
Input: An $m \times n$ matrix X with entries $x_{i,j} \in \{-1,+1\}$, a length m vector $v \in \{-1,+1\}^m$ of labels, a positive integer $q < m$, an integer m'.
Output: An $m' \times n$ matrix Y, and a length m' vector w of labels.

- *For each $i \in [m']$, randomly choose a set $T \subset [m]$, with $|T| = q$, create row y_i of Y, by assigning the jth component of y_i to be $\prod_{\ell \in T} x_{\ell,j}$, and letting the ith label be $\prod_{j \in T} v_j$.*

Ideally, we would be able to apply the algorithm LEARN NOISY PARITIES in a black-box fashion to the output of running MAKE MORE EXAMPLES on a small number of actual examples, as was done in [83]. Unfortunately, because the noise in the generated examples will increase with q in the exponent, we will not be able to take sufficiently large q so as to yield the necessary claim that the distribution of resulting examples is close in total variation distance to the desired uniform distribution.

Instead, we argue that the distribution of a small number (namely, k) of the columns of the generated examples are close to uniform. The idea is that we will argue that the distribution of the values in the k parity columns are close to uniform, which will let us apply our Chernoff bound to argue that with very high probability, the "good" entries $c_{S,S'}$ of the matrix C generated in LEARN NOISY PARITIES, corresponding to S, S' subsets of the true parity set, will be "large". For all the "bad" entries of C, we will not be able to apply Chernoff bounds; however, using the fact that the rows are pairwise independent, we will apply Chebyshev's inequality to argue that with probability at least $1/2$, each "bad" element will be small. Thus after running the whole algorithm $\log(\binom{n}{k/3})$ times, we can argue that with high probability, in *every* run, the "good" coordinates will be large, whereas for a "bad" element, in each run of the algorithm, it will be small with probability at least $1/2$. Thus after $\log(\binom{n}{k/3})$ runs, with high probability the only elements that were never "small" will correspond to entries whose row and column correspond to subsets of the true parity set, as desired. We now make this roadmap rigorous. We begin by defining what it means for a family of hash functions to be *universal*, and state the Leftover Hash Lemma.

Definition 10.12. *Let $\mathcal{H}$ be a family of hash functions from A to B, and let $H \in \mathcal{H}$ be chosen uniformly at random. $\mathcal{H}$ is a* universal *family of hash functions if for all distinct $a, a' \in A$, $\Pr[H(a) = H(a')] \leq \frac{1}{|B|}$.*

Lemma 10.13 (Leftover Hash Lemma [70])**.** *For $A \subset \{0,1\}^m$, with $|A| \geq 2^r$, and $|B| = \{-1,+1\}^{r-\ell}$ for some $\ell > 0$, if $\mathcal{H}$ is a universal family of hash functions from A to B, then with probability at least $1-2^{-\ell/4}$, a uniformly random $h \in \mathcal{H}$ will satisfy $D_{tv}[h(a), Unif(B)] \leq 2^{-\ell/4}$, where a is a random element of A, $Unif(B)$ denotes the uniform distribution on B, and D_{tv} is the statistical distance (total variation distance).*

The following basic fact will also be useful.

Fact 10.14. *Given a vector $v \in \{-1,+1\}^m$, such that $m(\frac{1}{2}+p)$ indices of v are $+1$, then for a random set $T \subset [m]$, with $|T| = q$,*

$$\Pr[\prod_{i \in T} v_i = 1] \geq \frac{1}{2}\left(1 + \left(\frac{2mp - q + 1}{m - q + 1}\right)^q\right).$$

Proposition 10.15. *Given an $m \times n$ matrix X and vector of labels v consisting of m examples from an instance of parity with noise with noise rate η, integer $q \leq \frac{m(1-2\eta)}{4}$, and integer m', for any fixed set $S \subset [n]$ with $|S| = k$, with probability at least $1 - 2^{-\frac{q \log \frac{m}{q} - k}{4}}$, the algorithm* MAKE MORE EXAMPLES *on input X, v, q, m', will output a matrix Y such that the $m' \times k$ submatrix Y_S defined as the subset of the columns of Y corresponding to indices in S, will have total variation distance at most $m'2^{-\frac{q \log \frac{m}{q} - k}{4}}$ from the distribution on matrices given by assigning each element to be ± 1 independently with probability $1/2$.*

Additionally, with probability at least $1 - 2^{-\frac{(q-1)\log \frac{m}{q-1} - k}{4}}$, the distribution of the rows of Y_S corresponding to the set of correct labels, will differ from that corresponding to the set of incorrect labels by statistical distance at most $2m'2^{-\frac{(q-1)\log \frac{m}{q-1} - k}{4}}$. Finally, provided $1 - 2\eta > 4m^{-0.4}$, with probability at least $1 - o(1/m)$, the number of correct labels will be at least $m'\left(\frac{1}{2} + \frac{1}{2}\left(\frac{1-2\eta}{4}\right)^q\right) - m'^{0.6}$.

Proof. We will first apply the Leftover Hash Lemma (Lemma 10.13). Note that each choice of matrix X defines a hash function from the set $A := \{T : T \subset [m], |T| = q\}$ to the set $B = \{-1,+1\}^k$, via the mapping that considers the columns of X corresponding to indices in set S, and XORs each coordinate of the rows of X with indices in set T (as described in the algorithm MAKE MORE EXAMPLES). Trivially, this family of hash functions is universal, since for two sets $T \neq T'$, supposing that $i \in T$, $i \notin T'$, the image of T and T' will differ XORing with a uniformly random string (namely, the ith row of X). Next, note that $|A| = \binom{m}{q} \geq 2^{q \log \frac{m}{q}}$ and thus Lemma 10.13 implies that with probability at least $1 - 2^{-\frac{q \log \frac{m}{q} - k}{4}}$ over the choice of matrix X, we will have that the distance between each row of Y and the uniform distribution over $\{-1,+1\}^k$ is at most $2^{-\frac{q \log m - 2k}{8}}$. A union bound over our m' rows yields the desired claim.

We now reason about labels. With probability at least $1 - o(1/m)$, the number of correct labels in the original vector v of labels will be at least $\frac{m}{2} + \frac{m(1-2\eta)}{2} - m^{0.6} > \frac{m}{2} + \frac{m(1-2\eta)}{4}$. Thus by Fact 10.14, with this probability the expected number of correct labels in vector w will be at least

$$m'\left(\frac{1}{2}\left(1 + \left(\frac{m(1-2\eta)/2 - q + 1}{m - q + 1}\right)^q\right)\right) \geq m'\left(\frac{1}{2} + \frac{1}{2}\left(\frac{1-2\eta}{4}\right)^q\right),$$

and thus with probability at least $1 - o(1/m)$ over the initial choice of the v labels, and the choice of the sets that generate the m' new examples, at least $m'\left(\frac{1}{2} + \frac{1}{2}\left(\frac{1-2\eta}{4}\right)^q\right) - m'^{0.6}$ of the labels w will be correct.

We now argue that for a given "manufactured" example, the correctness of the label is essentially independent of the values of the chosen set of k indices. We proceed as in [83], and note that, assuming there is at least one incorrectly labelled example in v (if not, then the independence is trivial), letting X_{odd}, X_{even} denote the sets of subsets $T \subset [m]$ with $|T| = q$ for which the number of corresponding label is incorrect (correct). With probability $1 - o(1/m)$, $|X_{even}| > |X_{odd}| > \binom{m}{q-1}$, and thus (since the correctness of the original labels are chosen independently of the corresponding example) we may apply Lemma 10.13 as above, to conclude that the distribution of the values of the k bits is distributed nearly uniformly over the 2^k values. In particular, with probability at least $1 - 2^{-\frac{(q-1)\log\frac{m}{q-1}-k}{2}}$, the distribution of the k bits conditioned on the label being correct will differ from the distribution conditioned on the label being incorrect by at most statistical distance $1 - 2^{-\frac{(q-1)\log\frac{m}{q-1}-k}{4}}$. □

We now describe our algorithm for solving instances of parity with noise, that uses few examples.

Algorithm 10.16. LEARN WITH FEW EXAMPLES
Input: Positive integers k, r, q, m' an $r \cdot m \times n$ matrix X with entries $x_{i,j} \in \{-1, +1\}$, and a length $r \cdot m$ vector $v \in \{-1, +1\}^m$ of labels.
Output: A set $S \subset [n]$ with $|S| = k$.

- *For $i = 1$ to r*
  - *Let matrix X', and labels v' be the output of running algorithm* MAKE MORE EXAMPLES *on input X^i, v^i, q, m', where X^i is the $m \times n$ submatrix of X consisting of rows $i \cdot m + 1$ through rows $(i+1)m$, and v^i is the corresponding vector of labels.*
  - *Let matrix Y be the result of running* MAKE BIASED EXAMPLES *on matrix X with $\alpha = \frac{1}{\sqrt{n}}$ and $t = \frac{n}{2} + \frac{k \log n}{12}\sqrt{n}$.*
  - *Remove all rows of Y with labels -1, and denote the resulting smaller $m'' \times n$ matrix Y'.*
  - *Generate the $m'' \times \binom{n}{k/3}$ matrix Z by taking each row of Y', and generating a row of length $\binom{n}{k/3}$ with each position $z_{\ell,S}$ corresponding to a set $S \subset [n]$ of $k/3$ (distinct) indices, and setting $z_{\ell,S} = \prod_{j \in S} y'_{\ell,j}$.*
  - *Compute the $\binom{n}{k/3} \times \binom{n}{k/3}$ matrix $C^i = Z^t Z$, and let $m^i := m''$.*
- *Let the set $ParityBits$ be the union of all pairs of disjoint sets of size $k/3$, S, S', that have the property that $c^i_{S,S'} > \frac{m^i (2/\sqrt{n})^{k/3}}{3}$ for each $i \in [r]$, where $c^i_{S,S'}$ denotes the index of matrix C^i indexed by the sets S, S', as in algorithm* LEARN PARITY WITH NOISE.
- *If $|ParityBits| \neq k$ output $FAIL$, otherwise output the set $ParityBits$.*

Theorem 10.3. *The algorithm* Learn With Few Examples*, when run on input k, $r := 100k \log n$, q, $m' := \frac{n^{\frac{2k}{3}(1+\epsilon)}}{(1-2\eta)^{3q}}$, and $m = 100 \cdot rq \frac{n^{2k/q}}{(1-2\eta)^6}$ examples from an (n, k, η) instance of parity with noise, will return the correct set of k parity bits with probability at least $1-o(1)$. Additionally, the number of examples used is m, and the total runtime of the algorithm is bounded by*

$$poly\left(\frac{1}{(1-2\eta)^q}, 2^q\right) \cdot n^{\frac{k}{3}(1+\epsilon)\omega},$$

where $\omega < 2.38$ is the matrix multiplication exponent.

Proof. The proof follows from noting first that $m' < \binom{m/r}{q}^4$, and thus with probability $1 - o(1)$, in all r runs, no two choices of random subsets of $[m]$ of size q chosen in the construction of X' will be equal, and thus with this probability, the rows of X' (and thus Y and Y') will all be pairwise independent. Thus, from the proof of Theorem 10.2 and Chebyshev's inequality, for each pair S, S' of disjoint sets of size $k/3$ that are not both subsets of the true set of parity bits, $c^i_{S,S'} \leq \frac{m^i(2/\sqrt{n})^{k/3}}{3}$ with probability at least $1/2$. Since each of the r runs are independent, the probability that such a bad pair of sets remains after all $r = 100k \log n$ runs is at most $\frac{1}{n^{100k}}$, and thus via a union bound over the at most $\binom{n}{k/3}^2$ such pairs of bad sets, with probability $1 - o(1)$, no such bad pairs of sets will appear in the final output set $ParityBits$.

By Proposition 10.15, and our choice of parameters, with probability $1 - o(1/n)$, the total variation distance between the assignment of the values to the k true parity columns of matrix X' in a given run, and if they were chosen uniformly is at most $o(1/n)$, and thus with probability $1 - o(1)$, after the r runs, the algorithm must perform identically to the performance in the case that these columns were chosen uniformly at random, and thus the arguments of the proof of Theorem 10.2, and, in particular, the Chernoff bound, guarantees that with probability $1 - o(1)$ in *all* r runs, every pair of disjoint sets S, S' of size k that are subsets of the parity bits, will satisfy $c^i_{S,S'} > \frac{m^i(2/\sqrt{n})^{k/3}}{3}$, as desired. □

Part III

Learning Mixtures of Gaussians

Chapter 11

Learning Univariate Mixtures of Gaussians

The problem of estimating the parameters of a mixture of Gaussians has a long history of study in statistics and more recently, computer science. Given a sample drawn from a single Gaussian distribution, it is easy to accurately estimate the mean and covariance of the true distribution, since the sample mean and covariance converge very quickly to the true mean and covariance. In contrast, consider the setting in which, for example, half the sample points are drawn from one Gaussian, and half are drawn from a different Gaussian. How can one obtain accurate estimates of the means and covariances of these two Gaussians? From a practical perspective, this problem arises in many settings across a number of fields, including agriculture, economics, medicine, and genetics [122, 88], and represents one of the most basic mathematical formulations of the pervasive problem of clustering high dimensional data.

In this chapter we tackle the univariate version of this problem: learning mixtures of Gaussians in one dimension. In Chapter 12 we leverage the results of this chapter via a dimension reduction approach to yield an algorithm for the high dimensional analog of this problem.

Consider a mixture of k *different* univariate distributions, each with *mean* $\mu_i \in \mathbb{R}$, *variance* $\sigma_i^2 \in \mathbb{R}^+$, and *mixing weight* $w_i > 0$. The mixture is referred to as a Gaussian Mixture Model (GMM), and if the univariate density of the i^{th} Gaussian component is $F_i = \mathcal{N}(\mu_i, \sigma_i^2)$, then the GMM density is,

$$F = \sum_i w_i F_i.$$

The problem of learning the mixture is that of estimating w_i, μ_i, and σ_i^2 from a sample consisting of m independent draws from the GMM.

In this chapter we prove that the parameters w_i, μ_i, σ_i^2 can be estimated at an inverse polynomial rate. Given a desired accuracy ϵ, we give an algorithm for recovering the parameters to within this accuracy whose runtime and required sample size is polynomial in $1/\epsilon$, under provably minimal assumptions on the GMM, namely that the mixing weights w_i

and the total variational distance between the Gaussian components are all bounded away from 0. Previously, even in the case of mixtures of just two components, to the best of our knowledge, no subexponential bounds on the required sample size were known.

The guarantees of the following theorem, which is the main theorem of this chapter, are in terms of the error of the recovered parameters; we rely on this result on parameter recovery in Chapter 12 to yield our more general theorem on learning GMMs in arbitrary dimension with the stronger success guarantee that the recovered components are close in total variational distance (ℓ_1 distance) to the true components.

Theorem 11.1. *Suppose we are given access to independent draws from a GMM F of variance in $[1/2, 2]$, consisting of at most k Gaussian components, $F = \sum_i w_i \mathcal{N}(\mu_i, \sigma_i^2)$, where for all i, $w_i \geq \epsilon$, and for all $i \neq j$, $|\mu_i - \mu_j| + |\sigma_i^2 - \sigma_j^2| \geq \epsilon$.*

There is an algorithm that for any fixed k, uses a sample of size $poly(\frac{1}{\epsilon}, \log \frac{1}{\delta})$ and has runtime at most $poly(\frac{1}{\epsilon}, \log \frac{1}{\delta})$, such that with probability at least $1 - \delta$ it will output mixture parameters $\hat{w}_i, \hat{\mu}_i, \hat{\sigma}_i{}^2$, with the property that there is a permutation $\pi : [k] \to [k]$ such that

$$|w_i - \hat{w}_{\pi(i)}| \leq \epsilon, \quad |\mu_i - \hat{\mu}_{\pi(i)}| \leq \epsilon, \quad |\sigma_i^2 - \hat{\sigma}^2_{\pi(i)}| \leq \epsilon \text{ for each } i = 1, \ldots, k \ .$$

Our approach is via the method of moments. We show that noisy estimates of the first $4k - 2$ moments of a univariate mixture of k Gaussians suffice to recover accurate estimates of the mixture parameters, as conjectured by Pearson in 1894 [105] in the case that $k = 2$, and that these estimates converge at an inverse polynomial rate.

The correctness of our algorithm rests on what we term the *polynomially robust identifiability* of GMMs (Theorem 11.2). The *identifiability* of GMMs is well known: any two different GMMs F, F' (where the components of F differ from those of F') have different probability distributions [120]. We show that this identifiability is "polynomially robust"; if the components of F and F' differ by ϵ, then the densities of F and F' differ in total variation distance by at least $poly(\epsilon)$. Our proof of this robust identifiability is based on a series of convolutions and "deconvolutions", which could also be interpreted via the univariate heat equation.

The running time (and sample complexity) of our algorithm is a fixed polynomial in $\frac{1}{\epsilon}$ for any constant number of mixture components, k. The dependence on k, however, is disappointing—the exponent of the polynomial is exponential in k. Nevertheless, in Section 11.4 we prove the following proposition demonstrating that a polynomial dependence on k is information theoretically impossible.

Proposition. *There exists mixtures F_1, F_2 of at most k Gaussians such that all mixing weights are at least $1/4k$, for every pair of components in the same mixture, their total variational distance is at least $1/4k$, one of the mixtures has a component with variance 2, and the other mixture consists entirely of components with variance 1, yet*

$$D_{tv}(F_1, F_2) < e^{-\Theta(k)}.$$

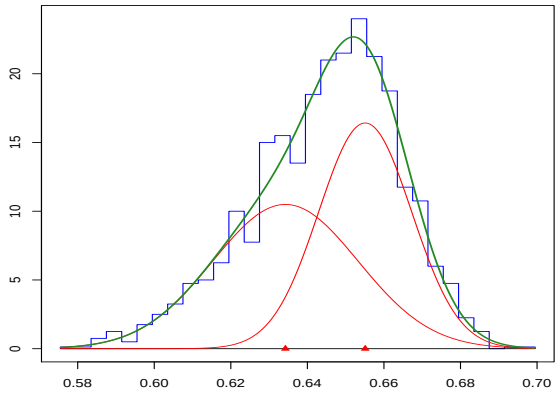

Figure 11.1: A fit of a mixture of two univariate Gaussians to Pearson's data on Naples crabs [105]. This density plot was created by P. Macdonald using *R* [84].

In particular, F, F' are mixtures of significantly different Gaussian components, yet the final condition shows that we cannot hope to distinguish between these two mixtures using a sample of size $poly(k)$.

A Brief History of GMMs

In the 1890's, Karl Pearson was given a dataset consisting of the ratio of the length to the breadth of 1000 crabs found near Naples, and conjectured that the dataset arose as a GMM of two components, corresponding to two crab species. Pearson then attempted to recover estimates of the parameters of the two hypothesized species, using the *method of moments*. He computed empirical estimates of the first six (raw) moments $\mathrm{E}[x^i] \approx \frac{1}{m}\sum_{j=1}^{m} x_j^i$, for $i = 1, 2, \ldots, 6$; then, using only the first five moments, he solved a cleverly constructed ninth-degree polynomial, *by hand*, from which he derived a set of candidate mixture parameters. Finally, he heuristically chose the candidate among them whose sixth moment most closely agreed with the empirical estimate. [105]

The potential problem with this approach, which Pearson acknowledged, was the issue of robust identifiability. Perhaps there exist two different mixtures, where the components of one mixture are very different from the components of the other mixture, but nevertheless the densities and the moments of the two mixtures are extremely similar.

Later work showed that "identifiability" is theoretically possible—if there are two different GMMs (i.e. the components of one of the mixtures differ from the components of the other mixture) then they have different probability densities [120]. The proof of this fact argued that if the components of the two GMMs of largest variance do not match, then this disparity will be exposed in the tails of the GMMs. If they do match, one can peel away the pair of components, and proceed inductively. The issue with this approach is that it sheds

no light on convergence *rates*, as it is based on differences in the density of the tails of the distributions, which would require exponential amounts of data to discern. In particular, to ϵ-approximate the Gaussian parameters in the sense that we will soon describe, previous work left open the possibility that it might require an amount of data that grows exponentially in $1/\epsilon$. In Section 11.2 we show that GMMs are *polynomially robust*, and hence a polynomial amount of data is sufficient. Our proof of this robustness is via the method of moments, in some sense validating Pearson's approach.

Dasgupta introduced the problem of learning GMMs to the theoretical computer science community in the setting in which the GMM is in high dimensional, and the components are extremely well separated in comparison with their covariances (essentially non-overlapping densities) [43]. Dasgupta considered the case where all the components are spherical Gaussians with radii 1, and required that the separation between the components' center be $O(d^2)$, where d is the dimension of the space. He gave a polynomial time algorithm for learning the mixture, which proceeded by first accurately clustering the sample points according to which component they were drawn from. Given such a clustering of the data, one can then simply return the sample mean and covariance for each cluster.

This work initiated a line of work on polynomial-time algorithms for clustering GMMs in high dimensions [12, 46, 134, 76, 3, 31]. As this work progressed, the assumptions that the components be identical and spherical was removed, and the separation assumptions were relaxed slightly; however, any approach to learning GMMs via clustering must assume that the components have essentially no overlap.

More recently, a polynomial-time *density estimation* algorithm was given for *axis-aligned* GMMs (i.e. GMMs whose components have diagonal covariance matrices), which did not require any separation assumption [53]. The problem of density estimation is to return a GMM whose probability density function is close to that of the true GMM from which the sample was drawn, as opposed to the potentially more ambitious goal of returning estimates of each of the constituent components. The results of this chapter, however, imply that any two GMMs with sufficiently similar densities must also have similar components (provided the number of components in each GMM is bounded, and the minimum mixing weight is not too small.)

Independently from this work, using ideas from algebraic geometry, it was recently shown that the method of moments can be used to provably learn mixture parameters in the more general context of mixtures of any family of distributions whose moments are given by polynomials of a finite parameter set [23]. For the case of GMMs, our results of Chapter 12 are stronger than the more general results of [23] in two senses: first, our approach allows one to obtain explicit bounds on the exponent of the polynomial bounding the required sample sample, in contrast to the existential results that simply show that the exponent of the polynomial dependence is finite for any fixed k. Secondly, and more significantly, we recover components that are accurate in a variational sense (i.e. the recovered components are close to the true components in ℓ_1 distance), and thus our algorithm can be used to accurately cluster the sample points (up to the statistical overlap in the components) or for

density estimation. In contrast, the results of [23] are in terms of a different metric which is not affine-invariant; their algorithm does not yield components that are necessarily close to the actual components in total variational distance (ℓ_1 distance), and hence their approach does not yield algorithms for clustering or density estimation.

Subsequent to the publication of the work presented in this chapter, there has been some progress on developing practically viable algorithms for learning GMMs and mixtures of other distributions, that also use a method of moments [8]. In contrast to our results, these algorithms only consider third or fourth moments, and have provable guarantees under certain non-degeneracy assumptions on the geometry of the components.

