Towards Musical Query-by-Semantic-Description using the CAL500 Data Set

Douglas Turnbull[1], Luke Barrington[2], David Torres[1], Gert Lanckriet[2]
Department of Computer Science and Engineering[1]
Department of Electrical and Computer Engineering[2]
University of California, San Diego
La Jolla, CA, USA
[dturnbul@cs, lbarrington@, datorres@cs, gert@ece].ucsd.edu

ABSTRACT

Query-by-semantic-description (QBSD) is a natural paradigm for retrieving content from large databases of music. A major impediment to the development of good QBSD systems for music information retrieval has been the lack of a cleanly-labeled, publicly-available, heterogeneous data set of songs and associated annotations. We have collected the Computer Audition Lab 500-song (CAL500) data set by having humans listen to and annotate songs using a survey designed to capture 'semantic associations' between music and words. We adapt the supervised multi-class labeling (SML) model, which has shown good performance on the task of image retrieval, and use the CAL500 data to learn a model for music retrieval. The model parameters are estimated using the *weighted mixture hierarchies expectation-maximization* algorithm which has been specifically designed to handle real-valued semantic association between words and songs, rather than binary class labels. The output of the SML model, a vector of class-conditional probabilities, can be interpreted as a *semantic multinomial* distribution over a vocabulary. By also representing a semantic query as a *query multinomial* distribution, we can quickly rank order the songs in a database based on the Kullback-Leibler divergence between the query multinomial and each song's semantic multinomial. Qualitative and quantitative results demonstrate that our SML model can both annotate a novel song with meaningful words and retrieve relevant songs given a multi-word, text-based query.

Categories and Subject Descriptors

I.2.m [**Computing Methodologies**]: Artificial Intelligence—*Miscellaneous*; J.5 [**Computer Applications**]: Arts and Humanities—*Music*

General Terms

Algorithms, Design, Experimentation

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
Copyright 2007 ACM 978-1-59593-597-7/07/0007 ...$5.00.

Keywords

Query-by-semantic-description, supervised multi-class classification, content-based music information retrieval

1. INTRODUCTION

An 80-gigabyte personal MP3 player can store about 20,000 songs. Apple iTunes, a popular Internet music store, has a catalogue of over 3.5 million songs[1]. Query-by-semantic-description (QBSD) is a natural paradigm for navigating such large databases of music. For example, one may wish to retrieve songs that "have strong folk roots, feature a banjo, and are uplifting." We propose a content-based QBSD music retrieval system that learns a relationship between acoustic features and words using a heterogeneous data set of songs and associated annotations. Our system directly models the relationship between audio content and words and can be used to search for music using text-based queries composed of one or more words from a large vocabulary.

While QBSD has been studied in computer vision research for both content-based image and video retrieval [1–4], it has received far less attention within the music information retrieval (MIR) community [5]. One major impediment has been the lack of a cleanly-labeled, publicly-available, data set of annotated songs. The first contribution of this paper is such a data set; the *Computer Audition Lab 500-Song* (CAL500) data set[2]. CAL500 consists of 500 popular music songs each of which has been annotated by a minimum of three listeners. A subset of the songs are taken from the publicly-available Magnatunes dataset [6], while the remaining songs can be downloaded from any number of web-based music retailers (such as Rhapsody or Apple iTunes). For all songs, we also provide various features that have been extracted from the audio. Each annotation was collected by playing music for human listeners and asking them to fill out a survey about their auditory experience. The results of the survey were then converted into annotation vectors over a 159-word vocabulary of musically-relevant, semantic concepts.

Our second contribution is showing that the CAL500 data set contains useful information which can be used to train a QBSD music retrieval system. This system is based on a supervised multi-class labeling (SML) probabilistic model [1], which has shown good performance on the task of image retrieval. The SML model estimates a Gaussian Mixture Model (GMM) distribution over an audio feature space con-

[1] Statistics from www.apple.com/itunes, January 2007.

[2] CAL500 can be download at http://cosmal.ucsd.edu/cal.

ditioned on each word in a semantic vocabulary. Parameter estimation is done using the efficient mixture hierarchies expectation-maximization (MH-EM) algorithm. However, for the task of music retrieval, we have to modify this parameter estimation technique to handle real-valued *semantic weights*, as opposed to binary class labels. Weights are more appropriate in the context of music since music is inherently subjective; each individual has their own personal experience when listening to a song. For example, we find that three out of four college students annotate Elvis Presley's "Heartbreak Hotel" as being a 'blues' song while everyone identified B.B. King's "Sweet Little Angel" as being a blues song. By extending the MH-EM algorithm to handle real-valued *semantic weights* between songs and words, we can explicitly model their respective strengths of association.

Our third contribution is showing how the SML model can be used to handle multiple-word queries. (See Table 1 for an illustrative example.) When annotating a novel song, the SML model produces a vector of class-conditional probabilities for each word in a vocabulary. Using Bayes rule, we obtain the set of posterior probabilities that represents a *semantic multinomial* distribution over the vocabulary. If we formulate a user-specified query as a *query multinomial* distribution over the same vocabulary, we can efficiently rank-order all the songs in a large database by calculating the Kullback-Leibler (KL) divergence between the query-multinomial and each song's semantic-multinomial.

The following section discusses how this work fits into the field of music information retrieval and relates to research on semantic retrieval of images and audio. Section 3 formulates the SML model used to solve the related problems of semantic audio annotation and retrieval, explains how to formulate multiple-word semantic queries, and describes how to estimate the parameters of the model using the *weighted* mixture hierarchies algorithm. Section 4 describes the methods for collecting human semantic annotations of music and the creation of the CAL500 data set. Section 5 reports qualitative and quantitative results for annotation and retrieval of music, including retrieval using multiple-word queries. The final section outlines a number of future directions for this research.

