

# Exploring the Semantic Annotation and Retrieval of Sound

Technical Report CAL-2007-01

Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet \*

February 16, 2007

©University of California San Diego, 2007

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of the Computer Audition Laboratory of the University of California, San Diego; an acknowledgment of the authors and individual contributors to the work; and all applicable portions of the copyright notice. Copying, reproducing, or republishing for any other purpose shall require a license with payment of fee to the University of California, San Diego. All rights reserved.

Computer Audition Laboratory (CAL) Technical reports are available on the CALs web page at <http://cosmal.ucsd.edu/cal> or you may contact us by mail at:

Prof. Gert Lanckriet  
EBU1, Room 5604  
University of California, San Diego  
9500 Gilman Drive, Mail code 0407  
La Jolla, CA 92093-0407

---

\*Corresponding Author: Douglas Turnbull - [dturnbul AT cs.ucsd.edu](mailto:dturnbul@cs.ucsd.edu)

## Abstract

We present a computer audition system that can both *annotate* novel audio tracks with semantically meaningful words and use a semantic query to *retrieve* relevant tracks from database of unlabeled audio content. We consider the related tasks of content-based audio annotation and retrieval as one supervised multi-class problem in which we model the joint probability of acoustic features and words. We collect a data set of 1700 human-generated musical annotations that describe 500 popular western musical 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.

## 1 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 use a semantic query to *retrieve* relevant audio tracks from database of unannotated tracks.

We view semantic annotation and retrieval of audio as supervised multi-class learning problems based on a heterogeneous data set.‘ Each data point (e.g., a song) is represented by both an audio track and descriptive text. The

audio content is represented as a set of feature vectors that are extracted by passing a short-time window over the audio signal. The text descriptions are represented by *annotation vectors*, vectors of weights where each element indicates how strongly a semantic concept applies to the audio track. An annotation vector represents an audio track as a multinomial distribution over semantic concepts where each parameter of the multinomial represents the probability that a word relates to the track.

Our probabilistic model is a set of *word-level* distributions over the audio feature space for each word in a 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 data points that are labeled with a positive semantic association with a word. To make parameter estimation and inference tractable, we make the naïve Bayes assumption that each feature vector is conditionally independent given a word. Using this *supervised multi-class naïve Bayes* (SMC-NB) 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 from a retrieval database.

Table 1 displays some qualitative annotations of songs produced by our system. Placing the most likely words from each semantic concept in a natural language context demonstrates how our annotation system could be used to generate automatic music reviews. Table 2 shows the top songs that the system retrieves from our data set, given a semantic query.

The SMC-NB algorithm was recently proposed for the task of 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 classification performance. We confirm these findings for audio data and extend this estimation technique to handle real-valued (rather than binary) class labels. Real-value 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 about three out of four people annotate Elvis Presley’s “Heartbreak Hotel” as being a ‘blues’ song.

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

Table 1: Automatic annotations. Words in **bold** are output by our system.

<p style="text-align: center;">Frank Sinatra - Fly me to the moon</p> <p>This is a <b>jazzy, singer / songwriter</b> song that is <b>calming</b> and <b>sad</b>. It features <b>acoustic guitar, piano, saxophone</b>, a nice <b>male vocal solo</b>, and <b>emotional, high-pitched vocals</b>. It is a song with a <b>light beat</b> and a <b>slow tempo</b> that you might like listen to while <b>hanging with friends</b>.</p>
<p style="text-align: center;">Creedence Clearwater Revival - Travelin' Band</p> <p>This is a <b>rockin', classic rock</b> song that is <b>arousing</b> and <b>powerful</b>. It features <b>clean electric guitar, backing vocals, distorted electric guitar</b>, a nice <b>distorted electric guitar solo</b>, and <b>strong, duet vocals</b>. It is a song with a <b>catchy</b> feel and is <b>very danceable</b> that you might like listen to while <b>driving</b>.</p>
<p style="text-align: center;">New Order - Blue Monday</p> <p>This is a <b>poppy, electronica</b> song that is <b>not emotional</b> and <b>not tender</b>. It features <b>sequencer, drum machine, synthesizer</b>, a nice <b>male vocal solo</b>, and <b>altered with effects, high-pitched</b> vocals. It is a song with a <b>synthesized texture</b> and with <b>positive feelings</b> that you might like listen to while <b>at a party</b>.</p>
<p style="text-align: center;">Dr. Dre (feat. Snoop Dogg) - Nuthin' but a 'G' thang</p> <p>This is <b>dance poppy, hip-hop</b> song that is <b>arousing</b> and <b>exciting</b>. It features <b>drum machine, backing vocals, male vocal</b>, a nice <b>acoustic guitar solo</b>, and <b>rapping, strong</b> vocals. It is a song that is <b>very danceable</b> and with a <b>heavy beat</b> that you might like listen to while <b>at a party</b>.</p>

as emotional content, genre, instrumentation, and vocal characterizations. The music data used is a set of 500 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.

