A Nonparametric Bayesian Model for Multiple Clustering with Overlapping Feature Views

Donglin Niu
Northeastern University
dniu@ece.neu.edu

Jennifer G. Dy
Northeastern University
jdy@ece.neu.edu

Zoubin Ghahramani
University of Cambridge
zoubin@eng.cam.ac.uk

Abstract

Most clustering algorithms produce a single clustering solution. This is inadequate for many data sets that are multi-faceted and can be grouped and interpreted in many different ways. Moreover, for high-dimensional data, different features may be relevant or irrelevant to each clustering solution, suggesting the need for feature selection in clustering. Features relevant to one clustering interpretation may be different from the ones relevant for an alternative interpretation or view of the data. In this paper, we introduce a probabilistic nonparametric Bayesian model that can discover multiple clustering solutions from data and the feature subsets that are relevant for the clusters in each view. In our model, the features in different views may be shared and therefore the sets of relevant features are allowed to overlap. We model feature relevance to each view using an Indian Buffet Process and the cluster membership in each view using a Chinese Restaurant Process. We provide an inference approach to learn the latent parameters corresponding to this multiple partitioning problem. Our model not only learns the features and clusters in each view but also automatically learns the number of clusters, number of views and number of features in each view.

1 Introduction

Clustering is the process of grouping objects based on some notion of similarity. It is commonly applied for exploratory analysis, segmentation, preprocessing and data summarization. Traditional clustering algorithms [15] find one partitioning of the data. However, richly-structured high-dimensional data can often be multi-faceted. Instead of having a single way of partitioning all the data, there may be multiple reasonable clustering interpretations. For example, human face images can be grouped based on identity or their pose; news webpages can be grouped by topic or by region. In such cases, a single partitioning model (such as a mixture model) is not sufficient.

Similarity is dependent on the features describing data. The presence of noisy and irrelevant features can degrade clustering performance, making feature selection an important factor in cluster analysis [9]. However, most approaches to feature selection in clustering only try to find one clustering solution and as such only one subset of features relevant to that task. But when data is multi-faceted, there may exist multiple clustering interpretations; and different feature subsets may be relevant to each interpretation. Features irrelevant to one clustering interpretation may be relevant to another clustering solution. For example, in news webpage clustering, some terms may be relevant to the clustering view based on topic and others may be appropriate for clustering based on region. Moreover, there is no reason to require these feature subsets to be disjoint.

In this paper, we call each clustering solution or interpretation a *view* of the data. We introduce a nonparametric Bayesian model for simultaneously discovering multiple clustering views and the corresponding features in each view. In this model, data instances that belong to the same cluster in one view can belong to different clusters in other views. Moreover, our model allows overlap of features among the views; i.e., two or more views may share common features. Because we utilize a nonparametric model, our approach can also automatically learn the number of clustering views, the number of clusters in each view, and the number of features in each view.

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

2 Related Work

There is a growing recent interest in methods for finding more than one clustering interpretation. These methods can be categorized as either sequential (iterative) or simultaneous approaches. Sequential or iterative methods find alternative clustering solutions given one or more existing solution. Gondek and Hofmann [11] finds an alternative non-redundant clustering by a conditional information bottleneck approach. Bae and Bailey [1] utilize cannot-link constraints and agglomerative clustering. Qi et al. [21] minimize the Kullback-Leibler divergence between the distribution of the original space and the projection subject to the constraint that sum squared error between samples in the projected space with the means of the clusters they belong to is smaller than a pre-specified threshold. Dang and Bailey in [7] maximizes the mutual information between the new clusters and data at the same time minimizing the alternative from a reference clustering; while in [6], they apply maximum likelihood and mutual information to find quality and alternative clusterings. Cui et al. [5] finds multiple alternative views by clustering in the subspace orthogonal to the clustering solutions found in previous iterations.

Simultaneous approaches discovers multiple clustering solutions simultaneously. Meta clustering [4] generates a diverse set of clustering solutions by either random initialization or random feature weighting then hierarchically clusters these solutions. Jain et al. in [16] learns two disparate clusterings by minimizing two k-means type sum-squared error objective while at the same time minimizing the correlation between these two clusterings. Poon et al. [20] build a latent pouch tree model for selecting features in each clustering solution. Dasgupta and Ng [8] utilizes each eigenvector in spectral clustering to generate multiple clusterings. Niu et al. [18] simultaneously learns multiple subspaces that provide multiple views and clusterings in each view by augmenting a spectral clustering objective function to incorporate dimensionality reduction and penalize dependency among views based on a kernel independence criterion. The approach we introduce in this paper is also a simultaneous method. However, unlike all these methods, we provide a probabilistic generative nonparametric model which can learn the features and clustering solutions in each view simultaneously.

