Multimedia Systems manuscript No.
(will be inserted by the editor)

Michael I. Mandel · Graham E. Poliner · Daniel P.W. Ellis

Support Vector Machine Active Learning for Music Retrieval

Abstract Searching and organizing growing digital music collections requires a computational model of music similarity. This paper describes a system for performing flexible music similarity queries using SVM active learning. We evaluated the success of our system by classifying 1210 pop songs according to *mood* and *style* (from an online music guide) and by the performing artist. In comparing a number of representations for songs, we found the statistics of mel-frequency cepstral coefficients to perform best in precision-at-20 comparisons. We also show that by choosing training examples intelligently, active learning requires half as many labeled examples to achieve the same accuracy as a standard scheme.

Keywords Support vector machines · active learning · music classification

1 Introduction

As the size of digital music libraries grows, identifying music to listen to from a personal collection or to buy from an online retailer becomes increasingly difficult. Since finding songs that are similar to each other is time consuming and each user has unique opinions, we would like to create a flexible, open-ended approach to music retrieval.

Our solution is to use relevance feedback, specifically Support Vector Machine (SVM) active learning, to learn a classifier for each query. A search is then a mapping from low level audio features to higher level concepts, customized to each user. To begin a search, the user presents the system with one or more examples of songs of interest or "seed" songs. The system then iterates between training a new classifier on labeled songs and soliciting new labels from the user for informative examples. Search proceeds quickly, and at every stage the system supplies its best estimate of appropriate songs. Since it takes a significant amount of time to listen to each song returned by a search, our system attempts to minimize the number of songs that a user must label for a query.

Active learning has two main advantages over conventional SVM classification. First, by presenting the user with and training on the most informative songs, the algorithm can achieve the same classification performance with fewer labeled examples. Second, by allowing the user to dynamically label the set of instances, a single system may perform any number of classification and retrieval tasks using the same precomputed set of features and classifier framework. For example, the system may be used to search for female artists, happy songs, or psychedelic music.

This flexibility depends on the information the acoustic feature vectors make explicit, leading to our comparison of various features for these tasks. On a collection of 1210 pop songs, the features that produced the most accurate artist classifier were the statistics of MFCCs calculated over entire songs. These same features achieved the best precision on the top 20 ranked results for music categorizations culled from allmusic.com.

We have also developed an automatic tester for our SVM active learning system, showing that an SVM active learner trained with half as many examples can perform as well as a normal SVM or, alternately, can increase its precision by ten percentage points with the same number of labeled examples.

1.1 Music Similarity

The idea of judging the similarity of music by a direct comparison of audio content was proposed by [13]. For computational simplicity, his system used discrete distributions over

This work was supported by the Fu Foundation School of Engineering and Applied Science via a Presidential Fellowship, the Columbia Academic Quality Fund, and the National Science Foundation (NSF) under Grant No. IIS-0238301. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

M. Mandel · G. Poliner · D. Ellis
Department of Electrical Engineering, 1312 S.W. Mudd,
500 West 120th Street, New York, NY 10027
Tel.: 212-854-3105, Fax: 212-932-9421
E-mail: {mim,graham,dpwe}@ee.columbia.edu

vector-quantizer symbols and was evaluated on a database of a few hundred 7-second excerpts. [21] was able to compare continuous distributions over thousands of complete songs, using the Earth Mover's Distance to calculate dissimilarity between mixtures of Gaussians. There have followed a number of papers refining the features, distance measures, and evaluation techniques, including our own work [3–5,12]; [2] provides an excellent review, where they characterize these approaches as "timbre similarity" to emphasize that they are based on distributions of short-term features and ignore most temporal structure.

Particular tasks for music similarity are largely defined by the availability of ground truth. [30] popularized the use of genre classification, whereas [31] proposed artist identification as a more interesting task, with the attraction of having largely unambiguous ground-truth. Here, we consider versions of both these tasks.

Most work has sought to define a low-dimensional feature space in which similarity is simply Euclidean distance, or measured by the overlap of feature distributions. Here, we use a more complex classifier (the SVM) on top of an implicit feature space of very high dimension; the related regularized least squares classifier was used for music similarity by [32]. The Fisher Kernel technique we use was introduced for audio classification by [22].

1.2 Relevance Feedback

While the idea of relevance feedback had been around for a number of years, [28] first described using support vector machines for active learning. [29] discussed the *version space* of all possible hyperplanes consistent with labeled data along with methods for reducing it as quickly as possible to facilitate active learning. [27,28] applied SVM active learning to text and image retrieval.

Recently, improvements in SVM active learning have been made in the areas of sample selection, scalability, and multimodal search. [7] described a number of methods for selecting the most informative database items to label, with the *angle diversity* selection criterion producing the best active retrieval results. The same paper describes a multimodal search in which the user can limit the pool of images to search by supplying relevant keywords. In order to scale to large databases, [19] describes methods for disambiguating search concepts and using a hierarchical data structure to more efficiently find data points.

[15,16] used relevance feedback for music retrieval, but their approach suffers from some limitations. Their system was based on the TreeQ vector quantization from [13], with which they must re-quantize the entire music database for each query. Relevance feedback was incorporated into the model by modifying the quantization weights of desired vectors. Our approach calculates the features of a song offline and uses SVM active learning, which has a strong theoretical justification, to incorporate user feedback.

1. Seed the search with representative song(s).
2. Acquire initial negative examples by e.g. presenting randomly selected songs for labeling
3. Train an SVM on all labeled examples
4. Present the user with the most relevant songs (those with the greatest positive distance to the decision boundary)
5. If the user wishes to refine the search further, present the most informative songs (those closest to the decision boundary) for labeling and repeat 3-5.

Fig. 1 Summary of SVM active learning algorithm.