There is a vast literature on heuristics for learning GMMs, including the popular Expectation-Maximization and k-means algorithms, which lack provable success guarantees. While these approaches are effective in practice in some contexts, they can suffer slow convergence rates or terminate at local optima when run on high-dimensional data (see, e.g. [108]). These heuristics are orthogonal to our current goal of describing algorithms with provable success guarantees, and we refer the reader to the two books [122, 88], for a treatment of such heuristic approaches to learning GMMs.

11.1 Notation and Definitions

We use $\mathcal{N}(\mu, \sigma^2)$ to denote the univariate Gaussian of mean μ and variance σ^2. Correspondingly, we denote the probability density function of such a Gaussian by $\mathcal{N}(\mu, \sigma^2, x) := \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

Given two distributions, F, F', with probability density functions $F(x), F'(x)$, we denote their *total variation distance* by $D_{tv}(F, F') := \frac{1}{2}\int |F(x) - F'(x)|dx$. For a function $f : \mathbb{R} \to \mathbb{R}$, its ℓ_2 norm is denoted $||f(x)||_2 := \int_{-\infty}^{\infty} (f(x))^2 dx$, and its ℓ_∞ norm is denoted by $||f(x)||_\infty := sup_{x\in\mathbb{R}}|f(x)|$. We denote the i^{th}-raw moment of a distribution F, as $M_i(F) := E_F[x^i]$.

We define the *condition number* of a GMM, which is a parameter expressing the information theoretic difficulty of learning the given mixture:

Definition 11.1. *The condition number $\kappa(F)$ of GMM $F = \sum_{i=1}^{k} w_i F_i$ is defined to be,*

$$\kappa(F) = \frac{1}{\min\big(\{w_1, w_2, \ldots, w_k\} \cup \{D_{tv}(F_i, F_j) \mid i \neq j\}\big)}.$$

Any estimation algorithm requires, at a minimum, a sample size proportional to $\kappa(F)$ to have a constant probability of accurately estimating each component. This is simply because one requires a sample of size at least $1/w_i$ to have a constant probability of encountering a single example generated by F_i. Hence, for very small w_i, a large sample size is necessary. Similarly, even if one knows the distributions of two components, F_1 and F_2, one requires a sample of size at least $1/D_{tv}(F_1, F_2)$ to have a constant probability of distinguishing between

the case that all sample points arise from F or all sample points arise from F'. Thus, at least a linear dependence on $\kappa(F)$ is required; our results will have a polynomial dependence on $\kappa(F)$.

Definition 11.2. *The* parameter distance *between GMMs* $F(x) = \sum_{i=1}^{n} w_i \mathcal{N}(\mu_i, \sigma_i^2, x)$, *and* $F'(x) = \sum_{i=1}^{k} w_i' \mathcal{N}(\mu_i', \sigma_i'^2, x)$ *is defined to be*

$$D_{par}(F, F') := \min_{\pi} \sum_{i} \left(|w_i - w'_{\pi(i)}| + |\mu_i - \mu'_{\pi(i)}| + |\sigma_i^2 - \sigma'^2_{\pi(i)}| \right),$$

where the minimization is taken over all mappings $\pi : \{1, \ldots, n\} \to \{1, \ldots, k\}$.

11.2 Polynomially Robust Identifiability

Our main technical result, which we prove in this section, is the following theorem, showing that GMMs are *polynomially robustly* identifiable.

Theorem 11.2. *There is a constant* $c > 0$ *such that, for any any* $\epsilon < c$ *and any GMMs* F, F', *of* n *and* k *components, respectively with condition numbers* $\kappa(F), \kappa(F') \leq \frac{1}{\epsilon}$, *if* $D_{par}(F, F') \geq \epsilon$, *then*

$$\max_{i \leq 2(n+k-1)} |M_i(F) - M_i(F')| \geq \epsilon^{(O(k))^k}.$$

Before giving a formal proof of Theorem 11.2, we first sketch the rough intuition. Our proof will be via induction on $\max(n, k)$. We start by considering the constituent Gaussian of minimal variance in the mixtures. Assume without loss of generality that this minimum variance component is a component of F, and denote it by $\mathcal{N}$. If there is no component of F' whose mean, variance, and mixing weight very closely match those of $\mathcal{N}$, then we argue that there is a significant disparity in the low order moments of F and F', no matter what the other Gaussian components are. (This argument is the crux of the proof, and we will give the high-level sketch in the next paragraph.) If there *is* a component $\mathcal{N}'$ of F' whose mean, variance, and mixture weight very closely match those of $\mathcal{N}$, then we argue that we can remove $\mathcal{N}$ from F and $\mathcal{N}'$ from F' with only negligible effect on the discrepancy in the low-order moments. More formally, let H be the mixture of $n-1$ Gaussians obtained by removing $\mathcal{N}$ from F, and rescaling the weights so as to sum to one, and define H', a mixture of $k-1$ Gaussians derived analogously from F'. Then, assuming that $\mathcal{N}$ and $\mathcal{N}'$ are very similar, the disparity in the low-order moments of H and H' is almost the same as the disparity in low-order moments of F and F'. We can then apply the induction hypothesis to the mixtures H and H'.

We now return to the problem of showing that if the smallest variance Gaussian in F cannot be paired with a component of F' with similar mean, variance, and weight, that there must be a polynomially-significant discrepancy in the low-order moments of F and F'. This step relies on "deconvolving" by a Gaussian with an appropriately chosen variance (this

corresponds to running the heat equation in reverse for a suitable amount of time). We define the operation of "deconvolving" by a Gaussian of variance α as $\mathcal{J}_\alpha$; applying this operator to a mixture of Gaussians has a particularly simple effect: subtract α from the variance of each Gaussian in the mixture (assuming that each constituent Gaussian has variance at least α). If α is negative, this operation is simply convolution by a Gaussian of variance $-\alpha$.

Definition 11.3. *Let $F(x) = \sum_{i=1}^{n} w_i \mathcal{N}(\mu_i, \sigma_i^2, x)$ be the probability density function of a mixture of Gaussian distributions, and for any $\alpha < \min_i \sigma_i^2$, define*

$$\mathcal{J}_\alpha(F)(x) = \sum_{i=1}^{n} w_i \mathcal{N}(\mu_i, \sigma_i^2 - \alpha, x).$$

The key step will be to show that if the smallest variance Gaussian in one of the mixtures cannot be paired with a nearly identical Gaussian in the other mixture, then there is some α for which the difference in the probability densities of the resulting mixtures, after applying the operation $\mathcal{J}_\alpha$, has large ℓ_∞ norm. Intuitively, this deconvolution operation allows us to isolate Gaussians in each mixture and then we can reason about the total variational distance between the two mixtures locally, without worrying about the other Gaussians in the mixture.

Given this ℓ_∞ distance between the transformed pair of mixtures, we use the fact that there are relatively few zero-crossings in the difference in probability density functions of two mixtures of Gaussians (Proposition 11.5) to show that this ℓ_∞ distance gives rise to a discrepancy in at least one of the low-order moments of the pair of transformed distributions. To complete the argument, we then show that applying this transform to a pair of distributions, while certainly not preserving total variational distance, roughly preserves the combined disparity between the low-order moments of the pair of distributions.

We now formalize the above high-level outline of the proof approach. The following lemma argues that if the smallest variance Gaussian in mixture F can not be matched with a sufficiently similar component in the mixture F', then there is some α, possibly negative, such that $\max_x |\mathcal{J}_\alpha(F)(x) - \mathcal{J}_\alpha(F')(x)|$ is significant. Furthermore, every component in the transformed mixtures has a variance that is not too small.

Lemma 11.4. *Let F, F' be GMMs with at most k components and condition numbers bounded by $1/\epsilon$. Suppose without loss of generality that the Gaussian component of minimal variance is $\mathcal{N}(\mu_1, \sigma_1^2)$, and that there is some positive $\gamma < \epsilon/8$ such that for every $i \in \{1, \ldots, k\}$, at least one of the following holds:*

- $|\mu_1 - \mu_i'| > \gamma^5$
- $|\sigma_1^2 - \sigma_i'^2| > \gamma^5$
- $|w_1 - w_i'| > \gamma$.

Then there is some $\alpha > -\gamma^4$ such that either

- $\max_x(|\mathcal{J}_\alpha(F)(x) - \mathcal{J}_\alpha(F')(x)|) \geq \frac{1}{8\gamma}$ *and the minimum variance in any component of* $\mathcal{J}_\alpha(F)$ *and* $\mathcal{J}_\alpha(F')$ *is at least* γ^4,

or

- $\max_x(|\mathcal{J}_\alpha(F)(x) - \mathcal{J}_\alpha(F')(x)|) \geq \frac{2}{\gamma^5}$ *and the minimum variance in any component of* $\mathcal{J}_\alpha(F)$ *and* $\mathcal{J}_\alpha(F')$ *is at least* γ^{12}.

Proof. We start by considering the case when there is no Gaussian in F' that matches both the mean and variance to within γ^5. Consider applying $\mathcal{J}_{\sigma_1^2-\gamma^{12}}$ to both mixtures. Observe that

$$\mathcal{J}_{\sigma_1^2-\gamma^{12}}(F)(\mu_1) \geq \epsilon\mathcal{N}(0,\gamma^{12},0) = \frac{\epsilon}{\gamma^6\sqrt{2\pi}} \geq \frac{8\gamma}{\gamma^6\sqrt{2\pi}}.$$

We now argue that $\mathcal{J}_{\sigma_1^2-\gamma^{12}}(F')(\mu_1)$ cannot be too large, as all of its components must either have large variance (and hence small maximum value of the probability density function), or has small variance but a mean that is far from μ_1. Corollary A.4, makes this argument rigorous, showing the following:

$$\mathcal{J}_{\sigma_1^2-\gamma^{12}}(F')(\mu_1) \leq \frac{2}{\gamma^5\sqrt{2\pi e}}.$$

Thus

$$\mathcal{J}_{\sigma_1^2-\gamma^{12}}(F)(\mu_1) - \mathcal{J}_{\sigma_1^2-\gamma^{12}}(F')(\mu_1) \geq \frac{8\gamma}{\gamma^6\sqrt{2\pi}} - \frac{2}{\gamma^5\sqrt{2\pi e}} \geq \frac{2}{\gamma^5}.$$

Next, consider the case where we have at least one Gaussian component of F' that matches both μ_1 and σ_1^2 to within γ^5, but whose weight differs from w_1 by at least γ. By the bounds on the condition number, there can be at most one such Gaussian component, say the i^{th}. If $w_1 \geq w_i' + \gamma$, then $\mathcal{J}_{\sigma_1^2-\gamma^4}(F)(\mu_1) - \mathcal{J}_{\sigma_1^2-\gamma^4}(F')(\mu_1) \geq \frac{1}{\gamma\sqrt{2\pi}} - \frac{2}{\epsilon\sqrt{2\pi e}}$, where the second term is a bound on the contribution of the other Gaussian components to $\mathcal{J}_{\sigma_1^2-\gamma^4}(F')(\mu_1)$, using the fact that F, F' have condition numbers at most $1/\epsilon$ and Corollary A.4. Since $\gamma < \epsilon/8$, this quantity is at least $\frac{3}{4\gamma\sqrt{2\pi}} > \frac{1}{8\gamma}$.

If $w_1 \leq w_i' - \gamma$, then consider applying $\mathcal{J}_{\sigma_1^2-\gamma^4}$ to the pair of distributions. Using the fact that $\frac{1}{\sqrt{1+x}} \geq 1 - x/2$, and using the fact that $\sigma_i'^2 \leq \sigma_1^2 + \gamma^5$, we have

$$\begin{aligned}
\mathcal{J}_{\sigma_1^2-\gamma^4}(F')(\mu_i') - \mathcal{J}_{\sigma_1^2-\gamma^4}(F)(\mu_i') &\geq \frac{1}{\sqrt{\gamma^4+\gamma^5}\sqrt{2\pi}}(w_1+\gamma) - \frac{1}{\gamma^2\sqrt{2\pi}}w_1 - \frac{2}{\epsilon\sqrt{2\pi e}} \\
&\geq \frac{1-\gamma/2}{\gamma^2\sqrt{2\pi}}(w_1+\gamma) - \frac{1}{\gamma^2\sqrt{2\pi}}w_1 - \frac{2}{8\gamma\sqrt{2\pi e}} \\
&\geq \frac{1}{8\gamma}.
\end{aligned}$$

□

The above lemma guarantees that we can pick some α such that $\mathcal{J}_\alpha(F)$ and $\mathcal{J}_\alpha(F')$ contain Gaussian components whose variances are at least γ^{12}, and whose probability densities differ significantly in the ℓ_∞ norm. We now show that this poly(γ) in the ℓ_∞ norm gives rise to a poly(γ) disparity in one of the first $2(k+n-1)$ raw moments of the distributions. To accomplish this, we first show that there are at most $2(k+n-1)$ zero-crossings of the difference in densities, $f(x) = \mathcal{J}_\alpha(F)(x) - \mathcal{J}_\alpha(F')(x)$, and construct a degree $2(k+n-1)$ polynomial $p(x)$ that always has the same sign as $f(x)$, and when integrated against $f(x)$ is at least poly(γ). We construct this polynomial so that the coefficients are bounded, and this implies that there is some raw moment i (at most the degree of the polynomial) for which the difference between the i^{th} raw moment of $\mathcal{J}_\alpha(F)$ and of $\mathcal{J}_\alpha(F')$ is large.

We start by showing that the difference in density functions, $\mathcal{J}_\alpha(F)(x) - \mathcal{J}_\alpha(F')(x)$, has relatively few zeros, for any α.

Proposition 11.5. *Given $f(x) = \sum_{i=1}^m a_i \mathcal{N}(\mu_i, \sigma_i^2, x)$, the linear combination of m one-dimensional Gaussian probability density functions, such that for $i \neq j$ either $\sigma_i^2 \neq \sigma_j^2$ or $\mu_i \neq \mu_j$ and for all i, $a_i \neq 0$, the number of solutions to $f(x) = 0$ is at most $2(m-1)$. Furthermore, this bounds is tight.*

Using only the facts that quotients of probability density functions of Gaussians are themselves Gaussian density functions and that the number of zeros of a function is at most one more than the number of zeros of its derivative, one can prove that linear combinations of m Gaussians have at most 2^m zeros (see Lemma 11.8). However, since the number of zeros dictates the number of moments that we must consider in our univariate estimation problem, we will use slightly more powerful machinery to prove the tighter bound of $2(m-1)$ zeros. Our proof of Proposition 11.5 will hinge upon the following fact:

Fact 11.6 (See, e.g. [69, 14])**.** *Given $f(x) : \mathbb{R} \to \mathbb{R}$, that is analytic and has n zeros, then for any $\sigma^2 > 0$, the function $g(x) = f(x) \circ \mathcal{N}(0, \sigma^2, x)$ has at most n zeros.*

The intuition for the above fact is that $g(x)$ is the solution to the following differential equation (known as the *heat equation*) for an appropriately chosen value of t:

$$h(x,0) = f(x), \qquad \frac{d}{dt}h(x,t) = \frac{d^2}{dx^2}h(x,t).$$

Intuitively, the above dynamics imply that local optima get smoothed out, rather than reinforced; in particular, for any second order zero of $h(x,t)$ (viewed as a function of x), the dynamics will remove that zero by increasing $h(x,t)$ according to $\frac{d^2}{dx^2}h(x,t)$, rather than creating an additional zero.

The following trivial lemma will be helpful in our proof of Proposition 11.5:

Lemma 11.7. *Given a linear combination of Gaussians, $f(x) := \sum_{i=1}^m a_i \mathcal{N}(\mu_i, \sigma_i^2, x)$, with r zeros, there exists $\epsilon > 0$ such that at least one of the following holds:*

- *For all positive $\epsilon' < \epsilon$, the function $f_{\epsilon'}(x) := (a_1 - \epsilon')\mathcal{N}(\mu_1, \sigma_1^2, x) + \sum_{i=2}^m a_i \mathcal{N}(\mu_i, \sigma_i^2, x)$ has at least r zeros, with $|\frac{d}{dx} f_{\epsilon'}(x)| > 0$ for at least r zeros.*
- *For all positive $\epsilon' < \epsilon$, the function $f_{\epsilon'}(x) := (a_1 + \epsilon')\mathcal{N}(\mu_1, \sigma_1^2, x) + \sum_{i=2}^m a_i \mathcal{N}(\mu_i, \sigma_i^2, x)$ has at least r zeros, with $|\frac{d}{dx} f_{\epsilon'}(x)| > 0$ for at least r zeros.*

Proof. For any analytic function $g(x)$ that is not identically zero, with zeros at $x_1, \ldots, x_r$, there exists $\delta > 0$ such that neither $g(x)$ nor $\frac{d}{dx} g(x)$ have any zeros within any of the sets $[x_i - \delta, x_i + \delta]$. Consider the setting in which

$$|\{i : g(x) \geq 0, \forall x \in [x_i - \delta, x_i + \delta]\}| \geq |\{i : g(x) \leq 0, \forall x \in [x_i - \delta, x_i + \delta]\}|,$$

and set $\alpha > 0$ such that $\alpha \leq |g(x_i \pm \delta)|$. For any $\epsilon > 0$ chosen such that $\max_x (\epsilon \cdot \mathcal{N}(\mu_1, \sigma_1^2, x)) < \alpha$, the function $g(x) + \epsilon \cdot \mathcal{N}(\mu_1, \sigma_1^2, x)$ will have at least r zeros, as the zeros of even multiplicity that are tangent to the axis from the upper half plane will each become a pair of zeros. Additionally, the derivative at at least r zeros will be nonzero.

The setting in which

$$|\{i : g(x) \geq 0, \forall x \in [x_i - \delta, x_i + \delta]\}| < |\{i : g(x) \leq 0 \forall x \in [x_i - \delta, x_i + \delta]\}|$$

yields the corresponding statement with the function $g(x) - \epsilon \cdot \mathcal{N}(\mu_1, \sigma_1^2, x)$, from which the lemma follows. □

Before proving Proposition 11.5, it will be helpful to establish that the number of zeros of a linear combination of Gaussian density functions is bounded.

Lemma 11.8. *The function $f(x) = \sum_{i=1}^m a_i \mathcal{N}(\mu_i, \sigma_i^2, x)$ has at most 2^m zeros.*

Proof. First observe that for $\sigma_1^2 < \sigma_2^2$, for any μ_1, μ_2, we have

$$q(x) := \frac{\mathcal{N}(\mu_1, \sigma_1^2, x)}{\mathcal{N}(\mu_2, \sigma_2^2, x)} = c \cdot \mathcal{N}(\mu, \sigma^2, x),$$

for $\mu = \frac{\sigma_2^2 \mu_1 - \sigma_1^2 \mu_2}{\sigma_2^2 - \sigma_1^2}$, and $\sigma = \frac{\sigma_1^2 \sigma_2^2}{\sigma_2^2 - \sigma_1^2}$, and some constant c.

We begin by proving the lemma in the case that $\sigma_i^2 > \sigma_j^2$ for all $i < j$. We then consider the case that several components have equal variances. In addition to the above fact about the quotients of Gaussians, we will use the following two elementary facts:

1. The number of zeros of any analytic function $g(x)$ is at most one more than the number of zeros of its derivative, $\frac{d}{dx} g(x)$.
2. For some degree d polynomial $p(x)$,

$$\frac{d}{dx} \mathcal{N}(\mu, \sigma^2, x) \cdot p(x) = \mathcal{N}(\mu, \sigma^2, x) \cdot q(x),$$

for some degree $d + 1$ polynomial $q(x)$.

We now iteratively define the functions $f_j(x)$, which will have the form

$$f_j(x) = \sum_{i=1}^{m-j} \mathcal{N}(\mu_{j,i}, \sigma_{j,i}^2, x) \cdot p_{j,i}(x)$$

for some polynomials $p_{j,i}$.

- Let $f_0(x) = f(x) = \sum_{i=1}^{m} a_i \mathcal{N}(\mu_i, \sigma_i^2, x)$.
- For $j \geq 1$, define

$$f_j := \frac{d^{\alpha_j+1}}{dx^{\alpha_j+1}} \left(\frac{f_{j-1}(x)}{\mathcal{N}(\mu, \sigma^2, x)} \right),$$

  where $\mathcal{N}(\mu, \sigma^2, x)$ is the Gaussian component with maximal variance in the expression for f_{j-1}, and α_j is $\max_i(deg(p_{j-1,i}))$.

By the second item above, $p_{j,i}(x)$ is a polynomial with degree at most $\alpha_j + 1$ more than the degree of $p_{j-1,i}(x)$, and since $\alpha_1 = 0$, $\alpha_j \leq 2^{j-1}$. From the first item above, $f_j(x)$ has at most $\alpha_j + 1$ fewer zeros than $f_{j-1}(x)$. Since $f_m = 0$, the number of zeros of $f_0(x)$ is bounded by $\sum_{j=1}^{m} \alpha_j \leq 2^m$, as claimed.

To conclude, we consider a linear combination of m Gaussians, $f(x)$, with the property that $\sigma_i^2 = \sigma_j^2$ for some distinct pair i, j. Assume for the sake of contradiction that $f(x)$ has at least $2m - 1$ zeros. Applying Lemma 11.7 yields that there is another linear combination of at most m Gaussians, $g(x)$, with at least $2m - 1$ zeros and nonzero derivative at these zeros. Since $\mathcal{N}(0, \sigma^2, x)$ is continuous in σ^2, for any sufficiently small $\delta > 0$ the linear combination resulting from modifying $g(x)$ by increasing the variance of one of its components by δ will still have at least $2m - 1$ zeros with nonzero derivatives, and thus one can transform $g(x)$ into a linear combination of m components with distinct variances and having at least $2m-1$ zeros, which contradicts the first part of our proof. □

Proof of Proposition 11.5. We proceed by induction on m. The base case, where $f(x) = a\mathcal{N}(\mu, \sigma^2, x)$ is trivial. The intuition behind the induction step is that we will consider the linear combination of the $m - 1$ Gaussians of largest variance, except with all variances decreased by that of the excluded component, of variance σ^2. The addition of this mth component, as essentially a Dirac delta function will add at most 2 zeros; we can then convolve all components by $\mathcal{N}(0, \sigma^2, x)$ to obtain the original linear combination $f(x)$; Fact 11.6 guarantees that this final convolution cannot increase the number of zeros. To make this sketch rigorous, we must be slightly careful, as we cannot actually add a true delta function.

We now prove the induction step. For ease of exposition, we describe the case in which, for $i < m$, $\sigma_i^2 > \sigma_m^2$; the general case is identical, except that we will add in all the Gaussians of variance equal to σ_m^2, rather than just the single component. Let $f(x) = \sum_{i=1}^{m} a_i \mathcal{N}(\mu_i, \sigma_i^2, x)$ denote the linear combination of m Gaussians. By Lemma 11.8 $f(x)$ has are a finite number of zeros, and hence we may apply Lemma 11.7 to yield that there is some constant $c_1 > 0$ such that for any $\epsilon < c_1$, the linear combination obtained by decreasing a_1 by ϵ will have

at least as many zeros as $f(x)$, with nonzero derivative at all zeros (the argument in the case that a_1 is incremented by ϵ, as in the second case of the statement of the Lemma 11.7 is analogous). Let $g(x) := (a_1 - c_1/2)\mathcal{N}(\mu_1, \sigma_1^2 - \sigma_m^2, x) + \sum_{i=2}^{m-1} a_i \mathcal{N}(\mu_i, \sigma_i^2 - \sigma_m^2, x)$; by our induction hypothesis, $g(x)$ has at most $2(m-2)$ zeros. As this number is finite, there exists some positive $c_2 < c_1/2$ such that $g_{c_2}(x)$, the linear combination of $m-1$ Gaussians, whose first coefficient is $a_1 - \frac{c_1}{2} - c_2$, has nonzero derivative at all its zeros, and for which $g_{c_2}(\mu_m) \neq 0$, as the zero set of g_ϵ is finite, and the location of each zero is a monotone function of ϵ within any sufficiently small interval around 0.