There are several nonparametric Bayesian models introduced for unsupervised learning [19, 3, 12, 23, 14]. The Chinese Restaurant Process (CRP) [19] and the stick-breaking Dirichlet Process Model [3] only assume one underlying partitioning of the data samples. The Indian Buffet Process (IBP) assumes that each sample is generated from an underlying latent set of features sampled from an infinite menu of features. There are also nonparametric Bayesian models for co-clustering [23]. None of these model multiple clustering solutions. There are, however, a few recent papers that provide a nonparametric Bayesian model for finding multiple partitionings: one called cross-categorization [17] utilizes a CRP-CRP construction and Gibbs sampling for inference, and another one [13] utilizes a multiple clustering stick-breaking construction and provide a variational inference approach to learn the model parameters and latent variables. Both [13, 17] assume that the features in each view are disjoint. However, in many applications, some features maybe shared among views. For example, in face images, an intensity feature can be useful for grouping images based on identity; it can also be useful for distinguishing pose. *We, thus, would like a model that allows feature overlap among views.* Recognizing the need for feature sharing in natural language processing [22], the authors in [22] first applied latent dirichlet allocation to assign features to views, then applied CRP on each view, as two separate processing steps. In contrast, *this paper provides a single nonparametric Bayesian generative model that can discover multiple clustering views where the views are allowed to share features.*

3 Nonparametric Multiple Clustering Model with Overlapping Feature Views

Given data $X \in \mathbb{R}^{N \times D}$, where N is the number of samples and D is the number of features. $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_D]$, where $\mathbf{x}_n \in \mathbb{R}^D$ are the samples. Our goal is to build a model that can learn multiple possible interesting alternative clustering solutions and at the same time learn the features describing the partitioning in each view, v. Each view is generated from a subset of the original features, where the features among views are allowed to overlap.

Figure 1 illustrates the type of latent partitioning structure that we would like to learn from our data. The n axis indexes the data samples, d the features and v the views respectively. The cube face with features and views axes in Figure 1 shows which feature belongs to which view. Let us call this the *latent view structure*. If a feature belongs to view v, the corresponding tile is shown in black. It is white otherwise. We allow features to overlap (i.e., features that belong to one view can also belong to another view). The cube face with instances and views axes in Figure 1 shows the partitioning of the samples in each view. Let us call this the *latent cluster structure* in each view. Samples that belong to the same cluster are coded with the

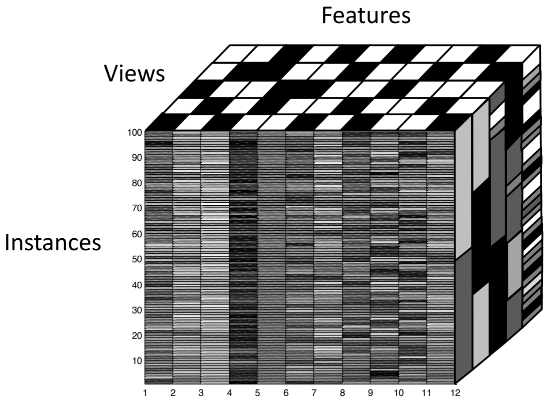

Figure 1: Instances, features and views form a cube. The instances and features face is the data. The features and views face indicate the membership of features in views. The instances and views face indicate multiple cluster labelings.

same shade of gray in each view. Note that samples that belong together in one view may belong to different clusters in other views. We would like to design a nonparametric Bayesian model that can generate such a latent structure (i.e., learn the features in each view allowing feature overlap and the partitioning of the samples in each view).

3.1 Latent View Structure Model

Let $Y \in \mathbb{R}^{D \times V}$ be a matrix of binary latent indicator variables, such that its elements $y_{d,v} = 1$, if the feature d belongs to view v and 0 otherwise. V here is the total number of views. In [13, 17], they assume that the features in each view are disjoint and accordingly model this as a partitioning problem with the constraint that the $\sum_v y_{d,v} = 1$ (i.e., a feature can only belong to one view). However in our case, we would like to allow sharing/overlap of features in different views. To allow the feature subsets belonging to different views to overlap, we need to remove this constraint and allow multiple ones in each row of Y. Instead of assigning each feature to only V views, we now allow 2^V possible assignments; each $y_{d,v}$ can be either one or zero. We model this as coming from a Bernoulli distribution and the parameters of the Bernoulli as coming from a Beta distribution, its conjugate prior. Rather than fixing the number of views, we allow the number of views to go to infinity and apply the Beta-Bernoulli model underlying the Indian Buffet Process (IBP) [12], enabling us to automatically learn the number of views. Heller and Ghahramani [14] also utilizes IBP to model overlap; however, this work only find one clustering interpretation/view and the overlap they consider is in the clusters (i.e., on the instances); whereas, we allow overlap on the features among views.

The Indian Buffet Process (IBP) [12] defines a distribution on infinite binary matrices. It uses a culinary metaphor in explaining the stochastic process. Many Indian buffets seem to offer an infinite number of dishes. Customers enter a restaurant and each customer sequentially samples dishes from infinitely many dishes arranged in a line. In our case, the features are the D customers and the views are the dishes. We assume that each feature d belongs to view v with probability π_v, and that the features are generated independently. The probability of a matrix Y given $\pi = \pi_1, \pi_2, \ldots, \pi_V$ is

$$P(Y|\pi) = \prod_{v=1}^{V} \prod_{d=1}^{D} P(y_{d,v}|\pi_v) = \prod_{v=1}^{V} \pi_v^{n_v} (1-\pi_v)^{D-n_v}$$

where $n_v = \sum_d y_{d,v}$ is the number of $y_{d,v} = 1$ in each view v. We use the conjugate prior for the binomial which is a beta distribution Beta(r, s) to generate π_v. Here we assume $r = \frac{\alpha}{V}$ and $s = 1$. The probability model is then

$$\begin{aligned} \pi_v|\alpha &\sim \text{Beta}(\frac{\alpha}{V}, 1) \\ y_{d,v}|\pi_v &\sim \text{Bernoulli}(\pi_v) \end{aligned}$$

