Support Vector Machine Active Learning for Music Retrieval

Michael Mandel, Graham Poliner, and Dan Ellis

April 16, 2005

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 evaluate the success of our system by classifying 1200 pop music songs according to *moods* and *styles* from an online music guide and the performing artist. In comparing a number of feature 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.

1 Introduction

With the sizes of personal digital music collections growing, choosing music appropriate to a particular situation is getting increasingly dif-

ficult. Furthermore, finding music one would like to listen to from a personal collection or an online music store is also a difficult task. 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 this task.

Our solution is to use relevance feedback, specifically Support Vector Machine (SVM) active learning, to learn a classifier for each query. A search is both a mapping from low level audio features to higher level concepts and a way for the user to clarify the search to him or herself. To begin a search, the user presents the system with one or more examples of songs of interest, or "seed" songs. After this, the system iterates between training a new classifier on labeled data and soliciting new labels from the user for informative examples. The search proceeds quickly, and at every stage the system supplies its best estimate of songs similar to the seed 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.

On a dataset of 1210 songs pop songs using music categorizations culled from Allmusic.com along with artist information as our ground truth, we compared a number of acoustic features for retrieval precision in the top 20 ranked results, finding the mean and covariance statistics of MFCC vectors to give the best performance. We have also developed an automatic tester for our SVM active learning system, showing that an SVM active learner can achieve the same precision as a normal SVM with less than half the number of training examples or, alternately, can achieve a ten percentage point increase in precision with the same number of labeled examples.

1.1 Previous Work

1.2 Music Similarity

The idea of judging the similarity of music by a direct comparison between the waveform content was proposed by Foote in 1997 [11]. For computational simplicity, his system used discrete distributions over vector-quantizer symbols, and was evaluated over a database of a few hundred 7-second excerpts. In 2001, Logan and Salomon were able to compare continuous distributions over thousand of complete songs, using the Earth Mover's Distance to calculate dissimilarity between mixtures of Gaussians [17]. There have followed a number of papers refining the features, distance measures, and evaluation techniques including our own work [3, 10, 4, 6]; Aucouturier and Pachet provide an excellent review in [2], 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. Tzanetakis and Cook popularized the use of genre classification [25], whereas Whitman et al. proposed artist identification as a more interesting task, with the attraction of having largely unambiguous ground-truth [26]. 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 RLSC classifier was used for music similarity in [27]. The Fisher Kernel technique we tried was introduced for audio classification

in [18].

1.2.1 Relevance Feedback

While the idea of relevance feedback had been around for a number of years, Simon Tong first described using support vector machines for active learning in [23]. In [24], Tong and Koller describe the *version space* of all possible hyperplanes consistent with labelled data along with methods for reducing it as quickly as possible to facilitate active learning. He has applied SVM active learning to text and image retrieval [23, 22].

Hoashi et al. [14, 13] use relevance feedback for music retrieval, but their approach has a number of disadvantages when compared with SVM active retrieval. Their system is based on Foote's TreeQ vector quantization[11], with which they must re-quantize the entire music database for each query. Relevance feedback is incorporated into the model by modifying the quantization weights of desired vectors. Our approach uses the same feature vectors for every query and SVM active learning, which has a strong theoretical justification, to incorporate user feedback.

2 Algorithm

Our system's success depends on the harmonious interaction between two components. The feature vectors used to describe each song must make explicit information from the original audio relevant to the classification task at hand. The classifier must then be able to take advantage of this information to successfully classify songs with as few labeled training examples as possible.

1. The user seeds the system with representative songs.
2. If this is the first feedback round, present the user with randomly selected songs to label.
   Else, present the user with the most informative songs to label, i.e. those closest to the decision boundary.
3. Train an SVM on all labeled training instances.
4. Return to the user the most relevant songs, i.e. those with the greatest positive distances from the decision boundary.
5. Repeat steps 2 to 4 until the user is satisfied with the results.

Figure 1: Summary of SVM active learning algorithm.

2.1 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.

2.1.1 Support Vector Machines

The support vector machine (SVM) is a supervised classification system that uses a hypothesis space of linear functions in a high dimensional feature space in order to learn separating hyperplanes. As such, SVM classification attempts to generalize an optimal decision boundary between classes. Labeled training data in a given space is separated by a maximum margin hyperplane through SVM classification.