Define the function

$$h(x, \epsilon) := (a_1 - \frac{c_1}{2} - c_2)\mathcal{N}(\mu_1, \sigma_1^2 - \sigma_m^2 + \epsilon, x) + \sum_{i=2}^{m-1} a_i \mathcal{N}(\mu_i, \sigma_i^2 - \sigma_m^2 + \epsilon, x)$$

and note that there exists some $\epsilon, \delta > 0$ such that for any positive $\epsilon' < \epsilon$, the following conditions hold:

- The magnitude of the derivative of $h(x, \epsilon')$ with respect to x within distance δ of any zero of the function is at least δ.
- The minimum magnitude of $h(x, \epsilon')$ outside of a δ-ball around any zero of the function is at least δ.
- For $x \in [\mu_m - \delta, \mu_m + \delta]$, $|h(x, \epsilon')| > 0$.

The above three conditions guarantee that for a sufficiently small ϵ', the function $h(x, \epsilon') + a_m \mathcal{N}(\mu_m, \epsilon', x)$ has at most 2 more zeros than $h(x, \epsilon')$, and hence at most $2(m-1)$ zeros. Consider the linear combination obtained by convolving each Gaussian with $\mathcal{N}(0, \sigma_m^2 - \epsilon', x)$; by Fact 11.6 and the choice of c_1, c_2, the resulting linear combination has at least as many zeros as the original linear combination $f(x)$, completing the induction step.

To see that this bound is tight, consider

$$f(x) := m\mathcal{N}(0, m^2, x) - \sum_{i=1}^{m-1} \mathcal{N}(i, \frac{1}{100}, x),$$

which is easily seen to have $2(m-1)$ zeros. □

Having bounded the number of zeros of the difference in probability densities of two GMMs, we now argue that we can leverage the ℓ_∞ distance between $\mathcal{J}_\alpha(F)$ and $\mathcal{J}_\alpha(F')$ guaranteed by Lemma 11.4 into a discrepancy between the low order moments of these two mixtures.

Lemma 11.9. *Let F, F' be GMMs of n and k components, respectively, with condition numbers at most $1/\epsilon$, and variances bounded by 2, and consider a positive constant $\gamma \leq$*

$\min\left(\epsilon^2, \frac{1}{100}, \frac{1}{(n+k)^2}\right)$. *Suppose without loss of generality that the Gaussian component of minimal variance is* $\mathcal{N}(\mu_1, \sigma_1^2)$, *and that for every* $i \in \{1, \ldots, k\}$, *at least one of the following holds:*

- $|\mu_1 - \mu_i'| > \gamma^5$
- $|\sigma_1^2 - \sigma_i'^2| > \gamma^5$
- $|w_1 - w_i'| > \gamma$.

Then there is some choice of $\alpha \geq -\gamma^4$ *and some moment* $i \in [2(n+k-1)]$ *such that*

$$|E_{\mathcal{J}_\alpha(F)}[x^i] - E_{\mathcal{J}_\alpha(F')}[x^i]| \geq \frac{\gamma^{16(n+k)}}{4(n+k-1)},$$

and in particular is $\text{poly}(\gamma)$ *for fixed* n, k.

Proof. For some specific choice of α, define the function $f(x) := \mathcal{J}_\alpha(F)(x) - \mathcal{J}_\alpha(F')(x)$. We start by briefly sketching the idea of the proof: the proof will follow from applying Lemma 11.4, which shows that an α can be chosen such that there exists some x^* for which $|f(x^*)|$ is large; additionally, no component of $f(x)$ will have variance that is too small, and hence the magnitude of the derivative of $f(x)$ can be uniformly bounded, and hence the integral of the absolute value of $f(x)$ will be large. We then apply Proposition 11.5 which guarantees that $f(x)$ has few zero crossings, and thus there exists a low degree polynomial $p(x)$ (defined by the zeros of $f(x)$, such that $f(x) \cdot p(x) \geq 0$ for all x, allowing us to bound $\int_{-\infty}^{\infty} f(x) \cdot p(x)$. We then note that this integral is some linear combination of the discrepancies in moments; indeed if $p(x) = x^j$, then this integral would simply be the discrepancy in jth moments. In order to bound the discrepancy in moments, we must ensure that the coefficients of $p(x)$ are not too large.

We now make the above sketch rigorous. Define the interval $I := [-\frac{2}{\gamma}, \frac{2}{\gamma}]$, and assume that there exists $x^* \in [-\frac{3}{2\gamma}, \frac{3}{2\gamma}] \subset I$, constant $b > 0$, and positive $c_1 < c_2$ satisfying:

$$|f(x^*)| \geq \frac{1}{b \cdot \gamma^{c_1}}, \text{ and } \sup_{x \in \mathbb{R}} \left|\frac{d}{dx} f(x)\right| \leq \frac{1}{\gamma^{c_2}}.$$

Consider the interval $J := [x^* - \frac{\gamma^{c_2 - c_1}}{2b}, x^* + \frac{\gamma^{c_2 - c_1}}{2b}] \subset I$, and observe that from our lower bound on $|f(x^*)|$ and our upper bound on the derivative of $|f(x)|$, the following two properties must clearly hold:

1. $f(x)$ has no zeros within the interval $J' := [x^* - \frac{\gamma^{c_2 - c_1}}{b}, x^* + \frac{\gamma^{c_2 - c_1}}{b}] \supset J$,
2. $\min_{x \in J} |f(x)| \geq \frac{|f(x^*)|}{2} \geq \frac{1}{2b\gamma^{c_1}}$,

Define the polynomial $p(x) = \pm\Pi_{z_i}(x - z_i)$ for all zeros $z_i \in I$. We can then choose the sign so that $p(x)f(x) \geq 0$ for any $x \in I$. Thus

$$\int_I p(x)f(x)dx \geq \int_J p(x)f(x)dx.$$

From Proposition 11.5, $f(x)$ has at most $2(n+k-1)$ zero crossings, and thus the polynomial $p(x)$ has degree at most $2(n+k-1)$. Since $f(x)$ has no zeros within J', for all $x \in J$, $|p(x)| \geq \left(\frac{\gamma^{c_2-c_1}}{2b}\right)^{2(n+k-1)}$. Combining this with the second fact above yields that

$$\int_J p(x)f(x)dx \geq \left(\frac{\gamma^{c_2-c_1}}{b}\right)\frac{1}{2b\gamma^{c_1}}\left(\frac{\gamma^{c_2-c_1}}{2b}\right)^{2(n+k-1)} \geq \frac{\gamma^{(c_2-c_1)2(n+k)}}{(2b)^{2(n+k)}}.$$

From the above, and the fact that each coefficient of $p(x)$ is at most $2^{2(n+k-1)}(2/\gamma)^{2(n+k-1)} = \frac{2^{2(n+k-1)}}{\gamma^{2(n+k-1)}}$, we conclude that there is some $i \in [2(n+k-1)]$ such that

$$|\int_I x^i f(x)dx| \geq \frac{1}{2(n+k-1)} \cdot \frac{\gamma^{(c_2-c_1+1)2(n+k)}}{8^{n+k}b^{2(n+k)}}. \tag{11.1}$$

We now consider what values of c_1, c_2 are implied by Lemma 11.4. Noting that

$$\sup_x |\frac{d\mathcal{N}(\mu,\sigma^2,x)}{dx}(x)| \leq \frac{1}{\sigma^2\sqrt{2\pi e}} \leq \frac{1}{2\sigma^2},$$

Lemma 11.4 guarantees that α can be chosen so as to either have $c_1 = 1$, $c_2 = 4$, and $b = 8$, or $c_1 = 5$, $c_2 = 12$, and $b = 1/2$. In either case, Equation 11.1 is at least $\frac{\gamma^{16(n+k)}}{2(n+k-1)}$.

Using Corollary A.8 which gives bounds on the contributions of the tails of Gaussians to the i^{th} moment, since the mixtures F, F' have condition numbers at most $1/\epsilon$ and hence mixing weights at least ϵ, and have variance at most 2, all component means lie within $[-2/\epsilon, 2/\epsilon]$, and variances are at most $2/\epsilon$, and hence we have that $\int_{\mathbb{R}\setminus I} x^i f(x)dx \leq \cdot 4^i i\sqrt{i!}\gamma^{-2i}e^{-\frac{1}{4\gamma^2}}$. The lemma now follows from noting that for γ in the prescribed range this is easily bounded by $\frac{\gamma^{16(n+k)}}{4(n+k-1)}$. □

We now consider what effect the transformation $\mathcal{J}_\alpha$ has on the distance between a pair of GMMs. Unfortunately, the transformation $\mathcal{J}_\alpha$ does not preserve the total variational distance between the two distributions. However, we show that it, at least roughly, preserves (up to a polynomial) the disparity in low-order moments of the distributions.

Lemma 11.10. *Given GMMs F, F', and some $\alpha \leq 1$ that is at most the minimum variance of any Gaussian component of F or F', then*

$$|M_k\left(\mathcal{J}_\alpha(F)\right) - M_k\left(\mathcal{J}_\alpha(F')\right)| \leq 2\frac{(k-1)!}{\lfloor k/2\rfloor!}\sum_{i=1}^{k}|M_i(F) - M_i(F')|,$$

The proof of the above lemma follows easily from the observation that the moments of F and $\mathcal{J}_\alpha(F)$ are related by a simple linear transformation.

Proof. Let X be a random variable with distribution $\mathcal{J}_\alpha\left(\mathcal{N}(\mu, \sigma^2)\right)$, and Y a random variable with distribution $\mathcal{N}(\mu, \sigma^2)$. From definition 11.3 and the fact that the sum of two independent Gaussian random variables is also a Gaussian random variable, it follows that $M_i(Y) = M_i(X + Z)$, where Z is a random variable, independent from X with distribution $\mathcal{N}(0, \alpha)$. From the independence of X and Z we have that

$$M_i(Y) = \sum_{j=0}^{i} \binom{i}{j} M_{i-j}(X) M_j(Z).$$

Since each moment $M_i(\mathcal{N}(\mu, \sigma^2))$ is some polynomial of μ, σ^2, which we shall denote by $m_i(\mu, \sigma^2)$, and the above equality holds for some interval of parameters, the above equation relating the moments of Y to those of X and Z is simply a polynomial identity:

$$m_i(\mu, \sigma^2) = \sum_{j=0}^{i} \binom{i}{j} m_{i-j}(\mu, \sigma^2 - \beta) m_j(0, \beta).$$

Given this polynomial identity, if we set $\beta = -\alpha$, we can interpret this identity as

$$M_i(X) = \sum_{j=0}^{i} \binom{i}{j} M_{i-j}(Y) \left(c_j M_j(Z)\right),$$

where $c_j = \pm 1$ according to whether j is a multiple of 4 or not.

Consider $|M_i\left(\mathcal{J}_\alpha(F)\right) - M_i\left(\mathcal{J}_\alpha(F')\right)|$; from above, and by linearity of expectation, we get

$$\begin{aligned} |M_i\left(\mathcal{J}_\alpha(F)\right) - M_i\left(\mathcal{J}_\alpha(F')\right)| &\leq \sum_{j=0}^{i} \binom{i}{j} \left(M_{i-j}(F) - M_{i-j}(F')\right) c_j M_j(\mathcal{N}(0, \alpha)) \\ &\leq \left(\sum_{j=0}^{i} \binom{i}{j} |M_{i-j}(F) - M_{i-j}(F')|\right) \max_{j \in \{0,1,\ldots,k-1\}} |M_j(\mathcal{N}(0, \alpha))|. \end{aligned}$$

In the above we have used the fact that $M_k(\mathcal{N}(0, \alpha)))$ can only appear in the above sum along with $|M_0(F) - M_0(F')| = 0$. Finally, using the facts that $\binom{i}{j} < 2^j$, and expressions for the raw moments of $\mathcal{N}(0, \alpha)$ given by Equation (A.1), the above sum is at most $2^k \cdot \frac{(k-1)!\alpha^{k/2}}{2^{k-1}\lfloor k/2 \rfloor!} \sum_{i=0}^{k} |M_{j-i}(F) - M_{j-i}(F')|$, which completes the proof. □

We are now equipped to put the pieces together and begin our induction proof of our main technical theorem.

Proof of Theorem 11.2. The base case for our induction is when $n = k = 1$, and follows from the fact that given parameters $\mu, \mu', \sigma^2, \sigma'^2$ such that $|\mu - \mu'| + |\sigma^2 - \sigma'^2| \geq \epsilon$, by definition, one of the first two moments of $\mathcal{N}(\mu, \sigma^2)$ differs from that of $\mathcal{N}(\mu', \sigma'^2)$ by at least $\epsilon/2$.

For the induction step, assume the for all pairs of mixtures where one element of the pair has at most n components, and the other has at most k components, that satisfy the conditions of the theorem, at least one of the first $2(n + k - 1)$ moments differ by at least $f(\epsilon, n+k) = O(\epsilon^{16^{n+k}})$, where the hidden constant is a function of n, k and is independent of ϵ. Consider mixtures F, F', mixtures of n', k' Gaussians, respectively, where either $n' = n+1$, or $k' = k + 1$, and either $n' = n$ or $k' = k$, and assume they satisfy the conditions of the theorem. Assume without loss of generality that σ_1^2 is the minimal variance in the mixtures, and that it occurs in mixture F.

We first consider the case that there exists a component of F' whose mean, variance, and weight match μ_1, σ_1^2, w_1 to within an additive x, where x is chosen so that each of the first $2(n + k)$ moments of any pair of Gaussians whose parameters are within x of each other, differ by at most $f(\epsilon/2, n + k - 1)/2$. Hence by Lemma A.17,it suffices to choose an $x = O(\epsilon^{n+k-1} f(\epsilon/2, n + k - 1))$. Since Lemma A.17 requires that $\sigma_1^2 \geq \sqrt{x}$, if this is not the case, we convolve the pair of mixtures by $\mathcal{N}(0, \epsilon)$, which by Lemma 11.10 changes the disparity in low-order moments by a constant factor (dependent on n, k).

Now, consider the mixtures H, H', obtained from F, F' by removing the two nearly-matching Gaussian components, and rescaling the weights so that they still sum to 1. The pair H, H' will now be mixtures of $k' - 1$ and $n' - 1$ components, and will have condition numbers at most $1/(\epsilon - \epsilon^2)$, and the discrepancy in their first $2(n' + k' - 2)$ moments is at most $f(\epsilon/2, n + k - 1)/2$ different from the discrepancy in the pair F, F'. By our induction hypothesis, there is a discrepancy in one of the first $2(n' + k' - 3)$ moments of at least $f(\epsilon/2, n + k - 1)$ and thus the original pair F, F' will have discrepancy in moments at least half of this.

In the case that there is no component of F' that matches μ_1, σ_1^2, w_1, to within the desired accuracy x, we can apply Lemma 11.4 with $\gamma = x$, and thus by Lemma 11.9 there exists some α such that in the transformed mixtures $\mathcal{J}_\alpha(F), \mathcal{J}_\alpha(F')$, there is a $O(x^{16(j+k)})$ disparity in the first $2(k + n - 1)$ moments. By Lemma 11.10, this disparity in the first $2(k + n - 1)$ moments is related to the disparity in these first $2(k + n - 1)$ moments of the original pair of mixtures, by a constant factor (dependent on j, k). Thus, up to constant factors, we must have $f(x, m) < (f(x/2, m - 2))^{16m}$, and thus taking $f(x, m) = x^{(O(m))^{m/2}}$ suffices. □

11.3 The Basic Univariate Algorithm

In this section we formally state the Basic Univariate Algorithm, and prove its correctness. In particular, we will prove the following corollary to the polynomially robust identifiability of GMMs (Theorem 11.2).

Theorem 11.3. *For $\epsilon < 1/k$, suppose we are given access to independent draws from a*

GMM

$$F = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \sigma_i^2)$$

with condition number $\kappa(F) \leq \frac{1}{\epsilon}$, *with variance in* $[1/2, 2]$. *The Basic Univariate Algorithm, when run with appropriate parameters will have runtime and sample size bounded by* $\epsilon^{(O(k))^k} \log \frac{1}{\delta}$, *and with probability at least* $1 - \delta$ *will output mixture parameters* $\hat{w}_i, \hat{\mu}_i, \hat{\sigma_i}^2$, *such that there is a permutation* $\pi : [k] \to [k]$ *for which*

$$|w_i - \hat{w}_{\pi(i)}| \leq \epsilon, \quad |\mu_i - \hat{\mu}_{\pi(i)}| \leq \epsilon, \quad |\sigma_i^2 - \hat{\sigma}^2_{\pi(i)}| \leq \epsilon \text{ for each } i = 1, \ldots, k \ .$$

Algorithm 11.11. Basic Univariate Algorithm
Input: k, m, γ, ϵ, *probability of failure* δ, *and sample oracle* $\mathrm{SA}(F)$.

1. *Take* m *draws from* $\mathrm{SA}(F)$, *and compute the first* $4k - 2$ *sample moments,* $\hat{m}_1, \ldots, \hat{m_{4k-2}}$.
2. *Iterate through the entire set of candidate parameter vectors of the form* $\tilde{F} = (\tilde{w_1}, \tilde{\mu_1}, \tilde{\sigma_1}^2, \ldots, \tilde{w_k}, \tilde{\mu_k}, \tilde{\sigma_k}^2)$ *satisfying:*
   - *All the elements are multiples of* γ,
   - $\tilde{w_i} \geq \epsilon/2$, *and* $\sum_i \tilde{w_i} = 1$,
   - *each pair of components has parameter distance at least* $\epsilon/2$.
   - $|\tilde{\mu_i}|, |\tilde{\sigma_i}^2| \leq 2/\epsilon$.
3. *Compute the first* $4k - 2$ *moments of mixture* $\tilde{F}$, $\tilde{m}_1, \ldots, \tilde{m_{4k-2}}$.
4. *If for all* $i \in \{1, \ldots, 4k-2\}$, $|\tilde{m}_i - \hat{m}_i| \leq \alpha$, *then return* $\tilde{F}$, *which will have the property that each returned parameters will match the corresponding true parameters to within* $\epsilon/2$, *with high probability.*

If the above algorithm outputs $\epsilon/2$*-accurate parameters with probability of success* > 0.9, *to boost the probability of success to* $1 - \delta$, *repeat the entire previous algorithm* $\log \frac{1}{\delta}$ *times; letting* $\tilde{F}_i$ *denote the parameter set returned by the* i*th run, for each candidate parameter vector* (μ, σ^2, w) *given in a* $\tilde{F}_i$, *output that parameter vector if there are at least* $\frac{1}{4} \log \frac{1}{\delta}$ *runs for which* $\tilde{F}_i$ *contains a component whose mean, variance, and mixing weight all match to within* $\epsilon/2$, *and for which no parameters matching to within* ϵ *has previously been output.*

Our proof of the above theorem will follow from these three step: first, basic concentration bounds will show that with high probability, the first $4k - 2$ sample moments will be close to the corresponding true moments. Next, we show that it suffices to perform a brute-force search over a polynomially–fine mesh of parameters in order to ensure that at least one point $(\hat{w_1}, \hat{\mu_1}, \hat{\sigma_1}^2, \ldots, \hat{w_k}, \hat{\mu_k}, \hat{\sigma_k}^2)$ in our parameter-mesh will have moments that are each sufficiently close to those of the true parameters. Finally, we will use Theorem 11.2

to conclude that the recovered parameter set $(\hat{\mu_1}, \hat{\sigma_1}^2, \ldots, \hat{\mu_k}, \hat{\sigma_k}^2)$ must be close to the true parameter set, because the first $4k-2$ moments nearly agree. We now formalize these pieces.

Lemma 11.12. *Let $x_1, x_2, \ldots, x_m$ be independent draws from a univariate GMM F with variance at most 2, and each of whose components has weight at least ϵ. With probability $\geq 1-\beta$,*

$$\left| \frac{1}{m} \sum_{i=1}^{m} x_i^k - \mathrm{E}_{x \sim F}[x^k] \right| \leq \frac{1}{m\beta^2} O(\epsilon^{-2k}),$$

where the hidden constant in the big-Oh notation is a function of k.

Proof. By Chebyshev's inequality, with probability at least $1-\beta$,

$$\left(\frac{1}{m} \sum_{i=1}^{m} x_i^j - \mathrm{E}_{x \sim F}[x^j] \right)^2 \leq \frac{1}{\beta} \mathrm{E} \left[\left(\frac{1}{m} \sum_{i=1}^{m} x_i^j - \mathrm{E}_{x \sim F}[x^j] \right)^2 \right].$$

We now bound the right hand side. By definition, $\mathrm{E}_{x_1,\ldots,x_m} \left[\frac{1}{m} \sum_{i=1}^{m} x_i^j - \mathrm{E}_{x \sim F}[x^j] \right] = 0$. Since the variance of a sum of independent random variables is the sum of the variances,

$$\begin{aligned} \mathrm{E} \left[\left(\frac{1}{m} \sum_{i=1}^{m} x_i^j - \mathrm{E}_{x \sim F}[x^j] \right)^2 \right] &= \frac{1}{m} \mathrm{E}_{x \sim F} \left[\left(x^j - \mathrm{E}_{x \sim F}[x^j] \right)^2 \right] \\ &\leq \frac{1}{m} \mathrm{E}_{x \sim F}[x^{2j}]. \end{aligned}$$

To conclude, we give a very crude upper bound on the qth moment of F; since F has variance at most 2 and mixing weights at least ϵ, the mean and variance of each component has magnitude at most $2/\epsilon$. Since the qth moment of a $\mathcal{N}(\mu, \sigma^2)$ is a polynomial in μ, σ of total degree at most q, and coefficients given by a function of q (see Claim A.16), $\mathrm{E}_{x \sim F}[x^q]$ can be bounded by $O((2/\epsilon)^q)$ where the constant in the big-Oh notation hides a function of q. □

We now argue that a polynomially-fine mesh suffices to guarantee that there is some parameter set in our mesh whose first $4k-2$ moments are all close to the corresponding true moments.