Integrating over all values for π, we get the marginal probability

$$P(Y) = \prod_{v=1}^{V} \frac{\frac{\alpha}{V}\Gamma(n_v + \frac{\alpha}{V})\Gamma(D - n_v + 1)}{\Gamma(D + 1 + \frac{\alpha}{V})}$$

By taking $V \to \infty$, we can get the distribution of Y with infinite V. The distribution is exchangeable. D customers enter a restaurant one after another. Each customer encounters a buffet consisting of infinitely many dishes arranged in a line. The first customer starts at the left of the buffet and takes a serving from each dish stopping after a Poisson(α) number of dishes. The dth customer moves along the buffet, sampling dishes in proportion to their popularity with probability $\frac{n_v}{d}$, where n_v is the number of previous customers who have sampled the dish. Having reached the end of all the previously sampled dishes, the dth customer then tries a Poisson$(\frac{\alpha}{d})$ number of new dishes.

In IBP, the expected number of views per feature and the total number of views are determined by α. However, this original Indian Buffet Process does not have any control over view *stickiness*. In [10], the authors present a two-parameter generalization of the IBP which lets us tune independently the average number of views each feature possesses and the overall number of views used in a set of D features. In the original IBP, α controls both the number of latent views per feature, and the amount of overlap between these latent views. Keeping the average number of views per feature at α as before, the two-parameter IBP defines a model in which the overall number of views can range from α (extreme stickiness/ herding, where all views are shared among all features) to $D\alpha$ (extreme repulsion/individuality). The generalization is as follows: in the beta distribution, the parameter r takes $(\alpha\beta)/V$ and s takes β (setting $\beta = 1$ leads to the one-parameter IBP). The joint distribution of the latent feature indicators becomes

$$p(Y) = \prod_{v=1}^{V} \frac{B(n_v + \frac{\alpha\beta}{V}, D - n_v + \beta)}{B(\frac{\alpha\beta}{V}, \beta)}$$

$$= \prod_{v=1}^{V} \frac{\Gamma(n_v + \frac{\alpha\beta}{V})\Gamma(D - n_v + \beta)}{\Gamma(D + \frac{\alpha\beta}{V} + \beta)} \frac{\Gamma(\frac{\alpha\beta}{V} + \beta)}{\Gamma(\frac{\alpha\beta}{V})\Gamma(\beta)}$$

β is introduced to preserve the average number of views per feature. This time, the dth customer samples a dish with probability $\frac{n_v}{\beta-1+d}$ and a new dish with probability $\frac{\alpha\beta}{\beta-1+d}$. The expected overall number of views is $\overline{V_+} = \alpha \sum_{d=1}^{D} \frac{\beta}{\beta+d-1}$, and that the distribution of this number is Poisson. We can see from this that the total number of views used increases as β increases, so we can interpret β as the view repulsion, or $1/\beta$ as the view stickiness. For finite β, the asymptotic of $\overline{V_+}$ for large D is $\overline{V_+} \sim \alpha\beta \ln D$. In our probabilistic multiple clustering setting, this two-parameter IBP is used to model the prior probability for indicating which feature belongs to which view: α controls the expected number of views a feature belongs; β controls the view *stickiness* or sharing of features among different views.

3.2 Latent Cluster Structure Model

Given the features in each view, we partition the instances into clusters based only on the features in view v by modeling these instances as coming from a mixture model. One may opt to use the standard finite mixture model; but in this paper, we utilize the Chinese Restaurant Process (CRP) [19] which allows us to also automatically learn the number of clusters in each view. Let $Z \in \mathbb{R}^{N \times K \times V}$ be a matrix of latent cluster indicator variables representing the partitioning of samples in all views, where each element, $z_{n,k,v} = 1$ if sample $\mathbf{x}_n$ belongs to cluster k in view v and $z_{n,k,v} = 0$ otherwise. K is the number of clusters in the view with the maximum number of clusters, and K_v is the number of unique and nonempty clusters in view v.

The Chinese Restaurant Process [19] defines a distribution on partitions in a Dirichlet process mixture whose name is inspired by the seemingly infinite tables in San Francisco's Chinatown restaurants [19]. The infinite set of tables are analogous to our clusters and the customers, our data samples. The first customer always sits at the first table. The nth customer sits at table k in view v with probability $\frac{n_k}{n+\eta}$, proportional to the number of customers (samples) already seated at table k, or will sit at a new table with probability $\frac{\eta}{n+\eta}$, proportional to the scalar concentration hyperparameter η. The CRP is an exchangeable distribution on partitions; meaning that, the distribution is invariant to the order in which customers (samples) are assigned to tables (clusters). By serving each table a separate dish, CRP provides a representation for a Dirichlet process mixture [19].

3.3 Cluster Component Model

For each cluster in each view, we assume a component density model with parameter $\boldsymbol{\gamma}$ and hyperparameter $\boldsymbol{\lambda}$ depending on the data. In our experiments, we apply a Gaussian component model for data with real-valued features and multinomial on data based on text. We describe these two common cluster component models in the inference Section 4. In general, one may apply other component models as appropriate.

3.4 Overall Model

Figure 2 shows a graphical model of our nonparametric multiple clustering model.