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 3 and 4 formulate the related problems of semantic audio annotation and retrieval, present the SMC-NB model, and describe three parameter estimation techniques including the *weighted* mix-

Table 2: Music retrieval examples. Words in ‘quotes’ are semantic concepts from our vocabulary used as a music retrieval query.

Query	Top 5 Retrieved Songs
‘Tender / Soft’ (Emotion)	Chet Baker - These foolish things Saras - Prelude Norah Jones - Don’t know why Art Tatum - Willow weep for me Crosby Stills and Nash - Guinnevere
‘Hip Hop / Rap’ (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

ture hierarchies algorithm. Section 5 describes the study used to collect human semantic annotations of a music data set. Section 6 describes the sound effects data set. Section 7 reports qualitative and quantitative results for annotation and retrieval of music and sound effects. 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 [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. This type of metadata can also be used to augment acoustic similarity between songs[4]. We consider a more general definition of musical metadata as any non-acoustic 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* [5], *-beatboxing* [6], and *-tapping* [7]. 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 [5]. A natural alternative is to describe music using words, an interface that anyone who has used an Internet search engine will be familiar with. A good deal of research has focused on content-based classification of music by genre [8], emotion [9], and instrumentation [10]. 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 [11] due to the fact that music is inherently subjective. A more flexible approach [12] considers similarity between songs in a semantic ‘anchor space’ where each dimension is a musical genre.

We propose a content-based *query-by-text* music retrieval system that learns a relationship between acoustic features and words using a heterogeneous data set of audio tracks and annotation vectors. 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 adopt a supervised multi-class naïve Bayes [1] model since it has performed well on the task of image annotation. This approach views semantic annotation as one multi-class problem rather than a set of binary one-vs-all problems. A com-

parative summary of alternative supervised one-vs-all [13] and unsupervised [14, 15] 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.[16, 17, 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* annotations for each song in our corpus. Second, they takes 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. The generative multi-class approach we propose outputs a natural ranking of words [1].

Other query-by-text audition systems [19, 20] have been developed for annotation and retrieval of sound effects. Slaney's Semantic Audio Retrieval system [21, 22] 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 audio annotation and retrieval

This section formalizes the related problems of semantic audio annotation and retrieval as supervised, multi-class classification tasks where each word in a vocabulary represents a class. 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 Figure 1.

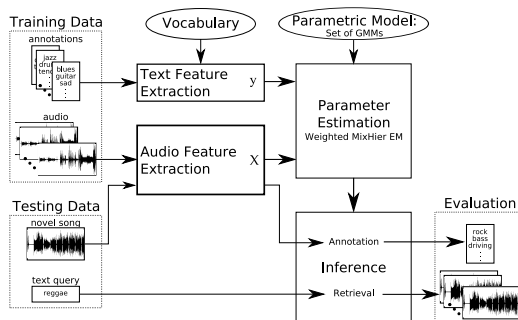


Figure 1: Semantic annotation and retrieval model diagram.

### 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’, ‘creaky door’, etc. The goal in annotation is to find a set  $\mathcal{W} = \{w_1, \dots, w_A\}$  of  $A$  semantically meaningful words that describe a query audio track  $s_q$ . Retrieval involves rank ordering a set of tracks (e.g., songs)  $\mathcal{S} = \{s_1, \dots, s_R\}$  given a query  $\mathcal{W}_q$ . It will be convenient to represent the text data describing each song as an *annotation* vector  $\mathbf{y} = (y_1, \dots, y_M)$  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 set  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$  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 song. Our data set  $\mathcal{D}$  is a collection of track-document pairs  $\mathcal{D} = \{(\mathcal{X}_1, \mathbf{y}_1), \dots, (\mathcal{X}_D, \mathbf{y}_D)\}$ .