Lemma 11.13. *Given a GMM F with k components, whose means and variances are bounded in magnitude by $2/\epsilon$, and weights are least ϵ, for $\gamma = \alpha \cdot \epsilon^{4k}$ there exists a GMM $\hat{F}$ with at most k components, all of whose components' means, variances, and mixing weights multiples of γ, such that each of the first $2(2k-1)$ moments of F and $\hat{F}$ differ in magnitude by at most $O(\alpha)$, where the hidden constant is a function of k.*

Proof. Consider $\hat{F}$ defined by rounding the means, variances, and weights of the components of F (ensuring that the weights still sum to 1). As above, the i^{th} moment of each component is a polynomial in μ, σ of total degree at most i, and coefficients bounded by $i!$. Thus changing the mean or variance by at most γ will change the ith moment by at most.

$$\begin{aligned} i! \cdot i \left((2/\epsilon + \gamma)^i - (2/\epsilon)^i\right) &\leq i^i (2/\epsilon)^i \left((1 + \gamma\epsilon/2)^i - 1\right) \\ &\leq i^i (2/\epsilon)^i (e^{2i\gamma\epsilon} - 1) \\ &\leq (2i/\epsilon)^i (4i\gamma\epsilon), \text{ since } e^x - 1 < 2x \text{ for } x < 1/2 \end{aligned}$$

Thus if we used the true mixing weights, the error in each moment of the entire mixture would be at most k times this. To conclude, note that for each mixing weight $|w_j - \hat{w}_j| \leq \gamma$, and since, as noted in the proof of the previous lemma, each moment is at most $O(\epsilon^{-i})$ (where the hidden constant depends on i), thus the rounding of the weight will contribute at most an extra $O(\gamma\epsilon^{-i})$. Adding these bounds together, yields the lemma. □

We now piece together the above two lemmas to prove Theorem 11.3.

Proof of Theorem 11.3. Given a desired moment accuracy $\alpha \leq \epsilon$, by applying a union bound to Lemma 11.12, $O(\alpha\epsilon^{-8k}\delta^{-2})$ examples suffices to guarantee that with probability at least $1-\delta$, the first $4k-2$ sample moments are within α from the true moments. By Lemma 11.13, setting $\gamma = \alpha\epsilon^{O(k)}$ yields mesh of parameters that are multiples of γ and suffice to recover a set of parameters $(\hat{w}_1, \hat{\mu}_1, \hat{\sigma_1}^2, \ldots, \hat{w}_k, \hat{\mu}_k, \hat{\sigma_k}^2)$ whose first $4k-2$ sample moments will all be within α from the sample moments, and hence within 2α from the true moments, with probability at least $1-\delta$.

To conclude, note that the pair of mixtures $F, \hat{F}$, have condition numbers at most $2/\epsilon$, and thus if their first $4(k-1)$ moments agree to within the accuracy specified by Theorem 11.2, the theorem will guarantees that the recovered parameters must be accurate to within ϵ; thus it suffices to set $\alpha = \epsilon^{(O(k))^k}$. □

11.4 Exponential Dependence on k is Inevitable

In this section, we present a lower bound, showing that an exponential dependence on the number of Gaussian components in each mixture is necessary, even for mixtures in just one dimension. We show this by giving a simple construction of two 1-dimensional GMMs, F_1, F_2 that are mixtures of at most m Gaussians, have condition numbers at most $2m$, and the parameter distance between the pair of distributions is at least $1/(2m)$, but nevertheless $D_{tv}(F_1, F_2) \leq e^{-\Theta(m)} = e^{-\Theta(\kappa(F))}$, for sufficiently large m. The construction hinges on the inverse exponential (in $k \approx \sqrt{m}$) total variational distance between $\mathcal{N}(0, 2)$, and the mixtures of infinitely many Gaussians of unit variance whose components are centered at multiples of $1/k$, with the weight assigned to the component centered at i/k being given by $\mathcal{N}(0, 1, i/k)$. Verifying that this is true is a straight-forward exercise in Fourier analysis. The final construction truncates the mixture of infinitely many Gaussians by removing all

the components with centers a distance greater than k from 0. This truncation clearly has negligibly small effect on the distribution. Finally, we alter the pair of distributions by adding to both distributions, Gaussian components of equal weight with centers at $-k, (-k^2+1)/k, (-k^2+2)/k, \ldots, k$, which ensures that in the final pair of distributions, all components have significant weight.

Proposition 11.14. *There exists a pair F_1, F_2 of mixtures of at most k^2+1 Gaussians with $\kappa(F_1), \kappa(F_2) \leq 4k^2+2$, and parameter distance $D_{par}(F_1, F_2) \geq \frac{1}{4k^2+2}$ but for which*

$$D_{tv}(F_1, F_2) \leq 11ke^{-k^2/24}.$$

The following lemma will be helpful in the proof of correctness of our construction.

Lemma 11.15. *Let $H_k(x) := c_k \sum_{i=-\infty}^{\infty} \mathcal{N}(0, 1/2, i/k)\mathcal{N}(i/k, 1/2, x)$, where c_k is a constant chosen so as to make H_k a distribution.*

$$||H_k(x), \mathcal{N}(0, 1, x)||_1 \leq 10ke^{-k^2/24}.$$

Proof. The probability density function $H_k(x)$ can be rewritten as

$$H_k(x) = \left(c_k C_{1/k}(x)\mathcal{N}(0, 1/2, x)\right) \circ \mathcal{N}(0, 1/2, x),$$

where $C_{1/k}(x)$ denotes the infinite comb function, consisting of delta functions spaced a distance $1/k$ apart, and $\circ$ denotes convolution. Considering the Fourier transform, we see that

$$\hat{H}_k(s) = c_k k \left(C_k(s) \circ \mathcal{N}(0, 2, s)\right) \mathcal{N}(0, 2, s).$$

It is now easy to see that why the lemma should be true, since the transformed comb has delta functions spaced at a distance k apart, and we're convolving by a Gaussian of variance 2 (essentially yielding nonoverlapping Gaussians with centers at multiples of k), and then multiplying by a Gaussian of variance 2. The final multiplication will nearly kill off all the Gaussians except the one centered at 0, yielding a Gaussian with variance 1 centered at the origin, whose inverse transform will yield a Gaussian of variance 1, as claimed.

To make the details rigorous, observe that the total Fourier mass of $\hat{H}_k$ that ends up within the interval $[-k/2, k/2]$ contributed by the delta functions aside from the one at the origin, even before the final multiplication by $\mathcal{N}(0, 2)$, is bounded by the following:

$$\begin{aligned} 2c_k k \sum_{i=1}^{\infty} \int_{(i-1/2)k}^{\infty} \mathcal{N}(0, 2, x)dx &= 2c_k k \sum_{i=1}^{\infty} \int_{(i-1/2)k/\sqrt{2}}^{\infty} \mathcal{N}(0, 1, x)dx \\ &\leq 2c_k k \sum_{i=1}^{\infty} \frac{1}{\sqrt{\pi}(i-1/2)k} e^{-(i-1/2)^2k^2/2} \\ &\leq 4c_k e^{-k^2/8} \leq 4e^{-k^2/8}. \end{aligned}$$

Additionally, this ℓ_1 fourier mass is an upper bound on the ℓ_2 Fourier mass. The total ℓ_1 Fourier mass (which bounds the ℓ_2 mass) outside the interval $[-k/2, k/2]$ contributed by the delta functions aside from the one at the origin is bounded by

$$\begin{aligned} 2c_k \int_{k/2}^{\infty} 2 \max_y(\mathcal{N}(0,2,y))\mathcal{N}(0,2,x)dx &\leq 4c_k \int_{k/2}^{\infty} \mathcal{N}(0,2,x)dx \\ &\leq 4c_k \int_{k/(2\sqrt{2})}^{\infty} \mathcal{N}(0,1,x)dx \\ &\leq 4c_k \frac{2}{k\sqrt{\pi}} e^{-k^2/8} \leq 4\frac{2}{k\sqrt{\pi}} e^{-k^2/8} \end{aligned}$$

Thus we have that

$$\begin{aligned} ||\hat{H}_k - c_k k\mathcal{N}(0,2)\mathcal{N}(0,2)||_2 &= ||\hat{H}_k - c_k k\frac{1}{2\sqrt{2\pi}}\mathcal{N}(0,1)||_2 \\ &\leq 4e^{-k^2/8} + 4\frac{2}{k\sqrt{\pi}} e^{-k^2/8} \end{aligned}$$

From Plancherel's Theorem: H_k, the inverse transform of $\hat{H}_k$, is a distribution, whose ℓ_2 distance from a single Gaussian (possibly scaled) of variance 1 is at most $8e^{-k^2/8}$. To translate this ℓ_2 distance to ℓ_1 distance, note that the contributions to the ℓ_1 norm from outside the interval $[-k, k]$ is bounded by $4\int_k^{\infty} \mathcal{N}(0,1,x)dx \leq 4\frac{1}{k\sqrt{2\pi}} e^{-k^2/2}$. Since the magnitude of the derivative of $H_k - c_k k\frac{1}{2\sqrt{2\pi}}\mathcal{N}(0,1)$, is at most 2 and the value of $H_k(x) - c_k k\frac{1}{2\sqrt{2\pi}}\mathcal{N}(0,1,x)$ is close to 0 at the endpoints of the interval $[-k, k]$, we have

$$\left(\max_{x\in[-k,k]}(|H_k(x) - c_k k\frac{1}{2\sqrt{2\pi}}\mathcal{N}(0,1,x)|)\right)^3/(12) \leq \int_{-k}^{k} |H_k(x) - c_k k\mathcal{N}(0,1,x)|^2 dx,$$

which, combined with the above bounds on the ℓ_2 distance, yields $\max_{x\in[-k,k]}(|H_k(x) - c_k k\frac{1}{2\sqrt{2\pi}}\mathcal{N}(0,1,x)|) \leq (72e^{-k^2/8})^{1/3}$. Thus we have

$$||H_k(x) - c_k k\frac{1}{2\sqrt{2\pi}}\mathcal{N}(0,1,x)||_1 \leq 4\frac{1}{k\sqrt{2\pi}} e^{-k^2/2} + (2k)(72e^{-k^2/8})^{1/3}.$$

The lemma follows from the additional observation that

$$||\mathcal{N}(0,1) - c_k k\frac{1}{2\sqrt{2\pi}}\mathcal{N}(0,1)||_1 = \min_{p(x)}(||c_k k\frac{1}{2\sqrt{2\pi}}\mathcal{N}(0,1) - p(x)||_1),$$

where the minimization is taken to be over all functions that are probability density functions. □

Proof of Proposition 11.14. We will construct a pair of GMMs F_1, F_2, that are mixtures of $k^2 + 1$ Gaussians, whose total variational distance is inverse exponential in $O(k^2)$, yet whose

condition numbers are $O(k^2)$, and parameter distances are at least $1/2k$. Let

$$F_1 = \frac{1}{2}\mathcal{N}(0,1/2) + \frac{1}{2(2k^2+1)}\sum_{i=-k^2}^{k^2}\mathcal{N}(i/k,1/2),$$

$$F_2 = \frac{1}{2}c'_k\sum_{i=-k^2}^{k^2}\mathcal{N}(0,1/2,i/k)\mathcal{N}(i/k,1/2) + \frac{1}{2(k^2+1)}\sum_{i=-k^2}^{k^2}\mathcal{N}(i/k,1/2),$$

where c'_k is a constant chosen so as to make $c'_k\sum_{i=-k^2}^{k^2}\mathcal{N}(0,1/2,i/k)\mathcal{N}(i/k,1/2)$ a distribution. Clearly the pair of distributions has condition number at least $4k^2+2$, since all weights are at least $1/(4k^2+2)$, and the components have means that differ by $1/k$. Finally, the Gaussian component of F_1 centered at 0 can not be paired with any component of F_2 without having a discrepancy in parameters of at least $1/2k$.

We now argue that F_1, F_2 are close in variational distance. Let

$$F'_2 = c'_k\sum_{i=-k^2}^{k^2}\mathcal{N}(0,1/2,i/k)\mathcal{N}(i/k,1/2).$$

Note that $\int_k^\infty H_k(x)dx \leq \int_k^\infty \mathcal{N}(0,1/2,x)2\max_y(\mathcal{N}(0,1/2,y))dx \leq \frac{2\sqrt{2}}{k\sqrt{\pi}}e^{-k^2} \leq 2e^{-k^2}$, and thus $||F'_2 - H_k||_1 \leq 8e^{-k^2}$, and our claim follows from Lemma 11.15. □

Chapter 12

Learning Mixtures of Gaussians in High Dimension

We describe a polynomial-time GMM learning algorithm—we emphasize that throughout, we favor clarity of presentation and ease of analysis at the cost of impracticality. The algorithm is based on the random projection method (see, e.g., [133]). Since the projection of a multivariate Gaussian onto one dimension is a Gaussian distribution, the projection of a GMM is a GMM in one dimension. Roughly, our algorithm proceeds by projecting the data onto a sequence of vectors, solving the associated sequence of one–dimensional learning problems, and then reconstructing the solution to the original high–dimensional GMM problem, as depicted in Figure 12.1.

There are several obstacles that must be surmounted to consummate this approach. First and foremost, is the question of solving the problem in one dimension, which we have accomplished in Chapter 11. Supposing one has an efficient algorithm for the one dimensional problem, a second obstacle in our high–level approach is ensuring that the projected data that are given as inputs to the one–dimensional algorithm are meaningful. Consider, for example, a GMM that consists of two Gaussian components that have identical covariances, but different means. If, unluckily, we project the data onto a vector orthogonal to the difference in the means, then the resulting one–dimensional mixture will have just a single component. Further complicating this concern is the existence of GMMs, such as that depicted in Figure 12.2, for which two or more essentially non-overlapping components will, with very high probability, project to nearly identical Gaussians in a random projection. How can we hope to disentangle these components if, in nearly every projection, they are indistinguishable?

We demonstrate that this problem can only arise for mixtures in which some components are extremely "skinny" (i.e. have small minimum eigenvalue of their covariance matrices), in comparison with the overall covariance of the mixture. When this condition holds, however, we show that a clustering-based approach will be successful. Specifically, when this condition is met, we will be able to partition the sample points into two sets, such that the Gaussian components from which the sample points in the first set were drawn are (nearly) disjoint

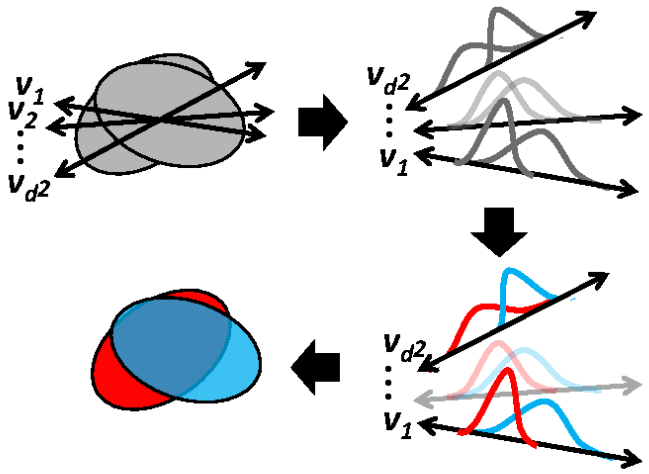

Figure 12.1: Illustration of the high–level approach: 1. project the data onto a series of vectors and learn the parameters of the resulting one dimensional GMMs, 2. determine a consistent labeling between the components of the recovered one dimensional GMMs, and 3. for each component, combine the recovered one dimensional parameters to reconstruct an estimate of the high dimensional parameters.

from the set of components from which the second set of sample points were drawn. Thus this partition of the sample corresponds to a partition of the GMM into two sub-mixtures; we can then apply our algorithm recursively to each of these two sets.

For clarity of exposition, in Section 12.1 we first describe a simplified version of our algorithm that does not require the clustering and recursion steps, though has slightly weaker performance guarantees. In Section 12.2, we describe the full algorithm.

Statement of Main Theorem

The input to our algorithm is a sample set of n points in d dimensions, drawn independently from GMM $F = \sum_{i=1}^{k} w_i F_i$, where each $F_i = \mathcal{N}(\mu_i, \Sigma_i)$ is a distinct d-dimensional Gaussian with mean $\mu_i \in \mathbb{R}^d$ and covariance matrix $\Sigma_i \in \mathbb{R}^{d \times d}$.

To measure the distance between Gaussians $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu', \Sigma')$, we employ the total variational distance, as opposed to the discrepancy in parameters as in Chapter 11. The main strength of total variational distance as a metric is that it measures the information theoretic similarity of two distributions. As such, total variational distance is scale invariant and affine invariant. This will prove particularly useful in this chapter, as we will perform several affine transformations as we peel apart the GMM in question.

Theorem 12.1. *For every $k \geq 1$, there is a constant, c_k, dependent on k, such that the following holds: for any $\epsilon, \delta > 0$, and d-dimensional GMM $F = \sum_{i=1}^{k'} w_i F_i$ with $k' \leq k$ components, and $n > \left(\frac{d}{\epsilon}\right)^{c_k} \log \frac{1}{\delta}$, the estimation algorithm when run on a sample consisting*

of n points independently drawn from F, outputs GMM $\hat{F} = \sum_{i=1}^{\hat{k}} \hat{w}_i \hat{F}_i$ such that, with probability $\geq 1 - \delta$, the following conditions hold:

- $D_{tv}(F, \hat{F}) \leq \epsilon$.
- *If, for all $i \in [k']$, $w_i > \epsilon$, and for all $i \neq j$, $D_{tv}(F_i, F_j) > \epsilon$, then $\hat{k} = k'$ and there is a permutation π of $[k']$ such that for all $i \leq k'$:*

$$|w_i - \hat{w_{\pi(i)}}| \leq \epsilon, \text{ and } D_{tv}(F_i, \hat{F_{\pi(i)}}) \leq \epsilon.$$

Additionally, the runtime of the algorithm is polynomial in the sample size, n.

While our success metric is affine-invariant, our proofs will need to refer to specific discrepancies in parameters. We will often work with the discrepancy in covariance matrices in terms of the Frobenius distance, which we define below. While the Frobenius norm is clearly not invariant to scaling, it is invariant to orthonormal changes of basis.

Definition 12.1. *The* Frobenius norm *of a real-valued matrix A is defined as*

$$||A||_{Fr} := \sqrt{\sum_{i,j} A_{i,j}^2}.$$

Additionally, the Frobenius distance between matrices A, B is defined to be $D_{Fr}(A, B) := ||A - B||_{Fr}$.

12.1 A Simple Algorithm

We start by describing a simple algorithm that illustrates our approach to learning GMMs. While the performance guarantees of this algorithm are slightly weaker than those of the full algorithm, described in Section 12.2, the mechanics of the reduction of the high–dimensional problem into a series of one–dimensional learning problems is more clear.

The goal is to obtain, for each component in the mixture, an estimate of this component's mean and covariance matrix when projected onto many different directions. For each component, we can then use these estimates to set up a system of linear constraints on the high–dimensional mean and covariance matrix of the component. This system can then be solved, yielding good estimates for these high–dimensional parameters.

We choose a vector v uniformly at random from the unit sphere and d^2 perturbations $v_{1,1} \ldots, v_{d,d}$ of v. For each direction $v_{i,j}$, we project the mixture onto direction $v_{i,j}$ and run our one–dimensional learning algorithm. Hence we obtain a set of d^2 parameters of one–dimensional mixtures of k Gaussians. We must now label the components of these d^2 mixtures consistently, such that for each $i = 1, \ldots, k$, the ith Gaussian component in one of the d^2 one–dimensional mixtures corresponds to the projection of the same high-dimensional component as the ith component of all the other one–dimensional mixtures.

In the general setting to which the full-algorithm of Section 12.2 applies, we will not always be able to do this consistent labeling. For the purposes of our simple high–dimensional learning algorithm, we will assume two conditions that will make consistent labeling easy. Specifically, we assume that all pairs of components differ in total variational distance by at least ϵ, and that all components are sufficiently "fat"—the minimum eigenvalue of any components' covariance matrix is bounded below. These conditions together imply that the parameters of each pair of components must differ significantly (by some $poly(\epsilon)$).

We then show that if each pair of components – say, $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu', \Sigma')$ – has either $||\mu - \mu'||_2 > \gamma$, or $||\Sigma - \Sigma'||_{Fr} > \gamma$, then with high probability over a randomly chosen direction v, the projected means or projected variances will differ by at least poly$(\gamma, \frac{1}{d})$. Thus we will be able to run our basic univariate algorithm, Algorithm 11.11, on the projected GMM. Additionally, this discrepancy in the parameters of the projected mixtures makes consistent labeling easy. Since these parameters differ in our projection onto v, and each $v_{i,j}$ is a small perturbation of v, the projection of any high-dimensional component onto v and $v_{i,j}$ will be similar, and thus we simply need to ensure that with high probability the discrepancy in the projected components will be significantly larger than the variation in parameters of components between their projection onto v and $v_{i,j}$.

For each component in the original mixture, after labeling, we have an estimate of the component's mean and variance when projected onto each direction $v_{i,j}$. The one–dimensional parameters of the projection of a Gaussian are related to the high–dimensional parameters by a system of linear equations. We show in Lemma 12.9 that this system is sufficiently well conditioned so as ensure that if our one–dimensional estimates are sufficiently accurate, we can solve this system to obtain good estimates for the high–dimensional parameters.

Algorithm 12.2. THE SIMPLE HIGH DIMENSIONAL ALGORITHM
Given a sample from a GMM in d dimensions with at most k components, target accuracy and probability of failure ϵ, δ:
Let $\epsilon_2 = \left(\frac{\epsilon}{d}\right)^{10}$, $\epsilon_3 = (\epsilon_2)^{10}$.

- *Choose a random orthonormal basis $(b_1, \ldots, b_d)$, and let $v := \frac{1}{\sqrt{d}} \sum_{i=1}^{d} b_i$.*
- *For all pairs $i, j \in \{1, \ldots, d\}$, let $v_{i,j} := v + \epsilon_2(b_i + b_j)$.*
- *For all pairs i, j, project the sample points onto $v_{i,j}$, run the* Basic Univariate Algorithm *(Algorithm 11.11) on the resulting one--dimensional data with target accuracy ϵ_3 and probability of failure $< \epsilon_2$, and let $P_{i,j} := (\{\mu_1, \sigma_1, w_1\}, \ldots)$ be the returned parameters.*
- *For each $m = 1, 2, \ldots, k$ let $\mu_m^{(0,0)}, \sigma_m^{(0,0)}, w_m^{(0,0)}$ be the recovered parameters of the mth component of $P_{0,0}$. For each pair $i, j \geq 1$ let $\mu_m^{(i,j)}, \sigma_m^{(i,j)}, w_m^{(i,j)}$ be the recovered parameters from $P_{i,j}$ of the component whose parameters are closest, in Euclidean distance, to $(\mu_m^{(0,0)}, \sigma_m^{(0,0)}, w_m^{(0,0)})$.*
- *For each $m = 1, \ldots, k$, let $\widehat{w_m} = median(w_m^{(i,j)})$, let $\widehat{\mu_m}, \widehat{\Sigma_m}$ be the output of running* RECONSTRUCT *(Algorithm 12.3) on input $\mu_m^{(i,j)}, \sigma_m^{(i,j)}, \epsilon_2$*
- *If $w_i < \epsilon$, disregard component i.*

If the above algorithm outputs $\epsilon/2$-accurate parameters with probability of success > 0.9, to boost the probability of success to $1 - \delta$, repeat the entire previous algorithm $\log \frac{1}{\delta}$ times; letting $\tilde{F}_i$ denote the parameter set returned by the ith run, for each candidate parameter vector (μ, Σ, w) given in a $\tilde{F}_i$, output that parameter vector if there are at least $\frac{1}{4} \log \frac{1}{\delta}$ runs for which $\tilde{F}_i$ contains a component whose mixing weight matches to within $\epsilon/2$, and that is at most $\epsilon/2$ far in total variational distance, and for which no component whose total variational distance is within ϵ has previously been output.