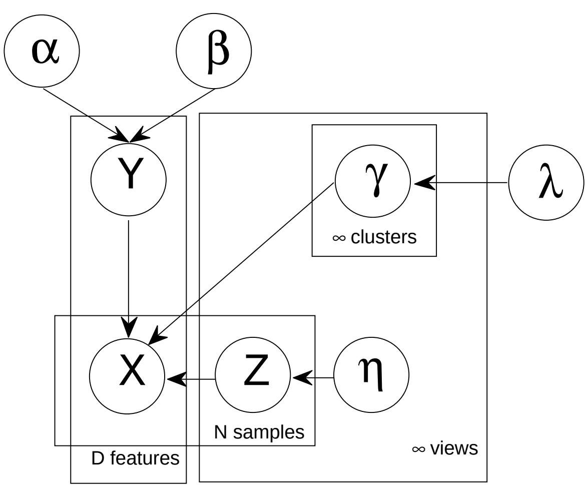

Figure 2: Graphical model for nonparametric multiple clustering model.

The joint distribution for all variables $p(X, Y, Z, \boldsymbol{\gamma})$ is:

$$
\begin{aligned}
&p(X|Y, Z, \boldsymbol{\gamma})p(Y|\alpha, \beta)p(Z|\alpha_v)p(\boldsymbol{\gamma}|\boldsymbol{\lambda}) \\
&= \frac{1}{c}\left[\prod_{v=1}^{V}\prod_{n=1}^{N} p(\mathbf{x}_{n,v}|\boldsymbol{\gamma}_v, Y, Z)\right]\left[\prod_{v=1}^{V}\prod_{d=1}^{D} p(y_{d,v}|\alpha, \beta)\right] \\
&\left[\prod_{v=1}^{V}\prod_{n=1}^{N} p(z_{n,v}|\eta_v)\right]\left[\prod_{v=1}^{V}\prod_{k=1}^{K_v} p(\boldsymbol{\gamma}_{k,v}|\boldsymbol{\lambda})\right]
\end{aligned}
$$

where $\mathbf{x}_{n,v} = (x_{n,d} : y_{d,v} = 1)$ is a vector representing sample n in view v comprised of only the corresponding features selected in that view, $\boldsymbol{\gamma}$ are the cluster parameters with hyperparameter $\boldsymbol{\lambda}$, V is the number of views and K_v is the number of clusters in view v, N is the number of data points and c is the normalizing constant which is needed to ensure that the density integrates to one. We express $p(X|Y, Z, \boldsymbol{\gamma})$ as a product of the densities for each view. Note that because we allow features to overlap in each view, the constant c makes sure the probability normalizes to 1.

The overall generative process for generating the multiple views is as follows:

1. Latent view, Y: Generate Y indicating which feature belongs to which view from a two-parameter Indian Buffet Process $IBP(\alpha, \beta)$.

2. Latent clusters, Z: For each view $v = \{1, 2, \ldots\}$, generate a partitioning for instances $z_{n,k,v}$ from a Chinese Restaurant Process $CRP(\eta)$.

3. Observation data, X: Data are generated from a multiplicative mixture of views, where each view is a mixture of cluster components. For each cluster in each view, we draw cluster parameter $\boldsymbol{\gamma}_{k,v}$ from the prior parameter distribution with hyperparameter $\boldsymbol{\lambda}$: $\boldsymbol{\gamma} \sim p(\boldsymbol{\gamma}|\boldsymbol{\lambda})$. This gives us the cluster component densities in each view, $p(\mathbf{x}_{n,v}|\boldsymbol{\gamma}_{z_{n,v},v})$, where $z_{n,v}$ is equal to the cluster k that sample n belongs to in view v. We now have V alternative CRP mixture distribution views. We combine the V views by assuming a multiplicative mixture to generate our observation X: $\mathbf{x}_n \sim p(\mathbf{x}_n|\boldsymbol{\gamma}, Y, Z) = \frac{1}{c}\prod_{v=1}^{V} p(x_{n,v}|\boldsymbol{\gamma}_v, Y, Z)$, where c is a normalization term.

Note that our model deals with cluster quality and non-redundancy simultaneously. Our model handles the cluster quality by estimating the joint distribution for each cluster in each view. The model handles clustering redundancy in an implicit way. Redundant clusterings increase the model complexity by adding more overlapping features which is penalized by the Indian Buffet Process prior and the multiplicative mixture model.

4 Inference by Gibbs Sampling

In this section, we describe a Gibbs sampling algorithm for estimating our posterior distribution. The sampling process has three steps. Step one is the feature assignment step, sampling Y conditioned on Z, X and $\boldsymbol{\gamma}$. Step two involves instance assignment, sampling Z conditioned on Y, X and $\boldsymbol{\gamma}$. And step three involves updating the parameters $\boldsymbol{\gamma}$. We iteratively repeat these steps until convergence to a stable distribution.