2 SVM Active Retrieval

SVM active learning combines the maximum margin classification of SVMs with ideas from relevance feedback. See Figure 1 for a summary of the active learning algorithm, which lends itself to both direct user interaction and automated testing.

2.1 Support Vector Machines

The support vector machine (SVM) is a supervised classification system that minimizes an upper bound on its expected error. It attempts to find the hyperplane separating two classes of data that will generalize best to future data. Such a hyperplane is the so called maximum margin hyperplane, which maximizes the distance to the closest points from each class.

More concretely, given data points $\{\mathbf{X}_0, \ldots \mathbf{X}_N\}$ and class labels $\{y_0, \ldots y_N\}$, $y_i \in \{-1, 1\}$, any hyperplane separating the two data classes has the form

$$y_i(\mathbf{w}^T \mathbf{X}_i + b) > 0 \; \forall i. \tag{1}$$

Let $\{\mathbf{w}_k\}$ be the set of all such hyperplanes. The maximum margin hyperplane is defined by

$$\mathbf{w} = \sum_{i=0}^{N} \alpha_i y_i \mathbf{X}_i, \tag{2}$$

and b is set by the Karush Kuhn Tucker conditions [6] where the $\{\alpha_0, \alpha_1, \ldots, \alpha_N\}$ maximize

$$L_D = \sum_{i=0}^{N} \alpha_i - \frac{1}{2} \sum_{i=0}^{N} \sum_{j=0}^{N} \alpha_i \alpha_j y_i y_j \mathbf{X}_i^T \mathbf{X}_j, \tag{3}$$

subject to

$$\sum_{i=0}^{N} \alpha_i y_i = 0 \quad \alpha_i \geq 0 \; \forall i. \tag{4}$$

For linearly separable data, only a subset of the α_is will be non-zero. These points are called the *support vectors* and all classification performed by the SVM depends on only these points and no others. Thus, an identical SVM would result from a training set that omitted all of the remaining examples. This makes SVMs an attractive complement to relevance feedback: if the feedback system can accurately identify the critical samples that will become the support vectors, training time and labeling effort can, in the best case, be reduced drastically with no impact on classifier accuracy.

Since the data points $\mathbf{X}$ only enter calculations via dot products, one can transform them to another *feature space* via a function $\Phi(\mathbf{X})$. The representation of the data in this feature space need never be explicitly calculated if there is an appropriate Mercer kernel operator for which

$$K(\mathbf{X}_i, \mathbf{X}_j) = \Phi(\mathbf{X}_i) \cdot \Phi(\mathbf{X}_j). \tag{5}$$

Data that is not linearly separable in the original space, may become separable in this feature space. In our implementation, we selected a radial basis function (RBF) kernel

$$K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\gamma D^2(\mathbf{X}_i, \mathbf{X}_j)}, \tag{6}$$

Where $D^2(\mathbf{X}_i, \mathbf{X}_j)$ could be any distance function. See Table 1 for a list of the distance functions used in our experiments and Section 4 for a discussion of them. Thus the space of possible classifier functions consists of linear combinations of weighted Gaussians around key training instances [9].

2.2 Active Learning

In an active learning system, the user becomes an integral part of the learning and classification process. As opposed to conventional ("passive") classification where a classifier is trained on a large pool of randomly selected labeled data, an active learning system asks the user to label only those instances that would be most informative to classification. Learning proceeds based on the feedback of the user and relevant responses are determined by the individual user's preferences and interpretations.

The duality between points and hyperplanes in feature space and *parameter space* enables SVM active learning. Notice that that Equation (1) can be interpreted with $\mathbf{X}_i$ as points and $\mathbf{w}_k$ as hyperplanes normals, but it can also be interpreted with $\mathbf{w}_k$ as points and $\mathbf{X}_i$ as normals. This second interpretation of the equation is known as parameter space. Within parameter space, the set $\{\mathbf{w}_k\}$ is known as *version space*, a convex region bounded by the hyperplanes defined by the $\mathbf{X}_i$. Finding the maximum margin hyperplane in the original space is equivalent to finding the point at the center of the largest hypersphere in version space.

The user's desired classifier corresponds to a point in parameter space that the SVM active learner would like to locate as quickly as possible. Labeled data points place constraints in parameter space, reducing the size of the version space. The fastest way to shrink the version space is to halve it with each labeled example, finding the desired classifier most efficiently. When the version space is nearly spherical, the most informative point to label is that point closest to the center of the sphere, i.e. closest to the decision boundary [29]. In pathological cases, this is not true, nor is it true that the greedy strategy of selecting all of the points closest to the decision boundary shrinks the version space most quickly.

In practice, however, we find that these simple strategies perform well. [7] described a number of heuristics for finding the most informative points to label and determines that the *angle diversity* strategy performs best. Angle diversity balances closeness to the decision boundary with coverage of the feature space, while avoiding extra classifier retrainings. The sparseness of our songs in feature space might obviate the need for an explicit enforcement of diversity in the examples chosen for labeling.

Since the user only seeds the system with positive examples, the first set of songs to be labeled cannot be chosen by a classifier. While there are many methods for choosing these first songs, we find the simplest, random selection, to work well in practice. Since positive examples are relatively rare in the database, many of the randomly chosen examples will be negative. Other selection methods include choosing songs that maximally cover the feature space, songs farthest from the seeds, songs closest to the seeds, and so forth.

Any active retrieval system may suffer from certain limitations. The small training set means that performance could suffer from poor seeding or insufficient sampling of the data.

3 Audio Feature Components

Since the flexibility of an SVM active learner depends on the descriptive power of the features on which it operates, we experimented with a number of features for song representation. All of these features have the property that they reduce every song, regardless of its original length, into a fixed-size vector. They are all based on Gaussian mixture models (GMMs) of mel-frequency cepstral coefficients (MFCCs).