The training data is projected into a higher dimensional feature space via a Mercer kernel operator $K(\cdot)$ [22]. For nonlinearly sep-

arable data, the original feature space may be mapped to a higher-dimensional, separable space by choosing a different kernel function. In our implementation, we selected a Gaussian kernel:

$$K(\mathbf{u}, \mathbf{v}) = e^{-\gamma||\mathbf{u}-\mathbf{v}||^2}, \tag{1}$$

thus the space of possible classifier functions consists of placing weighted Gaussians on key training instances[8]; the SVM training algorithm chooses these instances (the "support vectors") and weights to optimize the margin between classifier boundary and training examples.

Because only a subset of the training samples end up being used in the final classifier, an identical SVM would result from a training set that omitted all the remaining examples. This makes SVMs an attractive complement to relevance feedback: if the feedback system can accurately identify the critical samples that will form the support vectors, training time and labeling effort can, in the best case, be reduced drastically with no impact on classifier accuracy.

2.1.2 Active Learning

In an active learning system, the user becomes an integral part of the learning and classification process. As opposed to conventional ("passive") SVM classification where a classifier is trained on a large pool of randomly selected labeled data, in an active learning system the user is asked 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 in feature space and hyperplanes in

version space enables active learning with SVMs. Version space is defined as the set of all hyperplanes consistent with the labeled data. The user's desired division of the feature space corresponds to a point in version space that the SVM active learner would like to find as quickly as possible. The hyperplanes of labeled instances divide version space, restricting the range of consistent classifiers and homing in on the desired classifier. The most efficient way to find this classifier is to halve the version space with every labeled data point.

The simplest approximation to this goal is to select for labeling those points in feature space closest to the SVM decision boundary. Other methods exist for selecting points [24] but require expensive searches or multiple retrainings of the classifier. Ideally, points would be labeled individually, with the classifier retrained after each labeling, allowing an optimally efficient splitting of the version space. User convenience, however, dictates that multiple points be selected for labeling at once, a situation that leads to suboptimal shrinking of the version space, but reasonable performance.

The first round of active learning is treated as special. The user only seeds our system with positive examples so the first group of examples presented to the user for labeling cannot be chosen by a classifier. These first examples are simply chosen at random, with the expectation that since positive examples are relatively rare in the database, many of the randomly chosen examples will be negative. One could imagine choosing the first group of examples that would maximally cover the feature space. Since our features are precomputed, this group of songs would be the same for every query.

This active learning implementation has a number of advantages over conventional SVM classification. Primarily, by presenting the

user with and training on the most informative data, the algorithm can achieve the same classification performance with fewer labeled examples. Secondly, 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 a specific "mood", jazz music, or female artists. As a result, the active SVM system uniformly maps the range of high-level descriptors people use to describe their perception of music to low-level features extractable from digital music.

2.2 Audio Features

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. All of the features were based on mel-frequency cepstral coefficients (MFCCs), and most of them incorporated Gaussian mixture models (GMMs) trained on the MFCCs. It should be noted that the features compared in this paper only model stationary spectral characteristics of music, averaged across time, and ignore the higher-order temporal structure. In order to compare the features to one another, we used simple SVM cross-validation on the artist and style classification tasks (see Section 3). The rest of this section describes these audio features and how they were calculated.

MFCCs are a short-time spectral decomposition of an audio signal that conveys the general frequency characteristics important to human hearing. While originally developed to decouple vocal excitation

from vocal tract shape for automatic speech recognition [21], they have found applications in other auditory domains including music retrieval [16, 11].

In order to calculate MFCCs, the signal is first broken into overlapping frames, each approximately 20ms 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, depending on the length of the song, approximately 30,000 temporal frames.

2.2.1 MFCC Statistics

A combination of the mean and unwrapped covariance matrix of all of the MFCC frames in each song outperformed all other features. These features are commonly used in speech processing e.g. for segmenting a recording according to speaker turns [12, 7].

Since MFCC frames describe general spectral aspects of a signal, their covariance describes the co-occurrence of pairs of features across time frames, a descriptive quantity. Assuming that all of the MFCC frames of a song are independent, identically distributed samples from a single Gaussian, this feature represents a sufficient statistic for estimating that Gaussian from all of the samples in the song. While Gaussianity is a strong assumption to make, the success of these features in our experiments shows that they contain much information

about artists, styles, and moods. Further gains could be seen by modeling songs as mixtures of Gaussians over MFCC frames.

2.2.2 Gaussian Mixture Models

Other features were based on a Gaussian mixture model trained on MFCC frames collected from all of the songs in the training set. A single model described the entire music corpus.