### 3.2 Annotation

Annotation can be thought of as a multi-class 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 song. Our approach involves modeling a word-level distribution over audio features,  $P(\mathbf{x}|i), i \in \{1, \dots, |\mathcal{V}|\}$  for each word  $w_i \in \mathcal{V}$ . Given a query track represented by the set of audio feature vectors  $\mathcal{X}_q = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ , we use Bayes’ rule to calculate the posterior probability of each word in the

vocabulary, given the audio features:

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

where  $P(i)$  is the prior probability that word  $w_i$  will appear in an annotation. If we assume that  $\mathbf{x}_a$  and  $\mathbf{x}_b$  ( $\forall a, b \leq T, a \neq b$ ) are conditionally independent given word  $w_i$ , then

$$P(i|\mathcal{X}_q) = \frac{[\prod_{t=1}^T P(\mathbf{x}_t|i)] \cdot P(i)}{P(\mathcal{X}_q)}. \quad (2)$$

The naïve Bayes assumption implies that there is no temporal relationship between audio features. 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 prior,  $P(i) = 1/|\mathcal{V}|$ , for all  $i = 1, \dots, |\mathcal{V}|$  since the  $T$  factors in the product will dominate the word prior. The track prior,  $P(\mathcal{X}_q)$ , is the mean over all the word-level distributions so the  $1/|\mathcal{V}|$  terms cancel:

$$P(i|\mathcal{X}_q) = \frac{\prod_{t=1}^T P(\mathbf{x}_t|i)}{\sum_{d=1}^{|\mathcal{V}|} \prod_{t=1}^T P(\mathbf{x}_t|d)}. \quad (3)$$

Using word-level distributions,  $P(\mathbf{x}|i) \forall i \in 1..|\mathcal{V}|$ , to calculate the posterior probabilities of each word with Equation 3 produces a natural ranking of the words in the vocabulary. Normalizing these word posteriors so that they sum to one results in a *semantic multinomial* distribution description of the query track where each parameter of the multinomial represents a concept in the vocabulary. Each track in our database can now be compactly represented as a vector in a ‘semantic space’. An example of such a semantic multinomial is given in Figure 2. To annotate an audio track with the  $A$  best words, we use the word-level models to generate the track’s semantic distribution and then choose the  $A$  peaks of the multinomial, i.e., the words with maximum posterior probability.

### 3.3 Retrieval

For retrieval, we want to rank all songs in a test set based on their conditional probability given a single-word query  $w_q$ . We find empirically that using the posterior probability of the track’s audio features given the word (i.e., the

word-level distribution  $P(\mathcal{X}|q)$  always returns the same ranking under every trained word model since some tracks are much more likely than others. The first reason for this is that longer tracks (with more features) have lower likelihoods resulting from the product of additional probability terms (i.e.,  $T$  is larger in Equation 2). It has been argued that the underestimation of the likelihood is due to the poor conditional independence (naïve Bayes) assumption between the audio feature vectors [23]. The standard solution is to calculate the geometric-mean posterior for each track

$$G(P(\mathcal{X}|q)) = \left[ \prod_{t=1}^T P(\mathbf{x}_t|q) \right]^{\frac{1}{T}},$$

where  $T$  is proportional to the length of the song.

The second, more subtle, problem with using word-level distributions to rank tracks is due to the fact that many word-level distributions  $P(\mathbf{x}|q)$  are similar (in the Kulback-Leibler sense) to the track prior  $P(\mathbf{x})$ . This creates a *track bias* in which generic tracks that have high likelihood under the track prior will also have high likelihood under many of the word-level distributions.

Both of these problems can be solved by using the semantic annotations derived above as the basis for text-based audio information retrieval. Dividing  $P(\mathcal{X}|q)$  by the track prior  $P(\mathcal{X})$  normalizes for both the length and track bias. If we again include a uniform word prior (which doesn't affect the relative ranking), we are ranking by *word-posterior*,  $P(q|\mathcal{X})$ , calculated in Equation 3, for *retrieval*:

$$\frac{P(\mathcal{X}|q) \cdot P(q)}{P(\mathcal{X})} = P(q|\mathcal{X}). \quad (4)$$

Given query  $w_q$ , we find the tracks that maximize the word-posterior. Dividing by the track prior,  $P(\mathcal{X})$ , normalizes for both the track and length biases and effectively allows each song to place more weight on the words that have highest *relative* word posterior. This is equivalent to ranking tracks by the  $q$ -th parameter of each track's semantic multinomial distribution.