3.1 Mel-Frequency Cepstral Coefficients

MFCCs are a short-time spectral decomposition of audio that convey the general frequency characteristics important to human hearing. While originally developed to decouple vocal excitation from vocal tract shape for automatic speech recognition [24], they have found applications in other auditory domains including music retrieval [13, 20].

In order to calculate MFCCs, the signal is first broken into overlapping frames, each approximately 25ms long, a time scale at which the signal is assumed to be stationary. The log-magnitude of the discrete Fourier transform of each window is warped to the Mel frequency scale, imitating human frequency and amplitude sensitivity. The inverse discrete cosine transform decorrelates these "auditory spectra" and the so called "high time" portion of the signal, corresponding to fine spectral detail, is discarded, leaving only the general spectral shape. The MFCCs we calculated for the songs in our database contain 13 coefficients each and,

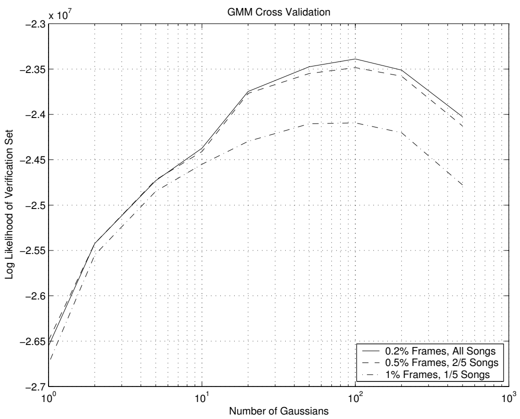


Fig. 2 Cross-validation of Gaussian mixture models using different numbers of Gaussians and different training sets.

at a rate of 100 per second, a typical five minute song will generate 30,000 temporal frames.

3.2 Gaussian Mixture Models

Certain features were based on a single Gaussian mixture model trained on samples from all of the songs in the training set, while others built individual GMMs for each song. In the former case, we trained a GMM with diagonal covariances on 0.2% of the MFCC frames selected at random from every song.

In order to estimate the correct number of Gaussians for our mixture, we measured the log-likelihood of models with different numbers of Gaussians on our validation dataset. Results of this test can be seen in Figure 2. The three lines in that figure correspond to three different methods for collecting training samples. While keeping the total number of samples the same, the number of songs sampled and the number of samples per song was varied. From the figure it is clear that the model that best fits the data is a mixture of somewhere between 50 and 100 Gaussians, independent of the number of songs used in training. This result probably does not hold beyond the case of pop music MFCCs modeled with Gaussian Mixtures, but it is interesting to see such a consistent result even for this case. The jumps in likelihood for GMMs with more Gaussians are due to overfitting of certain sound files that were broken in a characteristic way.

4 Audio Features

See Table 1 for a summary of features evaluated in the experiments. Each feature uses its own distance function in the RBF kernel of Equation (6). We go into detail on each of them in the following sections. The first three use Gaussian models trained on individual songs, while second three relate each song to a global Gaussian mixture model of the entire corpus. Both of these approaches only model stationary spectral characteristics of music, averaged across time, and ignore the higher-order temporal structure.

In the following sections, $\mathbf{X}$ denotes matrices of MFCCs, x_t denotes individual MFCC frames. Songs are indexed by i and j, GMM components by k. MFCC frames are indexed in time by t and MFCC frames drawn from a probability distribution are indexed by n.

4.1 MFCC Statistics

This first feature is based on the mean and covariance of the MFCC frames of individual songs. In fact, it models a song as just a single Gaussian, but uses a non-probabilistic distance measure between songs. The feature itself is the concatenation of the mean and the unwrapped covariance matrix of a song's MFCC frames. These features are commonly used in speech processing e.g. for segmenting a recording according to speaker turns [8, 14]. These low order statistics can only represent simple relationships between MFCCs, but our experiments show that they contain much useful information about artists, styles, and moods.

The feature vector is shown in Table 1, where the $\text{vec}(\cdot)$ function unwraps or rasterizes an $N \times N$ matrix into a $N^2 \times 1$ vector. Feature vectors are compared to one another using a Mahalanobis distance, where the Σ_μ and Σ_Σ variables are diagonal matrices containing the variances of the feature vectors over all of the songs.

4.2 Song GMMs

The second feature listed in Table 1, models songs as single Gaussians. The maximum likelihood Gaussian describing the MFCC frames of a song is parameterized by the sample mean and sample covariance. To measure the distance between two songs, one can calculate the Kullback-Leibler (KL) divergence between the two Gaussians. While the KL divergence is not a true distance measure, the symmetrized KL divergence is, and can be used in the RBF kernel of Equation (6) [23].

For two distributions, $p(x)$ and $q(x)$, the KL divergences is defined as,

$$KL(p\,||\,q) \equiv \int p(x) \log \frac{p(x)}{q(x)}\, dx = E_p \left\{ \log \frac{p(X)}{q(X)} \right\}. \tag{7}$$

There is a closed form for the KL divergence between two Gaussians, $p(x) = \mathcal{N}(x; \mu_p, \Sigma_p)$ and $q(x) = \mathcal{N}(x; \mu_q, \Sigma_q)$, [25]

$$\begin{aligned} 2KL(p\,||\,q) = &\log \frac{|\Sigma_q|}{|\Sigma_p|} + Tr(\Sigma_q^{-1} \Sigma_p) \\ &+ (\mu_p - \mu_q)^T \Sigma_q^{-1} (\mu_p - \mu_q) - d, \end{aligned} \tag{8}$$

Where d is the dimensionality of the Gaussians. The symmetrized KL divergence shown in Table 1 is simply

$$D^2(\mathbf{X}_i, \mathbf{X}_j) = KL(\mathbf{X}_i\,||\,\mathbf{X}_j) + KL(\mathbf{X}_j\,||\,\mathbf{X}_i) \tag{9}$$

Table 1 Summary of the features compared in the experiments. See text for explanation of variables.