Using 0.2% of the MFCC frames selected at random from every song, we trained a Gaussian mixture model with diagonal covariances. This GMM is a generative model and assumes that all time frames are independent identically distributed samples from a single underlying mixture distribution.

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 ways of 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 for such a model.

In the two groups of models trained on fewer frames from more songs, there is a large jump in likelihood between the GMM with 10 Gaussians and that with 20. It turns out that damaged mp3 files in the training and validation sets caused one Gaussian in all of the larger

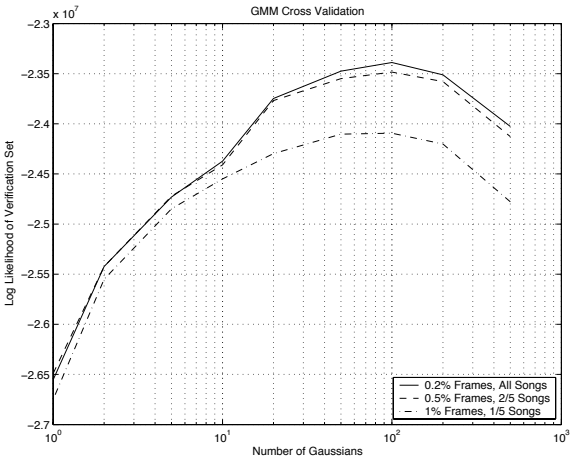

Figure 2: Cross-validation of Gaussian mixture models using different numbers of Gaussians and different training sets.

models to latch on to the spurious audio that resulted from decoding these files. These models make good detectors of the damaged files and will be used to remove those files from the database for future experiments. The damaged mp3s and their detector do not alter the optimal number of Gaussians much, if at all, as can be seen by comparing the peak in those two lines to that in the third, which managed to avoid the few damaged files altogether.

2.2.3 Average GMM Posteriors

Given the GMM trained on all of the songs, the task of extracting feature vectors from individual songs still remains. The first such feature we used was the mean log posterior over each song. We evaluated the posterior probability of each Gaussian in the GMM given every MFCC frame in the song, generating one feature dimension per Gaussian in the model. The features for a particular song were then the mean of

the log-posterior of all of the frames in that song. If each Gaussian in the GMM represents one cluster of MFCC frames, then this feature describes the proportion of each cluster in each song.

Since the number of MFCC frames in a song depends on the song's length, and since no Gaussian fits every frame particularly well, the sum of the log posteriors (corresponding to the joint probability of IID MFCC frames) would most closely correspond to the length of the song. The mean log posterior removes this length dependence from the features to focus on more interesting aspects of the audio. For these experiments, we extracted features for all songs based on GMMs with 50 and 100 Gaussians, based on the above GMM cross-validation experiment.

2.2.4 Fisher Kernel

Jaakkola and Haussler[15] describe the Fisher kernel 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 GMM, reducing the dimensionality of the data to the partial derivatives of the log posterior probabilities of each Gaussian with respect to the data. From [18],

$$\nabla_{\mu_k} \log(P(X|\mu_k)) = \sum_{t=1}^{m} P(k|\mathbf{x}_t)\Sigma_k^{-1}(\mathbf{x}_t - \mu_k). \tag{2}$$

where $P(k|\mathbf{x}_t)$ is the posterior probability of the kth Gaussian in the mixture given data point $\mathbf{x}_t$, 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 deriva-

tive with respect to changes in each dimension of each Gaussian's mean. A more compact feature would be 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.

3 Experiments

In order to thoroughly test the SVM active music retrieval system, we varied our SVM parameters, features, and the number of training examples per active retrieval round.

3.1 Dataset

We ran our experiments on a subset of the *uspop2002* collection [5, 9]. To avoid the so called "producer effect" [26] in which songs from the same album share overall spectral characteristics that could swamp any similarities between albums, we selected artists who had enough albums in *uspop2002* to designate entire albums as training, testing, or validation. Such a division required each artist to have three albums for training and two for testing, each with at least eight tracks to get enough data points per album. The validation set was made up of any albums the selected artists had in *uspop2002* in addition to those five. In total there were 18 artists (out of 400) who met these criteria, see Table 4 for a complete list of the artists and albums included in our experiments. In total, we used 90 albums by 18 artists which contained a total of 1210 songs divided into 656 training, 451 testing, and 103 validation songs.

Table 1: 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

3.2 Evaluation

Since the goal of SVM active learning is to quickly learn an arbitrary classification task, any categorization of the data points can be used as ground truth for testing. We chose to test our system by classifying AMG moods, AMG styles, and artist.

