Semantic Annotation and Retrieval of Music and Sound Effects

Douglas Turnbull, *Student Member, IEEE*, Luke Barrington, David Torres, and Gert Lanckriet

***Abstract*—We present a computer audition system that can both *annotate* novel audio tracks with semantically meaningful words and *retrieve* relevant tracks from a database of unlabeled audio content given a text-based query. We consider the related tasks of content-based audio annotation and retrieval as one supervised multiclass, multilabel problem in which we model the joint probability of acoustic features and words. We collect a data set of 1700 human-generated annotations that describe 500 Western popular music tracks. For each word in a vocabulary, we use this data to train a Gaussian mixture model (GMM) over an audio feature space. We estimate the parameters of the model using the *weighted mixture hierarchies expectation maximization* algorithm. This algorithm is more scalable to large data sets and produces better density estimates than standard parameter estimation techniques. The quality of the music annotations produced by our system is comparable with the performance of humans on the same task. Our "query-by-text" system can retrieve appropriate songs for a large number of musically relevant words. We also show that our audition system is general by learning a model that can annotate and retrieve sound effects.**

***Index Terms*—Audio annotation and retrieval, music information retrieval, semantic music analysis.**

I. Introduction

MUSIC is a form of communication that can represent human emotions, personal style, geographic origins, spiritual foundations, social conditions, and other aspects of humanity. Listeners naturally use words in an attempt to describe what they hear even though two listeners may use drastically different words when describing the same piece of music. However, words related to some aspects of the audio content, such as instrumentation and genre, may be largely agreed upon by a majority of listeners. This agreement suggests that it is possible to create a computer audition system that can learn the relationship between audio content and words. In this paper, we describe such a system and show that it can both *annotate* novel audio content with semantically meaningful words and *retrieve* relevant audio tracks from a database of unannotated tracks given a text-based query.

Manuscript received December 16, 2006; revised November 8, 2007. This work was supported by the National Science Foundation (NSF) under Grants IGERT DGE-0333451 and DMS-MSPA 062540922. Some of the material presented in this paper was presented at SIGIR'07 and ISMIR'06. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mark Sandler.

D. Turnbull and D. Torres are with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: dturnbul@cs.ucsd.edu; datorres@cs.ucsd.edu).

L. Barrington and G. Lanckriet are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: lbarrington@ucsd.edu; gert@ece.ucsd.edu).

Digital Object Identifier 10.1109/TASL.2007.913750

TABLE I
AUTOMATIC ANNOTATIONS GENERATED USING THE AUDIO CONTENT. WORDS IN **BOLD** ARE OUTPUT BY OUR SYSTEM AND THEN PLACED INTO A MANUALLY CONSTRUCTED NATURAL LANGUAGE TEMPLATE

Frank Sinatra - Fly me to the moon This is a **jazz**y, **singer / songwriter** song that is **calming** and **sad**. It features **acoustic guitar**, **piano**, **saxophone**, a nice **male vocal solo**, and **emotional**, **high-pitched vocals**. It is a song with a **light beat** and a **slow tempo** that you might like listen to while **hanging with friends**.
Creedence Clearwater Revival - Travelin' Band This is a **rock**in', **classic rock** song that is **arousing** and **powerful**. It features **clean electric guitar**, **backing vocals**, **distorted electric guitar**, a nice **distorted electric guitar solo**, and **strong, duet vocals**. It is a song with a **catchy** feel and is **very danceable** that you might like listen to while **driving**.
New Order - Blue Monday This is a **pop**py, **electronica** song that is **not emotional** and **not tender**. It features **sequencer**, **drum machine**, **synthesizer**, a nice **male vocal solo**, and **altered with effects**, **high-pitched** vocals. It is a song with a **synthesized texture** and with **positive feelings** that you might like listen to while **at a party**.
Dr. Dre (feat. Snoop Dogg) - Nuthin' but a 'G' thang This is **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**.

We view the related tasks of semantic annotation and retrieval of audio as one supervised multiclass, multilabel learning problem. We learn a joint probabilistic model of audio content and words using an annotated corpus of audio tracks. Each track is represented as a set of feature vectors that is extracted by passing a short-time window over the audio signal. The text description of a track is represented by an *annotation vector*, a vector of weights where each element indicates how strongly a semantic concept (i.e., a word) applies to the audio track.

Our probabilistic model is one *word-level* distribution over the audio feature space for each word in our vocabulary. Each distribution is modeled using a multivariate Gaussian mixture model (GMM). The parameters of a word-level GMM are estimated using audio content from a set of training tracks that are positively associated with the word. Using this model, we can infer likely semantic annotations given a novel track and can use a text-based query to rank-order a set of unannotated tracks. For illustrative purposes, Table I displays annotations of songs produced by our system. Placing the most likely words from specific semantic categories into a natural language context demonstrates how our annotation system can be used to generate automatic music reviews. Table II shows some of the top songs that the system retrieves from our data set, given various text-based queries.

Our model is based on the supervised multiclass labeling (SML) model that has been recently proposed for the task of

TABLE II
MUSIC RETRIEVAL EXAMPLES. EACH WORD (IN QUOTES) REPRESENTS A TEXT-BASED QUERY TAKEN FROM A SEMANTIC CATEGORY (IN PARENTHESIS)

Query	Top 5 Retrieved Songs
'Tender' (Emotion)	Chet Baker - These foolish things Saros - Prelude Norah Jones - Don't know why Art Tatum - Willow weep for me Crosby Stills and Nash - Guinnevere
'Hip Hop' (Genre)	Nelly - Country Grammar C+C Music Factory - Gonna make you sweat Dr. Dre (feat. Snoop Dogg) - Nuthin' but a 'G' thang 2Pac - Trapped Busta Rhymes - Woo hah got you all in check
'Sequencer' (Instrument)	Belief Systems - Skunk werks New Order - Blue Monday Introspekt - TBD Propellerheads - Take California Depeche Mode - World in my eyes
'Exercising' (Usage)	Red Hot Chili Peppers - Give it away Busta Rhymes - Woo hah got you all in check Chic - Le freak Jimi Hendrix - Highway chile Curtis Mayfield - Move on up
'Screaming' (Vocals)	Metallica - One Jackalopes - Rotgut Utopia Banished - By mourning Bomb the Bass - Bug powder dust Nova Express - I'm alive

image annotation and retrieval by Carneiro and Vasconcelos [1]. They show that their *mixture hierarchies* expectation-maximization (EM) algorithm [2], used for estimating the parameters of the word-level GMMs, is superior to traditional parameter estimation techniques in terms of computational scalability and annotation performance. We confirm these findings for audio data and extend this estimation technique to handle real-valued (rather than binary) class labels. Real-valued class labels are useful in the context of music since the strength of association between a word and a song is not always all or nothing. For example, based on a study described below, 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. Our *weighted* mixture hierarchies EM algorithm explicitly models these respective strengths of associations when estimating the parameters of a GMM.