GMM over	Feature	Parameters	Representation	Distance measure $D^2(\mathbf{X}_i, \mathbf{X}_j)$
	MFCC Stats	104	$[\mu^T \ \mathrm{vec}(\Sigma)^T]$	$(\mu_i - \mu_j)^T \Sigma_\mu^{-1} (\mu_i - \mu_j) + \mathrm{vec}(\Sigma_i - \Sigma_j)^T \Sigma_\Sigma^{-1} \mathrm{vec}(\Sigma_i - \Sigma_j)$
Song	KL 1G	104	μ, Σ	$tr(\Sigma_i^{-1}\Sigma_j + \Sigma_j^{-1}\Sigma_i) + (\mu_i - \mu_j)^T (\Sigma_i^{-1} + \Sigma_j^{-1})(\mu_i - \mu_j) - 2d$
	KL 20G	520	$\{\mu_k, \Sigma_k\}_{k=1\ldots20}$	$\frac{1}{N}\sum_{n=1}^N \log \frac{p_i(\mathbf{x}_{ni})}{p_j(\mathbf{x}_{ni})} + \frac{1}{N}\sum_{n=1}^N \log \frac{p_j(\mathbf{x}_{nj})}{p_i(\mathbf{x}_{nj})}$
	GMM Posteriors	100	$\{\frac{1}{T}\sum_{t=1}^T \log p(k \mid \mathbf{x}_t)\}_{k=1\ldots50}$	$\sum_{k=1}^{50} \log^2 \frac{p(X_i \mid k)^{1/T_i}}{p(X_j \mid k)^{1/T_j}}$
Corpus	Fisher	650	$\{\nabla_{\mu_k}\}_{k=1\ldots50}$	$\sum_{k=1}^{50} \left[\nabla_{\mu_k} \log p(\mathbf{X}_i \mid \mu_k) - \nabla_{\mu_k} \log p(\mathbf{X}_j \mid \mu_k)\right]^2$
	Fisher Mag	50	$\{\lvert\nabla_{\mu_k}\rvert\}_{k=1\ldots50}$	$\sum_{k=1}^{50} \left[\lvert\nabla_{\mu_k} \log p(\mathbf{X}_i \mid \mu_k)\rvert - \lvert\nabla_{\mu_k} \log p(\mathbf{X}_j \mid \mu_k)\rvert\right]^2$

The third feature models songs as mixture of Gaussians learned using the EM algorithm and still compares them using the KL divergence. Unfortunately, there is no closed form for the KL divergence between GMMs, so it must be approximated using Monte Carlo methods. The expectation of a function over a distribution, $p(x)$, can be approximated by drawing samples from $p(x)$ and averaging the values of the function at those points. In this case, by drawing samples $x_1, \ldots, x_N \sim p(x)$, we can approximate

$$E_p \left\{ \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \right\} \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i)}. \tag{10}$$

The distance function shown in Table 1 for the "KL 20G" features is the symmetric version of this expectation, where appropriate functions are calculated over N samples from each distribution. We used the Kernel Density Estimation toolbox from [17] for these calculations. As the number of samples used for each calculation grows, variance of the KL divergence estimate shrinks. We use $N = 2500$ samples for each distance estimate to balance computation time and accuracy.

4.3 Anchor Posteriors

The fourth feature listed in Table 1 compares each song to the GMM modeling our entire music corpus. If the Gaussians of the global GMM correspond to clusters of related sounds, one can characterize a song by the probability it came from each of these clusters. This feature corresponds to measuring the posterior probability of each Gaussian in the mixture, given the frames from each song. To calculate the posterior over the whole song from the posteriors for each frame,

$$P(k \mid \mathbf{X}) \propto p(\mathbf{X} \mid k) P(k) = P(k) \prod_{t=1}^{T} p(\mathbf{x}_t \mid k) \tag{11}$$

This feature tends to saturate, generating a nonzero posterior for only a single Gaussian. In order to prevent this saturation, we take the geometric mean of the frame probabilities instead of the product. This does not give the true class posteriors, but only a "softened" version of them

$$f(k) = P(k) \prod_{t=1}^{T} p(\mathbf{x}_t \mid k)^{1/T} \propto \prod_{t=1}^{T} p(k \mid \mathbf{x}_t)^{1/T}. \tag{12}$$

Since they are not proper probability functions, there is little justification for comparing them with anything but the Euclidean distance.

4.4 Fisher Kernel

The final two features are based on the Fisher kernel, which [18] described as a method for summarizing the influence of the parameters of a generative model on a collection of samples from that model. In this case, the parameters we consider are the means of the Gaussians in the global GMM. This process describes each song by the partial derivatives of the log likelihood of the song with respect to each Gaussian mean. From [22],

$$\nabla_{\mu_k} \log P(\mathbf{X} \mid \mu_k) = \sum_{t=1}^{T} P(k \mid \mathbf{x}_t) \Sigma_k^{-1} (\mathbf{x}_t - \mu_k). \tag{13}$$

where $P(k \mid \mathbf{x}_t)$ is the posterior probability of the kth Gaussian in the mixture given MFCC frame $\mathbf{x}_t$, and μ_k and Σ_k are the mean and variance of the kth Gaussian. This process then reduces arbitrarily sized songs to 650 dimensional feature vectors (50 means with 13 dimensions each).

Since the Fisher kernel is a gradient, it measures the partial derivative with respect to changes in each dimension of each Gaussian's mean. A more compact feature is the magnitude of the gradient with respect to each Gaussian's mean. While the full Fisher kernel creates a 650 dimensional vector, the Fisher kernel Magnitude is only 50 dimensional.