In step one, the hyperparameters α and β for Y are assumed known. Due to the exchangeability of IBP, we can derive the following Gibbs sampler as follows. The dth object can be taken to be the last customer to visit the buffet, then $P(y_{d,v} = 1|Y_{-(d,v)}, Z, X, \boldsymbol{\gamma}) \propto P(y_{d,v} = 1|Y_{-(d,v)})P(X|Y, Z, \boldsymbol{\gamma})$ and $P(y_{d,v} = 0|Y_{-(d,v)}, Z, X, \boldsymbol{\gamma}) \propto P(y_{d,v} = 0|Y_{-(d,v)})P(X|Y, Z, \boldsymbol{\gamma})$, where $Y_{-(d,v)}$ are the Y entries other than $y_{d,v}$, $P(y_{d,v} = 1|Y_{-(d,v)}) = \frac{n_v}{\beta+D-1}$ and $P(y_{d,v} = 0|Y_{-(d,v)}) = 1 - P(y_{d,v} = 1|Y_{-(d,v)})$. $p(X|Y, Z, \boldsymbol{\gamma})$ is the observation likelihood calculated as:

$$
p(X|Y, Z, \boldsymbol{\gamma}) = \frac{1}{c}\prod_{v=1}^{V}\prod_{n=1}^{N} p(\mathbf{x}_{n,v}|\boldsymbol{\gamma}, Y, Z) \qquad (1)
$$

where c is the normalizing constant. To form new views, we use $P(y_{d,v_{new}} = 1|Y_{-(d,v_{new})}, Z, X, \boldsymbol{\gamma}) \propto P(y_{d,v_{new}} = 1)P(X|Y, Z, \boldsymbol{\gamma})$, where $P(y_{d,v_{new}} = 1) = \frac{\alpha\beta}{\beta+D-1}$ from the two-parameter IBP and $P(X|Y, Z, \boldsymbol{\gamma})$ is the likelihood Equation 1 where $p(\mathbf{x}_{n,v_{new}}|\boldsymbol{\gamma}, Y, Z)$ is calculated from a CRP clustering process for the single feature d.

In step two, we assign the data samples to clusters in each view by sampling the cluster indicator variables. We update the latent cluster indicator variables as follows. $P(z_{n,v} = k|z_{-n,v}, Y, X, \boldsymbol{\gamma}) \propto p(X|Y, Z, \boldsymbol{\gamma})P(z_{n,v} = k|z_{-n,v})$, where $P(z_{n,v} = k|z_{-n,v}) = \frac{n_k}{N+\eta}$ and $P(z_{n,v} = k_{new}|z_{-n,v}) = \frac{\eta}{N+\eta}$ to form new cluster k_{new}. The observation likelihood $p(X|Y, Z, \boldsymbol{\gamma})$ is calculated as in Equation 1.

In step three, we update the cluster component models based on the posterior distribution of $\boldsymbol{\gamma}$. We know the form of the posterior distribution of $\boldsymbol{\gamma}$ because we assumed that $\boldsymbol{\gamma}$ was generated from its corresponding conjugate prior. In particular, we provide the update equations for two common cluster component models, the Gaussian and the multinomial components.

Gaussian Component. For the Gaussian distribution, the parameter $\boldsymbol{\gamma}_{v,z_{n,v}}$ comprises the mean $\boldsymbol{\mu}_{v,z_{n,v}}$ and precision matrix $\Lambda_{v,z_{n,v}}$ ($\Sigma^{-1}_{v,z_{n,v}}$) in view v and cluster $z_{n,v}$. We use a Gaussian-Wishart distribution $p(\boldsymbol{\gamma}_{v,z_{n,v}}|\boldsymbol{\lambda})$ for $\boldsymbol{\gamma}$ since it is conjugate to the Gaussian distribution. For each view, the hyper-parameter $\boldsymbol{\lambda}$ is a vector composed of $\mathbf{m}_{0,v}$, $W_{0,v}$, $\beta_{0,v}$, and $\nu_{0,v}$. The Gaussian likelihood distribution, $p(X|Y,Z,\boldsymbol{\gamma})$ is:

$$p(X|Y,Z,\boldsymbol{\gamma}) = \prod_{v=1}^{V}\prod_{n=1}^{N} \frac{1}{(2\pi)^{D_v/2}|\Sigma_{v,z_{n,v}}|^{1/2}}$$
$$\exp\left(-\frac{1}{2}(x_{n,v}-\boldsymbol{\mu}_{v,z_{n,v}})^T\Sigma_{v,z_{n,v}}^{-1}(x_{n,v}-\boldsymbol{\mu}_{v,z_{n,v}})\right)$$

where the indices indicate the corresponding view v and cluster component $z_{n,v}$ and D_v is the number of features in view v. $\mathbf{m}_{k,v}$, $W_{k,v}$, $\beta_{k,v}$ and $\nu_{k,v}$ are the parameters of the posterior Gaussian-Wishart distribution for cluster k in view v. And the update equations are:

$$\begin{aligned}
\beta_{k,v} &= \beta_{0,v} + N_{k,v} \\
\mathbf{m}_{k,v} &= \frac{1}{\beta_{k,v}}(\beta_{0,v}m_{0,v} + N_{k,v}\bar{x}_{k,v}) \\
W_{k,v}^{-1} &= W_{0,v}^{-1} + N_{k,v}S_{k,v} \\
&\quad + \frac{\beta_{0,v}N_{k,v}}{\beta_{0,v}+N_{k,v}}(\bar{x}_{k,v}-m_{0,v})(\bar{x}_{k,v}-m_{0,v})^T \\
\nu_{k,v} &= \nu_{0,v} + N_{k,v}
\end{aligned}$$

where $\bar{\mathbf{x}}_{k,v}$ and $S_{k,v}$ are the sample mean and sample covariance of $\mathbf{x}$ in cluster k of view v and $N_{k,v}$ are the number of samples in cluster k in view v.