2. RELATED WORK

A central goal of the music information retrieval community is to create systems that efficiently store and retrieve songs from large databases of musical content [7]. The most common way to store and retrieve music uses metadata such as the name of the composer or artist, the name of the song or the release date of the album. We consider a more general definition of musical metadata as any non-acoustic representation of a song. This includes genre, song reviews, ratings according to bipolar adjectives (e.g., happy/sad), and purchase sales records. These representations can be used as input to retrieval systems that help users search for music. The drawback of these systems is that they require a novel song to be *manually* annotated before it can be retrieved.

Table 1: Qualitative music retrieval results for our SML model. Results are shown for 1-, 2- and 3-word queries.

Query	Returned Songs
Pop	The Ronettes- Walking in the Rain
	The Go-Gos - Vacation
	Spice Girls - Stop
	Sylvester - You make me feel mighty real
	Boo Radleys - Wake Up Boo!
Female Lead Vocals	Alicia Keys - Fallin'
	Shakira - The One
	Christina Aguilera - Genie in a Bottle
	Junior Murvin - Police and Thieves
	Britney Spears - I'm a Slave 4 U
Tender	Crosby Stills and Nash - Guinnevere
	Jewel - Enter from the East
	Art Tatum - Willow Weep for Me
	John Lennon - Imagine
	Tom Waits - Time
Pop AND Female Lead Vocals	Britney Spears - I'm a Slave 4 U
	Buggles - Video Killed the Radio Star
	Christina Aguilera - Genie in a Bottle
	The Ronettes - Walking in the Rain
	Alicia Keys - Fallin'
Pop AND Tender	5th Dimension - One Less Bell to Answer
	Coldplay - Clocks
	Cat Power - He War
	Chantal Kreviazuk - Surrounded
	Alicia Keys - Fallin'
Female Lead Vocals AND Tender	Jewel - Enter from the East
	Evanescence - My Immortal
	Cowboy Junkies - Postcard Blues
	Everly Brothers - Take a Message to Mary
	Sheryl Crow - I Shall Believe
Pop AND Female Lead Vocals AND Tender	Shakira - The One
	Alicia Keys - Fallin'
	Evanescence - My Immortal
	Chantal Kreviazuk - Surrounded
	Dionne Warwick - Walk on by

Another retrieval approach, *query-by-similarity*, takes an audio-based query and measures the similarity between the query and all of the songs in a database [7]. A limitation of query-by-similarity is that it requires a user to have a useful audio exemplar in order to specify a query. For cases in which no such exemplar is available, researchers have developed *query-by-humming* [8], *-beatboxing* [9], and *-tapping* [10]. However, it can be hard, especially for an untrained user, to emulate the tempo, pitch, melody, and timbre well enough to make these systems viable [8]. A natural alternative is *query-by-semantic-description* (QBSD), describing music with semantically meaningful words. A good deal of research has focused on content-based classification of music by genre [11], emotion [12], and instrumentation [13]. These classification systems effectively 'annotate' music with class labels (e.g., 'blues', 'sad', 'guitar'). The assumption of a predefined taxonomy and the explicit (i.e., binary) labeling of songs into (often mutually exclusive) classes can give rise to a number of problems [14] due to the fact that music is inherently subjective. A more flexible approach [15] considers similarity between songs in a semantic 'anchor space' where each dimension reflects a strength of association to a musical genre.

The QBSD paradigm has been largely influenced by work on the similar task of image annotation. Our system is based on Carneiro et. al.'s SML [1] model, the state-of-the-art in image annotation. Their approach views semantic annotation as one multi-class problem rather than a set of binary one-vs-all problems. A comparative summary of alternative supervised one-vs-all [4] and unsupervised [2, 3] models for image annotation is presented in [1].

Despite interest within the computer vision community, there has been relatively little work on developing text queries for content-based music information retrieval. One exception is the work of Whitman et al. [16–18]. Our approach differs from theirs in a number of ways. First, they use a set of web-documents associated with an *artist* whereas we use multiple *song*-specific annotations for each song in our corpus. Second, they take a one-vs-all approach and learn a discriminative classifier (a support vector machine or a regularized least-squares classifier) for each term in the vocabulary. The disadvantage of the one-vs-all approach is that it results in binary decisions for each class. Our generative multi-class approach outputs a natural ranking of words based on a more interpretable probabilistic model [1].

Other QBSD audition systems [19, 20] have been developed for annotation and retrieval of sound effects. Slaney's Semantic Audio Retrieval system [19, 21] creates separate hierarchical models in the acoustic and text space, and then makes links between the two spaces for either retrieval or annotation. Cano and Koppenberger propose a similar approach based on nearest neighbor classification [20]. The drawback of these non-parametric approaches is that inference requires calculating the similarity between a query and every training example. We propose a parametric approach that requires one model evaluation per semantic concept. In practice, the number of semantic concepts is orders of magnitude smaller than the number of potential training data points, leading to a more scalable solution.

3. SEMANTIC MULTI-CLASS LABELING

This section formalizes the related problems of semantic audio annotation and retrieval as supervised, multi-class labeling tasks where each word in a vocabulary represents a class. We learn a *word-level* (i.e., class-conditional) distribution over an audio feature space for each word in a vocabulary by training only on the songs that are positively associated with that word. This set of word-level distributions is then used to 'annotate' a novel song, resulting in a semantic multinomial distribution. We can then retrieve songs by ranking them according to a their (dis)similarity to a multinomial that is generated from a text-based query. A schematic overview of our model is presented in Figure 1.