5 Experiments

In order to thoroughly test the SVM active music retrieval system, we compared the various features to one another and, using the best feature, examined the relationship between precision at 20 and number of examples labeled per active retrieval round.

5.1 Dataset

We ran our experiments on a subset of the *uspop2002* collection [5,11]. To avoid the so called "producer effect" [31]

Table 2 The moods and styles with the most songs

Mood	Songs	Style	Songs
Rousing	527	Pop/Rock	730
Energetic	387	Album Rock	466
Playful	381	Hard Rock	323
Fun	378	Adult Contemporary	246
Passionate	364	Rock & Roll	226

in which songs from the same album share overall spectral characteristics that could overwhelm any similarities between albums, we designated all of the songs from an album as training, testing, or validation. To be able to separate albums in this way, we chose artists who had at least five albums in *uspop2002*, three albums for training and two for testing, each with at least eight tracks. The validation set was made up of any albums the selected artists had in *uspop2002* in addition to those five and used for tuning model parameters. In total there were 18 artists (out of 400) with enough albums, see Table 5 for a complete list of the artists and albums included in our experiments. In total, we used 90 albums by those 18 artists, containing 1210 songs divided into 656 training, 451 testing, and 103 validation songs.

5.2 Evaluation

Since the goal of SVM active learning is to quickly learn an arbitrary classification, any binary categorization of songs can be used as ground truth. The categories we tested our system on were AMG moods, AMG styles, and artists.

The All Music Guide (AMG) is a website and book that reviews, rates, and categorizes music and musicians [1]. Two of our ground truth categorizations came from AMG, the "mood" and "style" of the music. In their glossary, AMG defines moods as "adjectives that describe the sound and feel of a song, album, or overall body of work," for example acerbic, campy, cerebral, hypnotic, rollicking, rustic, silly, and sleazy. While AMG never explicitly defines them, styles are sub-genre categories such as "punk-pop," "prog-rock/art rock," and "speed metal." In our experiments, we used styles and moods that included 50 or more songs, which amounted to 32 styles and 100 moods. See Table 2 for a list of the most popular moods and styles.

Due to the diversity of the 18 artists chosen for evaluation, some moods contain identical categorizations of our music, leaving 78 distinct moods out of the original 100. For the same reason, 12 of the 32 styles contain all of the work of only a single artist.

While AMG in general only assigns moods and styles to albums and artists, for the purposes of our testing, we assumed that all of the songs on an album could be described by the same moods and styles, namely those attributed to that album. This assumption does not necessarily hold, for example with a ballad on an otherwise upbeat album. We are looking into ways of inferring these sorts of labels for individual songs from collections of album labels and a measure of acoustic similarity.

Artist identification is the task of identifying the performer of a song given only the audio of that song. While a song can have many styles and moods, it can have only one artist, making this the ground truth of choice for our N-way classification test of the various feature sets. Note that a system based on this work, using conventional SVM classification of single Gaussian KL divergence, outperformed all other artist identification systems at an international competition, MIREX 2005 [10].

5.3 Experiments

The first experiment compared the features on passive artist, mood, and style retrieval to determine if one clearly dominated the others. For the artist ground truth, the system predicted the unique performing artist (out of the possible 18) for each song in the test set, after training on the entire training set. Instead of a binary SVM, it used a DAGSVM to perform the multi-class learning and classification [26]. We provide these results to compare against other authors' systems and to compare features to one another, but they are not directly applicable to the SVM active learning task, which only learns one binary categorization at a time.

For the mood and style ground truth and for the active retrieval tasks, we evaluated the success of retrieval by the precision on the top 20 songs returned from the test set. In order to rank songs, they were sorted by their distance from the decision boundary, as in [27]. Precision-at-20 focuses on the positive examples of each class, because for sparse categories a classifier that labels everything as negative is quite accurate, but neither helpful in retrieving music nor interesting. Scores are aggregated over all categories in a task by taking the mean, for example, the mood score is the mean of the scores on all of the individual mood.

One justification for this evaluation metric is that when searching large databases, users would like the first results to be most relevant (precision), but do not care whether they see all of the relevant examples (recall). We chose the number 20 because the minimum number of songs in each ground truth category was 50, and the training set contains roughly 40% of the songs, giving a minimum of approximately 20 correct results in each test category. This threshold is of course adjustable and one may vary the scale of the measured performance numbers by adjusting it. It also happens that these features' precision-at-20 scores are quite distinct from one another, facilitating meaningful comparison.

The second experiment compares different sized training sets in each round of active learning on the best-performing features, MFCC Statistics. Active learning should require fewer labeled examples to achieve the same accuracy as passive learning because it chooses more informative examples to be labeled first. To measure performance, we compared mean precision on the top 20 results on the same separate test albums.

In this experiment we compared five different training group sizes. In each trial, an active learner was randomly

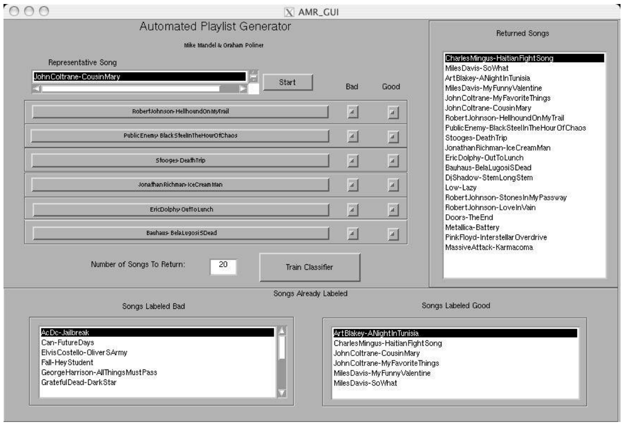


Fig. 3 Active Learning Graphical User Interface.

seeded with 5 elements from within the class, corresponding to a user supplying initial exemplars. The learner then performed simulated relevance feedback with 2, 5, 10, and 20 songs per round. A final classifier performed only one round of learning with 50 examples, equivalent to conventional SVM learning. The simulations stopped once the learner had labeled 50 results so that the different training sets could be compared.