The semantic annotations used to train our system come from a user study in which we asked participants to annotate songs using a standard survey. The survey contained questions related to different semantic categories, such as emotional content, genre, instrumentation, and vocal characterizations. The music data used is a set of 500 "Western popular" songs from 500 unique artists, each of which was reviewed by a minimum of three individuals. Based on the results of this study, we construct a vocabulary of 174 "musically relevant" semantic keywords. The resulting annotated music corpus, referred to as the *Computer Audition Lab 500* (CAL500) data set, is publicly available[1] and may be used as a common test set for future research involving semantic music annotation and retrieval.

Though the focus of this work is on music, our system can be used to model other classes of audio data and is scalable in terms of both vocabulary size and training set size. We demonstrate that our system can successfully annotate and retrieve sound effects using a corpus of 1305 tracks and a vocabulary containing 348 words.

The following section discusses how this work fits into the field of music information retrieval (MIR) and relates to research on semantic image annotation and retrieval. Sections III and IV formulate the related problems of semantic audio annotation and retrieval, present the SML model, and describe three parameter estimation techniques including the *weighted* mixture hierarchies algorithm. Section V describes the collection of human annotations for the CAL500 data set. Section VI describes the sound effects data set. Section VII reports qualitative and quantitative results for annotation and retrieval of music and sound effects. The final section presents a discussion of this research and outlines future directions.

II. 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 [3]. 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 nonacoustic representation of a song. This includes genre and instrument labels, song reviews, ratings according to bipolar adjectives (e.g., happy/sad), and purchase sales records. These representations can be used as input to collaborative filtering 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.

Another retrieval approach, called *query-by-similarity*, takes an audio-based query and measures the similarity between the query and all of the songs in a database [3]. 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* [4], *-beatboxing* [5], and *-tapping* [6]. 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 [4]. A natural alternative is to describe music using words, an interface that is familiar to anyone who has used an Internet search engine. A good deal of research has focused on content-based classification of music by genre [7], emotion [8], and instrumentation [9]. These classification systems effectively "annotate" music with class labels (e.g., "blues," "sad," "guitar"). The assumption of a predefined taxonomy and the explicit labeling of songs into (mutually exclusive) classes can give rise to a number of problems [10] due to the fact that music is inherently subjective.

We propose a content-based *query-by-text* audio retrieval system that learns a relationship between acoustic features and words from a data set of annotated audio tracks. Our goal is to create a more general system that directly models the relationship between audio content and a vocabulary that is less constrained than existing content-based classification systems. The query-by-text paradigm has been largely influenced by work on the similar task of image annotation. We adapt an SML

[1]The CAL500 data set can be downloaded from http://cosmal.ucsd.edu/cal.

model [1] since it has performed well on the task of image annotation. This approach views semantic annotation as one multiclass problem rather than a set of binary one-versus-all problems. A comparative summary of alternative supervised one-versus-all [11] and unsupervised ([12], [13]) models for image annotation is presented in [1].

Despite interest within the computer vision community, there has been relatively little work on developing "query-by-text" for audio (and specifically music) data. One exception is the work of Whitman *et al.* ([14]–[16]). 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* annotations for each song in our corpus. Second, they take a one-versus-all approach and learn a discriminative classifier (a support vector machine or a regularized least-squares classifier) for each word in the vocabulary. The disadvantage of the one-versus-all approach is that it results in binary decisions for each word. We propose a generative multiclass model that outputs a semantic multinomial distribution over the vocabulary for each song. As we show in Section III, the parameters of the multinomial distribution provide a natural ranking of words [1]. In addition, semantic multinomials are a compact representation of an audio track which is useful for efficient retrieval.

Other query-by-text audition systems ([17], [18]) have been developed for annotation and retrieval of sound effects. Slaney's Semantic Audio Retrieval system ([17], [19]) 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 [18]. The drawback of these nonparametric 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.

III. Semantic Audio Annotation and Retrieval

This section formalizes the related tasks of semantic audio annotation and retrieval as a supervised multiclass, multilabel classification problem where each word in a vocabulary represents a class and each song is labeled with multiple words. We learn a *word-level* (i.e., class-conditional) distribution for each word in a vocabulary by training only on the audio tracks that are positively associated with that word. A schematic overview of our model is presented in Fig. 1.

A. 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," "creaky door," etc. The goal in annotation is to find a set $\mathcal{W} = \{w_1, \ldots, w_A\}$ of A semantically meaningful words that describe a query audio track s_q. Retrieval involves rank ordering a set of tracks (e.g., songs) $\mathcal{S} = \{s_1, \ldots, s_R\}$ given a set of query words $\mathcal{W}_q$. It will be convenient to represent the text data describing each song as an

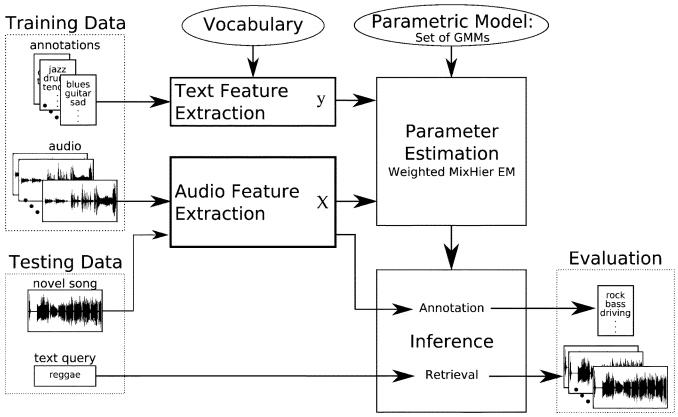


Fig. 1. Semantic annotation and retrieval model diagram.

annotation vector $\mathbf{y} = (y_1, \ldots, y_{|\mathcal{V}|})$, where $y_i > 0$ if w_i has a positive semantic association with the audio track, and $y_i = 0$ otherwise. The y_i's are called *semantic weights* since they are proportional to the strength of the semantic association. If the semantic weights are mapped to $\{0, 1\}$, then they can be interpreted as class labels. We represent an audio track s as a bag $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ of T real-valued feature vectors, where each vector $\mathbf{x}_t$ represents features extracted from a short segment of the audio content, and T depends on the length of the track. Our data set $\mathcal{D}$ is a collection of track-annotation pairs $\mathcal{D} = \{(\mathcal{X}_1, \mathbf{y}_1), \ldots, (\mathcal{X}_{|\mathcal{D}|}, \mathbf{y}_{|\mathcal{D}|})\}$.