3.1 Problem formulation

Consider a vocabulary $\mathcal{V}$ consisting of $|\mathcal{V}|$ unique words. Each 'word' $w_i \in \mathcal{V}$ is a semantic concept such as 'happy', 'blues', 'electric guitar', 'falsetto', etc. The goal in annotation is to find a set $\mathcal{W} = \{w_1, ..., w_A\}$ of A semantically meaningful words that describe a query song s_q. Retrieval involves rank ordering a set of songs $\mathcal{S} = \{s_1, ..., s_R\}$ given a query $\mathcal{W}_q$. It will be convenient to represent the text data describing each song as an *annotation* vector $\mathbf{y} = (y_1, ..., y_{|\mathcal{V}|})$ where $y_i > 0$ if w_i has a positive semantic association with the song and $y_i = 0$ otherwise. The y_i's are called *semantic weights* since they are proportional to the strength of the semantic association between a word and a song. If the semantic weights are mapped to $\{0, 1\}$, then they can be interpreted as class labels. We represent the audio content of a song s as a set $\mathcal{X} = \{\mathbf{x_1}, ..., \mathbf{x_T}\}$ of T real-valued feature vectors, where each vector $\mathbf{x_t}$ represents features extracted from a short segment of the audio and T depends on the length of the song. Our data set $\mathcal{D}$ is a collection of song-annotation pairs $\mathcal{D} = \{(\mathcal{X}_1, \mathbf{y}_1), ..., (\mathcal{X}_D, \mathbf{y}_D)\}$.

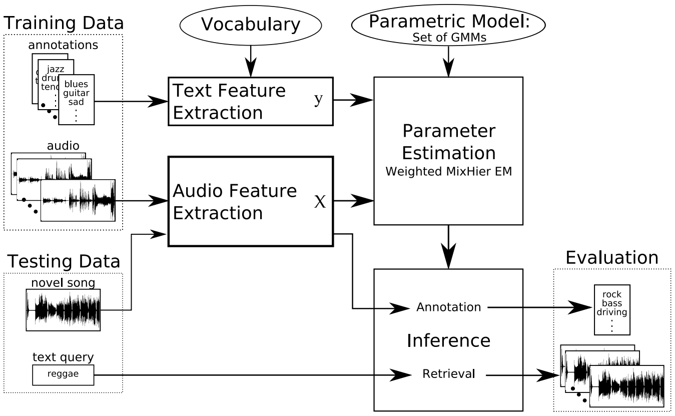

Figure 1: SML model diagram.

3.2 Annotation

Annotation can be thought of as a multi-class, multi-label classification problem in which each word $w_i \in \mathcal{V}$ represents a 'class' and the goal is to 'label' a song with a subset of words. Our approach involves modeling a word-level distribution over audio features, $P(\mathbf{x}|i), i \in \{1, ..., |\mathcal{V}|\}$ for each word $w_i \in \mathcal{V}$. Given a song represented by the set of audio feature vectors $\mathcal{X} = \{\mathbf{x_1}, ..., \mathbf{x_T}\}$, we use Bayes' rule to calculate the posterior probability of each word in the vocabulary, given the audio features:

$$P(i|\mathcal{X}) = \frac{P(\mathcal{X}|i)P(i)}{P(\mathcal{X})}, \tag{1}$$

where $P(i)$ is the prior probability that word w_i will appear in an annotation. If we assume that the feature vectors in $\mathcal{X}$ are conditionally independent given word w_i, then

$$P(i|\mathcal{X}) = \frac{\left[\prod_{t=1}^{T} P(\mathbf{x_t}|i)\right] \cdot P(i)}{P(\mathcal{X})}. \tag{2}$$

Note that this naïve Bayes assumption implies that there is no temporal relationship between audio feature vectors, given word i. While this assumption of conditional independence is unrealistic, attempting to model the temporal interaction between feature vectors may be infeasible due to computational complexity and data sparsity. We assume a uniform word prior, $P(i) = 1/|\mathcal{V}|$, for all $i = 1, .., |\mathcal{V}|$ since, in practice, the T factors in the product will dominate the word prior when calculating the numerator of Equation 2. We estimate the song prior $P(\mathcal{X})$ by $\sum_{v=1}^{|\mathcal{V}|} P(\mathcal{X}|v)P(v)$ and arrive at our final *annotation* equation:

$$P(i|\mathcal{X}) = \frac{\prod_{t=1}^{T} P(\mathbf{x_t}|i)}{\sum_{v=1}^{|\mathcal{V}|} \prod_{t=1}^{T} P(\mathbf{x_t}|v)}. \tag{3}$$

Note that by assuming a uniform word prior, the $1/|\mathcal{V}|$ factor cancels out of the equation.

Using word-level distributions ($P(\mathbf{x}|i), \forall i = 1, ..., |\mathcal{V}|$) and Bayes rule, we use Equation 3 to calculate the parameters of a *semantic multinomial* distribution over the vocabulary. That is, each song in our database is compactly represented as a vector of posterior probabilities $\mathbf{p} = \{p_1, ..., p_{|\mathcal{V}|}\}$ in a 'semantic space', where $p_i = P(i|\mathcal{X})$ and $\sum_i p_i = 1$. To annotate a song with the A best words, we use the word-level models to generate the song's semantic distribution and then choose the A peaks of the multinomial distribution, i.e., the A words with maximum posterior probability.