5.4 User Interface

In addition to testing the system with fixed queries, we also developed a graphical interface for users to interact with the system in real time with real queries. A few colleagues were encouraged to evaluate the system (on a different database of songs) with queries of their choosing, including jazz, rap, rock, punk, female vocalists, etc.

The graphical user interface is displayed in Figure 3. The user selects a representative seed song to begin the search. The system presents six songs to label as similar or dissimilar to the seed song according to the categorization the user has in mind. A song may be left unlabeled, in which case it will not affect the classifier, but will be excluded from future labeling. Labeled songs are displayed at the bottom of the interface, and the best ranked songs are displayed in the list to the right. At any time, the user may click on a song to hear a representative segment of it. After each classification round, the user is presented with six new songs to label and may perform the process as many times as desired.

5.5 Results

The results of the feature comparison experiment can be seen in Table 3. In general, the MFCC statistics outperform the other features. In the mood identification task, the Fisher kernel's precision-at-20 is slightly higher, but the results are quite comparable. The single Gaussian KL divergence features worked well for multi-class classification, but less well

Table 3 Comparison of various audio features: accuracy on 18-way artist classification and precision-at-20 for mood and style identification.

Feature	Accuracy Artist 18-way	Precision-at-20 Mood ID	Style ID
MFCC Stats	.682	.497	.755
Fisher Kernel	.543	.508	.694
KL 1G	.640	.429	.666
Fisher Ker Mag	.398	.387	.584
KL 20G	.386	.343	.495
GMM Posterior	.319	.376	.463

Table 4 Precision-at-20 on test set of classifiers trained with different numbers of examples per round or conventional (passive) training, all trained with 50 examples total.

	Examples per round				
Ground Truth	2	5	10	20	Conv.
Style	.691	.677	.655	.642	.601
Artist	.659	.667	.637	.629	.571
Mood	.444	.452	.431	.395	.405

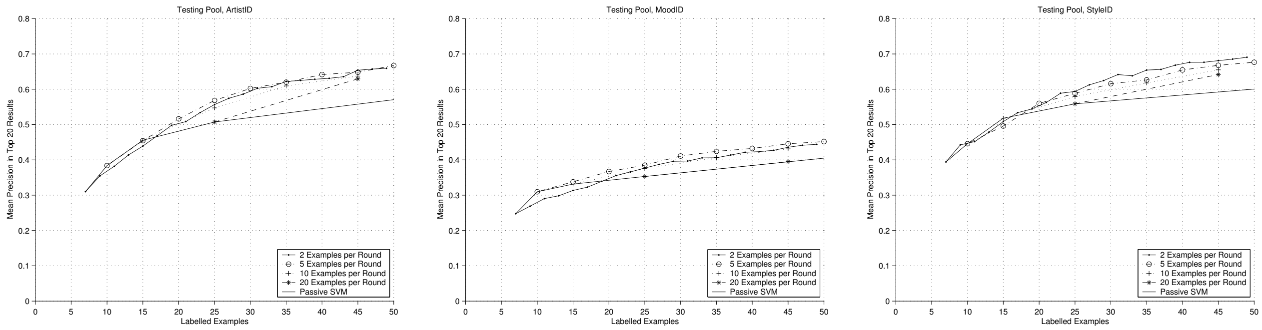
for the binary tasks which are more relevant to active learning.

The results of the active retrieval experiments can be seen in Figure 4. The figure shows that, as we expected, the quality of the classifier depends on the number of rounds of relevance feedback, not only on the absolute number of labeled examples. Specifically, a larger number of retrainings with fewer new labels elicited per cycle leads to a better classifier, since there are more opportunities for the system to choose the examples that will be most helpful in refining the classifier. This shows the power of active learning to select informative examples for labeling. Notice that the classifiers all perform at about the same precision below 15 labeled examples, with the smaller examples-per-round systems actually performing worse than the larger ones. Since the learner is seeded with five positive examples, it may take the smaller sample size systems a few rounds of feedback before a reasonable model of the negative examples can be built.

Comparing the ground truth sets to one another, it appears that the system performs best on the style identification task, achieving a maximum mean precision-at-20 of 0.691 on the test set, only slightly worse than the conventional SVM trained on the entire training set which requires more than 13 times as many labels. See Table 4 for a full listing of the precision-at-20 of all of the classifiers on all of the datasets after labeling 50 examples. On all of the ground truth sets, the active learner can achieve the same mean precision-at-20 with only 20 labeled examples that a conventional SVM achieves with 50.

6 Discussion and Future Work

As expected, labeling more songs per round suffers from diminishing returns; performance depends most heavily on the number of rounds of active learning instead of the number of



(a) (b) (c)

Fig. 4 Performance increase due to active learning for (a) artist identification, (b) mood classification, and (c) style classification. The plots show the mean precision in the top 20 results over the test set as the number of examples per round is varied. The solid line without symbols shows the performance of conventional SVMs trained on the same number of examples.

labeled examples. This result is a product of the suboptimal division of the version space when labeling multiple songs simultaneously.

Small feedback sets, however, do suffer from the initial lack of negative examples. Using few training examples per round of feedback can actually hurt performance initially because the classifier has trouble identifying examples that would be most discriminative to label. It might be advantageous, then, to begin training on a larger number of examples – perhaps just for the "special" first round – and then, once enough negative examples have been found, to reduce the size of the training sets in order to increase the speed of learning.