B. Annotation

Annotation can be thought of as a multiclass classification problem in which each word $w_i \in \mathcal{V}$ represents a class and the goal is to choose the best class(es) for a given audio track. Our approach involves modeling one word-level distribution over an audio feature space, $P(\mathbf{x}|i)$, for each word $w_i \in \mathcal{V}$. Given a track represented by the bag-of-feature-vectors $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, 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. We will assume a uniform word prior, $P(i) = 1/|\mathcal{V}|$ for all $i = 1, \ldots, |\mathcal{V}|$, to promote annotation using a diverse set of words.

To estimate $P(\mathcal{X}|i)$, we assume that $\mathbf{x}_a$ and $\mathbf{x}_b$ are conditionally independent given word w_i (i.e., $\mathbf{x}_a \perp \mathbf{x}_b | w_i, \forall a, b \leq T, a \neq b$) so that $P(\mathcal{X}|i) = \prod_{t=1}^{T} P(\mathbf{x}_t|i)$. While this naïve Bayes assumption is unrealistic, attempting to model interactions between feature vectors may be infeasible due to computational complexity and data sparsity. However, ignoring the temporal dependencies tends to underestimate $P(\mathcal{X}|i)$ [20]. One common solution is to estimate $P(\mathcal{X}|i)$ with the geometric average $(\prod_{t=1}^{T} P(\mathbf{x}_t|i))^{(1/T)}$. This solution has the added benefit of producing comparable probabilities for tracks with different lengths (i.e., when bags-of-feature-vectors do not contain the same number of vectors). That is, longer tracks (with large T) will be, in general, less likely than shorter tracks (with

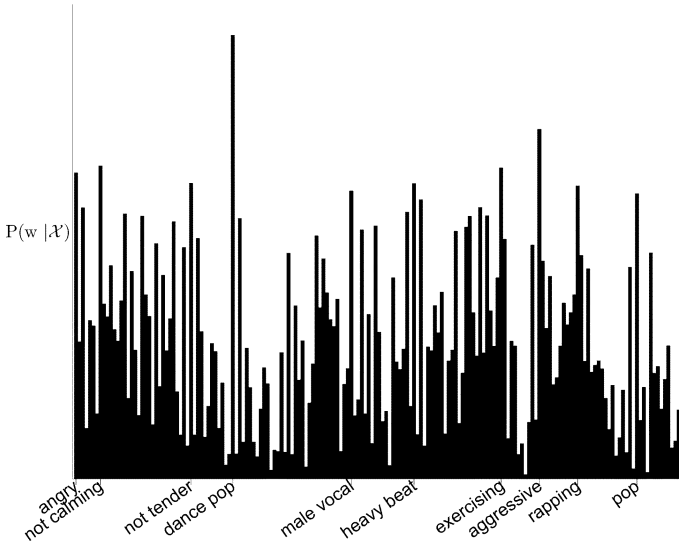


Fig. 2. Semantic multinomial distribution over all words in our vocabulary for the Red Hot Chili Pepper's *Give it Away*. The ten most probable words are labeled.

small T) if we use $\prod_{t=1}^{T} P(\mathbf{x}_t|i)$ to estimate $P(\mathcal{X}|i)$ instead of $(\prod_{t=1}^{T} P(\mathbf{x}_t|i))^{(1/T)}$.

We estimate the song prior $P(\mathcal{X})$ by $\sum_{v=1}^{|\mathcal{V}|} P(\mathcal{X}|v)P(v)$ and calculate our final *annotation* equation

$$P(i|\mathcal{X}) = \frac{\left(\prod_{t=1}^{T} P(\mathbf{x}_t|i)\right)^{\frac{1}{T}}}{\sum_{v=1}^{|\mathcal{V}|}\left(\prod_{t=1}^{T} P(\mathbf{x}_t|v)\right)^{\frac{1}{T}}}. \tag{2}$$

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

Using word-level distributions $(P(\mathbf{x}|i), \forall i = 1, \ldots, |\mathcal{V}|)$ and Bayes' rule, we use (2) 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 $\mathbf{p} = \{p_1, \ldots, p_{|\mathcal{V}|}\}$ in a "semantic space," where $p_i = P(i|\mathcal{X})$ and $\sum_i p_i = 1$. An example of such a semantic multinomial is given in Fig. 2. To annotate a track with the A best words, we first calculate the semantic multinomial distribution and then choose the A largest peaks of this distribution, i.e., the A words with maximum posterior probability.

C. Retrieval

Given the one-word query string w_q, a straightforward approach to retrieval involves ranking songs by $P(\mathcal{X}|q)$. However, we find empirically that this approach returns almost the same ranking for every word in our vocabulary. The problem is due to the fact that many word-level distributions $P(\mathbf{x}|q)$ are similar (in the Kullback–Leibler sense) to the generic distribution $P(\mathbf{x})$ over the audio feature vector space. This may be caused by using a general-purpose audio feature representation that captures additional information besides the specific semantic notion that we are attempting to model. For example, since most of the songs in our training corpus feature vocals, guitar, bass, and drums, we would expect most Rolling Stones songs to be more likely than most Louis Armstrong songs with respect to both the generic distribution $P(\mathbf{x})$ and most word-level distributions $P(\mathbf{x}|q)$. This creates a *track bias* in which generic tracks that have high likelihood under this generic distribution will also have high likelihood under many of the word-level distributions. Track bias is solved by dividing $P(\mathcal{X}|q)$ by the track prior $P(\mathcal{X})$ to normalize for track bias. Note that, if we assume a uniform word prior (which does not affect the relative ranking), this is equivalent to ranking by $P(q|\mathcal{X})$ which is calculated in (2) during annotation.

To summarize, we first annotate our audio corpus by estimating the parameters of a semantic multinomial for each track. For a one-word query w_q, we rank the tracks by the qth parameter of each track's semantic multinomial distribution. As described in [21], we can naturally extend this approach to multiword queries by constructing a *query multinomial* distribution from the words in the query string. We then rank songs by the Kullback–Leibler divergence between the query multinomial and the semantic multinomials for the tracks in our corpus.

IV. 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 tracks that have a positive association with word w_i. Each distribution is modeled with a R-component mixture of Gaussians distribution parameterized by $\{\pi_r, \mu_r, \Sigma_r\}$ for $r = 1, \ldots, R$. 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 $\mathcal{N}(\cdot|\mu, \Sigma)$ is a multivariate Gaussian distribution with mean μ, covariance matrix Σ, and mixing weight π_r. In this paper, 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 $|\mathcal{V}|$ models each have $\mathcal{O}(R \cdot D)$ parameters, where D is the dimension of feature vector $\mathbf{x}$.

We consider three parameter estimation techniques for learning the parameters of a word-level distributions: direct estimation, (weighted) modeling averaging, and (weighted) 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.