Algorithm 12.3. RECONSTRUCT
Given basis $B = (b_1, \ldots, b_d)$, *and for all pairs* $i, j \in [d] \cup \{(0,0)\}$, $\mu^{(i,j)}$, $\sigma^{(i,j)}$ *corresponding to projections onto vectors* $v^{i,j} := v + \epsilon(b_i + b_j)$*:*

- *Define*
$$\widehat{\mu} := \sum_i \frac{\mu^{(i,i)} - \mu^{(0,0)}}{2\epsilon} b_i.$$
- *Define* $S^i := \frac{1}{d}\sum_{j=1}^d \sigma^{(i,j)}$, *and* $S := \frac{1}{d^2}\sum_{i,j=1}^d \sigma^{(i,j)}$.
- *Define matrix* V *by setting the* i,j*th entry to be*
$$V_{i,j} := \frac{\sqrt{d}(S - S^i - S^j)}{2\epsilon^2(\epsilon + \sqrt{d})} - \frac{\sigma^{(i,i)} + \sigma^{(j,j)}}{4\epsilon^2(\epsilon + \sqrt{d})} - \frac{S}{2\epsilon\sqrt{d}} + \frac{\sigma^{(i,j)}}{2\epsilon^2}.$$
- *Output* $\widehat{\mu}, \widehat{\Sigma}$, *where*
$$\widehat{\Sigma} := B\left(\arg\min_{M \succeq 0} ||M - V||_{Fr}\right) B^\dagger,$$
is obtained by projecting V *onto the set of positive semidefinite (symmetric) matrices and then changing basis from* B *to the standard basis.*

Performance of the Simple Algorithm

The following proposition characterizes the performance of the *Simple High Dimensional Algorithm* (Algorithm 12.2).

Proposition 12.4. *There exists a constant,* c_k *dependent on* k*, such that given* n *independent draws from a GMM* $F = \sum_{i=1}^{k'} w_i F_i$*, with* $k' \leq k$ *components in* d *dimensions, with probability at least* $1 - \delta$ *the simple high dimensional algorithm, when run on inputs* k, ϵ, δ *and the* n *sample points, will return a GMM* $\hat{F} = \sum_{i=1}^{k'} \hat{w}_i \hat{F}_i$ *such that there exists a labeling of the components such that for all* i *:*

$$|w_i - \hat{w}_i| \leq \epsilon, \qquad D_{tv}(F_i, \hat{F}_i) \leq \epsilon \qquad ||\mu_i - \hat{\mu}_i|| \leq \epsilon, \qquad ||\Sigma_i - \hat{\Sigma}_i||_{Fr} < \epsilon,$$

provided:

- *for all* $i \leq k'$, $w_i > \epsilon$,
- *for all* $i \leq k'$, *for* $F_i = \mathcal{N}(\mu_i, \Sigma_i)$, *the minimum eigenvalue* Σ_i *is at least* ϵ.
- *for all* $i, j \leq k'$, $D_{tv}(F_i, F_j) > \epsilon$,
- $n > \left(\frac{d}{\epsilon}\right)^{c_k} \log \frac{1}{\delta}$,
- *the covariance of the mixture has all eigenvalues in the interval* $[1/2, 2]$.

The runtime of the estimation algorithm is polynomials in the size of the sample.

The performance guarantees of this algorithm differ from those of our more general algorithm, Algorithm 12.10, in two senses: first, the above algorithm requires that all mixing weights and pairwise variational distances between components is at least ϵ, and secondly, the above algorithm requires a lower bound on the minimum eigenvalue of each component's covariance matrix.

Proposition 12.4 follows from the following sequence of lemmas. First, in Lemma 12.5, we show the basic fact that, given the lower bound on the minimum eigenvalue of each component's covariance, the large total variational distance between components implies that either the mean or covariance are significantly different. Then, in Fact 12.6 and Lemma 12.7, we establish that if two multivariate Gaussians have sufficiently different means or covariances, with high probability their projections onto a randomly chosen unit vector will also have significantly different means and variances, and hence the output of running the basic univariate algorithm on the projections of the high dimensional data will be accurate. We then show that the returned set of 1-dimensional parameters corresponding to the projection of the GMM can be consistently partitioned so that for each high-dimensional component, we have a list of accurate estimates of the projection of the mean and covariance onto the d^2+1 vectors. Finally, in Lemma 12.9, we argue that these accurate estimates can be used to accurately reconstruct the high dimensional mean and covariance for each component. Since, by assumption, each of the components has a covariance matrix whose minimal eigenvalue is lower bounded, as Lemma 12.5 shows, if the recovered means and covariances are close in Euclidean and Frobenius distance, they are also close in total variational distance.

Lemma 12.5. *For $G_1 = \mathcal{N}(\mu_1, \Sigma_1), G_2 = \mathcal{N}(\mu_2, \Sigma_2)$, such that $D_{tv}(G_1, G_2) > \gamma$, if the minimum eigenvalue of Σ_1 is at least λ^*, then:*

$$||\mu_1 - \mu_2|| \geq \frac{\sqrt{\lambda^*}}{\sqrt{2}}\gamma, \text{ or } ||\Sigma_1 - \Sigma_2||_{Fr} \geq \frac{\lambda^*}{8d}\gamma^2.$$

Proof. By Fact A.15, we have

$$\gamma^2 \leq (D_{tv}(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2))^2 \leq \sum_{i=1}^{d}(\lambda_i + \frac{1}{\lambda_i} - 2) + (\mu_1 - \mu_2)^\dagger \Sigma_1^{-1}(\mu_1 - \mu_2),$$

where the λ_i are the eigenvalues of $\Sigma_1^{-1}\Sigma_2$. If the discrepancy in the means contributes at least $\gamma^2/2$ to this quantity, as the minimum eigenvalue of Σ_1 is at least λ^*, we have that $\frac{||\mu_1 - \mu_2||^2}{\lambda^*} \geq \frac{\gamma^2}{2}$, implying the claimed discrepancy in the means.

If the discrepancy in means contributes less than $\gamma^2/2$, then the contribution from the eigenvalues of $\Sigma_1^{-1}\Sigma_2$ must contribute at least $\gamma^2/2$, and hence trivially there is some eigenvalue λ_i for which $|\lambda_i - 1| \geq \frac{\gamma^2}{8d}$. Thus letting v be the corresponding unit eigenvector, $\Sigma_2 v = \lambda_i \Sigma_1 v$, hence $||(\Sigma_2 - \Sigma_1)v|| \geq \lambda^*|\lambda_i - 1|$, since the minimum eigenvalue of Σ_1 is bounded below by λ^*. Hence $||\Sigma_1 - \Sigma_2||_{Fr} \geq \lambda^*|\lambda_i - 1|$, from which the lemma follows. □

The following standard fact about random unit vectors guarantees that if the means of two components differ significantly, then with high probability, they will also differ in any projection.

Fact 12.6. *(See Lemma 1 of [45]) For any $\mu_1, \mu_2 \in \mathbb{R}^d$,*

$$\Pr_{v \in \mathbf{S}_{d-1}}\left[|v \cdot (\mu_1 - \mu_2)| \leq \delta \frac{||\mu_1 - \mu_2||}{\sqrt{d}}\right] \leq \delta.$$

The following basic lemma is the analogous fact about covariance matrices.

Lemma 12.7. *For any $d \times d$ symmetric matrix X,*

$$\Pr_{u \in \mathbf{S}_{d-1}}\left[u^\dagger X u \in [-\beta, \beta]\right] \leq \delta, \text{ for } \beta := \frac{||X||_{Fr}\delta^3}{27d^2}.$$

Proof. Consider drawing $v = (v_1, \ldots, v_d)$ from the spherical Gaussian, by choosing each coordinate independently from $\mathcal{N}(0,1)$. Consider $v^{dag} X v = \sum_{i,j} X_{i,j} v_i v_j$, and consider some pair i, j for which $X_{i,j} = \alpha \neq 0$. Fixing the choice of v_k for all $k \neq i, j$, the contribution of v_i, v_j to $v^{dag} X v$ is $v_i^2 X_{i,i} + v_j^2 X_{j,j} + 2\alpha v_i v_j + v_i \sum_{k \neq i,j} v_k X_{i,k} + v_j \sum_{k \neq i,j} v_k X_{j,k}$, whose derivative, with respect to v_i, is

$$2v_i X_{i,i} + 2\alpha v_j + \sum_{k \neq i,j} v_k X_{i,k}.$$

For any choice of v_i,

$$\Pr_{v_j}\left[|2v_i X_{i,i} + 2\alpha v_j + \sum_{k \neq i,j} v_k X_{i,k}| < \alpha\delta\right] < \delta,$$

and given such a v_j, for any interval I of length $\alpha\delta^2$,

$$\Pr_{v_i}[v^{dag} X v \in I] \leq \delta.$$

The distribution of v is related to a random unit vector simply by scaling, thus let $u := \frac{v}{||v||}$, and hence $u^\dagger X u = \frac{v^\dagger X v}{||v||^2}$. To conclude, note that, $E[||v||^2] = d$, and very crudely, $\Pr\left[||v||^2 > \frac{d}{\delta}\right] < \delta$, and hence by a union bound, for any interval I of length $\frac{\alpha\delta^3}{d}$,

$$\Pr[u^\dagger X u \in I] < 3\delta,$$

from which the lemma follows. □

Putting the above three pieces together, we now guarantee that with high probability, each of the recovered sets of parameters for the 1-dimensional mixtures obtained by running the one-dimensional algorithm on the projected data will be accurate.

Lemma 12.8. *With probability* $1-\epsilon$*, the recovered one dimensional parameters,* $\mu_m^{(i,j)}, \sigma_m^{(i,j)}, w_m^{(i,j)}$ *of Algorithm 12.2 will all be accurate to within error* ϵ_3 *of the corresponding true projected parameters, and*

Proof. For each pair of components $w_i\mathcal{N}(\mu_i, \Sigma_i), w_j\mathcal{N}(\mu_j, \Sigma_j)$, by assumption

$$D_{tv}\left(\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\mu_j, \Sigma_j)\right) > \epsilon,$$

and the minimum eigenvalues of their covariance matrices are bounded below by ϵ, hence by Lemma 12.5, either $||\mu_i - \mu_j|| > \epsilon^{3/2}/\sqrt{2}$, or $||\Sigma_i - \Sigma_j||_{Fr} > \frac{\epsilon^3}{8d}$. By Fact 12.6 and Lemma 12.7, in either of these cases, with probability at least $1-\gamma$ over random unit vectors v, the parameters of the projections will differ by at least $\frac{\gamma^3\epsilon^4}{(18)^3 d^3}$. From the bounds on $||\mu||$ and the maximum eigenvalue of Σ_i imposed by the assumption that the mixing weights are at least $1/\epsilon$ and the covariance of the mixture has all eigenvalues in the interval $[1/2, 2]$, it follows that the difference in the discrepancy between projected components in the projection v and the projection $v_{i,j} = v + \epsilon_2(b_k + b_\ell)$ is at most $O(\epsilon^9)$, and hence with probability at least $1 - \epsilon/2$, in all projections, the projected components will have parameter distance at least $\Omega(\frac{\epsilon^7}{d^3})$, in which case the one dimensional algorithm, Algorithm 11.11 when run with probability of failure at most $\frac{\epsilon}{2d^2}$, will return estimates of the claimed accuracy with the desired probability. A union bound assure that the probability of success is at least $1-\epsilon$. Given this accuracy, and the fact that the difference in true parameters between the different projections is at most $O(\epsilon^9)$, the returned 1-dimensional parameters can be consistently partitioned according to the (high dimensional) components to which they belong. □

We now argue that, provided all the recovered parameters of the projected mixtures are sufficiently accurate, the high dimensional parameters of each component can be accurately reconstructed. Given that the returned means are accurate in Euclidean distance, and covariances are accurate in the Frobenius norm, then Lemma 12.5 guarantees that the returned components are close in total variational distance to the actual components, since the minimum eigenvalue of each covariance matrix is lower bounded, by assumption.

Lemma 12.9. *Let* μ, Σ *be a mean and covariance matrix of a* d *dimensional Gaussian; as in Algorithms 12.2 and 12.3, let* $B = (b_1, \ldots, b_d)$ *be an orthonormal basis, and let vector* $v^{0,0} := v := \frac{1}{\sqrt{d}}\sum_i b_i$*, and* $v^{i,j} := v + \epsilon(b_i + b_j)$*. Assume that* $\mu^{i,i}, \sigma^{(i,j)}$ *satisfy, for all* $i, j \in [d] \cup \{(0,0)\}$*,*

$$|v^{i,i} \cdot \mu - \mu^{(i,i)}| < \gamma, \qquad |(v^{i,j})^\dagger \Sigma v^{i,j} - \sigma^{(i,j)}| < \gamma,$$

and define the parameters $\widehat{\mu}, \widehat{\Sigma}$ *as in Algorithm 12.3:*

$$\widehat{\mu} := \sum_i \frac{\mu^{(i,i)} - \mu^{(0,0)}}{2\epsilon} b_i,$$

and $\widehat{\Sigma}$ *defined by letting* $S^i := \frac{1}{d}\sum_{j=1}^d \sigma^{(i,j)}$, *and* $S := \frac{1}{d^2}\sum_{i,j=1}^d \sigma^{(i,j)}$, *and then defining matrix* V *by setting the* i,jth *entry to be*

$$V_{i,j} := \frac{\sqrt{d}(S - S^i - S^j)}{2\epsilon^2(\epsilon + \sqrt{d})} - \frac{\sigma^{(i,i)} + \sigma^{(j,j)}}{4\epsilon^2(\epsilon + \sqrt{d})} - \frac{S}{2\epsilon\sqrt{d}} + \frac{\sigma^{(i,j)}}{2\epsilon^2},$$

and

$$\widehat{\Sigma} := B\left(\arg\min_{M \succeq 0} ||M - V||_{Fr}\right) B^\dagger.$$

Then

$$||\mu - \widehat{\mu}||^2 \leq d\frac{\gamma^2}{\epsilon^2}, \qquad ||\Sigma - \widehat{\Sigma}||_{Fr} \leq 6d\frac{\gamma}{\epsilon^2}.$$

Proof. We first analyze the recovered mean:

$$\begin{aligned}
||\mu - \widehat{\mu}||^2 &= \sum_{i=1}^d (b_i \cdot \mu - b_i \cdot \widehat{\mu})^2 \\
&= \sum_{i=1}^d \left(b_i \cdot \mu - \frac{\mu^{(i,i)} - \mu^{(0,0)}}{2\epsilon}\right)^2 \\
&\leq \sum_{i=1}^d \left(|b_i \cdot \mu - \frac{v^{i,i} \cdot \mu - v^{0,0} \cdot \mu^{i,i}}{2\epsilon}| + \frac{2\gamma}{2\epsilon}\right)^2 \\
&= \sum_{i=1}^d \left(\frac{2\gamma}{2\epsilon}\right)^2 = d\frac{\gamma^2}{\epsilon^2}.
\end{aligned}$$

We now analyze the covariance estimate. Let $\langle x, y\rangle := x^\dagger \Sigma y$, and define the following quantities which will be the true values corresponding to the above approximations:

$$T^{i,j} := \langle v_{i,j}, v_{i,j}\rangle, \qquad T^i := \frac{1}{d}\sum_{j=1}^d T^{i,j}, \qquad T := \frac{1}{d}\sum_i T^i.$$

Expanding $T^{i,j} = \langle v, v\rangle + 2\epsilon\langle v, b_i + b_j\rangle + \epsilon^2\left(\langle b_i, b_i\rangle + \langle b_j, b_j\rangle + 2\langle b_i, b_j\rangle\right)$, we have the following, where we use the fact that $v = \frac{1}{\sqrt{d}}\sum_i b_i$:

$$\begin{aligned}
T^i &= \langle v, v\rangle + 2\epsilon\langle v, b_i\rangle + \frac{2\epsilon}{d}\sum_k \langle v, b_k\rangle + \frac{2\epsilon^2}{d}\sum_k \langle b_i, b_k\rangle + \epsilon^2\langle b_i, b_i\rangle + \frac{\epsilon^2}{d}\sum_k \langle b_k, b_k\rangle \\
&= \langle v, v\rangle + 2\epsilon\langle v, b_i\rangle + \frac{2\epsilon}{\sqrt{d}}\langle v, v\rangle + \frac{2\epsilon^2}{\sqrt{d}}\langle b_i, v\rangle + \epsilon^2\langle b_i, b_i\rangle + \frac{\epsilon^2}{d}\sum_k \langle b_k, b_k\rangle \\
&= \langle v, v\rangle\left(1 + \frac{2\epsilon}{\sqrt{d}}\right) + 2\epsilon\langle v, b_i\rangle\left(1 + \frac{\epsilon}{\sqrt{d}}\right) + \epsilon^2\langle b_i, b_i\rangle + \frac{\epsilon^2}{d}\sum_k \langle b_k, b_k\rangle
\end{aligned}$$

Thus we have:

$$\begin{aligned} T &= \langle v, v\rangle \left(1 + \frac{2\epsilon}{\sqrt{d}}\right) + \frac{2\epsilon}{d}\left(1 + \frac{\epsilon}{\sqrt{d}}\right)\sum_k \langle v, b_k\rangle + \frac{\epsilon^2}{d}\sum_k \langle b_k, b_k\rangle + \frac{\epsilon^2}{d}\sum_k \langle b_k, b_k\rangle \\ &= \left(1 + \frac{4\epsilon}{\sqrt{d}} + \frac{2\epsilon^2}{d}\right)\langle v, v\rangle + \frac{2\epsilon^2}{d}\sum_k \langle b_k, b_k\rangle. \end{aligned}$$

Hence:

$$T - T^i - T^j = \left(-1 + \frac{2\epsilon^2}{d}\right)\langle v, v\rangle - 2\epsilon\left(1 + \frac{\epsilon}{\sqrt{d}}\right)\langle v, b_i + b_j\rangle - \epsilon^2\langle b_i, b_i\rangle - \epsilon^2\langle b_j, b_j\rangle.$$

Some basic manipulation verifies the following expression for the i, jth entry of the covariance matrix Σ, when expressed in the basis B:

$$\langle b_i, b_j\rangle = \frac{1}{2\epsilon^2}T^{i,j} - \frac{1}{4\epsilon(2\epsilon + \sqrt{d})}(T^{i,i} + T^{j,j}) + \frac{\sqrt{d}}{2\epsilon^2(2\epsilon + \sqrt{d})}(T - T^i - T^j) - \frac{1}{2\epsilon\sqrt{d}}T^{0,0}.$$

By assumption $|T^{i,j} - \sigma^{(i,j)}| < \gamma$, and hence $|T^i - S^i| < \gamma$, and $T - S| < \gamma$, and hence from the above, we have

$$|\langle b_i, b_j\rangle - V_{i,j}| < \gamma\left(\frac{1}{2\epsilon^2} + \frac{2}{4\epsilon(2\epsilon\sqrt{d}} + \frac{3}{2\epsilon^2(2\epsilon + \sqrt{d})} + \frac{1}{2\epsilon\sqrt{d}}\right) \leq \frac{3\gamma}{\epsilon^2}.$$

Letting Q denote the covariance matrix Σ in the basis B, $Q := B^\dagger \Sigma B$, we have shown that $||V - Q||_{Fr} \leq 3d\frac{\gamma}{\epsilon^2}$, and hence, by the triangle inequality,

$$\left|\left|\arg\min_{M \succeq 0} ||V - M||_{Fr} - Q\right|\right|_{Fr} \leq 6d\frac{\gamma}{\epsilon^2}.$$

To conclude, note that a change of basis with respect to an orthonormal basis preserves the symmetry and Frobenius norm of a matrix. □

12.2 The Full High Dimensional Algorithm

We now motivate and describe our general algorithm for learning GMMs which, with high probability, returns a mixture whose components are accurate in terms of variational distance (ℓ_1 distance), without any assumptions on the minimum eigenvalue of the covariance matrices of the components. To get an intuitive sense for the types of mixtures for which the simple high–dimensional algorithm fails, consider the mixture of three components depicted in Figure 12.2. The two narrow components are very similar: both their means, and their covariance matrices are nearly identical. With overwhelming probability, the projection of this mixture onto any one–dimensional space will result in these two components becoming

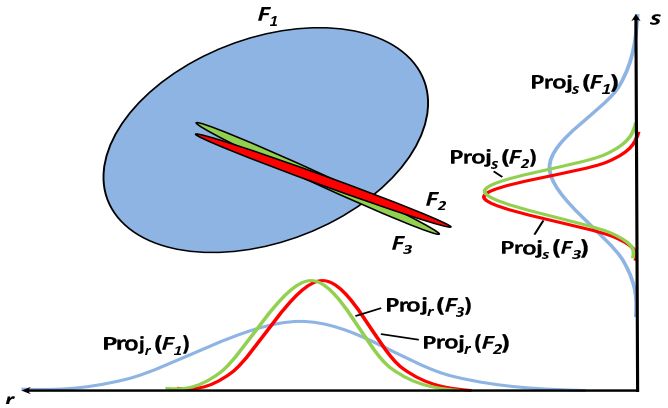

Figure 12.2: An example of a GMM with three components F_1, F_2, F_3, such that with high probability over random vectors, the one dimensional projections of F_2 and F_3 will be very similar, despite $D_{tv}(F_2, F_3) \approx 1$.

indistinguishable given any reasonable amount of data. Nevertheless, the variational distance between these two components is close to one, and thus, information theoretically, we *should* be able to distinguish them.

How can we hope to disentangle these two components if, in nearly every one–dimensional projection, these components are indistinguishable? The intuition for the solution is also provided in the example: we can cluster out these two components and recurse. In particular, there is a vector (corresponding to the direction of small variance of these two components) such that if we project all the data onto this direction, the pair of narrow Gaussians are almost completely "disentangled" from the third component. Almost all of the data corresponding to the two narrow components will be contained within a small interval when projected on this direction, and almost none of the data generated by the third component will be contained in this interval.

If we are able to successfully perform such a clustering of the original mixture into two sub-mixtures, we can recursively apply the entire algorithm to each of the two sub-mixtures. If we consider the sub-mixture corresponding to just the two narrow Gaussians, then we can re-scale the space by applying an affine transformation so that the resulting mean and variance are zero and one, respectively, in every direction. This re-scaling has the effect of stretching out this sub-mixture along the direction of small variance. In the resulting mixture of two Gaussians, if we project on a randomly chosen direction, the components *will* be noticeably different.

Our full algorithm will follow this general plan—in each step, our algorithm either learns a good estimate and outputs this estimate, or else will cluster the mixture into two proper sub-mixtures and recurse. The remainder of this section is devoted to explaining how we can learn a direction of small variance, and hence enable the clustering and recursion step if we are not able to directly apply the *Simple High Dimensional Algorithm* (Algorithm 12.2) to learn good estimates for the GMM components.