It is also interesting that the KL divergence features did not perform as well as either the Fisher kernel or the MFCC statistics. This is especially surprising because KL divergence between single Gaussians uses exactly the same feature vector to characterize each song as MFCC statistics and is more mathematically justified. Even more surprising is the performance degradation of GMMs with 20 components as compared to single Gaussians. This discrepancy could be due to inadequate Monte Carlo sampling when measuring the KL divergence between GMMs. More likely, however, is that the off-diagonal elements in the single Gaussian's full covariance matrix aid discrimination more than being able to use a mixture of diagonal covariance Gaussians.

We have also created a java demonstration of an alternative interface to the system, an automatic playlist generator. A screen shot can be seen in Figure 5, and the demo can be downloaded from our website[1]. This playlist generator seamlessly integrates relevance feedback with normal music-listening habits, for instance by interpreting the skipping of a song as a negative label for the current search, while playing it all the way through would label it as desirable. The classifier is retrained after each song is labeled, converging to the best classifier as quickly as possible.

Using this playlist generator can give quite startling results as the system accurately infers the user's criteria. It also highlights how neatly the active learning paradigm matches a music listening activity in two ways. First, the the user's labels can be obtained transparently from their choice over whether to skip a song. And second, for music, listening to the selected examples is not just a chore undertaken to supervise classifier training, but can also be the goal of the process.

[1] http://labrosa.ee.columbia.edu/projects/playlistgen/

Fig. 5 Screen shot of the SVM active learning automatic playlist generator.

MFCC statistics serve as a flexible representation for songs, able to adequately describe musical artists, moods, and styles. Moreover, we have shown that SVM active learning can improve the results of music retrieval searches by finding relevant results for a user's query more efficiently than conventional SVM retrieval.

Acknowledgements We would like to thank Dr. Malcolm Slaney for his useful discussion and Prof. Shih-Fu Chang for introducing us to the idea of SVM active learning. We would also like to thank the reviewers for their helpful suggestions.

Table 5 Artists and albums from uspop2002 included in experiments. Note that D1, D2, etc. refer to the first and second disc in a multidisc set.

Artist	Training	Testing	Validation
Aerosmith	A Little South of Sanity D1, Nine Lives, Toys in the Attic	A Little South of Sanity D2, Live Bootleg	
Beatles	Abbey Road, Beatles for Sale, Magical Mystery Tour	1, A Hard Day's Night	Revolver
Bryan Adams	Live Live Live, Reckless, So Far So Good	On a Day Like Today, Waking Up the Neighbors	
Creedence Clearwater Revival	Live in Europe, The Concert, Willy and the Poor Boys	Cosmo's Factory, Pendulum	
Dave Matthews Band	Live at Red Rocks D1, Remember Two Things, Under the Table and Dreaming	Before These Crowded Streets, Live at Red Rocks D2	Crash
Depeche Mode	Music for the Masses, Some Great Reward, Ultra	Black Celebration, People are People	Violator
Fleetwood Mac	London Live '68, Tango in the Night, The Dance	Fleetwood Mac, Rumours	
Garth Brooks	Fresh Horses, No Fences, Ropin' the Wind	In Pieces, The Chase	Garth Brooks
Genesis	From Genesis to Revelations, Genesis, Live: The Way We Walk Vol 1	Invisible Touch, We Can't Dance	
Green Day	Dookie, Nimrod, Warning	Insomniac, Kerplunk	
Madonna	Music, You Can Dance, I'm Breathless	Bedtime Stories, Erotica	Like A Prayer
Metallica	Live S—: Binge and Purge D1, Reload, S&M D1	Live S—: Binge and Purge D3, Load	S&M D2
Pink Floyd	Dark Side of the Moon, Pulse D1, Wish You Were Here	Delicate Sound of Thunder D2, The Wall D2	The Wall D1
Queen	Live Magic, News of the World, Sheer Heart Attack	A Kind of Magic, A Night at the Opera	Live Killers D1
Rolling Stones	Get Yer Ya-Ya's Out, Got Live if You Want It, Some Girls	Still Life: American Concert 1981, Tattoo You	
Roxette	Joyride, Look Sharp, Tourism	Pearls of Passion, Room Service	
Tina Turner	Live in Europe D1, Twenty Four Seven, Wildest Dreams	Private Dancer, Live in Europe D2	
U2	All That You Can't Leave Behind, Rattle and Hum, Under a Blood Red Sky	The Joshua Tree, The Unforgettable Fire	Zooropa