A. Direct Estimation

Direct estimation trains a model for each word w_i using the superset of feature vectors for all the songs that have word w_i in the associated human annotation: $\bigcup \mathcal{X}_d, \forall d$ such that $[\mathbf{y}_d]_i > 0$. Using this training set, we directly learn the word-level mixture of Gaussians distribution using the EM algorithm [see Fig. 3(a)]. The drawback of using this method is that computational complexity increases with training set size. We find that, in practice,

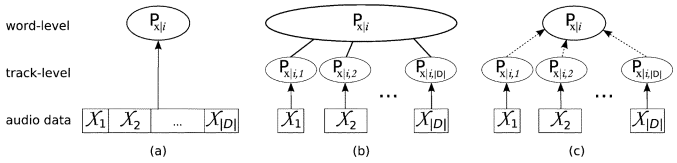


Fig. 3. (a) Direct, (b) naive averaging, and (c) mixture hierarchies parameter estimation. Solid arrows indicate that the distribution parameters are learned using standard EM. Dashed arrows indicate that the distribution is learned using mixture hierarchies EM. Solid lines indicate weighted averaging of track-level models.

we are unable to estimate parameters using this method in a reasonable amount of time since there are on the order of 100 000's of training vectors for each word-level distribution. One suboptimal work around to this problem is to simply ignore (i.e., subsample) part of the training data.

B. Model Averaging

Instead of directly estimating a word-level distribution for w_i, we can first learn *track-level* distributions, $P(\mathbf{x}|i,d)$ for all tracks d such that $[\mathbf{y}_d]_i > 0$. Here we use EM to train a track-level distribution from the feature vectors extracted from a single track. We then create a word-level distribution by calculating a weighted average of all the track-level distributions where the weights are set by how strongly each word w_i relates to that track

$$P_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|i) = \frac{1}{C}\sum_{d=1}^{|\mathcal{D}|}[\mathbf{y}_d]_i \sum_{k=1}^{K}\pi_k^{(d)}\mathcal{N}\left(\mathbf{x}|\mu_k^{(d)},\Sigma_k^{(d)}\right)$$

where $C = \sum_d [\mathbf{y}_d]_i$ is the sum of the semantic weights associated with word w_i, $|\mathcal{D}|$ is total number of training examples, and K is the number of mixture components in each track-level distribution [see Fig. 3(b)].

Training a model for each track in the training set and averaging them is relatively efficient. The drawback of this nonparametric estimation technique is that the number of mixture components in the word-level distribution grows with the size of the training database since there will be K components for each track-level distribution associated with word w_i. In practice, we may have to evaluate thousands of multivariate Gaussian distributions for each of the feature vectors $\mathbf{x}_t \in \mathcal{X}_q$ of a novel query track $\mathcal{X}_q$. Note that $\mathcal{X}_q$ may contain thousands of feature vectors depending on the audio representation.

C. Mixture Hierarchies Estimation

The benefit of direct estimation is that it produces a distribution with a fixed number of parameters. However, in practice, parameter estimation is infeasible without subsampling the training data. Model averaging efficiently produces a distribution but it is computationally expensive to evaluate this distribution since the number of parameters increases with the size of the training data set. Mixture hierarchies estimation is an alternative that efficiently produces a word-level distribution with a fixed number of parameters [2].

Consider the set of $|\mathcal{D}|$ track-level distributions (each with K mixture components) that are learned during model averaging estimation for word w_i. We can estimate a word-level distribution with R components by combining the $|\mathcal{D}| \cdot K$ track-level components using the mixture hierarchies EM algorithm. [see Fig. 3(c)]. This EM algorithm iterates between the E-step and the M-step as follows.

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

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

where N is a user-defined parameter. In practice, we set $N = K$ so that on average $\pi_k^{(d)}N$ is equal to 1.

M-step: Update the parameters of the word-level distribution

$$\pi_r^{\text{new}} = \frac{\sum_{(d),k} h^r_{(d),k}}{W \cdot K}, \quad \text{where} \quad W = \sum_{d=1}^{|\mathcal{D}|}[\mathbf{y}_d]_i$$

$$\mu_r^{\text{new}} = \sum_{(d),k} z^r_{(d),k}\mu_k^{(d)}, \quad \text{where } z^r_{(d),k} = \frac{h^r_{(d),k}\pi_k^{(d)}}{\sum_{(d),k} h^r_{(d),k}\pi_k^{(d)}}$$

$$\mathbf{\Sigma}_r^{\text{new}} = \sum_{(d),k} z^r_{(d),k}\left[\mathbf{\Sigma}_k^{(d)} + \left(\mu_k^{(d)} - \mu_t\right)\left(\mu_k^{(d)} - \mu_t\right)^T\right].$$

From a generative perspective, a track-level distribution is generated by sampling *mixture components* from the word-level distribution. The observed audio features are then samples from the track-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, yet we learn this model using all of the training data without subsampling. We have essentially replaced one computationally expensive (and often impossible) run of the standard EM algorithm with $|\mathcal{D}|$ 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.

Our formulation differs from that derived in [2] in that the responsibility $h^r_{(d),k}$ is multiplied by the semantic weight $[\mathbf{y}_d]_i$ between word w_i and audio track 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.

V. Semantically Labeled Music Data

Perhaps the fastest and most cost-effective way to collect semantic information about music is to mine web documents that relate to songs, albums, or artists [16], [22]. Whitman *et al.* collect a large number Web pages related to the artist when attempting to annotate individual songs [16]. 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 [22]. The drawback of this technique is that critics do not explicitly make decisions about the relevance of each individual 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 decided to collect a "clean" set of semantic labels by asking human listeners to explicitly label songs with acoustically relevant words. 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 [23], were annotated as relevant to the song or not; 18 emotions, found by Skowronek *et al.* [24] 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 [25] (e.g., "I would listen to this song while *driving, sleeping, etc*.").

The music corpus is a selection of 500 Western popular songs from the last 50 years by 500 different artists. This set was chosen to maximize the acoustic variation of the music while still representing some familiar genres and popular artists. The corpus includes 88 songs from the Magnatunes database [26] one from each artist whose songs are not from the classical genre.

To generate new semantic labels, we paid 66 undergraduate students to annotate our music corpus with the semantic concepts from our vocabulary. Participants were rewarded $10 per hour to listen to and annotate music in a university computer laboratory. The computer-based annotation interface contained an MP3 player and an HTML form. The form consisted of one or more radio boxes and/or check boxes for each of our 135 concepts. The form was not presented during the first 30 s of song playback to encourage undistracted listening. Subjects could advance and rewind the music and the song would repeat until they completed the annotation form. Each annotation took about 5 min, and most participants reported that the listening and annotation experience was enjoyable. We collected at least three semantic annotations for each of the 500 songs in our music corpus and a total of 1708 annotations. This annotated music corpus is referred to as the Computer Audition Lab 500 (CAL500) data set.