The All Music Guide (AMG) is a website (www.allmusic.com) and book that reviews, rates, and categorizes music and musicians [1]. Two of our ground truth datasets were AMG "moods" and "styles." 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 1 for a list of the most popular moods and styles.

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 had 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.

3.3 Experiments

Before beginning our experiments, we needed to set the SVM parameters γ (from equation 1), and C, the weighting used to trade-off between classifier margin and margin violations for particular points which are more efficiently treated as mislabeled via the so-called "slack variables". We used a simple cross-validation grid search to find well-performing values. We did not exhaustively compare these results for all combinations of features and ground truth, but only a representative sample. After normalizing all feature columns to be zero mean and unit variance, the best performing classifiers used $C = 10^4$ and $\gamma = 0.01$. Settings widely divergent from these tended to generate uninformative classifiers that labeled everything as a negative result.

The first experiment compared our different feature sets against artist and style ground truth in order to determine if one clearly dominated the others. The first test was 18-way artist identification, training and testing on separate albums. We provide these results to compare against other author's systems and to compare features to one another, but they are not directly applicable to the SVM active learning task which only learns one category at a time. We also compared the features to one another by measuring the mean precision on the top

20 results returned from the test albums after training on the training albums, the same metric used in the later experiments.

Since the ground truth categories typically have many more negative than positive examples, it is possible to achieve quite good accuracy by classifying all points as negative. In order to focus on positive examples, we evaluate the SVM active learner by examining the mean precision of its top 20 results, where results are sorted by distance from the decision boundary, most positive first, as in [22]. The mean is taken over all of the categories in a particular ground truth set, e.g. over all artists.

One justification for this evaluation metric is that as the size of music databases grow, the user wants good search results to be returned first, but may not care so much whether he or she sees all of the positive examples. 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 smoothly controls the measured numbers, it also happens that precision-at-20 distinguishes the features well from each other.

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

In this experiment we compared five different training group sizes. In each trial, an active learner was randomly seeded with 5 elements

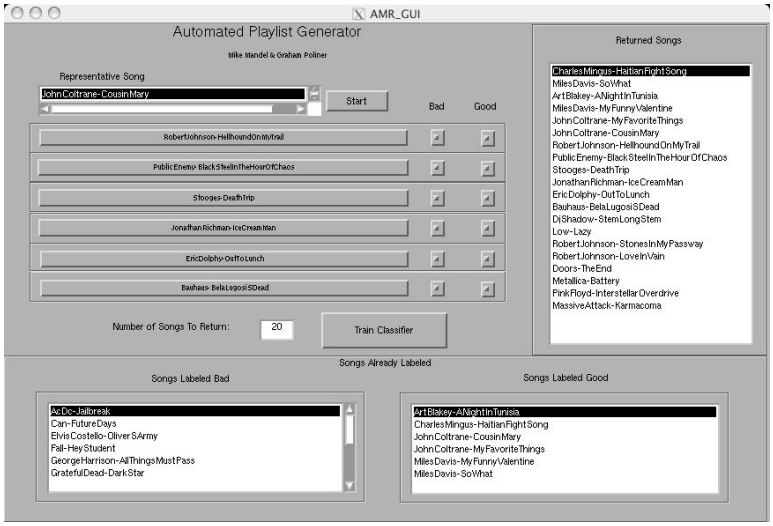

Figure 3: Active Learning Graphical User Interface.

from within the class, corresponding to a user supplying songs that they would like the results to be similar to. The learner then performed simulated relevance feedback with 2, 5, 10, and 20 songs per round, with a final classifier performing 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.

3.4 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 to generate relevant response to their queries. The evaluators searched for categories such as jazz, rap, rock, punk, female vocalists,

fast, etc.

The graphical user interface is displayed in Figure 3. The user selects a representative seed song and begins the active retrieval system. The system presents six songs to label as similar or dissimilar to the representative song according to the user's desired category (the song may also be left unlabeled). Next, the user selects the number of songs to return and begins the classification process. Labeled songs are displayed at the bottom of the interface, and songs returned by the classifier are displayed in the list to the right. At any time, the user may click on any song displayed in the interface to hear a representative segment of that song. After each classification round, the user is presented with six new songs to label and may perform the process iteratively as many times as desired.