3.3 Retrieval

For retrieval, we first annotate our database by calculating a semantic multinomial for each song. When a user enters a query, we construct a 'query multinomial' distribution, parameterized by the vector $\mathbf{q} = \{q_1, ..., q_{|\mathcal{V}|}\}$, by assigning $q_i = C$ if word w_i is in the text-based query, and $q_i = \epsilon$ where $1 \gg \epsilon > 0$ otherwise. We then normalize $\mathbf{q}$, making it's elements sum to unity so that it correctly parameterizes

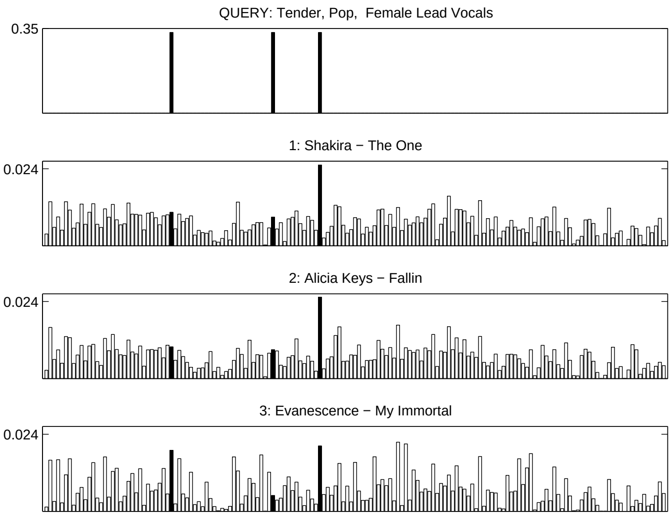

Figure 2: Multinomial distributions over the 159-word vocabulary. The top distribution represents the *query* multinomial for the three-word query presented in Table 1. The next three distribution are the *semantic* multinomials for top three retrieved songs.

a multinomial distribution. In practice, we set the $C = 1$ and $\epsilon = 10^{-6}$. However, we should stress C need not be a constant, rather it could be a function of the query string. For example, we may want to give more weight to words that appear earlier in the query string as is commonly done by Internet search engines for retrieving web documents. Examples of a semantic query multinomial and the retrieved song multinomials are given in Figure 2.

Once we have a query multinomial, we rank all the songs in our database by the Kullback-Leibler (KL) divergence between the query multinomial $\mathbf{q}$ and each semantic multinomial. The KL divergence between $\mathbf{q}$ and a semantic multinomial $\mathbf{p}$ is given by [22]:

$$KL(\mathbf{q}||\mathbf{p}) = \sum_{i=1}^{|\mathcal{V}|} q_i \log \frac{q_i}{p_i}, \qquad (4)$$

where the query distribution serves as the 'true' distribution. Since $q_i = \epsilon$ is effectively zero for all words that do not appear in the query string, a one-word query w_i reduces to ranking by the i-th parameter of the semantic multinomials. For a multiple-word query, we only need to calculate one term in Equation 4 per word in the query. This leads to a very efficient and scalable approach for music retrieval in which the majority of the computation involves sorting the D scalar KL divergences between the query multinomial and each song in the database.

3.4 Parameter Estimation

For each word $w_i \in \mathcal{V}$, we learn the parameters of the word-level (i.e., class-conditional) distribution, $P(\mathbf{x}|i)$, using the audio features from all songs that have a positive association with word w_i. Each distribution is modeled with an R-component Gaussian Mixture Model (GMM) distribution parameterized by $\{\pi_r, \mu_r, \Sigma_r\}$ for $r = 1, ..., R$. The word-level distribution for word w_i is given by:

$$P(\mathbf{x}|i) = \sum_{r=1}^{R} \pi_r \mathcal{N}(\mathbf{x}|\mu_r, \Sigma_r),$$

where $\sum \pi_r = 1$ are the mixture weights and $\mathcal{N}(\cdot|\mu, \Sigma)$ is a multivariate Gaussian distribution with mean μ and covariance matrix Σ. In this work, we consider only diagonal covariance matrices since using full covariance matrices can cause models to overfit the training data while scalar covariances do not provide adequate generalization. The resulting set of $|\mathcal{V}|$ distributions each have $\mathcal{O}(R \cdot F)$ parameters, where F is the dimension of feature vector $\mathbf{x}$.

Carneiro et al. [1] consider three parameter estimation techniques for learning a SML model: direct estimation, model averaging estimation, and mixture hierarchies estimation. The techniques are similar in that, for each word-level distribution, they use the expectation-maximization (EM) algorithm for fitting a mixture of Gaussians to training data. They differ in how they break down the problem of parameter estimation into subproblems and then merge these results to produce a final density estimate. Carneiro et al. found that mixture hierarchies estimation was not only the most scalable technique, but it also resulted in the density estimates that produced the best image annotation and retrieval results. We confirmed these finding for music annotation and retrieval during some initial experiments (not reported here).

The formulation in [1] assumes that the semantic information about images is represented by binary annotation vectors. This formulation is natural for images where the majority of words are associated with relatively 'objective' semantic concepts such as 'bear', 'building', and 'sunset'. Music is more 'subjective' in that two listeners may not always agree that a song is representative of a certain genre or generates the same emotional response. Even seemingly objective concepts, such as those related to instrumentation, may result in differences of opinion, when, for example, a digital synthesizer is used to emulate a traditional instrument. To this end, we believe that a real-valued annotation vector of associated 'strengths of agreement' is a more natural semantic representation. We now extend the mixture hierarchies estimation to handle real-value semantic weights, resulting in the *weighted mixture hierarchies algorithm.*

Consider the set of $|\mathcal{D}|$ song-level GMM distributions (each with K mixture components) that are estimated using the feature vectors that are extracted from each song. We can estimate a word-level distribution with R components using an extension of the EM algorithm:

E-step: Compute the responsibilities of each word-level component, r, to a song-level component, k from song d

$$h^r_{(d),k} = \frac{[\mathbf{y}_d]_i \left[\mathcal{N}(\mu_k^{(d)}|\mu_r, \mathbf{\Sigma}_r) e^{-\frac{1}{2}\mathrm{Tr}\{(\mathbf{\Sigma}_r)^{-1}\mathbf{\Sigma}_k^{(d)}\}}\right]^{\pi_k^{(d)} N} \pi_r}{\sum_l \left[\mathcal{N}(\mu_k^{(d)}|\mu_l, \mathbf{\Sigma}_l) e^{-\frac{1}{2}\mathrm{Tr}\{(\mathbf{\Sigma}_l)^{-1}\mathbf{\Sigma}_k^{(d)}\}}\right]^{\pi_k^{(d)} N} \pi_l},$$

where N is a user defined parameter. In practice, we set $N = K$ so that $E[\pi_k^{(d)} N] = 1$.