Multinomial Component. Assuming $p(\mathbf{x}_{n,v}|\boldsymbol{\gamma}_{v,z_{n,v}})$ comes from a multinomial distribution, the parameter $\boldsymbol{\gamma}_{v,z_{n,v}}$ comprises of the probability of occurrence for each feature in view v and cluster $z_{n,v}$. We use a Dirichlet distribution, the conjugate prior to a multinomial distribution as our prior $p(\boldsymbol{\gamma}_{v,z_{n,v}}|\boldsymbol{\lambda})$, with hyperparameters $\boldsymbol{\lambda} = \{\alpha_{0,1},\ldots,\alpha_{0,D_v}\}$. The multinomial likelihood distribution, $p(X|Y,Z,\boldsymbol{\gamma})$ is:

$$\prod_{v=1}^{V}\prod_{n=1}^{N} p(\mathbf{x}_{n,v}|\boldsymbol{\gamma}_{v,z_{n,v}}) = \prod_{v=1}^{V}\prod_{n=1}^{N}\prod_{l=1}^{D_v}\rho_{v,z_{n,v},l}^{x_{n,v,l}}$$

Here, we use the notation $x_{n,v,l}$ to be the value of the lth feature in view v in sample n. $\alpha_{k,v,l}$ is the parameter for the posterior Dirichlet distribution for the lth feature in cluster k of view v. And the posterior update equation is $\alpha_{k,v,l} = \alpha_{0,l} + x_{k,v,l}$, where $x_{k,v,l}$ is the count of the lth feature in cluster k of view v. We, then, repeat steps one (latent feature update), two (cluster assignment) and three (parameter update) until they converge to a stable distribution.

Initialization We initialize our Gibbs sampler using a multiple forward sequential search strategy. Our method extends the forward sequential search for one feature subset [9] that optimizes the likelihood of a mixture model as criterion for selecting features to multiple search. First, we perform a sequential forward search to select one subset of features. This becomes the initial set of features for view 1. Next, we select the first feature to include for the next view from the rest of the features. To allow sharing of features, we then perform sequential forward search on the whole set of features starting from this first feature. We continue this process until all the features belong to at least one view. Note that this process does not impose the disjoint subspaces constraint.

5 Experiments

In this section, we perform experiments on three synthetic data sets of varying feature overlap and on five real data sets (a face image, a machine sound and three text datasets) to investigate whether our algorithm gives reasonable multiple clustering solutions. We compare our method, nonparametric Bayesian multiple clustering with overlapping feature views (NBMC-OFV), against a nonparametric probabilistic multiple clustering method based on (CRP-CRP) [17], a simultaneous multiple clustering approach (de-correlated kmeans, D-kmeans) [16], a sequential approach (orthogonal projection, orthProj) method [5] and a baseline nonparametric Dirichlet process mixture model (DP-Gaussian or DP-Multinomial) [3]. The CRP-CRP [17], uses a Chinese restaurant process to cluster features into subspaces and Chinese restaurant process again to cluster instances into clusters in each subspace. In their model, features among views are assumed disjoint. In [16], they use two kmeans objective penalized by a correlation term between the views to find multiple clusterings. In [5], they apply orthogonal projection to discover an alternative subspace and clustering. We also compare with a nonparametric Dirichlet process mixture [3] to serve as a baseline. In all our experiments, we apply the Gaussian cluster component for the synthetic, image and sound data and the multinomial cluster component for text data. In the IBP, α is the expected number of clustering views each feature possesses, and β roughly controls the total number of clustering views in the data. η is set roughly according to the expected number of clusters in each view. Clustering results are based on the expected values from the Markov chain Monte-Carlo samples after burn-in.

To show how well our method compares against competing methods in discovering the "true" labeling, we report the normalized mutual information (NMI) [24] between the clusters found by these methods with the "true" class labels. Let C represent the clustering results and L the labels, $NMI = \frac{MI(C,L)}{\sqrt{H(C)H(L)}}$, where $MI(C,L)$ is the mutual information between random variables C and L and $H(\cdot)$ is the entropy. Note that in all our experiments, labeled information are not used for training; we only use them to measure the performance of our clustering algorithms. Higher NMI values mean the more similar the clustering results are with the labels; and it only reaches its maximum value of one when both clustering and labels are perfectly matched. Since we have multiple views and label interpretation, for all methods, we report the best one-to-one mapping between the clustering views to the different label views based on NMI. For unlabeled text data, we use *perplexity* to evaluate our model. We would like to achieve high likelihood for the dataset given our model. The *perplexity* is the inverse of the geometric mean per-word likelihood. A lower perplexity score indicates better performance. For a set X of N documents, the *perplexity* is defined as

$$perplexity(X) = \exp\left\{-\frac{\sum_{n=1}^{N}\log p(w_n)}{\sum_{n=1}^{N} M_n}\right\}$$

where w_n is the word vector in document n and M_n is the total number of words in document n.