By computing the likelihood of the acoustic features of a given audio track under each of the learned class-conditional word-level models, we assign probability to each word in a vocabulary and generate a multinomial distribution over semantic concepts to represent the track. We use this semantic distribution to annotate a given track with appropriate words or retrieve relevant tracks from a database, given a query word. This semantic representation

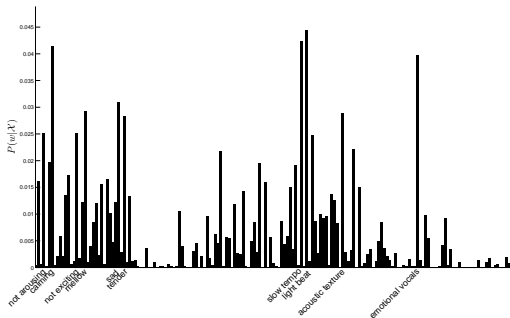


Figure 2: Semantic multinomial distribution over all vocabulary words for Frank Sinatra’s ”Fly me to the moon”. The 10 most probable words are labeled.

of audio information is very compact, having dimension equal to the size of the vocabulary, and allows rapid information retrieval. The complexity of adding new audio data is proportional to the size of the vocabulary, not the number of database items. New semantic concepts can easily be added to the vocabulary by learning a new word-level distribution, adding the posterior of each database item to the semantic representation and renormalizing.

## 4 Parameter Estimation

For each word  $w_i \in \mathcal{V}$ , we learn the parameters of the word-level 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  $r = 1, \dots, R$ . 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  $\mu$  and covariance matrix  $\Sigma$  and  $\pi_r$  is the contribution of component  $r$  to the overall mixture. 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

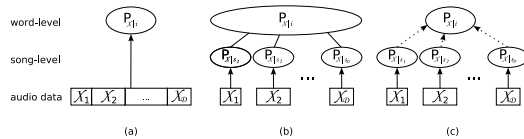


Figure 3: (a) Direct, (b) naive averaging, and (c) mixture hierarchies parameter estimation. Solid arrows indicate that the distribution parameters are learned using standard EM. Dashed arrows indicate that the distribution is learned using mixture hierarchies EM.

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 a supervised multi-class naïve Bayes model: direct estimation, (weighted) modeling averaging, and (weighted) mixture hierarchies. 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 sub-problems and then merge these results to produce a final density estimate.

## 4.1 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:  $\cup \mathcal{X}_d, \forall d$  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 Figure 3a). The drawback of using this method is that computational complexity increases with training set size. We find that, in practice, 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. Subsampling the training data is also not optimal since this does not utilize all of the available training data.

## 4.2 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:

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

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

Training a model for each track in the training set and averaging them is relatively efficient. The drawback of this non-parametric 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 on the order of 10,000 feature vectors depending on the audio representation.

### 4.3 Mixture hierarchies estimation

The benefit of direct estimation is that it produces a parametric distribution with a fixed number of parameters. However, in practice, parameter estimation is infeasible without subsampling the training data. Model averaging estimation can efficiently produce a parametric 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 efficient alternative that produces word-level distributions with a fixed number of parameters [2].

Consider the set of  $D$  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 using an extension of the EM algorithm where the track-level distributions are generated by sampling the world level distributions.(see Figure 3c.) This EM algorithm iterates between the E-step and the M-step:

**E-step:** Compute the responsibilities of each word-level component,  $r$ , to a

track level component,  $k$  from track  $d$

$$h_{(d),k}^r = \frac{[\mathbf{y}_d]_i \left[ \mathcal{N}(\mu_k^{(d)} | \mu_r, \Sigma_r) e^{-\frac{1}{2} \text{Tr}\{(\Sigma_r)^{-1} \Sigma_k^{(d)}\}} \right]^{\pi_k^{(d)} N} \pi_r}{\sum_l \left[ \mathcal{N}(\mu_k^{(d)} | \mu_l, \Sigma_l) e^{-\frac{1}{2} \text{Tr}\{(\Sigma_l)^{-1} \Sigma_k^{(d)}\}} \right]^{\pi_k^{(d)} N} \pi_l},$$

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 parameters of the word-level distribution

$$\begin{aligned} \pi_r^{new} &= \frac{\sum_{(d),k} h_{(d),k}^r}{C \cdot K}, \\ \mu_r^{new} &= \sum_{(d),k} z_{(d),k}^r \mu_k^{(d)}, \quad \text{where } z_{(d),k}^r = \frac{h_{(d),k}^r \pi_k^{(d)}}{\sum_{(d),k} h_{(d),k}^r \pi_k^{(d)}}, \\ \Sigma_r^{new} &= \sum_{(d),k} z_{(d),k}^r \left[ \Sigma_k^{(d)} + (\mu_k^{(d)} - \mu_t)(\mu_k^{(d)} - \mu_t)^T \right]. \end{aligned}$$

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  $D$  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,  $h_{(d),k}^r$ , 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.