M-step: Update the word-level distribution parameters

$$\begin{aligned}
\pi_r^{new} &= \frac{\sum_{(d),k} h^r_{(d),k}}{W \cdot K}, \text{ where } W = \sum_{d=1}^{|\mathcal{D}|} [\mathbf{y}_d]_i \\
\mu_r^{new} &= \sum_{(d),k} z^r_{(d),k} \mu_k^{(d)}, \quad \text{where } z^r_{(d),k} = \frac{h^r_{(d),k} \pi_k^{(d)}}{\sum_{(d),k} h^r_{(d),k} \pi_k^{(d)}}, \\
\mathbf{\Sigma}_r^{new} &= \sum_{(d),k} z^r_{(d),k} \left[\mathbf{\Sigma}_k^{(d)} + (\mu_k^{(d)} - \mu_t)(\mu_k^{(d)} - \mu_t)^T \right].
\end{aligned}$$

From a generative perspective, a song-level distribution is generated by sampling *mixture components* from the word-level distribution. The observed audio features are then samples from the song-level distribution. Note that the number of parameters for the word-level distribution is the same as the number of parameters resulting from direct estimation[3]. We have essentially replaced one computationally expensive (and often impossible) run of the standard EM algorithm with at most $|\mathcal{D}|$ computationally inexpensive runs and one run of the mixture hierarchies EM. In practice, mixture hierarchies EM requires about the same computation time as one run of standard EM for a song-level model. However, the main benefit of using the MH-EM algorithm is that it provides a form of regularization by first representing each song by a 'smooth' distribution, rather then a finite set of point estimates, before learning a word-level distribution.

Our formulation differs from that derived in [23] in that the responsibility, $h^r_{(d),k}$, is multiplied by the semantic weight $[\mathbf{y}_d]_i$ between word w_i and song s_d. This *weighted mixture hierarchies algorithm* reduces to the standard formulation when the semantic weights are either 0 or 1. The semantic weights can be interpreted as a relative measure of importance of each training data point. That is, if one data point has a weight of 2 and all others have a weight of 1, it is as though the first data point actually appeared twice in the training set.

4. THE CAL500 MUSIC DATA SET

Perhaps the easiest way to collect semantic information about a song is to mine text from web pages related to the song, album or artist [18, 24]. Whitman et al. collect a large number webpages related to the artist when attempting to annotate individual songs [18]. One drawback of this methodology is that it produces the same training annotation vector for all songs by a single artist. This is a problem for many artists, such as Paul Simon and Madonna, who have produced an acoustically diverse set of songs over the course of their careers. In previous work, we take a more song-specific approach by text-mining song reviews written by expert music critics [24]. The drawback of this technique is that critics do not explicitly make decisions about the relevance of given word when writing about songs and/or artists. In both works, it is evident that the semantic labels are a noisy version of an already problematic 'subjective ground truth.' To address the shortcomings of noisy semantic data mined from text-documents, we attempt to collect a 'clean' set of semantic labels by asking human listeners to explicitly label songs with acoustically-relevant words. In an attempt to overcome the problems arising from the inherent subjectivity involved in music annotation, we require that each song be annotated by multiple listeners.

4.1 Semantic Representation

Our goal is to collect training data from human listeners that reflect the strength of association between words and songs. We designed a survey that listeners used to evaluate songs in our music corpus. The corpus is a selection of 500 'western popular' songs composed within the last 50 years by 500 different artists, chosen to cover a large amount of acoustic variation while still representing some familiar genres and popular artists.

In the survey, we considered 135 musically-relevant concepts spanning six semantic categories: 29 instruments were annotated as present in the song or not; 22 vocal characteristics were annotated as relevant to the singer or not; 36 genres, a subset of the Codaich genre list [25], were annotated as relevant to the song or not; 18 emotions, found by Skowronek et al. [26] to be both important and easy to identify, were rated on a scale from one to three (e.g., "not happy", "neutral", "happy"); 15 song concepts describing the acoustic qualities of the song, artist and recording (e.g., tempo, energy, sound quality); and 15 usage terms from [27], (e.g., "I would listen to this song while *driving, sleeping, etc.*").

We paid 66 undergraduate students to annotate the CAL500 corpus with semantic concepts from our vocabulary. Participants were rewarded \$10 for a one hour annotation block spent listening to MP3-encoded music through headphones in a university computer laboratory. The annotation interface was an HTML form loaded in a web browser requiring participants to simply click on check boxes and radio buttons. The form was not presented during the first 30 seconds of playback to encourage undistracted listening. Listeners could advance and rewind the music and the song would repeat until all semantic categories were annotated. Each annotation took about 5 minutes and most participants reported that the listening and annotation experience was enjoyable. We collected at least 3 semantic annotations for each of the 500 songs in our music corpus and a total of 1708 annotations.

We expanded the set of 135 survey concepts to a set of 237 'words' by mapping all bipolar concepts to two individual words. For example, 'Energy Level' gets mapped to 'Low Energy' and 'High Energy'. We are left with a collection of human annotations where each annotation is a vector of numbers expressing the response of a human listener to a semantic keyword. For each word the annotator has supplied a response of +1 or -1 if the user believes the song is or is not indicative of the word, or 0 if unsure. We take all the human annotations for each song and compact them to a single annotation vector by observing the level of agreement over all annotators. Our final semantic weights $\mathbf{y}$ are

$$[\mathbf{y}]_i = \max\left(0, \left[\frac{\#(\text{Positive Votes}) - \#(\text{Negatives Votes})}{\#(\text{Annotations})}\right]_i\right).$$

For example, for a given song, if four listeners have labeled a concept w_i with +1, +1, 0, -1, then $[\mathbf{y}]_i = 1/4$.