Finding a Skinny Component

How does one find a vector in which direction some of the components have small variance? Intuitively, finding this direction seems to require knowledge of the true mixture. Our approach will be to first learn an estimate of the mixture that is close to some *partition* of the true components, and thus gain some insight into the general structure of the mixture.

To deal with the issue of the skinny components, suppose we add d-dimensional Gaussian noise to sample points drawn from the example GMM of Figure 12.2. This would have the effect of "fattening" each component. After "fattening", the two narrow components would have extremely small statistical distance. So we could run our simple learning algorithm on this "fattened" mixture. Even though this distribution is a mixture of three Gaussians, the mixture is extremely close in variational distance to a mixture of two Gaussians. Our simple learning algorithm will return an estimate mixture of two Gaussians with the property that each component is close to a sub-mixture of the "fattened" distribution.

Thus one of the components in this estimate will correspond to the sub-mixture of the two narrow components. By examining this component, we notice that it is "skinny" (after adjusting the covariance matrix to account for the noise that we artificially added). Hence if we compute the smallest eigenvector of this covariance matrix, we recover a direction which allows us to cluster the original mixture into two sub-mixtures and recurse.

If the *Simple High Dimensional Algorithm* is run on a sample from a GMM in which all components have large minimum eigenvalue (for example, if the sample points have been "fattened"), then the algorithm, when run with target accuracy ϵ, will successfully learn the mixture provided that for each pair of components, either the total variational distance is at least ϵ, or at most $\epsilon' << \epsilon$, where $\epsilon' = p(\epsilon)$ for some polynomial p, and similarly, either each mixing weight is at least ϵ, or at most ϵ'. In the case that some set of components all have pairwise variational distance at most ϵ', or mixing weights at most ϵ', then with high probability the outcome of the simple high dimensional algorithm will be indistinguishable from the case that it was run on input generated from a GMM in which these components are merged into a single component, and hence will simply return a single component in place of this set of components, or will be unaware of the existence of the component arising with negligible mixing weight ϵ'. The difficulty is when there exists some pair of components whose variational distance lies within this bad *window* $[p(\epsilon), \epsilon]$., or a component whose mixing weight is in this interval. In such an instance, the *Simple High Dimensional Algorithm* has no provable guarantees.

To avoid the potential difficulty of finding a target accuracy ϵ for which no mixing weights lie in this inadmissable window, and no pair of components have variational distance within the associated inadmissable window, one simply runs the high dimensional algorithm with a range of target accuracies, $\epsilon_1, \ldots, \epsilon_{k^2}$, with $\epsilon_i < p(\epsilon_{i-1})$. While we will never know which runs succeeded, there are at most $\binom{k}{2}$ pairwise variational distances, and each pairwise variational distance can fall into the inadmissible window of at most one run; similarly for the k mixing weights. Thus a majority of the runs will be successful. All that remains is to find a set of at least k^2 runs which are consistent: given two sets of parameters returned by runs with

target accuracies $\epsilon_1 < \epsilon_2$, we say they are consistent if there is some surjective mapping of the components returned by the ϵ_1 run into the components returned by the ϵ_2 run, such that each component has similar mean and covariance to its image. Thus, one can find such a chain of at least k^2 consistent runs, yielding a set of accurate parameters.

Algorithm 12.10. THE FULL HIGH DIMENSIONAL ALGORITHM
Given a sample from a GMM in d dimensions with at most k components, target accuracy ϵ and probability of failure δ:
Let $\tau = \epsilon^c$, (for a constant c dependent on k).

- *Rescale the set of sample points so as to have mean 0 and covariance the identity matrix.*
- *Create a* fattened *set of sample points: for each of the original sample points add an independent $x \leftarrow \mathcal{N}(0, I_{d\times d}/2)$.*
- *Define $\epsilon_1 > \ldots > \epsilon_{2k^2}$ with $\epsilon_1 = \tau$ and $\epsilon_i = \epsilon^{c'\cdot i}$, for a constant c' (dependent on k.) Run the* Simple High Dimensional Algorithm *$2k^2$ times, with the ith run having target accuracy ϵ_i, and taking $1/\epsilon_{i+1}$ fattened sample points as input; this yields $2k^2$ parameter sets $P_1, \ldots, P_k$.*
- *Find a* consistent *chain of at least k^2 parameter sets; we say P_i is consistent with P_j for $i < j$ if there exists a mapping of the components of P_i into the components of P_j such that the total variational distance between each component of P_i and its image in P_j is at most $\epsilon_i + \epsilon_j$.*
- *Let $P' = (\{\mu_1, \Sigma_1, w_1\}, \ldots)$ be one of these parameter sets in the chain, and let $P = (\{\mu_1, \Sigma_1 - I/2, w_1\}, \{\mu_2, \Sigma_2 - I/2, w_2\}, \ldots)$ be the* unfattened *parameters.*
- *Let $k' \leq k$ be the number of components of P. Let λ be the minimum over $i \in \{1, \ldots, k'\}$, of the minimum eigenvalue of Σ_i.*
  - *If $\lambda > \sqrt{\tau}$, output the recovered parameters and return* SUCCESS.
  - *Otherwise, run the* CLUSTER *algorithm (Algorithm 12.11) on the original (non-noisy) sample, the list of returned parameters, and the input parameter $\gamma \leftarrow \tau^{\frac{1}{4k}}$; this algorithm projects the sample onto the eigenvector corresponding to this minimum eigenvalue, and clusters the sample points into two clusters, Y, Z with each cluster containing points that nearly exclusively originated from distinct subsets of the components.*
  - *Recursively apply this entire algorithm to each of the two sets, Y, Z, with target accuracy ϵ, and probability of failure $\epsilon/2$ and number of components set to be at most $k - 1$.*

If the above algorithm outputs $\epsilon/2$-accurate parameters with probability of success > 0.9, to boost the probability of success to $1 - \delta$, repeat the entire previous algorithm $\log \frac{1}{\delta}$ times; letting $\tilde{F}_i$ denote the parameter set returned by the ith run, for each candidate parameter vector (μ, Σ, w) given in a $\tilde{F}_i$, output that parameter vector if there are at least $\frac{1}{4} \log \frac{1}{\delta}$ runs for which $\tilde{F}_i$ contains a component whose total variational distance and mixing weight are within $\epsilon/2$, and for which no previously output components has total variational distance within ϵ.

Algorithm 12.11. CLUSTER
Given a sample $x_1, \ldots, x_n$ from a GMM in d dimensions, and k recovered parameter sets, μ_i, Σ_i for $i = 1, \ldots, k$, and a parameter γ:

1. *Let λ^*, v^* be the minimum eigenvalue of any of the covariance matrices, and corresponding eigenvector. Without loss of generality, assume that it corresponds to the first component, μ_i, Σ_i.*
2. *If $\sqrt{\lambda^*} > \gamma^k$ then RETURN; otherwise, provisionally set $t := \gamma^k$, and initialize the set $A = \{1\}$.*
3. *For each component, $i \notin A$, if $|v^* \cdot \mu_i - v^* \cdot \mu_1| < t$ and $v^{*\dagger} \Sigma_i v^* < t^2$:*
   - *update $A \leftarrow A \cup \{i\}$,*
   - *update $t \leftarrow \frac{t}{\gamma}$,*
   - *return to Step 3.*
4. *Initialize the sets Y, Z to be empty.*
5. *For each point $x_i \in X$, if $|v^* \cdot x_i - v^* \cdot \mu_1| < t\sqrt{\gamma}$ then add x_i to set Y, otherwise add x_i to set Z.*
6. *Output sets Y, Z.*

12.3 Proof of Theorem 12.1

We start by ensuring that Algorithm 12.10 will make progress in each step of the recursion. The following lemma guarantees that in any GMM with covariance matrix that is close to the identity, provided that all pairs of components have total variational distance at least ϵ, there is some pair of components $\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\mu_j, \Sigma_j)$ for which either $||\mu_i - \mu_j||$ or $||\Sigma_i - \Sigma_j||$ is reasonably large, and hence at least two components will be recovered when the Simple High Dimensional algorithm is run. Hence in the recursion step, each of the sub-mixtures will have at most $k - 1$ components.

Lemma 12.12. *Given a GMM of at most k components with minimum eigenvalue of its covariance matrix bounded below by $1/2$, provided that the mixing weights are at least ϵ, and all pairs of components have total variational distance at least ϵ, then there exists some pair of components, $\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\mu_j, \Sigma_j)$ for which either $||\mu_i - \mu_j|| > \frac{\epsilon}{3}$ or $||\Sigma_i - \Sigma_j||_{Fr} > \frac{\epsilon^2}{32d}$,*

Proof. If the minimum eigenvalue of Σ_1 is less than $1/4$, then there must be another component in the mixture, $\mathcal{N}(\mu_i, \Sigma_i)$ for which $||\Sigma_1 - \Sigma_i||_{Fr} > 1/4$, since the entire mixture has variance at least $1/2$ in the direction corresponding to the minimum eigenvalue of Σ_1, and hence if we consider the orthonormal change of basis that diagonalizes Σ_1, in that basis it

is clear that there will be some component $\mathcal{N}(\mu_i, \Sigma_i)$ for which $||\Sigma_1, \Sigma_i||_{Fr} > 1/4$, and the Frobenius norm is invariant to orthonormal changes of basis.

In the case that the minimum eigenvalue of Σ_1 is at least $1/4$, Lemma 12.5 guarantees that either $||\mu_1 - \mu_2|| \geq \epsilon/\sqrt{8}$ or $||\Sigma_1 - \Sigma_2||_{Fr} \geq \frac{\epsilon^2}{32d}$, yielding the lemma. □

We next argue that if the clustering algorithm, Algorithm 12.11, is run, it will accurately partition the sample points into two sets corresponding to a partition of the components, provided the input parameters to the algorithm are sufficiently accurate.

Lemma 12.13. *Let $A \subset [k]$, Y, Z, v^*, γ be as defined in Algorithm 12.11. Given that the parameter sets input to the algorithm are at least γ^k-accurate, for a data point x that was drawn from the ith component, if $i \in A$, then with probability at least $1 - 2\sqrt{\gamma}$, x will be assigned to Y. Similarly, if $i \notin A$, then with probability at least $1 - 2\sqrt{\gamma}$, x will be assigned to Z.*

Proof. For any $i \in A$, letting $\mu := \mu_i \cdot v^*$, and $\sigma^2 := v^{*\dagger}\Sigma_i v^*$ denote the mean and variance of the projection of the ith component onto v^*, we have that

$$\Pr_{x \leftarrow \mathcal{N}(\mu_i, \Sigma_i)}[x \text{ assigned to } Z] \leq \Pr_{y \leftarrow \mathcal{N}(0,1)}\left[|y| > \frac{1}{2\sqrt{\gamma}}\right] \leq e^{-\frac{1}{8\gamma}} << 2\sqrt{\gamma}.$$

For any $i \notin A$, letting $\mu := \mu_i \cdot v^*$, and $\sigma^2 := v^{*\dagger}\Sigma_i v^*$ denote the mean and variance of the projection of the ith component onto v^*, we have that

$$\Pr_{x \leftarrow \mathcal{N}(\mu_i, \Sigma_i)}[x \text{ assigned to } Y] \leq \max_{\sigma^2} \Pr_{y \leftarrow \mathcal{N}(0,\sigma^2)}[y \in [1 - 2\sqrt{\gamma}, 1 + 2\sqrt{\gamma}]] \leq 2\sqrt{\gamma}.$$

□

We now put the pieces together to establish the correctness of Algorithm 12.10.

Proof of Theorem 12.1. First observe that by elementary Chernoff bounds, the affine transformation computed in the first step of the algorithm, putting the data into isotropic position, will have the property that all eigenvalues of the projection of the covariance of the actual GMM will, with the desired probability, lie in the interval $[1/2, 3/2]$, and henceforth assume this holds.

Since the recovered means and covariances from the successful runs of the Simple High Dimensional Algorithm are accurate to within τ in Euclidean and Frobenius distance, respectively. Any component parameters output by the algorithm will be computed in some projection in which the minimum eigenvalue of that component is at least $\sqrt{\tau_2}$, and thus by Lemma 12.5, the recovered components will be within total variational distance $\sqrt{8d}\tau^{1/4} < \epsilon$ from the corresponding actual component. As this metric is affine invariant, after inverting the series of transformations (in each recursive step) that placed the data into isotropic position, the accuracy of the recovered components in total variational distance still hold.

We now ensure that the $\epsilon_i = poly(\epsilon)$ can be set so as yield the claimed performance. Let $p(\alpha, k)$ denote the sample size required by the Simple High Dimensional Algorithm (Algorithm 12.2) in order to return α-accurate parameters, with probability of failure at most ϵ, when given as input a sample from a GMM with at most k components, in d dimensions, satisfying the conditions of Proposition 12.4. For two GMMs, F, F' satisfying $D_{tv}(F, F') \leq O(p(\alpha, k)k^2/\epsilon)$, the results of running any algorithm on a sample drawn from F will be indistinguishable from the results of drawing the sample from F' with probability at least $1 - \epsilon/k^2$. Thus we will set the $\epsilon_1, \ldots, \epsilon_{2k^2}$ of Algorithm 12.10 as follows: $\epsilon_1 = \tau$, and $\epsilon_i = \frac{1}{p(\epsilon_{i-1},k)}k^2/\epsilon$. Hence, the ith run of the Simple High Dimensional Algorithm will certainly perform correctly with the desired probability, when every pair of components of the input GMM either have total variational distance at least ϵ_{i-1}, or at most ϵ_i.

By Lemma 12.12, each run of the Simple High Dimensional Algorithm will output at least two component sets. As mentioned above, there are only $\binom{k}{2}$ pairwise variational distances between components, and hence at least half of the k^2 executions of the Simple High Dimensional Algorithm will perform correctly, in which case the consistent chain will be found, resulting in an accurate set of parameters for at least two components.

If each component (after subtracting the contribution to the covariance of the "fattening") has minimum eigenvalue at least, $\sqrt{\tau}$, then if each of the true mixing weights is at least ϵ, by Lemma 12.5, close approximations of the components in terms of variational distance will be found. In the case that the pairwise variational distances between components is at least ϵ, then all components will be recovered. If the minimum pairwise variational distance, or minimum mixing weight are less than ϵ, we still have performed density estimation.

If the minimum eigenvalue of any returned component is less than τ in a given run of the recursion, then we now argue that the success guarantee of Lemma 12.13 will ensure that the clustering algorithm partitions the data points sufficiently accurately so that the results of recursing the algorithm on each of the two data sets Y, Z with $k \leftarrow k - 1$, will be (with high probability) indistinguishable from the output of running the algorithm on a sample drawn from each of the sub-mixtures defined by the set A of Algorithm 12.11.

The probability of mis-clustering each data point, by Lemma 12.13 is at most $\tau^{\frac{1}{2k}}$, and hence we can pick τ such that $\tau^{\frac{1}{2k}} << \frac{1}{q(\epsilon,k-1)}$, where $q(\alpha, k')$ is the sample size required to learn GMMs of at most k' components to accuracy ϵ. Hence, by induction on k, using the fact from Proposition 12.4 that $p(\alpha, k)$ is a polynomial in α for any fixed k, we have that $q(\alpha, k)$ is a polynomial in α for any fixed k, as desired. $\square$

Bibliography

[1] J. Acharya, A. Orlitsky, and S. Pan. "The maximum likelihood probability of unique-singleton, ternary, and length-7 patterns". In: *IEEE Symposium on Information Theory*. 2009.

[2] J. Acharya et al. "Competitive closeness testing". In: *Conference on Learning Theory (COLT)*. 2011.

[3] D. Achlioptas and F. McSherry. "On spectral learning of mixtures of distributions". In: *Conference on Learning Theory (COLT)*. 2005, pp. 458–469.

[4] M. Ajtai, R. Kumar, and D. Sivakumar. "A sieve algorithm for the shortest lattice vector problem". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2001, pp. 601–610.

[5] N. Alon, Y. Matias, and M. Szegedy. "The space complexity of approximating the frequency moments". In: *Journal of Computer and System Sciences* 58 (1999), pp. 137–147.

[6] N. Alon and A. Naor. "Approximating the cut-norm via Grothendiecks inequality". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2004, pp. 72–80.

[7] N. Alon et al. "A combinatorial characterization of the testable graph properties: it's all about regularity". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2006.

[8] A. Anandkumar, D. Hsu, and S.M. Kakade. "A method of moments for mixture models and hidden Markov models". In: *Conference on Learning Theory (COLT)*. 2012.

[9] A. Andoni and P. Indyk. "Near–optimal hashing algorithms for approximate nearest neighbor in high dimensions". In: *Communications of the ACM* 51.1 (2008), pp. 117–122.

[10] A. Andoni and P. Indyk. "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2006, pp. 459–468.

[11] S. Arora and R. Ge. "New algorithms for learning in presence of errors". In: *International Colloquium on Automata, Languages and Programming (ICALP)*. 2011, pp. 403–415.

[12] S. Arora and R. Kannan. "Learning mixtures of arbitrary Gaussians". In: *Annals of Applied Probability* 15.1A (2005), pp. 69–92.

[13] M. Arumugam and J. Raes et al. "Enterotypes of the human gut microbiome". In: *Nature* 473.7346 (2011), pp. 174–180.

[14] J. Babaud et al. "Uniqueness of the Gaussian kernel for scale-space filtering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8.1 (1986), pp. 26–33.

[15] Z. Bar-Yossef, R. Kumar, and D. Sivakumar. "Sampling algorithms: lower bounds and applications". In: *Symposium on Theory of Computing (STOC)*. 2001.

[16] Z. Bar-Yossef et al. "Counting distinct elements in a data stream". In: *6th Workshop on Randomization and Approximation Techniques*. 2002.

[17] A. Barbour and L. Chen. *An Introduction to Stein's Method*. Singapore University Press, 2005.

[18] T. Batu. *Testing Properties of Distributions*. Ph.D. thesis, Cornell University, 2001.

[19] T. Batu et al. "Testing random variables for independence and identity". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2001.

[20] T. Batu et al. "Testing that distributions are close". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2000.

[21] T. Batu et al. "The complexity of approximating the entropy". In: *Symposium on Theory of Computing (STOC)*. 2002.

[22] T. Batu et al. "The complexity of approximating the entropy". In: *SIAM Journal on Computing* (2005).

[23] M. Belkin and K. Sinha. "Polynomial learning of distribution families". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2010.

[24] J.L. Bentley. "Multidimensional binary search trees used for associative searching". In: *Communications of the ACM* 18.9 (1975), pp. 509–517.

[25] K. Beyer et al. "On synopses for distinct-value estimation under multiset operations". In: *ACM SIGMOD International Conference on Management of Data*. 2007.

[26] R. Bhattacharya and S. Holmes. "An exposition of Götze's estimation of the rate of convergence in the multivariate central limit theorem". In: *Stanford Department of Statistics Technical Report* 2010-02 (2010).

[27] A. Blum, A. Kalai, and H. Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model". In: *Journal of the ACM (JACM)* 50.4 (2003), pp. 507–519.

[28] A. Blum et al. "Weakly learning DNF and characterizing statistical query learning using Fourier analysis". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 1994, pp. 253–262.

[29] M. Blum, M. Luby, and R. Rubinfeld. "Self–testing/correcting with applications to numerical problems". In: *Journal of Computer and System Sciences* 47.3 (1993), pp. 549–595.

[30] Z. Brakerski and V. Vaikuntanathan. "Efficient fully homomorphic encryption from (standard) LWE". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2011.

[31] S. C. Brubaker and S. Vempala. "Isotropic PCA and affine-invariant clustering". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2008, pp. 551–560.

[32] J. Bunge. "Bibliography of references on the problem of estimating support size, available at http://www.stat.cornell.edu/~bunge/bibliography.html". In: ().

[33] J. Bunge and M. Fitzpatrick. "Estimating the number of species: a review". In: *Journal of the American Statistical Association* 88.421 (1993), pp. 364–373.

[34] A. Carlton. "On the bias of information estimates". In: *Psychological Bulletin* 71 (1969), pp. 108–109.

[35] A. Chakrabarti, G. Cormode, and A. McGregor. "A near-optimal algorithm for computing the entropy of a stream". In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2007.

[36] A. Chao and T.J. Shen. "Nonparametric estimation of shannons index of diversity when there are unseen species in sample". In: *Environmental and Ecological Statistics* 10 (2003), pp. 429–443.

[37] M. Charikar et al. "Towards estimation error guarantees for distinct values". In: *Symposium on Principles of Database Systems (PODS)*. 2000.

[38] S. Chatterjee. *Personal communication, May 2010*.

[39] S. Chatterjee and E. Meckes. "Multivariate normal approximation using exchangeable pairs". In: *Latin American Journal of Probability and Mathematical Statistics (ALEA)* 4 (2008), pp. 257–283.

[40] L.H.Y. Chen, L. Goldstein, and Q. Shao. *Normal Approximation by Stein's Method*. Springer, 2011.

[41] K. Clarkson. "A randomized algorithm for closest–point queries". In: *SIAM Journal on Computing* 17.4 (1988), pp. 830–847.

[42] D. Coppersmith. "Rectangular matrix multiplication revisited". In: *Journal of Complexity* 13.1 (1997), pp. 42–49.

[43] S. Dasgupta. "Learning mixtures of Gaussians". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 1999, pp. 634–644.

[44] S. Dasgupta and A. Gupta. "An elementary proof of a theorem of Johnson and Lindenstrauss". In: *Random Structures and Algorithms* 22.1 (2003), pp. 60–65.

[45] S. Dasgupta, A. T. Kalai, and C. Monteleoni. "Analysis of perceptron-based active learning". In: *Journal of Machine Learning Research* 10 (2009), pp. 281–299.

[46] S. Dasgupta and L. J. Schulman. "A two-round variant of EM for Gaussian mixtures". In: *Conference on Uncertainty in Artificial Intelligence (UAI)*. 2000, pp. 152–159.

[47] C. Daskalakis and C. H. Papadimitriou. "Computing equilibria in anonymous games". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2007.

[48] C. Daskalakis and C. H. Papadimitriou. "Discretized multinomial distributions and Nash equilibria in anonymous games". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2008.

[49] M. Datar et al. "Locality–sensitive hashing scheme based on p-stable distributions". In: *Proceedings of the 20th ACM Symposium on Computational Geometry (SoCG)*. 2004, pp. 253–262.

[50] M. Dubiner. "Bucketing coding and information theory for the statistical high dimensional nearest neighbor problem". In: *CoRR* abs/0810.4182 (2008).

[51] B. Efron and C. Stein. "The jacknife estimate of variance". In: *Annals of Statistics* 9 (1981), pp. 586–596.

[52] B. Efron and R. Thisted. "Estimating the number of unseen species: how many words did Shakespeare know?" In: *Biometrika* 63.3 (1976), pp. 435–447.

[53] J. Feldman, R. A. Servedio, and R. O'Donnell. "PAC learning axis-aligned mixtures of Gaussians with no separation assumption". In: *Conference on Learning Theory (COLT)*. 2006, pp. 20–34.

[54] V. Feldman et al. "New results for learning noisy parities and halfspaces". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2006.

[55] R.A. Fisher, A. Corbet, and C.B. Williams. "The relation between the number of species and the number of individuals in a random sample of an animal population". In: *Journal of the British Ecological Society* 12.1 (1943), pp. 42–58.