## 5 Semantically Labeled Music Data

A preliminary analysis [24] of a semantically-labeled music data set using the models described in Section 3 demonstrated the ability of our system to learn relationships between audio features and corresponding semantic labels. In this initial study semantic information was extracted from song reviews which were mined from the web. However, these results revealed that the signal in this text data was weak and noisy. The semantic labels assigned by a professional reviewer or amateur blogger when describing a piece of music differ from those required to train a statistical model of the semantics of sound. In particular, the data set in [24] was *weakly labeled*: reviewers do not make explicit decisions about whether to include each word in their vocabulary in a natural language song review. Much of the review content refers to social, historical or anecdotal features related to the song, rather than a description of the audio content, making it impossible for a statistical model without such high-level background knowledge to predict these words from the audio alone. Furthermore, artifacts of the text feature extraction procedure meant that the binary document vector often contained misleading features. For example, if a review states that “this song does not rock”, the word “rock” would erroneously be added to the annotation of that song. Nevertheless, the fact that the results from [24] scored significantly above chance demonstrate that the model is capable of learning a relationship between audio and semantic descriptions, wherever such a relationship exists.

To address the shortcomings of relying on semantic data mined from the web, we collected an entirely new set of semantic labels created specifically for a music annotation task. 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.*”). A complete list of the questions used in our data collection survey can be found in [28].

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 [29], one from each artist whose songs are not from the classical genre.

To generate new semantic labels we paid 66 undergraduate music students to annotate our music corpus with the 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 a 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 song playback to encouraging undistracted listening. Subjects 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.

## 5.1 Semantic Features

We expand the set of 135 survey concepts to a set of 237 concepts by mapping all bipolar concepts to two individual concepts. 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 subject to a semantic keyword. For each semantic concept the annotator has supplied a response of +1 or -1 if the user believes the song is or is not indicative of the concept, 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 subjects 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 ‘ground truth’ annotation of binary annotation vectors. To generate binary vectors, we label a song with

a concept if a minimum of two people vote for the concept and there is at least  $[y]_i = .80$  agreement between all subjects. Finally, we prune all concepts that are represented by fewer than five songs. This reduces our set of 237 concepts to a set of 174 concepts.

## 5.2 Music Features

Each song is represented as a *bag-of-feature-vectors*: we extract an unordered set of feature vectors for every song, by extracting one feature vector for each short-time segment of audio data. We compute dynamic Mel-frequency cepstral coefficients (dMFCCs) from each half-overlapping, medium-time ( $\sim 743$  msec) segment of music audio[8]. This results in about 800 52-dimensional feature vectors for a five minute song.

## 6 Semantically Labeled Sound Effects Data

To confirm the general applicability of our model to any type of semantically labeled audio, we also test the system on sound effects. In this case, the data set is 1305 audio clips from the BBC sound effects library and the associated text captions that describe each clip. The vocabulary,  $\mathcal{V}$ , is automatically selected as the 348 words that appear 5 or more times in all the sound effects captions. We represent each caption as a *bag of words*: a set of words  $\mathcal{W}$  that are found in both the review and our vocabulary  $\mathcal{V}$ .

The acoustic features for sound effects are delta cepstrum vectors extracted from each half-overlapping short-time ( $\sim 12$  msec) audio segment [30]. The delta cepstrum vectors are generated by taking the cepstral coefficients of an audio signal and appending to it instantaneous first and second order derivative information. Every 30 seconds of audio content produces about 5000 39-dimensional feature vectors. dMFCC features are not appropriate for sound effects since some audio clips are too short.

## 7 Model evaluation

In this section, we quantitatively evaluate our supervised multi-class naïve Bayes model for audio annotation and retrieval. We find it hard to compare our results to previous work [21, 20] 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). To our knowledge, there has been very little work done on semantic music annotation [17] and virtually no work focused on semantic music retrieval.

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

It is informative to estimate the performance of humans on an annotation task. This is done by holding out a single human annotation from each of the 142 songs that had more than 3 annotations. To evaluate human performance, we compare the held out subjects’ semantic descriptions of songs to the “ground truth” labels obtained from the remaining annotations for those songs. We run a statistically significant number of these tests and report average findings in Table 3 under the heading, “Human.”