A. Semantic Feature Representation

We expand the set of *concepts* to a set of 237 *words* by mapping all bipolar concepts to two individual words. For example, "tender" gets mapped to "tender" and "not tender" so that we can explicitly learn separate models for tender songs and songs that are not tender. Note that, according to the data that we collected, many songs may be annotated as neither tender nor not tender. Other concepts, such as genres or instruments, are mapped directly to a single word.

For each song, we have a collection of human annotations where each annotation is a vector of numbers expressing the response of a subject to a set of words. For each word, the annotator has supplied a response of $+1$ or -1 if the annotator believes the song is or is not indicative of the word, or 0 if unsure. We take all the 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 annotators have labeled a concept w_i with $+1, +1, 0, -1$, then $[\mathbf{y}]_i = 1/4$. The semantic weights are used for parameter estimation.

For evaluation purposes, we also create a binary "ground truth" annotation vector for each song. To generate this vector, we label a song with a word if a minimum of two people vote for the word and there is a high level of agreement ($[\mathbf{y}]_i \geq \cdot 80$) between all subjects. This assures that each positive label is reliable. Finally, we prune all words that are represented by fewer than five songs. This reduces our set of 237 words to a set of 174 words.

B. Music Feature Representation

Each song is represented as a *bag-of-feature-vectors*: a set of feature vectors where each vector is calculated by analyzing a short-time segment of the audio signal. In particular, we represent the audio with a time series of *MFCC-Delta* feature vectors [27]. A time series of Mel-frequency cepstral coefficient (MFCC) [28] vectors is extracted by sliding a half-overlapping, short-time window ($\sim$23 ms) over the song's digital audio file. A MFCC-Delta vector is calculated by appending the first and second instantaneous derivatives of each MFCC to the vector of MFCCs. We use the first 13 MFCCs resulting in about 5200 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 vectors within the time series. We find that randomly subsampling 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 overall performance.

We have also explored a number of alternative feature representations, many of which have shown good performance on the task of genre classification, artist identification, song similarity, and/or cover song identification [29]. These include auditory filterbank temporal envelope [7], dynamic MFCC [7], MFCC (without derivatives), chroma features [30], and fluctuation patterns [31]. While a detailed comparison is beyond the scope of this paper, one difference between these representations is the amount of the audio content that is summarized by each feature vector. For example, a MFCC-Delta vector is computed from less than 80 ms of audio content, a dynamic MFCC vector summarizes MFCCs extracted over 3/4 of a second, and fluctuation patterns can represent information extracted from 6 s of audio content. We found that MFCC-Delta features outperformed the other representations with respect to both annotation and retrieval performance.

VI. Semantically Labeled Sound Effects Data

To confirm the general applicability of the SML model to other classes of audio data, we show that we can also annotate and retrieve sound effects. We use the BBC sound effects library which consists of 1305 sound effects tracks [17]. Each track has been annotated with a short 5–10 word caption. We automatically extract a vocabulary consisting of 348 words by including each word that occurs in five or more captions. Each caption for a track is represented as a 348-dimensional binary annotation vector where the ith value is 1 if word w_i is present in the caption, and 0 otherwise. As with music, the audio content of the sound effect track is represented as a time series of MFCC-Delta vectors, though we use a shorter short-time window ($\sim$11.5 ms) when extracting MFCC vectors. The shorter time window is used in an attempt to better represent important inharmonic noises that are generally present in sound effects.

VII. Model Evaluation

In this section, we quantitatively evaluate our SML model for audio annotation and retrieval. We find it hard to compare our results to previous work [15], [17], [18] since existing results are mainly qualitative and relate to individual tracks, or focus on a small subset of sound effects (e.g., isolated musical instruments or animal vocalizations).

For comparison, we evaluate our two SML models and compare them against three baseline models. The parameters for one SML model, denoted "MixHier," are estimated using the weighted mixture hierarchies EM algorithm. The second SML model, denoted "ModelAvg," results from weighted modeling averaging. Our three baseline models include a "Random" lower bound, an empirical upper bound (denoted "UpperBnd"), and a third "Human" model that serves as a reference point for how well an individual human would perform on the annotation task.

The "Random" model samples words (without replacement) from a multinomial distribution parameterized by the word prior distribution, $P(i)$ for $i = 1, \ldots, |\mathcal{V}|$, estimated using the observed word counts of a training set. Intuitively, this prior stochastically generates annotations from a pool of the most frequently used words in the training set. The "UpperBnd" model uses the ground truth to annotated songs. However, since we require that each model use a fixed number of words to annotate each song, if the ground truth annotation contains too many words, we randomly pick a subset of the words from the annotation. Similarly, if the ground truth annotation contains too few words, we randomly add words to the annotation from the rest of the vocabulary.

Lastly, we will compare an individual's annotation against a "ground truth" annotation that is found by averaging multiple annotations (i.e., an annotation based on group consensus). Specifically, the "Human" model is created by randomly holding out a single annotation for a song that has been annotated by four or more individuals. This model is evaluated against a "ground truth" that is obtained combining the remaining annotations for that song. (See Section V.A for the details of our summarization process.) It should be noted that each individual annotation uses on average 36 of the 174 words in our vocabulary. Each ground truth annotation uses on average only 25 words since we require a high level of agreement between multiple independent annotators for a word to be considered relevant. This reflects the fact that music is inherently subjective in that individuals use different words to describe the same song.

A. Annotation

Using (2), we annotate all test set songs with ten words and all test set sound effect tracks with six words. Annotation performance is measured using mean *per-word* precision and recall. Per-word precision is the probability that the model correctly uses the word when annotating a song. Per-word recall is the probability that the model annotates a song that should have been annotated with the word. More formally, for each word w, $|w_H|$ is the number of tracks that have word w in the human-generated "ground truth" annotation. $|w_A|$ is the number of tracks that our model automatically annotates with word w. $|w_C|$ 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 $|w_C|/|w_A|$.[2] While trivial models can easily maximize one of these measures (e.g., 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 a ground truth annotation is greater or lesser than the number of words that are output by our model. For example, if our system outputs ten words to annotate a test song where the ground truth annotation contains 25 words, mean per-word recall will be upper-bounded by a value less than one. The exact upper bounds for recall and precision depend on the relative frequencies of each word in the vocabulary and can be empirically estimated using the "UpperBnd" model which is described above.

It may seem more straightforward 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 ten best words to annotate each of the 500 songs, our system outputs 166 of the 174 words for at least one song.

Table III presents quantitative results for music and Table IV for sound effects. Table III also displays annotation results using only words from each of six semantic categories (emotion, genre, instrumentation, solo, usage, and vocal). All reported results are means and standard errors computed from tenfold cross-validation (i.e., 450-song training set, 50-song test set).