3.5 Results

The results of the feature comparison experiment can be seen in Table 2. It is clear that the MFCC Statistics outperform the other features in all tasks, with Fisher kernel magnitudes second best, while posterior features and full Fisher kernels perform similarly, favoring the posterior features for N-way artist identification but Fisher kernels for the binary style identification task which is 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 heavily 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

Table 2: Comparison of various audio features: number of dimensions, accuracy on 18-way artist classification and precision-at-20 for artist identification and style identification.

		Accuracy	Precision-at-20	
Feature	# dims	Artist 18-way	Artist ID	Style ID
Post050	50	.420	.350	.392
Post100	100	.463	.378	.450
Fisher Kernel	650	.373	.375	.494
Fisher Ker Mag	50	.463	.433	.497
MFCC Stats	91	.687	.656	.744

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 0.683 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 3 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.

Table 3: Precision-at-20 on test set of classifiers trained with different numbers of examples per round (Ex/R) or conventional (passive) training, all trained with 50 examples total.

Ground Truth	2 Ex/R	5 Ex/R	10 Ex/R	20 Ex/R	Conv.
Style	.683	.671	.663	.641	.587
Artist	.624	.629	.603	.583	.501
Mood	.478	.465	.447	.435	.412

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

Opposing the use of small training sets, however, is 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 would be interesting to see how well GMMs trained on individual songs would fare as song features. Our MFCC statistics can be equated to a GMM with only a single Gaussian, and their consistent structure makes them easy to compare by unwrapping the covariance matrix. Using mixtures of Gaussians, however, would require comparing more varied pdfs, for instance with the KL divergence between mixtures,

precomputing an $n \times n$ Gram matrix instead of just n feature vectors. Moreno et al. have applied this kernel to multimedia classification[19] but not to musical recommendation.

We are currently working on an online demonstration of the system, in addition to our current Matlab user interface. Another potential interface could be seamlessly integrated with current music players, 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 desireable. In order to train the classifier most effectively, the most desireable results could be interspersed with the most discriminative results in a ratio selectable by the user. This system would allow retraining of the classifier between every labelling, converging on the most relevant classifier as quickly as possible.

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

Acknowledgments

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. 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.

References

[1] All Music Guide. Site glossary. http://www.allmusic.com/cg/amg.dll?p=amg&sql=32:amg/info_pages/a_siteglossary.html.

[2] Aucouturier, J.-J., and 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., and Lawrence, S. Using voice segments to improve artist classification of music. In *AES-22 Intl. Conf. on Virt., Synth., and Ent. Audio.* (Espoo, Finland, June 2002).

[4] Berenzweig, A., Ellis, D. P. W., and Lawrence, S. Anchor space for classification and similarity measurement of music. In *ICME 2003* (2003).

[5] Berenzweig, A., Logan, B., Ellis, D., and Whitman, B. A large-scale evaluation of acoustic and subjective music similarity measures. In *International Symposium on Music Information Retrieval* (October 2003).

[6] Berenzweig, A., Logan, B., Ellis, D. P. W., and Whitman, B. A large-scale evalutation of acoustic and subjective music similarity measures. In *Proc. Int. Conf. on Music Info. Retrieval ISMIR-03* (2003).

[7] Chen, S., and 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).

[8] Cristianini, N., and Shawe-Taylor, J. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000.

[9] Ellis, D., Berenzweig, A., and Whitman, B. The"uspop200" pop music data set. http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html.

[10] Ellis, D. P. W., Whitman, B., Berenzweig, A., and Lawrence, S. The quest for ground truth in musical artist similarity. In *Proc. Int. Symposium on Music Inform. Retriev. (ISMIR)* (October 2002), pp. 170–177.

[11] Foote, J. T. Content-based retrieval of music and audio. In *Proc. SPIE Vol. 3229, p. 138-147, Multimedia Storage and Archiving Systems II* (Oct. 1997), C.-C. J. Kuo, S. F. Chang, and V. N. Gudivada, Eds., pp. 138–147.

[12] Gish, H., Siu, M.-H., and Rohlicek, R. Segregation of speakers for speech recognition and speaker identification. In *Proc. IEEE ICASSP* (1991).

[13] Hoashi, K., Matsumoto, K., and Inoue, N. Personalization of user profiles for content-based music retrieval based on relevance feedback. In *ACM international conference on Multimedia* (New York, NY, USA, 2003), ACM Press, pp. 110–119.

[14] Hoashi, K., Zeitler, E., and Inoue, N. Implementation of relevance feedback for content-based music retrieval based on user prefences. In *International ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2002), ACM Press, pp. 385–386.