References

1. All Music Guide: Site glossary. URL `http://www.allmusic.com/cg/amg.dll?p=amg&sql=32:amg/info_pages/a_siteglossary.html`
2. Aucouturier, J.J., Pachet, F.: Improving timbre similarity : How high's the sky? Journal of Negative Results in Speech and Audio Sciences **1**(1) (2004)
3. Berenzweig, A., Ellis, D.P.W., Lawrence, S.: Using voice segments to improve artist classification of music. In: Proc. AES Intl. Conf. on Virtual, Synthetic, and Entertainment Audio. Espoo, Finland (2002)
4. Berenzweig, A., Ellis, D.P.W., Lawrence, S.: Anchor space for classification and similarity measurement of music. In: Proc. IEEE Intl. Conf. on Multimedia & Expo (2003)
5. Berenzweig, A., Logan, B., Ellis, D.P.W., Whitman, B.: A large-scale evalutation of acoustic and subjective music similarity measures. In: Proc. Intl. Conf. on Music Info. Retrieval (2003)
6. Burgess, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery **2**(2), 121–167 (1998)
7. Chang, E.Y., Tong, S., Goh, K., Chang, C.W.: Support vector machine concept-dependent active learning for image retrieval. ACM Transactions on Multimedia (2005). In press
8. Chen, S., Gopalakrishnan, P.: Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop (1998)
9. Cristianini, N., Shawe-Taylor, J.: An introduction to support Vector Machines: and other kernel-based learning methods. Cambridge University Press, New York, NY (2000)
10. Downie, J.S., West, K., Ehmann, A., Vincent, E.: The 2005 music information retrieval evaluation exchange (mirex 2005): Preliminary overview. In: J.D. Reiss, G.A. Wiggins (eds.) Proc. Intl. Conf. on Music Info. Retrieval, pp. 320–323 (2005)
11. Ellis, D., Berenzweig, A., Whitman, B.: The "uspop2002" pop music data set (2003). URL `http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html`
12. Ellis, D.P.W., Whitman, B., Berenzweig, A., Lawrence, S.: The quest for ground truth in musical artist similarity. In: Proc. Intl. Conf. on Music Info. Retrieval, pp. 170–177 (2002)
13. Foote, J.T.: Content-based retrieval of music and audio. In: C.C.J.K. et al. (ed.) Proc. Storage and Retrieval for Image and Video Databases (SPIE), vol. 3229, pp. 138–147 (1997)
14. Gish, H., Siu, M.H., Rohlicek, R.: Segregation of speakers for speech recognition and speaker identification. In: Proc. Intl. Conf. on Acoustics, Speech and Signal Processing (1991)
15. Hoashi, K., Matsumoto, K., Inoue, N.: Personalization of user profiles for content-based music retrieval based on relevance feedback. In: Proc. ACM Intl. Conf. on Multimedia, pp. 110–119. ACM Press, New York, NY (2003)
16. Hoashi, K., Zeitler, E., Inoue, N.: Implementation of relevance feedback for content-based music retrieval based on user prefences. In: Intl. ACM SIGIR conf. on Research and development in information retrieval, pp. 385–386. ACM Press, New York, NY (2002)
17. Ihler, A.: Kernel density estimation toolbox for MATLAB (2005). URL `http://ssg.mit.edu/~ihler/code/`
18. Jaakkola, T.S., Haussler, D.: Exploiting generative models in discriminative classifiers. In: Advances in Neural Information Processing Systems, pp. 487–493. MIT Press, Cambridge, MA (1999)
19. Lai, W.C., Goh, K., Chang, E.Y.: On scalability of active learning for formulating query concepts. In: L. Amsaleg, B.T. Jónsson, V. Oria (eds.) Workshop on Computer Vision meets Databases, CVDB, pp. 11–18. ACM (2004)
20. Logan, B.: Mel frequency cepstral coefficients for music modelling. In: Proc. Intl. Conf. on Music Info. Retrieval (2000)

21. Logan, B., Salomon, A.: A music similarity function based on signal analysis. In: Proc. IEEE Intl. Conf. on Multimedia & Expo. Tokyo, Japan (2001)
22. Moreno, P., Rifkin, R.: Using the fisher kernel for web audio classification. In: Proc. Intl. Conf. on Acoustics, Speech and Signal Processing (2000)
23. Moreno, P.J., Ho, P.P., Vasconcelos, N.: A kullback-leibler divergence based kernel for SVM classification in multimedia applications. In: S. Thrun, L. Saul, B. Schölkopf (eds.) Advances in Neural Information Processing Systems. MIT Press, Cambridge, MA (2004)
24. Oppenheim, A.V.: A speech analysis-synthesis system based on homomorphic filtering. Journal of the Acostical Society of America **45**, 458–465 (1969)
25. Penny, W.D.: Kullback-Liebler divergences of normal, gamma, Dirichlet and Wishart densities. Tech. rep., Wellcome Department of Cognitive Neurology (2001)
26. Platt, J.C., Cristianini, N., Shawe-Taylor, J.: Large margin DAGs for multiclass classification. In: S. Solla, T. Leen, K.R. Mueller (eds.) Advances in Neural Information Processing Systems, pp. 547–553 (2000)
27. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: Proc. ACM Intl. Conf. on Multimedia, pp. 107–118. ACM Press, New York, NY (2001)
28. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. In: Proc. Intl Conf. on Machine Learning, pp. 999–1006 (2000)
29. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. Journal of Machine Learning Research **2**, 45–66 (2001)
30. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing **10**(5), 293–302 (2002)
31. Whitman, B., Flake, G., Lawrence, S.: Artist detection in music with minnowmatch. In: IEEE Workshop on Neural Networks for Signal Processing, pp. 559–568. Falmouth, Massachusetts (2001)
32. Whitman, B., Rifkin, R.: Musical query-by-description as a multiclass learning problem. In: Proc. IEEE Multimedia Signal Processing Conf. (2002)

Michael Mandel is a PhD candidate at Columbia University. He received his BS degree in Computer Science from the Massachusetts Institute of Technology in 2004 and his MS from Columbia University in Electrical Engineering in 2006. In addition to music recommendation and music similarity, he is interested in computational models of sound and hearing and machine learning.

Graham Poliner received his BS degree in Electrical Engineering from the Georgia Institute of Technology in 2002 and his MS degree in Electrical Engineering from Columbia University in 2004 where he is currently a PhD candidate. His research interests include the application of signal processing and machine learning techniques toward music information retrieval.

Daniel Ellis is an associate professor in the Electrical Engineering department at Columbia University in the City of New York. His Laboratory for Recognition and Organization of Speech and Audio (LabROSA) is concerned with all aspects of extracting high-level information from audio, including speech recognition, music description, and environmental sound processing. Ellis has a Ph.D. in Electrical Engineering from MIT, where he was a research assistant at the Media Lab, and he spent several years as a research scientist at the International Computer Science Institute in Berkeley, CA. He also runs the AUDITORY email list of 1700 worldwide researchers in perception and cognition of sound.