For evaluation purposes, we also create 'ground truth' binary annotation vectors. We generate binary vectors by labeling a song with a word if a minimum of two people express an opinion and there is at least 80% agreement between all listeners. We prune all concepts that are represented by fewer than eight songs. This reduces our vocabulary from 237 to 159 words.

[3] Direct estimation involves using the union of sets of audio feature vectors from relevant training songs to estimate a word-level GMM using the standard EM algorithm.

4.2 Musical Representation

We represent the audio with a time series of *delta cepstrum* feature vectors. A time series of Mel-frequency cepstral coefficient (MFCC) [28] vectors is extracted by sliding a half-overlapping short-time window (∼12 msec) over the song's digital audio file. A delta cepstrum vector is calculated by appending the instantaneous first and second derivatives of each MFCC to the vector of MFCCs. We use the first 13 MFCCs resulting in about 10,000 39-dimensional feature vectors per minute of audio content. The reader should note that the SML model (a set of GMMs) ignores the temporal dependencies between adjacent feature vector within the time series. We find that randomly sub-sampling the set of delta cepstrum feature vectors so that each song is represented by 10,000 feature vectors reduces the computation time for parameter estimation and inference without sacrificing much overall performance.

5. MODEL EVALUATION

In this section, we qualitatively and quantitatively evaluate our SML model for music annotation and retrieval. To our knowledge, there has been little previous work on these problems [16–18, 24]. It is hard to compare our performance against the work of Whitman et al. since their work focuses on vocabulary selection while the results in [24] are calculated using a different model on a different data set of words and songs.

Instead, we evaluate our system against two baselines: a 'random' baseline and a 'human' baseline. The random baseline is a system that samples words (without replacement) from a multinomial distribution parameterized by the word prior distribution, $P(i)$ for $i = 1, ..., |\mathcal{V}|$, estimated using the observed word counts from the training set. Intuitively, this prior stochastically generates annotations from a pool of the words used most frequently in the training set.

We can also estimate the performance of a human on the annotation task. This is done by holding out a single human annotation from each of the 142 songs in the CAL500 data set that had more than 3 annotations. To evaluate performance, we compare this human's semantic description of a song to the "ground truth" labels obtained from the remaining annotations for that song. We run a large number of simulations by randomly holding out different human annotations.

5.1 Annotation Performance

Given an SML model, we can effectively 'annotate' a novel song by estimating a semantic multinomial using Equation 3. Placing the most likely words into a natural language context demonstrates how our annotation system can be used to generate 'automatic music reviews' as illustrated in Table 2. It should be noted that in order to create these reviews, we made use of the fact that the words in our vocabulary can loosely be organized into semantic categories such as genre, instrumentation, vocal characteristic, emotions, and song usages.

Quantitative annotation performance is measured using mean *per-word* precision and recall [1, 2]. First, we annotate each test set song with a fixed number of words (e.g., A = 8) from our vocabulary of 159 words. For each word w in our vocabulary, $|w_H|$ is the number of songs that have word w in the "ground truth" annotation, $|w_A|$ is the number of songs that our model annotates with word w, and $|w_C|$ is the number of "correct" words that have been used both in the ground truth annotation and by the model. Per-word recall is $|w_C|/|w_H|$ and per-word precision is $|w_C|/|w_A|$. While trivial models can easily maximize one of these measures (e.g., by labeling all songs with a certain word or, instead, none of them), achieving excellent precision and recall simultaneously requires a truly valid model.

Mean per-word recall and precision is the average of these ratios over all the words in our vocabulary. It should be noted that these metrics range between 0.0 and 1.0, but one may be upper bounded by a value less than 1.0 if either the number of words that appear in the corpus is greater or lesser than the number of words that are output by our system. For example, if our system outputs 4000 words to annotate the 500 test songs for which the ground truth contains 6430 words, mean recall will be upper-bounded by a value less than one. The exact upper bounds (denoted "UpperBnd" in Table 3) for recall and precision depend on the relative frequencies of each word in the vocabulary and can be calculated empirically using a simulation where the model output exactly matches the ground truth.

It may seem more intuitive to use *per-song* precision and recall, rather than the per-word metrics. However, per-song metrics can lead to artificially good results if a system is good at predicting the few common words relevant to a large group of songs (e.g., "rock") and bad at predicting the many rare words in the vocabulary. Our goal is to find a system that is good at predicting all the words in our vocabulary. In practice, using the 8 best words to annotate each song, our SML model outputs 143 of the 159 words in the vocabulary at least once.

Table 3 presents quantitative results for music annotation. The results are generated using ten-fold cross validation. That is, we partition the CAL500 data set into ten sets of fifty songs and estimate the semantic multinomials for the songs in each set with an SML model that has been trained using the songs in the other nine sets. We then calculate the per-word precision and recall for each word and average over the vocabulary.

The quantitative results demonstrate that the SML model significantly outperforms the random baselines and is comparable to the human baseline. This does not mean that our model is approaching a 'glass ceiling', but rather, it illustrates the point that music annotation is a subjective task since an individual can produce an annotation that very different from the annotation derived from a population of listeners. This highlights the need for incorporating semantic weights when designing an automatic music annotation and retrieval system.

5.2 Retrieval Performance

We evaluate every one-, two-, and three-word text-based query drawn from our vocabulary of 159 words. First, we create query multinomials for each query string as described in Section 3.3. For each query multinomial, we rank order the 500 songs by the KL divergence between the query multinomial and the semantic multinomials generated during annotation. (As described in the previous subsection, the semantic multinomials are generated from a test set using cross-validation and can be considered representative of a novel test song.)