5.1 Synthetic Data

To get a better understanding of our method and test its applicability, we first perform our approach on three synthetic data sets. Each synthetic data set has 1000 instances and 100 features. The first synthetic data is generated from ten clustering views. Each clustering view has 10 features without overlap. Three Gaussian clusters are generated in each feature subspace. Clustering views are independent to each other. The second synthetic data is also generated from ten clustering views. However, each clustering view has 11 features with slight overlap: 10 features belong to more than one view. Three Gaussian clusters are generated in each feature subspace. The third synthetic data is generated from heavily overlapped clustering views. The binary matrix which indicates the feature membership on each view is generated from an IBP; then three Gaussian clusters are generated in each view.

Table 1: Average NMI Results of the Ten Views on Data 1, 2 and 3

	Data1	Data2	Data3	time(10^3s)
NBMC-OFV	**0.89**	**0.81**	**0.59**	1.1
orthProj	0.78	0.65	0.56	0.53
D-kmeans	0.64	0.41	0.48	0.65
CRP-CRP	0.87	0.66	0.34	0.87
DP-Gauss	0.24	0.27	0.23	0.016

Table 1 shows the average NMI results of the ten views compared to the true labels in each synthetic data and the average running time. Without feature overlap, both our method (NBMC-OFV) and CRP-CRP perform well on Data 1. With slight and heavy feature overlap, our method outperforms CRP-CRP and the other methods since our method allows feature overlap. For all the synthetic data, DP-Gaussian does not discover useful solutions since it can only output a single clustering and the features from the other views make it difficult for DP-Gaussian to discover any one of the views.

5.2 Real Data

We now test our method on five real-world data sets to see whether we can find meaningful clustering views. In particular, we test our method on a face image, a machine sound data and three text data. We select data that have high-dimensionality and multiple possible partitionings. We compare our results to true labeling. For text data without labeling, we evaluate our method with *perplexity.*

Face, Machine Sound and WebKB Data. The face dataset from UCI KDD repository [2] consists of 640 face images of 20 people taken at varying poses (straight, left, right, up). The two dominant views of this face data are: identity of the person and their pose. These two views are non-redundant, meaning that with the knowledge of identity, no prediction of pose can be made. We test our method to see whether we can find these two clustering views. Each person has 32 images with four equally distributed poses. The image resolution is 32×30 pixels, resulting in a data set with 640 instances and 960 features. Machine sound data is comprised of 280 sound instances collected from acoustic accelerometer signals of different machines inside buildings. The goal is to classify these sounds into three basic machine classes: `pump, fan, motor`. Each sound instance can be from one machine, or mixture of two or three machines. As such, this data has a multiple clustering view structure. In one view, data can be grouped as `pump` or `no pump`; the other two views are similarly defined. We represent each sound signal by its FFT (Fast Fourier Transform) coefficients, providing us with $100,000$ coefficients. We select the 1000 highest values in the frequency domain as our features. This data set [1] contains html documents from four universities: Cornell University, University of Texas, Austin, University of Washington and University of Wisconsin, Madison. We removed the miscellaneous pages and sub-sampled a total of 1041 pages from four web-page owner types: course, faculty, project and student. We pre-processed the data by removing rare

[1] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

Table 2: NMI Results on Face, Machine Sound, and WebKB Data

	Face		WebKB		Machine Sound		
	Identity	Pose	University	Owner	Pump	Motor	Fan
NBMC-OFV	**0.89**	**0.57**	**0.85**	0.70	**0.89**	**0.86**	**0.87**
orthProj [5]	0.83	0.48	0.78	**0.75**	0.73	0.68	0.47
D-kmeans [16]	0.70	0.40	0.48	0.57	0.64	0.58	0.75
CRP-CRP [17]	0.86	0.53	0.63	0.68	0.81	0.78	0.83
DP-Gaussian [3]	0.84	0.03	-	-	0.25	0.32	0.16
DP-Multinomial [3]	-	-	0.26	0.39	-	-	-

words, stop words, and words with low variances. The two views we want the algorithm to discover are either based on university or based on owner type.

Table 2 provides the NMI results of the different algorithms compared to the labeled views for these data sets. Results show that our method provided the best results compared to competing methods. NBMC-OFV is better than CRP-CRP because we allow feature overlap. It is better than orthogonal projection because orthogonal projection is a sequential approach and also because their feature views are strictly orthogonal; whereas, NBMC-OFV allow some overlap. De-correlated kmeans is worse because it does not perform feature selection in each view; it utilizes all the features in all views. Finally, DP-Gaussian for the real-valued data and DP-multinomial for the text data are limited because they can only discover one view.

NSF Abstract and NYTimes Data. The NSF dataset [2] consists of 129, 000 abstracts from year 1990 to 2003. Each text instance is represented by the frequency of occurrence of each word. This dataset is richly structured. It covers a wide range of research. In our method, we find two possible clusterings: one based research topic and the other based on research type (theoretical vsėxperimental). We found 102 research topics (clusters) in view 1. Example topics are "inference" with top words (statistic, inference, andersson, adg (acyclic directed graph), "optimization" with top words (optimize, programming, sqp (sequential quadratic programming), nonconvexity), "wireless" with top words (antenna, wireless, basestation, stap, mnp (mobile number portability)). We discovered two clusters based on research type in view 2: "theoretical" with top words (methods, mathematical, develop, equation) and "experimental" with top words (experiments, processes, technique, measurement).