[15] Jaakkola, T. S., and Haussler, D. Exploiting generative models in discriminative classifiers. In *Advances in neural information processing systems 11* (Cambridge, MA, USA, 1999), MIT Press, pp. 487–493.

[16] Logan, B. Mel frequency cepstral coefficients for music modelling. In *International Symposium on Music Information Retrieval* (2000).

[17] Logan, B., and Salomon, A. A music similarity function based on signal analysis. In *Proc. ICME 2001* (Tokyo, Japan, 2001).

[18] Moreno, P., and Rifkin, R. Using the fisher kernel for web audio classification. In *Proc. International Conference on Acoustics, Speech and Signa l Processing* (2000).

[19] Moreno, P. J., Ho, P. P., and Vasconcelos, N. A kullback-leibler divergence based kernel for SVM classification in multimedia applications. In *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. MIT Press, Cambridge, MA, 2004.

[20] Oppenheim, A. V. A speech analysis-synthesis system based on homomorphic filtering. *Journal of the Acostical Society of America 45* (February 1969), 458–465.

[21] Tong, S., and Chang, E. Support vector machine active learning for image retrieval. In *ACM International Conference on Multimedia* (New York, NY, USA, 2001), ACM Press, pp. 107–118.

[22] Tong, S., and Koller, D. Support vector machine active learning with applications to text classification. In *International Conference on Machine Learning* (2000), pp. 999–1006.

[23] Tong, S., and Koller, D. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research 2* (2001), 45–66.

[24] Tzanetakis, G., and Cook, P. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing 10*, 5 (July 2002), 293–302.

[25] Whitman, B., Flake, G., and Lawrence, S. Artist detection in music with minnowmatch. In *IEEE Workshop on Neural Networks for Signal Processing* (Falmouth, Massachusetts, September 10–12 2001), pp. 559–568.

[26] Whitman, B., and Rifkin, R. Musical query-by-description as a multi-class learning problem. In *Proc. IEEE Multimedia Signal Processing Conference (MMSP)* (December 2002).

Table 4: Artists and albums from uspop2002 included in experiments.

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 Shit: Binge and Purge D1, Reload, S&M D1	Live Shit: 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

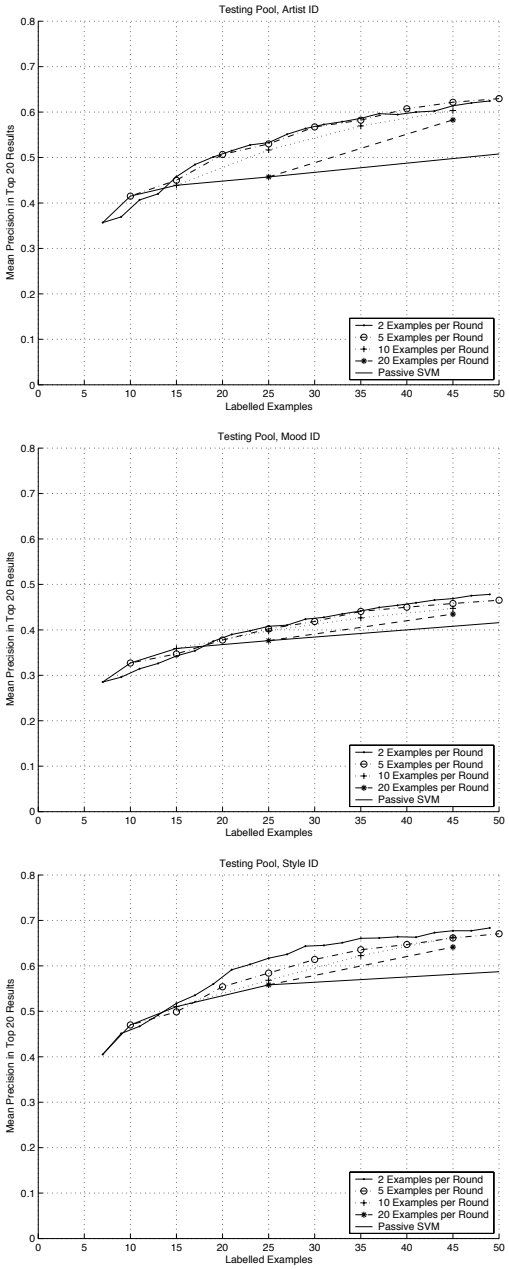

Figure 4: Performance increase due to active learning for (a) artist identification, (b) mood classification, and (c) style classification. The plots show the mean precision in the top 20 results over the test set as the number of examples per round is varied. The solid line without symbols shows the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