Table 1 shows the top 5 songs retrieved for a number of text-based queries. In addition to being (mostly) accurate, the reader should note that queries, such as 'Tender' and 'Female Vocals', return songs that span different genres and are composed using different instruments. As more words

Table 2: Automatically generated music reviews. Words in bold are output by our system.

White Stripes - Hotel Yorba
This is a **brit popp**y, **alternative** song that is **not calming** and **not mellow**. It features **male vocal**, **drum set**, **distorted electric guitar**, a nice **distorted electric guitar** solo, and **screaming**, **strong** vocals. It is a song **with high energy** and **with an electric texture** that you might like listen to while **driving**.

Miles Davis - Blue in Green
This is a **jazz**y, **folk** song that is **calming** and **not arousing**. It features **acoustic guitar**, **saxophone**, **piano**, a nice **piano** solo, and **emotional**, **low-pitched** vocals. It is a song **slow tempo** and **with low energy** that you might like listen to while **reading**.

Dr. Dre (feat. Snoop Dogg) - Nuthin' but a 'G' thang
This is a **dance pop**py, **hip-hop** song that is **arousing** and **exciting**. It features **drum machine**, **backing vocals**, **male vocal**, a nice **acoustic guitar solo**, and **rapping**, **strong** vocals. It is a song that is **very danceable** and with a **heavy beat** that you might like listen to while **at a party**.

Depeche Mode - World In My Eyes
This is a **funk**y, **dance pop** song that is **arousing** not **not tender**. It features **male vocal**, **synthesizer**, **drum machine**, a nice **male vocal** solo, and **altered with effects**, **strong** vocals. It is a song **with a synthesized texture** and **that was recorded in studio** that you might like listen to while **at a party**.

Table 3: Music annotation results: SML model learned from $K = 8$ component song-level GMMs, and composed of $R = 16$ component word-level GMMs. Each CAL500 song is annotated with $A = 8$ words from a vocabulary of $|\mathcal{V}|$ =159 words.

Model	Precision	Recall
Random	0.169	0.052
Human	0.342	0.110
UpperBnd	1.000	0.302
SML	0.312	0.142

are added to the query string, note that the songs returned are representative of all the semantic concepts in each of the queries.

By considering the "ground truth" target for a multiple-word query as all the songs that are associated with *all* the words in the query string, we can quantitatively evaluate retrieval performance. We calculate the mean average precision (MeanAP) [2] and the mean area under the receiver operating characteristic (ROC) curve (MeanAROC) for each query for which there is a minimum of 8 songs present in the ground truth. Average precision is found by moving down our ranked list of test songs and averaging the precisions at every point where we correctly identify a new song. An ROC curve is a plot of the true positive rate as a function of the false positive rate as we move down this ranked list of songs. The area under the ROC curve (AROC) is found by integrating the ROC curve and is upper bounded by 1.0. Random guessing in a retrieval task results in an AROC of 0.5. Comparison to human performance is not possible for retrieval since an individual's annotations do not provide a ranking over all retrievable songs. Columns 3 and 4 of Table 4 show MeanAP and MeanAROC found by averaging each metric over all testable one, two and three word queries. Column 1 of Table 4 indicates the proportion of all possible multiple-word queries that actually have 8 or more songs in the ground truth against which we test our model's performance.

Table 4: Music retrieval results for 1-, 2-, and 3-word queries. See Table 3 for SML model parameters.

Query Length	Model	MeanAP	MeanAROC
1-word	Random	0.173	0.500
(159/159)	SML	**0.307**	**0.705**
2-words	Random	0.076	0.500
(4,658/15,225)	SML	**0.164**	**0.723**
3-words	Random	0.051	0.500
(50,471/1,756,124)	SML	**0.120**	**0.730**

As with the annotation results, we see that our model significantly outperforms the random baseline. As expected, MeanAP decreases for multiple-word queries due to the increasingly sparse ground truth annotations (since there are fewer relevant songs per query). However, an interesting finding is that the MeanAROC actually increases with additional query terms, indicating that our model can successfully integrate information from multiple words.

5.3 Comments

The qualitative annotation and retrieval results in Tables 1 and 2 indicate that our system produces sensible semantic annotations of a song and retrieves relevant songs, given a text-based query. Using the explicitly annotated music data set described in Section 4, we demonstrate a significant improvement in performance over similar models trained using weakly-labeled text data mined from the web [24] (e.g., music retrieval MeanAROC increases from 0.61 to 0.71). The CAL500 data set, automatic annotations of all songs, and retrieval results for each word, can be found at the UCSD Computer Audition Lab website (http://cosmal.ucsd.edu/cal).

Our results are comparable to state-of-the-art content-based image annotation systems [1] which report mean per-word recall and precision scores of about 0.25. However, the relative objectivity of the tasks in the two domains as well as the vocabulary, the quality of annotations, the features, and the amount of data differ greatly between our audio annotation system and existing image annotation systems making any direct comparison dubious at best.

6. DISCUSSION AND FUTURE WORK

We have collected the CAL500 data set of cleanly annotated songs and offer it to researchers who wish to work on semantic annotation and retrieval of music. This data set may be considered small in comparison to standard data set that have been developed by the text-mining and computer vision communities. However, by developing a useful and efficient parameter estimation algorithm (weighted mixture hierarchies EM), we have shown how the CAL500 data set can be used to train a query-by-semantic-description system for music information retrieval that significantly outperforms the system presented in [24]. While direct comparison is impossible since different vocabularies and music corpora are used, both qualitative and quantitative results suggest that end user experience has been greatly improved. We have also shown that compactly representing a song as *semantic multinomial* distribution over a vocabulary is useful for both annotation and retrieval. More specifically, by representing a multi-word query string as a multinomial distribution, the KL divergence between this query multinomial and the semantic multinomals provides a natural and computationally inexpensive way to rank order songs in a database. The semantic multinomial representation is also useful for related music information tasks such as 'query-by-semantic-example' [15, 29].