The NYTimes dataset [2] consists of 300, 000 news articles from New York Times. We sample 10, 000 instances from the dataset. We represent each article by the frequency of occurrence of each word. This dataset is also very rich. Articles contain topics on politics, economics, business, sports and so on. We use perplexity to evaluate how well our model fits both the NSF and NYTimes data compared to other probabilistic models in Table 3. Results confirm that our NBMC-OFV nonparametric overlapping feature model performed the best compared to the other probabilistic models, CRP-CRP and DP-multinomial.

Table 3: Perplexity Results on NSF and NYTimes

	NSF	NYTimes
NBMC-OFV	**3541**	**8765**
CRP-CRP	3821	9344
DP-Multinomial	5042	12845

6 Conclusion

Standard clustering algorithms output a single clustering solution. However, data can have multiple clustering interpretations. Furthermore, only a subset of features may be important to discover each clustering interpretation or view. In this paper, we introduced a probabilistic nonparametric Bayesian model for this type of richly structured multi-faceted data that can automatically learn the features and clusters in each view for this model simultaneously. Unlike previous nonparametric models which assumes that the features in each view are disjoint, we provided a more flexible model that allows the features to overlap or be shared among views. We model feature relevance to each view using a two parameter Indian Buffet Process and the cluster membership in each view using a Chinese Restaurant Process. We provided an inference approach to learn the latent parameters corresponding to this multiple partitioning problem. Besides learning the latent features and clusters per view, our Bayesian formulation also allows us to automatically learn the number of views and the number of clusters in each view. Our results on synthetic and real-world data show that our method can find high quality multiple alternative clustering views, and outperform competing methods.

Acknowledgments

This work is supported by NSF IIS-0915910.

References

[1] E. Bae and J. Bailey. Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *Proc. of the IEEE Int'l Conf. on Data Mining*, pages 53–62, 2006.

[2] S. D. Bay. The UCI KDD archive, 1999.

[3] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006.

[4] R. Caruana, M. Elhawary, N. Nguyen, and C. Smith. Meta clustering. In *Proc. of the IEEE Int'l Conf. on Data Mining*, pages 107–118, 2006.

[5] Y. Cui, X. Z. Fern, and J. Dy. Non-redundant multi-view clustering via orthogonalization. In *Proc. of the IEEE Int'l Conf. on Data Mining*, pages 133–142, 2007.

[6] X. H. Dang and J. Bailey. Generation of alternative clusterings using the cami approach. In *SIAM Int'l Conf. on Data Mining*, pages 118–129, 2010.

[7] X. H. Dang and J. Bailey. A hierarchical information theoretic technique for the discovery of non linear alternative clusterings. In *Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining (KDD'10)*, pages 573–582, 2010.

[8] S. Dasgupta and V. Ng. Mining clustering dimensions. In *Proc. of the Int'l Conf. on Machine Learning*, pages 263–270, 2010.

[9] J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, August 2004.

[10] Z. Ghahramani, T. Griffiths, and P. Sollich. Bayesian nonparametric latent feature models (with discussion). In *Bayesian Statistics 8*, pages 201–226, Oxford, UK, July 2007.

[11] D. Gondek and T. Hofmann. Non-redundant data clustering. In *Proc. of the IEEE Int'l Conf. on Data Mining*, pages 75–82, 2004.

[12] T. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems 18*, pages 475–482. 2006.

[13] Y. Guan, J. G. Dy, D. Niu, and Z. Ghahramani. Variational inference for nonparametric multiple clustering. In *Workshop on Discovering, Summarizing and Using Multiple Clustering (MultiClust) at the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2010.

[14] K. A. Heller and Z. Ghahramani. A nonparametric baysian approach to modeling overlapping clusters. In *Proc. of the Int'l Conf. on Artificial Intelligence and Statistics, JMLR Proceedings Track 2*, pages 187–194, 2007.

[15] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[16] P. Jain, R. Meka, and I. S. Dhillon. Simultaneous unsupervised learing of disparate clustering. In *Proc. of SIAM Int'l Conf. on Data Mining*, pages 858–869, 2008.

[17] V. Mansinghka, E. Jonas, C. Petschulat, B. Cronin, P. Shafto, and J. Tenenbaum. Cross-categorization: A method for discovering multiple overlapping clusterings. In *Nonparametric Bayes Workshop at NIPS*, 2009.

[18] D. Niu, J. Dy, and M. Jordan. Multiple non-redundant spectral clustering views. In *Proc. of the Int'l Conf. on Machine Learning*, pages 831–838, 2010.

[19] J. Pitman. Combinatorial stochastic processes. Technical report, U.C. Berkeley, Department of Statistics, August 2002.

[20] L. K. M. Poon, N. L. Zhang, T. Chen, and Y. Wang. Variable selection in model-based clustering: To do or to facilitate. In *Proc. of the Int'l Conf. on Machine Learning*, pages 887–894, 2010.

[21] Z. J. Qi and I. Davidson. A principled and flexible framework for finding alternative clusterings. In *ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 717–726, 2009.

[22] J. Reisinger and R. Mooney. Cross-cutting models of lexical semantics. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pages 1405–1415, July 2011.

[23] H. Shan and A. Banerjee. Bayesian co-clustering. In *Proc. of the IEEE Int'l Conf. on Data Mining*, pages 530–539, 2008.

[24] A. Strehl and J. Ghosh. Cluster ensembles–a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 3:583–617, 2002.