## 7.1 Annotation

Using Equation 3, we annotate all test set songs with 10 words and all test set sound effect tracks with 6 words. Annotation performance is measured using mean *per-word* precision and recall. Per-word precision measures the fraction of all predictions of the corresponding word that are actually correct. Per-word recall measures the fraction of all correct annotations with the corresponding word that are actually predicted. More formally, for each word  $w$ ,  $|w_H|$  is the number of tracks that have word  $w$  in the “ground truth” annotation.  $|w_A|$  is the number of tracks that our model 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  $|w_C|/|w_A|$ . 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 the corpus is greater or lesser than the number of words that are output by our system. For example, if our

system outputs 500 words to annotate the 50 test songs from a corpus where the ground truth contains 643 words, mean 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 are displayed in the results tables below under the heading “UpperBnd”.

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 10 best words to annotate each song, our system outputs 144 of the 174 words in the vocabulary.

Table 3 presents quantitative results for music and Table 4 for sound effects. Table 3 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 10-fold cross-validation.

The quantitative results demonstrate that models trained using model averaging and, in particular, mixture hierarchies estimation significantly outperform the random baselines for both data sets. Furthermore, music performance is comparable to and, in many categories, exceeds human consistency. Annotations by a single individual are not necessarily predictive of the average semantic descriptions of the larger population that our model is trained on. For example, a friend’s description of a song may not agree with the average review on a web site, but both have merit. This highlights the need for incorporating weighted semantic descriptions when learning annotation and retrieval systems or for designing systems that are tailored to the preferences of individual users.

## 7.2 Retrieval

For each word  $w_q$ , we rank the test songs in  $\mathcal{S}$  according to Equation 4 and calculate the mean average precision (Mean AP) [15] and the mean area under the receiver operating characteristic (ROC) curve (Mean 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

Table 3: 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 / $ \mathcal{V} $	Model	Precision		Recall	
All Words	10 / 174	Random	0.195	(0.006)	0.068	(0.003)
		Human	0.363	(0.026)	0.141	(0.014)
		UpperBnd	<i>1.000</i>	(0.000)	<i>0.377</i>	(0.009)
		ModelAvg	0.186	(0.010)	0.121	(0.007)
		MixHier	<b>0.440</b>	(0.015)	<b>0.142</b>	(0.004)
Emotion	4 / 36	Random	0.379	(0.012)	0.113	(0.003)
		Human	0.443	(0.051)	0.177	(0.014)
		UpperBnd	<i>0.987</i>	(0.006)	<i>0.393</i>	(0.007)
		ModelAvg	0.410	(0.015)	0.169	(0.007)
		MixHier	<b>0.465</b>	(0.018)	<b>0.193</b>	(0.005)
Genre	2 / 31	Random	0.060	(0.008)	0.077	(0.009)
		Human	0.300	(0.054)	0.330	(0.051)
		UpperBnd	<i>0.676</i>	(0.018)	<i>0.792</i>	(0.015)
		ModelAvg	0.167	(0.012)	0.149	(0.015)
		MixHier	<b>0.278</b>	(0.022)	<b>0.234</b>	(0.017)
Instrumentation	4 / 24	Random	0.147	(0.008)	0.182	(0.013)
		Human	0.390	(0.020)	0.510	(0.040)
		UpperBnd	<i>0.616</i>	(0.018)	<i>0.888</i>	(0.016)
		ModelAvg	0.223	(0.011)	0.323	(0.031)
		MixHier	<b>0.332</b>	(0.017)	<b>0.341</b>	(0.011)
Solo	1 / 9	Random	0.040	(0.010)	0.152	(0.034)
		Human	0.060	(0.034)	0.283	(0.123)
		UpperBnd	<i>0.219</i>	(0.021)	<i>0.785</i>	(0.044)
		ModelAvg	0.051	(0.010)	<b>0.247</b>	(0.042)
		MixHier	<b>0.051</b>	(0.009)	0.220	(0.026)
Usage	2 / 15	Random	0.070	(0.007)	0.146	(0.020)
		Human	0.105	(0.027)	0.174	(0.054)
		UpperBnd	<i>0.353</i>	(0.015)	<i>0.815</i>	(0.023)
		ModelAvg	0.098	(0.010)	0.184	(0.024)
		MixHier	<b>0.145</b>	(0.015)	<b>0.238</b>	(0.021)
Vocal	2 / 16	Random	0.065	(0.003)	0.148	(0.012)
		Human	0.150	(0.040)	0.337	(0.100)
		UpperBnd	<i>0.331</i>	(0.018)	<i>0.783</i>	(0.023)
		ModelAvg	0.115	(0.019)	0.227	(0.019)
		MixHier	<b>0.188</b>	(0.026)	<b>0.277</b>	(0.021)

Table 4: Sound effects annotation results.  $A = 6$ ,  $|\mathcal{V}| = 348$ .