The quantitative results demonstrate that the SML models trained using model averaging (ModelAvg) and mixture hierarchies estimation (MixHier) significantly outperform the random

[2]If the model never annotates a song with word w, then per-word precision is undefined. In this case, we estimate per-word precision using the empirical prior probability of the word $P(i)$. Using the prior is similar to using the "Random" model to estimate the per-word precision, and thus, will in general hurt model performance. This produces a desired effect since we are interested in designing a model that annotates songs using many words from our vocabulary.

TABLE III
MUSIC ANNOTATION RESULTS. TRACK-LEVEL MODELS HAVE $K = 8$ MIXTURE COMPONENTS, WORD-LEVEL MODELS HAVE $R = 16$ MIXTURE COMPONENTS. A = ANNOTATION LENGTH (DETERMINED BY THE USER), $|\mathcal{V}|$ = VOCABULARY SIZE

Category	A / $\lvert\mathcal{V}\rvert$	Model	Precision		Recall	
All Words	10 / 174	Random	0.144	(0.004)	0.064	(0.002)
		Human	0.296	(0.008)	0.145	(0.003)
		UpperBnd	*0.712*	(0.007)	*0.375*	(0.006)
		ModelAvg	0.189	(0.007)	0.108	(0.009)
		MixHier	**0.265**	(0.007)	**0.158**	(0.006)
Emotion	4 / 36	Random	0.276	(0.012)	0.113	(0.004)
		Human	0.453	(0.014)	0.180	(0.006)
		UpperBnd	*0.957*	(0.005)	*0.396*	(0.010)
		ModelAvg	0.366	(0.012)	0.179	(0.005)
		MixHier	**0.424**	(0.008)	**0.195**	(0.004)
Genre	2 / 31	Random	0.055	(0.005)	0.079	(0.008)
		Human	0.268	(0.017)	0.290	(0.021)
		UpperBnd	*0.562*	(0.026)	*0.777*	(0.018)
		ModelAvg	0.122	(0.012)	0.161	(0.017)
		MixHier	**0.171**	(0.009)	**0.242**	(0.019)
Instrumentation	4 / 24	Random	0.141	(0.009)	0.195	(0.014)
		Human	0.416	(0.014)	0.522	(0.008)
		UpperBnd	*0.601*	(0.015)	*0.868*	(0.018)
		ModelAvg	**0.267**	(0.008)	0.320	(0.022)
		MixHier	0.259	(0.010)	**0.381**	(0.021)
Solo	1/ 9	Random	0.031	(0.007)	0.155	(0.035)
		Human	0.104	(0.020)	0.158	(0.034)
		UpperBnd	*0.197*	(0.019)	*0.760*	(0.052)
		ModelAvg	0.057	(0.012)	0.231	(0.033)
		MixHier	**0.060**	(0.012)	**0.261**	(0.050)
Usage	2 / 15	Random	0.073	(0.008)	0.154	(0.016)
		Human	0.125	(0.012)	0.175	(0.023)
		UpperBnd	*0.363*	(0.014)	*0.814*	(0.031)
		ModelAvg	0.103	(0.010)	0.170	(0.017)
		MixHier	**0.122**	(0.012)	**0.264**	(0.027)
Vocal	2 / 16	Random	0.062	(0.007)	0.153	(0.018)
		Human	0.188	(0.021)	0.304	(0.023)
		UpperBnd	*0.321*	(0.017)	*0.788*	(0.019)
		ModelAvg	0.102	(0.008)	0.226	(0.016)
		MixHier	**0.134**	(0.005)	**0.335**	(0.021)

TABLE IV
SOUND EFFECTS ANNOTATION RESULTS. $A = 6, |\mathcal{V}| = 348$

Model	Recall		Precision	
Random	0.018	(0.002)	0.012	(0.001)
UpperBnd	*0.973*	(0.004)	*0.447*	(0.009)
ModelAvg ($K = 4$)	**0.360**	(0.014)	**0.179**	(0.010)
MixHier ($K = 8, R = 16$)	0.306	(0.010)	0.145	(0.005)

baselines for both music and sound effects. For music, MixHier significantly outperforms ModelAvg in both precision and recall when considering the entire vocabulary as well as showing superior performance for most semantic categories, where "instrumentation precision" is the sole exception. However, for sound effects, ModelAvg significantly outperforms MixHier. This might be explained by interpreting model averaging as a nonparametric approach in which the likelihood of the query track is computed under every track-level model in the database. For our sound effects data set, it is often the case that semantically related pairs of tracks are acoustically very similar causing that one track-level model to dominate the average.

Over the entire music vocabulary, the MixHier model performance is comparable to the Human model. It is also interesting to note that MixHier model performance is significantly worse than the Human model performance for the more "objective" semantic categories (e.g., Instrumentation and Genre) but is comparable for more "subjective" semantic categories (e.g., Usage and Emotion). We are surprised by the low Human model precision, especially for some of these more objective categories, when compared against the UpperBnd model. Taking a closer look at precision for individual words, while there are some words with relatively high precision, such as "male lead vocals" (0.96) and "drum set" (0.81), there are many words with low precision. Low precision words arise from a number of causes including test subject inattentiveness (due to boredom or fatigue), nonexpert test-subjects (e.g., cannot detect a "trombone" in a horn section), instrument ambiguity (e.g., deciding between "acoustic guitar" versus. "clean electric guitar"), and our summarization process. For example, consider the word "clean electric guitar" and the song *Everything She Does is Magic* by The Police. Given four test subjects, two subjects positively associate the song with the word because the overall guitar sound is clean, one is unsure, and one says there is no "clean electric guitar" presumably because, technically, the guitarist makes use of a delay distortion.[3] Our summarization process would not use the word to label this songs despite the fact that half of the subjects used this word to describe the song. In Section VIII, we will discuss both ways to improve the survey process as well as an alternative data collection technique.

B. Retrieval

For each one-word query w_q in $\mathcal{V}$, we rank-order a test set of songs. For each ranking, we calculate the average precision (AP) [13] and the area under the receiver operating characteristic curve (AROC). 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 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 audio tracks. Mean AP and Mean AROC are found by averaging each metric over all the words in our vocabulary (shown in Tables V and VI).

As with the annotation results, we see that our SML models significantly outperform the random baseline and that MixHier outperforms ModelAvg for music retrieval. For sound effects retrieval, MixHier and ModelAvg are comparable if we consider Mean AROC, but MixHier shows superior performance if we consider Mean AP.

VIII. DISCUSSION AND FUTURE WORK

The qualitative annotation and retrieval results in Tables I and II indicate that our system can produce sensible semantic annotations for an acoustically diverse set of songs and can retrieve relevant songs given a text-based query. When comparing these results with previous results based on models trained using web-mined data [22], it is clear that using "clean" data (i.e., the CAL500 data set) results in much more intuitive music reviews and search results.