All qualitative and quantitative results reported are based on one SML model ($K = 8, R = 16$) trained using the weighted mixture hierarchies EM algorithm. Though not reported, we have conducted extensive parameter testing by varying the number of song-level mixture components (K), varying the number of word-level mixture components (R), exploring other parameter estimation techniques (direct estimation, model averaging, standard mixture hierarchies EM [1]), and using alternative audio features (such as dynamic MFCCs [11]). Some of these models show comparable performance for some evaluation metrics. For example, dynamic MFCC features tend to produce better annotations, but worse retrieval results than those based on delta cepstrum features reported here.

In all cases, it should be noted that we use a very basic frame-based audio feature representation. We can imagine using alternative representations, such as those that attempt model higher-level notions of harmony, rhythm, melody, and timbre. Similarly, our probabilistic SML model (a set of GMMs) is one of many models that have been developed for image annotation [2, 3]. Future work may involve adapting other models for the task of audio annotation and retrieval. In addition, one drawback of our current model is that, by using GMMs, we ignore all medium-term (> 1 second) and long-term (entire song) information that can be extracted from a song. Future research will involve exploring models, such as hidden Markov models, that explicitly model the longer-term temporal aspects of music.

Lastly, we are currently exploring a more scalable data collection approach that involves using web-based games[4] to collect semantic information about music [30]. This technique, referred to as *human computation*, has been successful used to collect semantic information about images [31].

Acknowledgements

We would like to thank A. Chan, A. Cont, G. W. Cottrell, S. Dubnov, C. Elkan, O. Lang, L. Saul, and N. Vasconcelos for their helpful comments.This work is supported by NSF IGERT DGE-0333451 and NSF grant DMS-MSPA 062540922.

7. REFERENCES

[1] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE PAMI*, 29(3):394–410, 2007.

[2] S. L. Feng, R. Manmatha, and Victor Lavrenko. Multiple bernoulli relevance models for image and video annotation. *IEEE CVPR*, 2004.

[3] D. M. Blei and M. I. Jordan. Modeling annotated data. *ACM SIGIR*, 2003.

[4] D. Forsyth and M. Fleck. Body plans. *IEEE CVPR*, 1997.

[5] International conferences of music information retrieval. http://www.ismir.net/.

[6] MIREX 2005. Music information retrieval evaluation exchange. http://www.music-ir.org/mirex2005.

[7] M. Goto and K. Hirata. Recent studies on music information processing. *Acoustical Science and Technology*, 25(4):419–425, 2004.

[8] R. B. Dannenberg and N. Hu. Understanding search performance in query-by-humming systems. *ISMIR*, 2004.

[9] A. Kapur, M. Benning, and G. Tzanetakis. Query by beatboxing: Music information retrieval for the dj. *ISMIR*, 2004.

[10] G. Eisenberg, J.M. Batke, and T. Sikora. Beatbank - an mpeg-7 compliant query by tapping system. *Audio Engineering Society Convention*, 2004.

[11] M. F. McKinney and J. Breebaart. Features for audio and music classification. *ISMIR*, 2003.

[12] T. Li and G. Tzanetakis. Factors in automatic musical genre classification of audio signals. *IEEE WASPAA*, 2003.

[13] S. Essid, G. Richard, and B. David. Inferring efficient hierarchical taxonomies for music information retrieval tasks: Application to musical instruments. *ISMIR*, 2005.

[14] F. Pachet and D. Cazaly. A taxonomy of musical genres. *RIAO*, 2000.

[15] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A large-scale evalutation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 2004.

[16] B. Whitman. *Learning the meaning of music.* PhD thesis, Massachusetts Institute of Technology, 2005.

[17] B. Whitman and D. Ellis. Automatic record reviews. *ISMIR*, 2004.

[18] B. Whitman and R. Rifkin. Musical query-by-description as a multiclass learning problem. *IEEE Workshop on Multimedia Signal Processing*, 2002.

[19] M. Slaney. Semantic-audio retrieval. *IEEE ICASSP*, 2002.

[20] P. Cano and M. Koppenberger. Automatic sound annotation. In *IEEE workshop on Machine Learning for Signal Processing*, 2004.

[21] M. Slaney. Mixtures of probability experts for audio retrieval and indexing. *IEEE Multimedia and Expo*, 2002.

[22] T. Cover and J. Thomas. *Elements of Information Theory.* Wiley-Interscience, 1991.

[23] N. Vasconcelos. Image indexing with mixture hierarchies. *IEEE CVPR*, pages 3–10, 2001.

[24] D. Turnbull, L. Barrington, and G. Lanckriet. Modelling music and words using a multi-class naïve bayes approach. *ISMIR*, 2006.

[25] C. McKay, D. McEnnis, and I. Fujinaga. A large publicly accessible prototype audio database for music research. *ISMIR*, 2006.

[26] J. Skowronek, M. McKinney, and S. ven de Par. Ground-truth for automatic music mood classification. *ISMIR*, 2006.

[27] Xiao Hu, J. S. Downie, and A. F. Ehmann. Exploiting recommended usage metadata: Exploratory analyses. *ISMIR*, 2006.

[28] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[29] L. Barrington, A. Chan, D. Turnbull, and G. Lanckriet. Audio information retrieval using semantic similarity. In *IEEE ICASSP*, 2007.

[30] D. Turnbull, R. Liu, L. Barrington, D. Torres, and G Lanckriet. Using games to collect semantic information about music. Technical report, 2007.

[31] L. von Ahn. Games with a purpose. *IEEE Computer Magazine*, 39(6):92–94, 2006.

[4] Listen Game: www.listengame.org