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

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 audio tracks. Columns 4 and 5 of Table 5 show Mean AP and Mean AROC found by averaging each metric over all the words in our vocabulary. As with the annotation results, we see that our models significantly outperform the random baseline and that mixture hierarchies estimation is superior to model averaging for music information retrieval. For sound effects, model averaging outperforms mixture hierarchies. This might be explained by interpreting model averaging as a non-parametric 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.

### 7.3 Comments

Our models significantly outperform the random baseline and, for annotation, are as good at predicting ground truth labels as a human. 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 5, we demonstrate a significant improvement in performance over the same models trained using weakly-labeled text data mined from the web [24] (e.g., music retrieval mean AROC increases from 0.61 to 0.69). 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.

Table 5: Music retrieval results.  $|\mathcal{V}| = 174$ .

Category	$ \mathcal{V} $	Model	MeanAP		MeanAROC	
All Words	174	Random	0.233	(0.004)	0.504	(0.003)
		ModelAvg	0.323	(0.010)	0.633	(0.009)
		MixHier	<b>0.383</b>	(0.007)	<b>0.688</b>	(0.005)
Emotion	36	Random	0.330	(0.006)	0.504	(0.003)
		ModelAvg	0.423	(0.013)	0.634	(0.010)
		MixHier	<b>0.482</b>	(0.009)	<b>0.687</b>	(0.005)
Genre	31	Random	0.132	(0.005)	0.500	(0.005)
		ModelAvg	0.235	(0.019)	0.363	(0.011)
		MixHier	<b>0.320</b>	(0.012)	<b>0.695</b>	(0.007)
Instrumentation	24	Random	0.221	(0.007)	0.502	(0.004)
		ModelAvg	0.314	(0.014)	0.641	(0.010)
		MixHier	<b>0.406</b>	(0.019)	<b>0.695</b>	(0.006)
Solo	9	Random	0.221	(0.007)	0.502	(0.004)
		ModelAvg	0.314	(0.014)	<b>0.641</b>	(0.010)
		MixHier	<b>0.406</b>	(0.019)	0.619	(0.006)
Usage	15	Random	0.145	(0.012)	0.501	(0.005)
		ModelAvg	0.202	(0.016)	0.631	(0.010)
		MixHier	<b>0.217</b>	(0.022)	<b>0.681</b>	(0.006)
Vocal	16	Random	0.137	(0.006)	0.502	(0.004)
		ModelAvg	0.198	(0.020)	0.634	(0.010)
		MixHier	<b>0.254</b>	(0.021)	<b>0.686</b>	(0.007)

Table 6: 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)	<b>0.785</b>	(0.005)
MixHier ( $K = 8, R = 16$ )	<b>0.331</b>	(0.008)	0.784	(0.006)

## 8 Discussion and Future Work

By collecting a cleanly annotated data set of songs and developing a useful and efficient parameter estimation algorithm (weighted mixture hierarchies EM), we have developed an automatic music annotation and retrieval system that significantly outperforms the system presented in [24]. While direct comparison is impossible since different vocabularies and music were used, both qualitative and quantitative results suggest that end user experience has been greatly improved. The compact *semantic multinomial* representation of a song, which is generated during annotation, is useful for related music information tasks such as ‘retrieval-by-semantic-similarity’ [12, 31].

We have shown that for estimating the parameters of a GMM, the mixture hierarchies EM algorithm is more efficient than direct estimation or model averaging and, for music, it produces better results. 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 bag-of-feature-vectors that are extracted from the audio signal. We then learn a mixture from 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 maximums, is more likely to converge to a better density estimate.

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 SMC-NB model (a set of GMMs) is one of many models that have been developed for image annotation [14, 15]. 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 track) information that can be extracted from an audio track. Future research will involve exploring models, such as hidden Markov models, that explicitly model the longer-term temporal aspects of music.

Additional future work involves modeling individual users (or subsets of users) with *user-specific* models. For example, during data collection, we had one user annotate 200 of the 500 songs in our data set. A preliminary study showed that we were better able to model some words (especially the ‘usage’ words) for this user using the 200 songs he annotated compared against models trained using all 500 songs. This is not surprising since we would expect an individual to be *self-consistent* when annotating songs with subjective concepts.