Our goal in collecting the CAL500 data set was to quickly and cheaply collect a small music corpus with reasonably accurate annotations for the purposes of training our SML model.

[3]A delay causes the sound to repeatedly echo as the sound fades away, but does not grossly distort the timbre of electric guitar.

TABLE V
MUSIC RETRIEVAL RESULTS. $|\mathcal{V}| = 174$

Category	$\|\mathcal{V}\|$	Model	MeanAP		MeanAROC	
All Words	174	Random	0.231	(0.004)	0.503	(0.004)
		ModelAvg	0.372	(0.008)	0.682	(0.006)
		MixHier	**0.390**	(0.004)	**0.710**	(0.004)
Emotion	36	Random	0.327	(0.006)	0.504	(0.003)
		ModelAvg	0.486	(0.013)	0.685	(0.010)
		MixHier	**0.506**	(0.008)	**0.710**	(0.005)
Genre	31	Random	0.132	(0.005)	0.500	(0.005)
		ModelAvg	0.309	(0.020)	0.695	(0.008)
		MixHier	**0.329**	(0.012)	**0.719**	(0.005)
Instrumentation	24	Random	0.221	(0.007)	0.502	(0.004)
		ModelAvg	0.372	(0.015)	0.694	(0.008)
		MixHier	**0.399**	(0.018)	**0.719**	(0.006)
Solo	9	Random	0.106	(0.014)	0.502	(0.004)
		ModelAvg	**0.190**	(0.028)	0.688	(0.008)
		MixHier	0.180	(0.025)	**0.712**	(0.006)
Usage	15	Random	0.169	(0.012)	0.501	(0.005)
		ModelAvg	0.231	(0.012)	0.684	(0.007)
		MixHier	**0.240**	(0.016)	**0.707**	(0.004)
Vocal	16	Random	0.137	(0.006)	0.502	(0.004)
		ModelAvg	0.234	(0.019)	0.680	(0.007)
		MixHier	**0.260**	(0.018)	**0.705**	(0.005)

TABLE VI
SOUND EFFECTS RETRIEVAL RESULTS. $|\mathcal{V}| = 348$

Model	Mean AP		Mean AROC	
Random	0.051	(0.002)	0.506	(0.004)
ModelAvg ($K = 4$)	0.183	(0.003)	**0.785**	(0.005)
MixHier ($K = 8, R = 16$)	**0.331**	(0.008)	0.784	(0.006)

The human experiments were conducted using (mostly) nonexpert college students who spent about 5 min annotating each song using our survey. While we think that the CAL500 data set will be useful for future content-based music annotation and retrieval research, it is not of the same quality as data that might be collected using a highly controlled psychoacoustics experiment. Future improvements would include spending more time training our test subjects and inserting consistency checks so that we could remove inaccurate annotations from test subjects who show poor performance.

Currently, we are looking at two extensions to our data collection process. The first involves vocabulary selection: if a word in the vocabulary is inconsistently used by human annotators, or the word is not clearly represented by the underlying acoustic representation, the word can be considered as *noisy* and should be removed from the vocabulary to denoise the modeling process. We explore these issues in [32], whereby we devise vocabulary pruning techniques based on measurements of human agreement and correlation of words with the underlying audio content.

Our second extension involves collecting a much larger annotated data set of music using web-based human computation games [33]. We have developed a web-based game called "Listen Game" which allows multiple "annotators" to label music through realtime competition. We consider this to be a more scalable and cost-effective approach for collecting high-quality music annotations than laborious surveys. We are also able to grow our vocabulary by allowing users to suggest words that describe the music.

Our weighted mixture hierarchies EM is more computationally efficient and produces better density estimates than direct estimation or modeling averaging. The improvement in performance may be attributed to the fact that we represent each track with a track-level distribution before modeling a word-level distribution. The track-level distribution is a smoothed representation of the bag-of-feature-vectors that are extracted from the audio signal. We then learn a mixture from the mixture components of the track-level distributions that are semantically associated with a word. The benefit of using smoothed estimates of the tracks is that the EM framework, which is prone to find poor local maxima, is more likely to converge to a better density estimate.

The *semantic multinomial* representation of a song, which is generated during annotation (see Section III-B), is a useful and compact representation of a song. In derivative work [21], we show that if we construct a *query multinomial* based on a multiword query string, we can quickly retrieve relevant songs based on the Kullback–Liebler (KL) divergence between the query multinomial and all semantic multinomials in our database of automatically annotated tracks. The semantic multinomial representation is also useful for related audio information tasks such as "retrieval-by-semantic-similarity" [34], [35].

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 to 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 [12], [13]. 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 temporal dependencies between audio feature vectors. Future research will involve exploring models, such as hidden Markov models, that explicitly model the longer term temporal aspects of music.

Lastly, our future work will involve modeling individual users (or subsets of similar users) with *user-specific* models. For example, during data collection, we had one test subject annotate 200 of the 500 songs in our data set. A preliminary study showed that we were better able to predict some words (especially "usage" words) for this subject using the 200-song subset when compared against models trained using the entire CAL500 data set. This is not surprising since we would expect an individual to be *self-consistent* when annotating songs with subjective concepts. We expect that user-specific models will offer us a chance to reduce the impact caused by subjectivity in music so that we can better model an individual's notions of audio semantics.

ACKNOWLEDGMENT

The authors would like to thank A. Chan, A. Cont, G. W. Cottrell, S. Dubnov, C. Elkan, L. Saul, N. Vasconcelos, and our anonymous reviewers for their helpful comments.

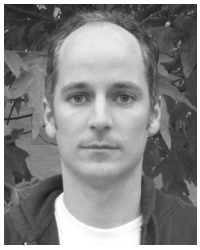
REFERENCES

[1] G. Carneiro and N. Vasconcelos, "Formulating semantic image annotation as a supervised learning problem," in *Proc. IEEE CVPR*, 2005, pp. 163–168.
[2] N. Vasconcelos, "Image indexing with mixture hierarchies," in *Proc. IEEE CVPR*, 2001, pp. 3–10.