[56] P. W. Glynn. "Upper Bounds on Poisson Tail Probabilities". In: *Operations Research Letters* 6.1 (1987), pp. 9–14.

[57] O. Goldreich, S. Goldwasser, and D. Ron. "Property Testing and Its Connection to Learning and Approximation". In: *IEEE Symposium on Foundations of Computer Science (FOCS), 1996*.

[58] O. Goldreich and L. Trevisan. "Three theorems regarding testing graph properties". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 1996.

[59] I. J. Good. "The population frequencies of species and the estimation of population parameters". In: *Biometrika* 40.16 (1953), pp. 237–264.

[60] I.J. Good and G.H. Toulmin. "The number of new species, and the increase in population coverage, when a sample is increased". In: *Biometrika* 43 (1956), pp. 45–63.

[61] F. Götze. "On the rate of convergence in the multivariate CLT". In: *Annals of Probability* 19.2 (1991), pp. 724–739.

[62] E. Grigorescu, L. Reyzin, and S. Vempala. "On noise-tolerant learning of sparse parities and related problems". In: *The 22nd International Conference on Algorithmic Learning Theory (ALT)*. 2011.

[63] S. Guha, A. McGregor, and S. Venkatasubramanian. "Streaming and sublinear approximation of entropy and information distances". In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2006.

[64] L. Gurvits. "On Newton(like) inequalities for multivariate homogeneous polynomials". http://www.optimization-online.org/DB_FILE/2008/06/1998.pdf. 2008.

[65] P. J. Haas et al. "Selectivity and cost estimation for joins based on random sampling". In: *Journal of Computer and System Sciences* 52.3 (1996), pp. 550–569.

[66] N.J.A. Harvey, J. Nelson, and K Onak. "Sketching and streaming entropy via approximation theory". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2008.

[67] N. J. Hopper and A. Blum. "Secure human identification protocols". In: *ASIACRYPT*. 2001, pp. 52–66.

[68] D.G. Horvitz and D.J. Thompson. "A generalization of sampling without replacement from a finite universe". In: *Journal of the American Statistical Association* 47.260 (1952), pp. 663–685.

[69] R. A. Hummel and B. C. Gidas. "Zero crossings and the heat equation". In: *Technical Report Number 111, Courant Institute of Mathematical Sciences at NYU* (1984).

[70] R. Impagliazzo and D. Zuckerman. "How to recycle random bits". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 1989, pp. 248–253.

[71] P. Indyk and R. Motwani. "Approximate nearest neighbors: towards removing the curse of dimensionality". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 1998.

[72] P. Indyk and D. Woodruff. "Tight lower bounds for the distinct elements problem". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2003.

[73] A. Kalai, A. Moitra, and G. Valiant. "Disentangling Gaussians". In: *Communications of the ACM (Research Highlights Section)* 55.2 (2011).

[74] A. Kalai, A. Moitra, and G. Valiant. "Efficiently learning mixtures of two Gaussians". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2010.

[75] D. Kane, J. Nelson, and D. Woodruff. "An optimal algorithm for the distinct elements problem". In: *Symposium on Principles of Database Systems (PODS)*. 2010.

[76] R. Kannan, H. Salmasian, and S. Vempala. "The spectral method for general mixture models". In: *SIAM Journal on Computing* 38.3 (2008), pp. 1141–1156.

[77] L. Kantorovich and G. Rubinstein. "On a functional space and certain extremal problems". In: *Dokl. Akad. Nauk. SSSR (Russian)* 115 (1957), pp. 1058–1061.

[78] N. Karmarkar. "A new polynomial time algorithm for linear programming". In: *Combinatorica* 4.4 (1984), pp. 373–395.

[79] M. Kearns. "Efficient noise-tolerant learning from statistical queries". In: *Journal of the ACM* 45.6 (1998), pp. 983–1006.

[80] A. Keinan and A. G. Clark. "Recent explosive human population growth has resulted in an excess of rare genetic variants". In: *Science* 336.6082 (2012), pp. 740–743.

[81] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. "Efficient search for approximate nearest neighbor in high dimensional spaces". In: *SIAM Journal on Computing* 30.2 (2000), pp. 457–474.

[82] L. Liu et al. "Robust singular value decomposition analysis of microarray data". In: *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* 100.23 (2003), pp. 13167–13172.

[83] V. Lyubashevsky. "The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem". In: *RANDOM*. 2005, pp. 378–389.

[84] P.D.M. Macdonald. *Personal communications, November 2009*.

[85] F. MacWilliams and N. Sloane. *The Theory of Error-Correcting Codes*. North–Holland Mathematical Library, Amsterdam, The Netherlands, 1977.

[86] D. A. McAllester and R.E. Schapire. "On the convergence rate of Good-Turing estimators". In: *Conference on Learning Theory (COLT)*. 2000.

[87] M.A. McDowell et al. *Anthropometric Reference Data for Children and Adults: United States, 2003–2006*. National Health Statistics Reports, No. 10. 2008.

[88] G.J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2009.

[89] S. Meiser. "Point location in arrangements of hyperplanes". In: *Information and Computation* 106.2 (1993), pp. 286–303.

[90] G. Miller. "Note on the bias of information estimates". In: *Information Theory in Psychology: Problems and Methods* (1955), pp. 95–100.

[91] A. Moitra and G. Valiant. "Settling the polynomial learnability of mixtures of Gaussians". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2010.

[92] E. Mossel, R. O'Donnell, and R. Servedio. "Learning functions of k relevant variables". In: *Journal of Computer and System Sciences* 69.3 (2004), pp. 421–434.

[93] M. R. Nelson and D. Wegmann et al. "An abundance of rare functional variants in 202 drug target genes sequenced in 14,002 people". In: *Science* 337.6090 (2012), pp. 100–104.

[94] R. O'Donnell, Y. Wu, and Y. Zhou. "Optimal lower bounds for locality sensitive hashing (except when q in tiny)". In: *Innovations in Theoretical Computer Science (ITCS)*. 2011, pp. 275–283.

[95] F. Olken and D. Rotem. "Random sampling from database files: a survey". In: *Proceedings of the Fifth International Workshop on Statistical and Scientific Data Management*. 1990.

[96] A. Orlitsky, N.P. Santhanam, and J. Zhang. "Always Good Turing: asymptotically optimal probability estimation". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2003.

[97] A. Orlitsky, N.P. Santhanam, and J. Zhang. "Always Good Turing: asymptotically optimal probability estimation". In: *Science* 302.5644 (2003), pp. 427–431.

[98] A. Orlitsky et al. "On modeling profiles instead of values". In: *Conference on Uncertainity in Artificial Intelligence (UAI)*. 2004, pp. 426–435.

[99] R. Pagh. "Compressed matrix multiplication". In: *Innovations in Theoretical Computer Science (ITCS)*. 2012.

[100] R. Panigraphy. "Entropy–based nearest neighbor search in high dimensions". In: *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2006.

[101] L. Paninski. "Estimating entropy on m bins given fewer than m samples". In: *IEEE Trans. on Information Theory* 50.9 (2004), pp. 2200–2203.

[102] L. Paninski. "Estimation of entropy and mutual information". In: *Neural Computation* 15.6 (2003), pp. 1191–1253.

[103] S. Panzeri and A Treves. "Analytical estimates of limited sampling biases in different information measures". In: *Network: Computation in Neural Systems* 7 (1996), pp. 87–107.

[104] Ramamohan Paturi, Sanguthevar Rajasekaran, and John H. Reif. "The light bulb problem." In: *Conference on Learning Theory (COLT)*. 1989, pp. 261–268.

[105] K. Pearson. "Contributions to the Mmathematical theory of evolution". In: *Philosophical Transactions of the Royal Society of London* (1894), pp. 71–110.

[106] C. Peikert. "Public–key cryptosystems from the worst-case shortest vector problem". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2009, pp. 333–342.

[107] S. Raskhodnikova et al. "Strong lower bounds for approximating distribution support size and the distinct elements problem". In: *SIAM Journal on Computing* 39.3 (2009), pp. 813–842.

[108] R. A. Redner and H. F. Walker. "Mixture densities, maximum likelihood and the EM algorithm". In: *SIAM Review* 26 (1984), pp. 195–239.

[109] O. Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM* 56.6 (2009), pp. 1–40.

[110] O. Regev. "The learning with errors problem". In: *Invited survey in IEEE Conference on Computational Complexity (CCC)* (2010).

[111] G. Reinert and A. Röllin. "Multivariate normal approximation with Stein's method of exchangeable pairs under a general linearity condition". In: *Annals of Probability* 37.6 (2009), pp. 2150–2173.

[112] T.J. Rivlin. *The Chebyshev Polynomials*. John Wiley and Sons, 1974.

[113] H.E. Robbins. "Estimating the total probability of the unobserved outcomes of an experiment". In: *Annals of Mathematical Statistics* 39.1 (1968), pp. 256–257.

[114] B. Roos. "On the rate of multivariate Poisson convergence". In: *Journal of Multivariate Analysis* 69 (1999), pp. 120–134.

[115] R. Rubinfeld and M. Sudan. "Robust Characterizations of Polynomials with Applications to Program Testing". In: *SIAM Journal on Computing* 25.2 (1996), pp. 252–272.

[116] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Elsevier, 2006.

[117] I.J. Schoenberg. "Positive definite functions on spheres". In: *Duke Mathematical Journal* 9.1 (1942), pp. 96–108.

[118] C. Stein. "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables". In: *Proc. Sixth Berkeley Symp. on Mathematical Statistics and Probability* 2 (1972).

[119] G. Szegö. "Orthogonal Polynomials, 4th edition". In: *American Mathematical Society, Colloquium Publications, 23. Providence, RI* (1975).

[120] H. Teicher. "Identifiability of mixtures". In: *Annals of Mathematical Statistics* 32.1 (1961), pp. 244–248.

[121] J. A. Tennessen, A.W. Bigham, and T.D. O'Connor et al. "Evolution and functional impact of rare coding variation from deep sequencing of human exomes". In: *Science* 337.6090 (2012), pp. 64–69.

[122] D.M. Titterington, A.F.M. Smith, and U.E. Makov. *Statistical analysis of finite mixture distributions*. Wiley, 1985.

[123] A Treves and S. Panzeri. "The upward bias in measures of information derived from limited data samples". In: *Neural Computation* 7 (1995), pp. 399–407.

[124] G. Valiant. "Finding correlations in subquadratic time, with applications to learning parities and juntas". In: *IEEE Symposium on Foundations of Computer Science (FOCS) (to appear)*. 2012.

[125] G. Valiant. "Finding correlations in subquadratic time, with applications to learning parities and juntas with noise". Available at: http://eccc.hpi-web.de/report/2012/006/. 2012.

[126] G. Valiant and P. Valiant. "A CLT and tight lower bounds for estimating entropy". Available at: http://www.eccc.uni-trier.de/report/2010/179/. 2010.

[127] G. Valiant and P. Valiant. "Estimating the unseen: a sublinear-sample canonical estimator of distributions". Available at: http://www.eccc.uni-trier.de/report/2010/180/. 2010.

[128] G. Valiant and P. Valiant. "Estimating the unseen: an $n/\log(n)$–sample estimator for entropy and support size, shown optimal via new CLTs". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2011.

[129] G. Valiant and P. Valiant. "The power of linear estimators". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. 2011.

[130] L. Valiant. "Functionality in neural nets". In: *First Workshop on Computational Learning Theory*. 1988, pp. 28–39.

[131] P. Valiant. "Testing symmetric properties of distributions". In: *Symposium on Theory of Computing (STOC)*. 2008.

[132] P. Valiant. "Testing symmetric properties of distributions". In: *SIAM Journal on Computing* 40.6 (2011), pp. 1927–1968.

[133] S. Vempala. "The Random Projection Method". In: *American Mathematical Society* (2004).

[134] S. Vempala and G. Wang. "A spectral algorithm for learning mixture models". In: *Journal of Computer and System Sciences* 68.4 (2004), pp. 841–860.

[135] K. A. Verbeurgt. "Learning DNF under the uniform distribution in quasipolynomial time." In: *Conference on Learning Theory (COLT)*. 1990, pp. 314–326.

[136] J. Victor. "Asymptotic bias in information estimates and the exponential (Bell) polynomials". In: *Neural Computation* 12 (2000), pp. 2797–2804.

[137] V.Q. Vu, B. Yu, and R.E. Kass. "Coverage-adjusted entropy estimation". In: *Statistics in Medicine* 26.21 (2007), pp. 4039–4060.

[138] A.B. Wagner, P. Viswanath, and S.R. Kulkami. "A better Good-Turing estimator for sequence Probabilities". In: *IEEE Symposium on Information Theory*. 2007.

[139] A.B. Wagner, P. Viswanath, and S.R. Kulkami. "Strong consistency of the Good-Turing estimator". In: *IEEE Symposium on Information Theory*. 2006.

[140] R. Weber, H.J. Schek, and S. Blott. "A quantitative analysis and performance study for similarity–search methods in high–dimensional spaces". In: *The 24th International Conference on Very Large Databases (VLDB)*. 1998.

[141] V. Vassilevska Williams. "Multiplying matrices faster than Coppersmith–Winograd". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2012.

[142] D. Woodruff. "The average-case complexity of counting distinct elements". In: *The 12th International Conference on Database Theory*. 2009.

[143] T. Yatsunenko et al. "Human gut microbiome viewed across age and geography". In: *Nature* 486.7402 (2012), pp. 222–227.

[144] S. Zahl. "Jackknifing an index of diversity". In: *Ecology* 58 (1977), pp. 907–913.

Appendix A

Basic Properties of Gaussian and Poisson Distributions

A.1 Basic Properties of Gaussians

This section contains several basic facts about Gaussians which are used throughout Parts I and III of the dissertation.

Fact A.1. *For $\epsilon > 0$,*

$$\max_x |\mathcal{N}(0, \sigma^2, x) - \mathcal{N}(0, \sigma^2 + \epsilon, x)| \leq \frac{\epsilon}{2\sigma^{3/2}\sqrt{2\pi}} \leq \frac{\epsilon}{5\sigma^{3/2}}.$$

Proof. It is easy to verify that the desired quantity is maximized at $x = 0$, from which the claim follows by noting that for $\alpha, \beta > 0$, $\frac{1}{\sqrt{\alpha}} - \frac{1}{\sqrt{\alpha+\beta}} \leq \frac{\beta}{2\alpha^{3/2}}$. □

Fact A.2. *For $\epsilon > 0$,*

$$\max_x |\mathcal{N}(0, \sigma^2, x) - \mathcal{N}(\epsilon, \sigma^2, x)| \leq \frac{\epsilon}{\sigma\sqrt{2\pi e}} \leq \frac{\epsilon}{4\sigma}.$$

Proof. This follows from noting that the maximum magnitude of the derivative of $\mathcal{N}(0, \sigma^2, x)$ is attained at $x = \pm\sigma$. □

Fact A.3.

$$\max_{\sigma^2} \mathcal{N}(0, \sigma^2, \gamma) = \frac{1}{\gamma\sqrt{2\pi e}}.$$

Proof. It is easy to verify that $\text{argmax}_{\sigma^2}\mathcal{N}(0, \sigma^2, \gamma) = \gamma^2$, from which the fact follows. □

Corollary A.4.

$$\max_{\mu,\sigma^2:\mu+\sigma^2\geq\gamma} \mathcal{N}(\mu, \sigma^2, 0) \leq \max\left(\frac{2}{\gamma\sqrt{2\pi e}}, \frac{1}{\sqrt{\pi\gamma}}\right).$$

Proof. Either $\mu \geq \gamma/2$, or $\sigma^2 \geq \gamma/2$. In the first case, using Fact A.3,

$$\max_{\mu\geq\gamma/2} \mathcal{N}(\mu, \sigma^2, 0) = \max_{\sigma^2} \mathcal{N}(0, \sigma^2, \gamma/2) = \frac{2}{\gamma\sqrt{2\pi e}}.$$

In the second case, we have

$$\max_{x,\sigma^2\geq\gamma/2} \mathcal{N}(0, \sigma^2, x) = \mathcal{N}(0, \gamma/2, 0) = \frac{1}{\sqrt{\pi\gamma}}.$$

□

Claim A.5. *For i odd:*

$$\begin{aligned} H_i(x, \sigma) := \int x^i e^{-\frac{x^2}{2\sigma^2}} dx &= -x^{i-1}\sigma^2 e^{-\frac{x^2}{2\sigma^2}} - (i-1)x^{i-3}\sigma^4 e^{-\frac{x^2}{2\sigma^2}} \\ &\quad -(i-1)(i-3)x^{i-5}\sigma^6 e^{-\frac{x^2}{2\sigma^2}} \ldots - (i-1)!!\sigma^{i+1}e^{-\frac{x^2}{2\sigma^2}} \end{aligned}$$

Proof. We can check

$$\begin{aligned}\frac{\partial}{\partial x}H_i(x,\sigma) &= x^i e^{-\frac{x^2}{2\sigma^2}} + (i-1)x^{i-2}\sigma^2 e^{-\frac{x^2}{2\sigma^2}} \ldots + (i-1)!!x\sigma^{i-1}e^{-\frac{x^2}{2\sigma^2}} \\ &\quad -(i-1)x^{i-2}\sigma^2 e^{-\frac{x^2}{2\sigma^2}} - (i-1)(i-3)x^{i-4}\sigma^4 e^{-\frac{x^2}{2\sigma^2}} \ldots - (i-1)!!xe^{-\frac{x^2}{2\sigma^2}} \\ &= x^i e^{-\frac{x^2}{2\sigma^2}}\end{aligned}$$

□

Claim A.6. *For i even:*

$$\begin{aligned}H_i(x,\sigma) := \int x^i e^{-\frac{x^2}{2\sigma^2}}dx &= -x^{i-1}\sigma^2 e^{-\frac{x^2}{2\sigma^2}} - (i-1)x^{i-3}\sigma^4 e^{-\frac{x^2}{2\sigma^2}} \\ &\quad -(i-1)(i-3)x^{i-5}\sigma^6 e^{-\frac{x^2}{2\sigma^2}} \ldots + (i-1)!!\sigma^i \int e^{-\frac{x^2}{2\sigma^2}}dx\end{aligned}$$

We can apply these identities to get bounds on the contribution of the tails - i.e. $|x| \geq \frac{\sigma}{\epsilon}$ for all finite moments $i \geq 0$.

Lemma A.7. *For $\epsilon \leq 1$,*

$$\int_{|x|\geq\sigma/\epsilon} |x|^i \mathcal{N}(0,\sigma^2,x)dx \leq i(i!)^{1/2}\sigma^i\epsilon^{-i}e^{-\frac{1}{2\epsilon^2}}.$$

Proof. We can immediately apply Claim A.5 and Claim A.6. Note that for constant i, σ^2, this bound is $O(e^{-1/\epsilon})$. □

Corollary A.8. *For $\epsilon \leq 1$,*

$$\int_{|x-\mu|\geq\sigma/\epsilon} |x|^i \mathcal{N}(\mu,\sigma^2,x)dx \leq 2^i i(i!)^{1/2}\max(\sigma^i,1)\max(|\mu|,\frac{1}{\epsilon})^i e^{-\frac{1}{2\epsilon^2}}.$$

Proof. Using Lemma A.7, the above bound follows by a change of variables and expanding the binomial term. □

For completeness, for the univariate normal distribution $\mathcal{N}(0,\sigma^2)$, the ith raw moment is,

$$\mathrm{E}_{x\sim\mathcal{N}(0,\sigma^2)}[x^i] = \begin{cases}0 & \text{if } i \text{ is odd} \\ (i-1)!!\sigma^i = \frac{i!}{2^i(i/2)!}\sigma^i & \text{if } i \text{ is even.}\end{cases} \tag{A.1}$$

Bounds on the Distance Between Multivariate Gaussians

This section contains a straightforward analysis of the statistical distance between two multivariate Gaussians.

Fact A.9. *Given independent real-valued random variables W, X, Y, Z the total variation distance satisfies $D_{tv}\left((W,X),(Y,Z)\right) \leq D_{tv}(W,Y) + D_{tv}(X,Z)$, where (W,X) and (Y,Z) denote joint distributions.*

Proof.

$$\begin{aligned}
D_{tv}\left((W,X),(Y,Z)\right) &= \frac{1}{2}\int\int |P_W(a)P_X(b) - P_Y(a)P_Z(b)|da\ db \\
&= \frac{1}{4}\int\int |(P_W(a) - P_Y(a))(P_X(b) + P_Z(b)) \\
&\qquad +(P_W(a) + P_Y(a))(P_X(b) - P_Z(b))|da\ db \\
&\leq \frac{1}{4}\int\int |(P_W(a) - P_Y(a))(P_X(b) + P_Z(b))|da\ db \\
&\qquad +\frac{1}{4}\int\int (P_W(a) + P_Y(a))(P_X(b) - P_Z(b))|da\ db \\
&= \frac{1}{2}\int |(P_W(a) - P_Y(a))|da + \frac{1}{2}\int (P_X(b) - P_Z(b))|db \\
&= D_{tv}(W,Y) + D_{tv}(X,Z).
\end{aligned}$$

□

Fact A.10. *Letting $\mathcal{N}(\mu,\sigma^2)$ denote the univariate Gaussian distribution,*

$$D_{tv}(\mathcal{N}(\mu,1),\mathcal{N}(\mu+\alpha,1)) \leq |\alpha|/\sqrt{2\pi}.$$

Fact A.11. *Letting $\mathcal{N}(\mu,\sigma^2)$ denote the univariate Gaussian distribution,*

$$D_{tv}(\mathcal{N}(\mu,1),\mathcal{N}(\mu,\sigma^2)) \leq \frac{\max(\sigma^2,1/\sigma^2)-1}{\sqrt{2\pi e}}.$$

Fact A.12. *Given two Gaussian distributions in m dimensions $G_1 = \mathcal{N}(\mu_1,\Sigma_1)$, and $G_2 = \mathcal{N}(\mu_2,\Sigma_2)$, where $\Sigma_1 = TT'$, is the Cholesky decomposition of Σ_1, then*

$$D_{tv}(G_1,G_2) \leq \sum_{i=1}^{m} \frac{\max(\lambda_i,1/\lambda_i)-1}{\sqrt{2\pi e}} + \frac{||T^{-1}(\mu_1-\mu_2)||}{\sqrt{2\pi}},$$

where λ_i is the ith eigenvalue of $T^{-1}\Sigma_2 T'^{-1}$.