Lastly, the text-based music retrieval framework presented here naturally extends to handling multi-word queries. We can imagine representing the query string as a semantic multinomial and ranking tracks based on the Kullback-Leibler distance between the query distribution and a track’s semantic multinomial distribution. Alternatively, we could combine the semantic weights for all songs associated with the multiple query words and then learn a ‘query-level’ distribution using the weighted mixture hierarchies algorithm. There are also a number of heuristic methods that involve com-

binning the results of single-word queries. This future work will also be useful for dealing with *heterogeneous queries* in which a user specifies a query with both words and audio examples.

## Acknowledgements

We would like to thank Antoni Chan, Arshia Cont, Garrison W. Cottrell, Shlomo Dubnov, Charles Elkan, Lawrence Saul, and Nuno Vasconcelos for their helpful comments. Luke Barrington and Douglas Turnbull receive support from NSF IGERT fellowship DGE-0333451.

## References

- [1] G. Carneiro and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. *IEEE CVPR*, 2005.
- [2] N. Vasconcelos. Image indexing with mixture hierarchies. *IEEE CVPR*, pages 3–10, 2001.
- [3] Masataka Goto and Keiji Hirata. Recent studies on music information processing. *Acoustical Science and Technology*, 25(4):419–425, 2004.
- [4] F. Vignoli and S. Pauws. A music retrieval system based on user-driven similarity and its evaluation. *ISMIR*, 2005.
- [5] R. B. Dannenberg and N. Hu. Understanding search performance in query-by-humming systems. *ISMIR*, 2004.
- [6] George Tzanetakis Ajay Kapur, Manjinder Benning. Query by beatboxing: Music information retrieval for the dj. *ISMIR*, 2004.
- [7] Gunnar Eisenberg, Jan-Mark Batke, and Thomas Sikora. Beatbank - an mpeg-7 compliant query by tapping system. *Audio Engineering Society Convention*, 2004.
- [8] M. F. McKinney and J. Breebaart. Features for audio and music classification. *ISMIR*, 2003.
- [9] Tao Li and George Tzanetakis. Factors in automatic musical genre classification of audio signals. *IEEE WASPAA*, 2003.

- [10] Slim Essid, Gaël Richard, and Bertrand David. Inferring efficient hierarchical taxonomies for music information retrieval tasks: Application to musical instruments. *ISMIR*, 2005.
- [11] Francois Pachet and Daniel Cazaly. A taxonomy of musical genres. *RIAO*, 2000.
- [12] Adam Berenzweig, Beth Logan, Daniel P.W. Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 2004.
- [13] D. Forsyth and M. Fleck. Body plans. *IEEE CVPR*, 1997.
- [14] D. M. Blei and M. I. Jordan. Modeling annotated data. *ACM SIGIR*, 2003.
- [15] S. L. Feng, R. Manmatha, and Victor Lavrenko. Multiple bernoulli relevance models for image and video annotation. *IEEE CVPR*, 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. Semantic-audio retrieval. *IEEE ICASSP*, 2002.
- [22] M. Slaney. Mixtures of probability experts for audio retrieval and indexing. *IEEE Multimedia and Expo*, 2002.
- [23] D. Reynolds, T.F. Quatieri, and R.B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.
- [24] Douglas Turnbull, Luke Barrington, and Gert Lanckriet. Modelling music and words using a multi-class naïve bayes approach. *ISMIR*, 2006.
- [25] Cory McKay, Daniel McEnnis, and Ichiro Fujinaga. A large publicly accessible prototype audio database for music research. *ISMIR*, 2006.

- [26] Janto Skowronek, Martin McKinney, and Steven ven de Par. Ground-truth for automatic music mood classification. *ISMIR*, 2006.
- [27] Xiao Hu, J. Stephen Downie, and Andreas F. Ehmann. Exploiting recommended usage metadata: Exploratory analyses. *ISMIR*, 2006.
- [28] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. UCSD technical report: Semantic annotation and retrieval of music and sound effects. Technical report, 2006.
- [29] Magnatune: free MP3 music and music licensing. Magnatune. <http://www.magnatune.com>.
- [30] C. R. Buchanan. Semantic-based audio recognition and retrieval. Master's thesis, School of Informatics, University of Edinburgh, 2005.
- [31] Luke Barrington, Antoni Chan, Douglas Turnbull, and Gert Lanckriet. UCSD technical report: Query by semantic example. Technical report, 2006.