[3] M. Goto and K. Hirata, "Recent studies on music information processing," *Acoust. Sci. Technol.*, vol. 25, no. 4, pp. 419–425, 2004.
[4] R. B. Dannenberg and N. Hu, "Understanding search performance in query-by-humming systems," in *Proc. ISMIR*, 2004, pp. 232–237.
[5] A. Kapur, M. Benning, and G. Tzanetakis, "Query by beatboxing: Music information retrieval for the dj," in *Proc. ISMIR*, 2004, pp. 170–178.
[6] G. Eisenberg, J. M. Batke, and T. Sikora, "Beatbank—An MPEG-7 compliant query by tapping system," *Audio Eng. Soc. Conv.*, 2004, paper 6136.
[7] M. F. McKinney and J. Breebaart, "Features for audio and music classification," in *Proc. ISMIR*, 2003, pp. 151–158.
[8] T. Li and G. Tzanetakis, "Factors in automatic musical genre classification of audio signals," in *IEEE WASPAA*, 2003, pp. 143–146.
[9] S. Essid, G. Richard, and B. David, "Inferring efficient hierarchical taxonomies for music information retrieval tasks: Application to musical instruments," in *Proc. ISMIR*, 2005, pp. 324–328.
[10] F. Pachet and D. Cazaly, "A taxonomy of musical genres," *RIAO*, 2000.
[11] D. Forsyth and M. Fleck, "Body plans," in *Proc. IEEE CVPR*, 1997, pp. 678–683.
[12] D. M. Blei and M. I. Jordan, "Modeling annotated data," in *Proc. ACM SIGIR*, 2003, , pp. 127–134.
[13] S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple bernoulli relevance models for image and video annotation," in *Proc. IEEE CVPR*, 2004, pp. II-1002–II-1009.
[14] B. Whitman, "Learning the meaning of music," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 2005.
[15] B. Whitman and D. Ellis, "Automatic record reviews," in *Proc. ISMIR*, 2004, pp. 470–477.
[16] B. Whitman and R. Rifkin, "Musical query-by- description as a multiclass learning problem," in *IEEE Workshop Multimedia Signal Process.*, 2002, pp. 153–156.
[17] M. Slaney, "Semantic-audio retrieval," in *Proc. IEEE ICASSP*, 2002, pp. IV-1408–IV-1411.
[18] P. Cano and M. Koppenberger, "Automatic sound annotation," in *Proc. IEEE Workshop Mach. Learn. Signal Process.*, 2004, pp. 391–400.
[19] M. Slaney, "Mixtures of probability experts for audio retrieval and indexing," in *Proc. IEEE Multimedia Expo.*, 2002, pp. 345–348.
[20] D. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, pp. 19–41, 2000.
[21] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Towards musical query-by-semantic description using the CAL500 data set," in *Proc. SIGIR'07*, 2007, pp. 439–446.
[22] D. Turnbull, L. Barrington, and G. Lanckriet, "Modelling music and words using a multi-class naïve bayes approach," in *Proc. ISMIR*, 2006, pp. 254–259.
[23] C. McKay, D. McEnnis, and I. Fujinaga, "A large publicly accessible prototype audio database for music research," in *Proc. ISMIR*, 2006, pp. 160–163.
[24] J. Skowronek, M. McKinney, and S. ven de Par, "Ground-truth for automatic music mood classification," in *Proc. ISMIR*, 2006, pp. 395–396.
[25] X. Hu, J. S. Downie, and A. F. Ehmann, "Exploiting recommended usage metadata: Exploratory analyses," in *Proc. ISMIR*, 2006, pp. 19–22.
[26] "Magnatune: Free MP3 music and music licensing," Magnatune [Online]. Available: http://www.magnatune.com
[27] C. R. Buchanan, "Semantic-based audio recognition and retrieval," M.S. thesis, School of Informatics, Univ. Edinburgh, Edinburgh, U.K., 2005.
[28] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[29] J. S. Downie, "Music information retrieval evaluation exchange (Mirex)," [Online]. Available: http://www.music-ir.org/mirex2006
[30] D. Ellis and G. Poliner, "Identifying cover songs with chroma features and dynamic programming beat tracking," in *Proc. IEEE ICASSP*, 2007, pp. IV-1429–IV-1432.
[31] E. Pampalk, "Computational models of music similarity and their application in music information retrieval," Ph.D. dissertation, Vienna Univ. Technol., Vienna, Austria, 2006.
[32] D. Torres, D. Turnbull, L. Barrington, and G. Lanckriet, "Identifying words that are musically meaningful," in *Proc. ISMIR'07*, 2007, pp. 405–410.
[33] D. Turnbull, R. Liu, L. Barrington, D. Torres, and G. anckriet, "Using games to collect semantic information about music," in *Proc. ISMIR'07*, 2007, pp. 535–538.
[34] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman, "A large-scale evalutation of acoustic and subjective music-similarity measures," *Comput. Music J.*, pp. 63–76, 2004.
[35] L. Barrington, A. Chan, D. Turnbull, and G. Lanckriet, "Audio information retrieval using semantic similarity," in *Proc. IEEE ICASSP*, 2007, pp. II-725–II-728.

Douglas Turnbull (S'08) received the B.S.E. degree (with honors) from Princeton University, Princeton, NJ, in 2001 and the M.S. degree from the University of California at San Diego (UCSD), La Jolla, in 2005. He is currently pursuing the Ph.D. degree and is a National Science Foundation Integrative Graduate Education and Research Traineeship (NSF IGERT) Fellow in the Computer Science and Engineering Department, UCSD.

He is a cofounder of the Computer Audition Laboratory (CAL) with L. Barrington and advisor G. Lanckriet. His primary research interests include machine learning, signal processing, and (music) information retrieval. In the summer of 2006, he was an NSF EASPI Fellow at the National Institute of Advance Industrial Science and Technology (AIST), Tsukuba, Japan. His favorite instrument is the banjo.



Luke Barrington received the B.E. degree in electrical engineering from the University College Dublin (UCD), Dublin, U.K., in 2001 and the M.S. degree from University of California, San Diego (UCSD), La Jolla, in 2004. He is currently pursuing the Ph.D. degree in the Electrical and Computer Engineering Department, UCSD, where he is a National Science Foundation Integrative Graduate Education and Research Traineeship (NSF IGERT) Fellow.

He is a cofounder of the Computer Audition Laboratory at UCSD. In 2005, he was an NSF EAPSI Fellow at the Advanced Telecommunications Research Institute, Kansai, Japan. He is an avid musician.

Mr. Barrington was the UCD Young Engineer of the Year in 2001.

David Torres received the B.S. degrees in both computer science and applied mathematics from the University of Texas at Austin in 2004. He is currently pursuing the M.S. degree in the Computer Science and Engineering Department, University of California at San Diego, La Jolla.

He is a member of the Computational Statistics and Machine Learning Group (COSMAL) and is one of the first members of the Computer Audition Laboratory at UCSD. His interests are in optimization and machine learning, concentrating on its applications to music analysis.

Gert Lanckriet received the Electrical Engineering degree from the Catholic University Leuven, Leuven, Belgium, in 2000 and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 2001 and 2005, respectively.

He is an Assistant Professor in the Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, and cofounder of the interdisciplinary Computer Audition Laboratory. He conducts research on machine learning, applied statistics, and convex optimization with applications in computer audition and music as well as biology and finance.