Proof. Since variational distance is affine-invariant, applying the affine transformation T^{-1}, we have $D_{tv}(G_1, G_2) = D_{tv}\left(\mathcal{N}(0, T^{-1}\Sigma_1 T'^{-1}), \mathcal{N}(T^{-1}(\mu_1 - \mu_2), T^{-1}\Sigma_2 T'^{-1})\right)$, where we have $T^{-1}\Sigma_1 T'^{-1} = I$, the $m \times m$ identity. Thus, by the triangle inequality, this distance is at most

$$D_{tv}\left(\mathcal{N}(0, I), \mathcal{N}(T^{-1}(\mu_1 - \mu_2), I)\right) + D_{tv}\left(\mathcal{N}(0, I), \mathcal{N}(0, T^{-1}\Sigma_2 T'^{-1})\right).$$

Viewing $\mathcal{N}(T^{-1}(\mu_1 - \mu_2), I)$ as the joint distribution of m independent univariate Gaussians, where the first $m-1$ distributions are $\mathcal{N}(0,1)$, and the mth distribution is $\mathcal{N}(||T^{-1}(\mu_1 - \mu_2)||, 1)$, by Facts A.9 and A.10 we get that

$$D_{tv}\left(\mathcal{N}(0, I), \mathcal{N}(T^{-1}(\mu_1 - \mu_2), I)\right) \leq \frac{||T^{-1}(\mu_1 - \mu_2)||}{\sqrt{2\pi}}.$$

To bound the other component, view $\mathcal{N}(0, T^{-1}\Sigma_2 T'^{-1})$ as the joint distribution of m independent univariate Gaussians, where the ith distribution is $\mathcal{N}(0, \lambda_i)$, with λ_i the ith eigenvalue of $T^{-1}\Sigma_2 T'^{-1}$, and use facts Facts A.9 and A.11, to yield the claimed result. □

Proposition A.13. *Given two m-dimensional Gaussians $G_1 = \mathcal{N}(\mu_1, \Sigma_1), G_2 = \mathcal{N}(\mu_2, \Sigma_2)$ such that for all $i, j \in [m]$, $|\Sigma_1(i,j) - \Sigma_2(i,j)| \leq \alpha$, and $min(eig(\Sigma_1)) > \lambda > \alpha$,*

$$D_{tv}(G_1, G_2) \leq \frac{||\mu_1 - \mu_2||}{\sqrt{2\pi\lambda}} + \frac{m\alpha}{\sqrt{2\pi e}(\lambda - \alpha)}.$$

Proof. Let $\Sigma_1 = PDDP'$, where D is a diagonal matrix, and P is a unitary matrix. Note that the minimum entry on the diagonal of D is $\sqrt{\lambda}$. We now write $\Sigma_2 = \Sigma_1 + A$, for some symmetric matrix A whose entries are bounded in magnitude by α. By Fact A.12, the contribution to $D_{tv}(G_1, G_2)$ from the discrepancy in the means is at most

$$\frac{||D^{-1}P'(\mu_1 - \mu_2)||}{\sqrt{2\pi}} \leq \frac{||\mu_1 - \mu_2||}{\sqrt{2\pi\lambda}}.$$

We now consider the contribution to $D_{tv}(G_1, G_2)$ from the discrepancy in the covariance matrices. We consider the eigenvalues of $D^{-1}P'\Sigma_2 PD^{-1} = I + D^{-1}P'APD^{-1}$. We have $\max_v \frac{||D^{-1}P'APD^{-1}v||}{||v||} \leq \frac{\alpha}{\lambda}$, and thus the maximum eigenvalue of $I + D^{-1}P'APD^{-1}$ is at most $1 + \frac{\alpha}{\lambda}$, and the minimum eigenvalue is at least $1 - \frac{\alpha}{\lambda}$; thus from Fact A.12 we have

$$\begin{aligned} D_{tv}(G_1, G_2) &\leq \frac{||\mu_1 - \mu_2||}{\sqrt{2\pi\lambda}} + \frac{m\left(\frac{1}{1-\alpha/\lambda} - 1\right)}{\sqrt{2\pi e}} \\ &= \frac{||\mu_1 - \mu_2||}{\sqrt{2\pi\lambda}} + \frac{m\alpha}{\sqrt{2\pi e}(\lambda - \alpha)}. \end{aligned}$$

□

Kullback-Leibler Divergence for Gaussians

Fact A.14. *Let $G_1 = \mathcal{N}(\mu_1, \Sigma_1), G_2 = \mathcal{N}(\mu_2, \Sigma_2)$ be two m dimensional Gaussian distributions.*

$$KL(G_1 \| G_2) = \frac{1}{2}\left(\log \frac{det(\Sigma_2)}{det(\Sigma_1)} + Tr(\Sigma_2^{-1}\Sigma_1) + (\mu_1 - \mu_2)^T \Sigma_2^{-1}(\mu_1 - \mu_2) - m\right).$$

Total Variance via Kullback-Leibler Divergence

Fact A.15. *Let $G_1 = \mathcal{N}(\mu_1, \Sigma_1)$ and $G_2 = \mathcal{N}(\mu_2, \Sigma_2)$ be two m-dimensional Gaussian distributions. Let $\lambda_1, \ldots, \lambda_n > 0$ be the eigenvalues of $\Sigma_1^{-1}\Sigma_2$. Then the total variational distance satisfies,*

$$(D_{tv}(G_1, G_2))^2 \leq \sum_{i=1}^{m}(\lambda_i + \frac{1}{\lambda_i} - 2) + (\mu_1 - \mu_2)^T \Sigma_1^{-1}(\mu_1 - \mu_2).$$

Proof. From Fact A.14, The Kullback-Leibler divergence (KL) between two Gaussians is,

$$\mathrm{KL}(G_1 \| G_2) = \frac{1}{2}\left(Tr(\Sigma_1^{-1}\Sigma_2) + \ln \frac{det(\Sigma_1)}{det(\Sigma_2)} - m + (\mu_1 - \mu_2)^T \Sigma_1^{-1}(\mu_1 - \mu_2)\right).$$

Note that $det(\Sigma_1^{-1}\Sigma_2) = \frac{det(\Sigma_2)}{det(\Sigma_1)} = \lambda_1 \ldots \lambda_n$, and hence $\ln \frac{det(\Sigma_1)}{det(\Sigma_2)} = \sum \ln \frac{1}{\lambda_i}$. Also, $Tr(\Sigma_1^{-1}\Sigma_2) = \lambda_1 + \ldots + \lambda_n$. By Pinsker's inequality, $D_{tv}(G_1, G_2) \leq \sqrt{\mathrm{KL}(G_1 \| G_2)/2}$. This gives,

$$(D(G_1, G_2))^2 \leq \sum_{i=1}^{m}(\lambda_i + \ln \frac{1}{\lambda_i} - 1) + (\mu_1 - \mu_2)^T \Sigma_1^{-1}(\mu_1 - \mu_2).$$

Using the fact that $\log x \leq x - 1$, we are done. □

Moment Bounds for Univariate Mixtures

Here we prove some basic moment bounds for univariate mixtures of Gaussians, used in Chapter 11.

Claim A.16. *The k^{th} raw moment of a univariate Gaussian, $M_k(\mathcal{N}(\mu, \sigma^2)) = \sum_{i=0}^{k} c_i \mu^i \sigma^{k-i}$, where $|c_i| \leq k!$, and $c_i = 0$ for odd i.*

Proof. Consider the moment generating function $M_X(t) = e^{t\mu + \sigma^2 t^2/2}$, and recall that $M_k(\mathcal{N}(\mu, \sigma^2))$ is given by $\frac{d^k M_X(t)}{dt^k}$ evaluated at $t = 0$. We will prove that

$$\frac{d^k M_X(t)}{dt^k} = M_X(t) \sum_{i,j \geq 0 \, s.t. \, i+j \leq k} c_{(i,j,k)} \mu^i \sigma^{2j} t^{2j+i-m},$$

where the coefficient $c_{(i,j,k)}$ is positive and at most $\frac{k!}{(2j+i-k)!}$, from which the claim follows. We proceed by induction on k. The base case, when $k = 1$, is clearly valid. For the induction step, assume the above claim holds for some value $k \geq 1$, and consider computing $\frac{d^{k+1}M_X(t)}{dt^{k+1}}$. To see that the exponents of μ, σ, and t are as claimed, note that the term $M_X(t)c_{(i,j,k)}\mu^i\sigma^{2j}t^{2j+i-m}$ when differentiated becomes

$$\begin{aligned} M_X(t)\left((2j+i-k)c_{(i,j,k)}\mu^i\sigma^{2j}t^{2j+i-(k+1)}\right. &+ c_{(i,j,k)}\mu^{i+1}\sigma^{2j}t^{2j+(i+1)-(k+1)} \\ &+ \left. c_{(i,j,k)}\mu^i\sigma^{2(j+1)}t^{2(j+1)+i-(k+1)}\right), \end{aligned}$$

all of whose terms have the form $M_X(t)c\mu^{i'}\sigma^{2j'}t^{2j'+i'-(k+1)}$, as desired, and thus the coefficient of any term with σ^i for odd i must be zero.

To see that $c_{(i,j,k)} \leq \frac{k!}{(2j+i-k)!}$ holds, note that the three terms that contribute to $c_{(i,j,k+1)}$ are $c_{(i,j,k)}$, $c_{(i-1,j,k)}$, and $c_{(i,j-1,k)}$, where for simplicity we assume that $c_{(i',j',k)} = 0$ if $i' < 0$ or $j' < 0$. In particular,

$$c_{(i,j,k+1)} = c_{(i,j,k)}(2j+i-k) + c_{(i-1,j,k)} + c_{(i,j-1,k)}.$$

By our inductive hypothesis, we have the following:

$$\begin{aligned} c_{(i,j,k+1)} &= c_{(i,j,k)}(2j+i-k) + c_{(i-1,j,k)} + c_{(i,j-1,k)} \\ &\leq \frac{k!}{(2j+i-k)!}(2j+i-k) + \frac{k!}{(2j+i-k-1)!} + \frac{k!}{(2j+i-k-2)!} \\ &= \frac{k!}{(2j+i-k-1)!}\left(1+1+(2j+i-k-1)\right) \\ &\leq \frac{(k+1)!}{(2j+i-(k+1))!}, \end{aligned}$$

as desired. □

Lemma A.17. *Let $M_j(X)$ denote the jth moment of the random variable X. Given γ, ϵ, μ, μ', σ^2, σ'^2 such that the following are satisfied:*

- $|\mu|, |\mu'|, \sigma^2, \sigma'^2 \leq \frac{1}{\epsilon}$,
- $|\mu - \mu'| + |\sigma^2 - \sigma'^2| \leq \gamma < \epsilon$,

then for $\epsilon < 1/j$,

$$|M_j(\mathcal{N}(\mu,\sigma)) - M_j(\mathcal{N}(\mu',\sigma'))| \leq 2(j+1)!\left(\frac{2}{\epsilon}\right)^j\gamma.$$

Proof. We prove this by bounding both $|M_j(\mathcal{N}(\mu,\sigma^2)) - M_j(\mathcal{N}(\mu,\sigma'^2))|$, and $|M_j(\mathcal{N}(\mu,\sigma'^2)) - M_j(\mathcal{N}(\mu',\sigma'^2))|$, by $(j+1)!\left(\frac{2}{\epsilon}\right)^j\gamma$.

From Claim A.16, and our assumption that $|\sigma^2 - \sigma'^2| \leq \gamma$, we have

$$\begin{aligned}
|M_j(\mathcal{N}(\mu,\sigma^2)) - M_j(\mathcal{N}(\mu,\sigma'^2))| &\leq \sum_{i=0}^{j} c_i \mu^i |\sigma^{j-i} - \sigma'^{j-i}| \\
&\leq \sum_{i=0}^{j} c_i \mu^i \left[(2/\epsilon+\gamma)^{(j-i)/2} - (2/\epsilon)^{(j-i)/2}\right] \\
&\leq \sum_{i=0}^{j} c_i \mu^i \left[(2/\epsilon+\gamma)^{j-i} - (2/\epsilon)^{j-i}\right] \\
&\leq \sum_{i=0}^{j} c_i \mu^i (j-i)(\frac{2}{\epsilon})^{j-i}\gamma.
\end{aligned}$$

Where the final inequality above is because $2/\epsilon > j$, and thus for $k \leq j$, the i^{th} term in the expansion of $(2/\epsilon+\gamma)^k$ is at most $k^i(2/\epsilon)^{k-i}\gamma^i \leq (2/\epsilon)^k\gamma$. Plugging in the bounds of $c_i \leq j!$ from Claim A.16 yields the desired bound. The same argument holds for bounding $|M_j(\mathcal{N}(\mu,\sigma'^2)) - M_j(\mathcal{N}(\mu',\sigma'^2))|$, from which the claim follows. □

A.2 Basic Properties of Poissons

In this section we collect the useful facts about the Poisson distribution, and the "Poisson functions," $poi(x,i) := \frac{x^i e^{-x}}{i!}$ that are used in Chapters 3 through 8.

Second Derivative of Poisson Functions

Proposition A.18. *Letting $poi_{xx}(x,j)$ denote the second derivative of the jth Poisson function, for all $x > 0$, $j \geq 0$ we have $|poi_{xx}(x,j)| \leq \min\{2, \frac{2}{x}\}$.*

Proof. Since $poi(x,j) \triangleq \frac{x^j e^{-x}}{j!}$, we have $poi_{xx}(x,j) = (x^j - 2jx^{j-1} + j(j-1)x^{j-2})\frac{e^{-x}}{j!}$.
Case 1: $j = 0$ or 1. We have from the above expression that $poi_{xx}(x,0) = e^{-x}$, which is easily seen to be less than $\min\{2, \frac{2}{x}\}$. Similarly, for $j = 1$ we have $poi_{xx}(x,1) = (x-2)e^{-x}$, where, for $x \in (0,1)$ we have that $|(x-2)e^{-x}| \leq 2e^{-x} \leq 2$. For $x \geq 1$, we must show that $|(x-2)e^{-x}| \leq \frac{2}{x}$, or equivalently, $|\frac{1}{2}x^2 - x| \leq e^x$. Since $|\frac{1}{2}x^2 - x| \leq \frac{1}{2}x^2 + x$, and this last expression is just two terms from the power series of e^x, all of whose terms are positive, it is thus bounded by e^x as desired.
Case 2: $x < 1$ and $j \geq 2$.

In this case we must show $|poi_{xx}(x,j)| \leq 2$. For $j \geq 2$, we note that we may simplify the above expression for $poi_{xx}(x,j)$ to $((x-j)^2 - j)\frac{x^{j-2}e^{-x}}{j!}$. Noting that for $x \in (0,1)$ we have $x^{j-2} \leq 1$ and $e^{-x} < 1$, we may bound the absolute value of this last expression by $\frac{|(x-j)^2 - j|}{j!}$. Since $(x-j)^2 \geq 0$ and $-j \leq 0$, we may bound this expression as $\max\left\{\frac{(x-j)^2}{j!}, \frac{j}{j!}\right\}$; since we have $j \geq 2$ and $x \in (0,1)$, we note that $\frac{(x-j)^2}{j!} \leq \frac{j^2}{j!} \leq 2$, and $\frac{j}{j!} \leq 1$, as desired.
Case 3: $x \geq 1$ and $j \geq 2$.

We reexpress $|poi_{xx}(x,j)|$ as $\left|(1-\frac{j}{x})^2 - \frac{j}{x^2}\right| \cdot poi(x,j)$, which we may bound by $\max\{(1-\frac{j}{x})^2, \frac{j}{x^2}\} \cdot poi(x,j)$.

We consider the second term first. For $j > x+1$, consider the ratio of the expression $\frac{j}{x^2}poi(x,j)$ for consecutive values of j:

$$\frac{j}{j-1}\frac{x^j(j-1)!}{x^{j-1}j!} = \frac{x}{j-1}$$

and note that this is always at most 1. Thus $\frac{j}{x^2}poi(x,j)$ attains its maximum (over j) for $j \leq x+1$. We may thus bound $\frac{j}{x^2}poi(x,j)$ by taking $j \leq x+1$ and noting that, since $poi(x,j) \leq 1$ we have $\frac{j}{x^2}poi(x,j) \leq \frac{x+1}{x^2} \leq \frac{2}{x}$ as desired.

We now consider the first term, $(1-\frac{j}{x})^2 poi(x,j)$ and show that it attains its maximum for j in the interval $[x - 2\sqrt{x}, x + 2\sqrt{x} + 1]$. Consider the ratio of $(1-\frac{j}{x})^2 poi(x,j)$ to $(1-\frac{j-1}{x})^2 poi(x,j-1)$:

$$\frac{(1-\frac{j}{x})^2}{(1-\frac{j-1}{x})^2}\frac{e^{-x}x^j(j-1)!}{e^{-x}x^{j-1}j!} = \left(\frac{x-j}{x-j+1}\right)^2\frac{x}{j} \tag{A.2}$$

We now show that this ratio is at most 1 for $j \geq x + 2\sqrt{x} + 1$, and at least 1 for $j \leq x - 2\sqrt{x} + 1$, thereby showing that $(1 - \frac{j}{x})^2 poi(x, j)$ attains its maximum in the interval $j \in [x - 2\sqrt{x}, x + 2\sqrt{x} + 1]$. We note that both $\frac{x-j}{x-j+1}$ and $\frac{x}{j}$ are decreasing functions of j, outside the interval $[x, x + 1]$, so it suffices to check the claim for $j = x + 2\sqrt{x} + 1$ and $j = x - 2\sqrt{x} + 1$. We have

$$\left(\frac{x - (x + 2\sqrt{x} + 1)}{x - (x + 2\sqrt{x} + 1) + 1}\right)^2 \frac{x}{x + 2\sqrt{x} + 1} = \frac{(2\sqrt{x} + 1)^2}{(2\sqrt{x} + 2)^2} \leq 1$$

and

$$\left(\frac{x - (x - 2\sqrt{x} + 1)}{x - (x - 2\sqrt{x} + 1) + 1}\right)^2 \frac{x}{x - 2\sqrt{x} + 1} = \frac{(2\sqrt{x} - 1)^2}{(2\sqrt{x} - 2)^2} \geq 1$$

Thus $(1 - \frac{j}{x})^2 poi(x, j)$ attains its maximum for j in the interval $[x - 2\sqrt{x}, x + 2\sqrt{x} + 1]$. We note that on the sub-interval $[x - 2\sqrt{x}, x + 2\sqrt{x}]$, we have $(1 - \frac{j}{x})^2 \leq (\frac{2\sqrt{x}}{x})^2 \leq \frac{4}{x}$, and that, for $x \geq 1$, $poi(x, j) \leq \frac{1}{e}$, implying that $(1 - \frac{j}{x})^2 poi(x, j) \leq \frac{2}{x}$ as desired. Finally, for the remainder of the interval, we have, since $x \geq 1$ that $(1 - \frac{j}{x})^2 \leq \frac{(2\sqrt{x}+1)^2}{x^2} \leq \frac{9}{x}$. On this sub-interval $j > x + 2\sqrt{x}$, and thus we have, since $x \geq 1$ and j is an integer, that $j \geq 4$. Since $poi(x, j)$ is maximized with respect to x when $x = j$, this maximum has value $\frac{j^j e^{-j}}{j!}$, which, by Stirling's approximation, is at most $\frac{1}{\sqrt{2\pi j}} < \frac{2}{9}$ (for $j \geq 4$). Combining these two bounds yields the desired bound of $\frac{2}{x}$. □

Tail Bounds for Poisson Distributions

Fact A.19. *(From [56]) For $\lambda > 0$, and an integer $n \geq 0$, if $n \leq \lambda$,*

$$\sum_{i=0}^{n} poi(\lambda, i) \leq \frac{poi(\lambda, n)}{1 - n/\lambda},$$

and for $n \geq \lambda$,

$$\sum_{i=n}^{\infty} poi(\lambda, i) \leq \frac{poi(\lambda, n)}{1 - \lambda/(n + 1)}.$$

Corollary A.20. *For any constant $\epsilon > 0$, there exists a constant $\delta_\epsilon > 0$ such that for any $\lambda \geq 1$, letting $X \leftarrow Poi(\lambda)$*

$$\Pr[|X - \lambda| > \lambda^{\frac{1}{2}+\epsilon}] \leq e^{-\lambda^{\delta_\epsilon}}.$$

Appendix B

The Reduction of Feldman et al. from Learning Juntas and DNF to Learning Parities

B.1 Learning Juntas and DNF via Sparse Parities

We formally state the results of Feldman et al. [54] which reduce the problem of learning Juntas and DNF to the problem of learning parity with noise. The main intuition, and proof approach of [54] is that the problem of learning parities with noise is the problem of finding a heavy Fourier coefficient, given the promise that one exists; in the case of learning a k-junta, one knows that there will be at most 2^k significant Fourier coefficients. The reduction proceeds by essentially peppering the labels with random XORs, so that after the peppering process, with some decent probability, exactly *one* Fourier coefficient will have survived, in which case the problem has been successfully transformed into the problem of learning a parity of size k with noise. It is worth stressing that this reduction results in an instance with a very large noise rate—noise $\frac{1}{2} - \frac{1}{2^k}$, thus highlighting the importance of considering the learning noisy parities problem with noise-rates that approach $1/2$. We conclude this section with formal statements of these reductions.

Theorem B.1 (Feldman et al. [54])**.** *Given an algorithm that learns noisy k-parities on length n strings (under the uniform distribution) with noise rate $\eta \in [0, \frac{1}{2})$ that runs in time $T(n, k, \eta)$, there exists an algorithm that learns k-juntas under the uniform distribution with noise rate η' that runs in time*

$$O\left(k2^{2k} \cdot T(n, k, \frac{1}{2} - \frac{1-2\eta'}{2^k})\right).$$

Theorem B.2 (Feldman et al. [54])**.** *Given an algorithm that learns noisy k-parities on length n strings (under the uniform distribution) with noise rate $\eta \in [0, \frac{1}{2})$ that takes $S(n, k, \eta)$ examples and runs in time $T(n, k, \eta)$, there exists an algorithm that (ϵ, δ)–PAC learns r-term DNF formulae under the uniform distribution that runs in time*

$$\tilde{O}\left(\frac{r^4}{\epsilon^2} \cdot T\left(n, \log\left(\tilde{O}(r/\epsilon)\right), \frac{1}{2} - \tilde{O}(\epsilon/r)\right) \cdot S\left(n, k, \log\left(\tilde{O}(r/\epsilon)\right), \frac{1}{2} - \tilde{O}(\epsilon/r)\right)^2\right).$$

Additionally, as Feldman observed, an improved algorithm for learning noisy k-parities can be used, via the reduction of Feldman et al. [54] to yield an improvement in runtime of the approach of Mossel et al [92] for the problem of learning k-juntas *without* noise. The key observation of Mossel et al. is that either a k-junta has a heavy Fourier coefficient of degree at most d, or, when represented as a polynomial over $\mathbb{F}_2$, has degree at most $k - d$. Their algorithm proceeds by brute force-searching for a heavy Fourier coefficients of order at most αk for some appropriately chosen α; if none are found, then the junta is found by solving a linear system over $n^{(1-\alpha)k}$ variables. Given an improved algorithm for learning noisy parities, using the reduction of Feldman et al., one improve upon the brute-force search component of the algorithm of Mossel et al. The following corollary quantifies this improvement.

Corollary B.1. *Given an algorithm that learns noisy j-parities on length n strings (under the uniform distribution) with noise rate $\eta \in [0, \frac{1}{2})$ that runs in time $T(n, j, \eta)$, for any*

$\alpha \in (0,1)$*, there exists an algorithm that learns* k*-juntas* without noise *under the uniform distribution in time*

$$\max\left(T(n, \alpha k, \frac{1}{2} - \frac{1}{2^{\alpha k}}), n^{\omega k(1-\alpha)}\right) poly(n).$$