SONG-LEVEL FEATURES AND SUPPORT VECTOR MACHINES FOR MUSIC CLASSIFICATION

Michael I. Mandel and Daniel P.W. Ellis
LabROSA, Dept. of Elec. Eng., Columbia University, NY NY USA
{mim,dpwe}@ee.columbia.edu

ABSTRACT

Searching and organizing growing digital music collections requires automatic classification of music. This paper describes a new system, tested on the task of artist identification, that uses support vector machines to classify songs based on features calculated over their entire lengths. Since support vector machines are exemplar-based classifiers, training on and classifying entire songs instead of short-time features makes intuitive sense. On a dataset of 1200 pop songs performed by 18 artists, we show that this classifier outperforms similar classifiers that use only SVMs or song-level features. We also show that the KL divergence between single Gaussians and Mahalanobis distance between MFCC statistics vectors perform comparably when classifiers are trained and tested on separate albums, but KL divergence outperforms Mahalanobis distance when trained and tested on songs from the same albums.

Keywords: Support vector machines, song classification, artist identification, kernel spaces

1 INTRODUCTION

In order to organize and search growing music collections, we will need automatic tools that can extract useful information about songs directly from the audio. Such information could include genre, mood, style, and performer. In this paper, we focus on the specific task of identifying the performer of a song out of a group of 18. Since each song has a unique performer, we use a single 18-way classifier.

While previous authors have attempted such classification tasks by building models of the classes directly from short-time audio features, we show that an intermediate stage of modeling entire songs improves classification. Further gains are also seen when using Support Vector Machines (SVMs) as the classifier instead of k-nearest neighbors (kNN) or other direct distance-based measures. These advantages become evident when comparing four combinations of classifiers and features. Not only does song-level modeling improve classification accuracy, it also decreases classifier training times, allowing rapid classifier construction for tasks such as active retrieval.

We also explore the space of song-level features by comparing three different distance measures for both SVM and kNN classification. The first distance measure is the Mahalanobis distance between so-called MFCC statistics features as used in Mandel et al. (2005). As recommended in Moreno et al. (2004), we also model songs as single, full-covariance Gaussians and mixtures of 20 diagonal-covariance Gaussians, measuring distances between them with the symmetric Kullback Leibler divergence.

Our dataset, a subset of *uspop2002* , contained 1210 songs from 18 artists. When it was broken up so that training and testing songs came from different albums, an SVM using the Mahalanobis distance performed the best, achieving a classification accuracy of 69%. When the songs were randomly distributed between cross validation sets, an SVM using the KL divergence between single Gaussians was able to classify 84% of songs correctly.

1.1 Previous Work

The popularity of automatic music classification has been growing steadily for the past few years. Many authors have proposed systems that either model songs as a whole or use SVMs to build models of classes of music, but to our knowledge none has combined the two ideas.

West and Cox (2004) use neither song level features nor SVMs. Instead, they train a complicated classifier on many types of audio features, but still model entire classes with frame-level features. They show promising results on 6-way genre classification tasks, with nearly 83% classification accuracy for their best system.

Aucouturier and Pachet (2004) model individual songs with GMMs and use Monte Carlo methods to estimate the KL divergence between them. Their system is designed as a music-retrieval system, and thus its performance is measured in terms of retrieval precision. They do not use

an advanced classifier, as their results are ranked by kNN. They do provide some useful parameter settings for various models that we use in our experiments, namely 20 MFCC coefficients and 20 Gaussian components in our GMMs.

Logan and Salomon (2001) also model individual songs as GMMs, trained using k-means instead of EM. They approximate the KL divergence between GMMs as the earth mover's distance based on the KL divergences of the individual Gaussians in each mixture. Since their system is described as a distance measure, there is no mention of an explicit classifier. They do, however, suggest generating playlists with the nearest neighbors of a seed song.

Tzanetakis and Cook (2002) also calculate song-level features. They classify songs into genre with kNN based on GMMs trained on song features. Even though they only had 100 feature vectors per class, they were still able to model these classes with GMMs having a small number of components because of their parsimonious use of feature dimensions.

Of the researchers classifying music with SVMs, Whitman et al. (2001) and Xu et al. (2003) both train SVMs on collections of short-time features from entire classes, classify individual frames in test songs, and then let the frames vote for the class of the entire song.

Moreno et al. (2004) use SVM classification on various file-level features for speaker identification and speaker verification tasks. They introduce the Symmetric KL divergence based kernel and also compare modeling a file as a single, full-covariance Gaussian or a mixture of Gaussians.

2 ALGORITHM

2.1 Song-Level Features

All of our features are based on mel-frequency cepstral coefficients (MFCCs). 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 (Oppenheim, 1969), they have found applications in other auditory domains including music retrieval (Logan, 2000; Foote, 1997). At the recommendation of Aucouturier and Pachet (2004), we used 20-coefficient MFCCs.

Our features are most accurately described as *timbral* because they do not model any temporal aspects of the music, only its short-time spectral characteristics. We make the strong assumption that songs with the same MFCC frames in a different order should be considered identical. Some authors call this type of modeling a "bag of frames", after the "bag of words" models used in text retrieval, which are based on the idea that each word is an independent, identically distributed (IID) sample from a bag containing many words in different amounts.

Once we have extracted the MFCCs for a particular song, we describe that song in a number of ways, comparing the effectiveness of each model. The mean and covariance of the MFCCs over the duration of the song describe the Gaussian with the maximum likelihood of generating those points under the "bag of frames" model. Those statistics, however, can also be unwrapped into a vector and compared using the Mahalanobis distance. Equivalently, the vectors can be normalized over all songs to be zero-mean and unit-variance, and compared to one another using the Euclidean distance. Going beyond the simple Gaussian model, a mixture of Gaussians, fit to the MFCCs of a song using the EM algorithm, is richer, able to model nonlinear correlations.

2.2 Support Vector Machines

The support vector machine is a supervised classification system that finds the maximum margin hyperplane separating two classes of data. If the data are not linearly separable in the feature space, as is often the case, they can be projected into a higher dimensional space by means of a Mercer kernel, $K(\cdot)$. In fact, only the inner products of the data points in this higher dimensional space are necessary, so the projection can be implicit if such an inner product can be computed directly.

The space of possible classifier functions consists of weighted linear combinations of key training instances in this kernel space (Cristianini and Shawe-Taylor, 2000). The SVM training algorithm chooses these instances (the "support vectors") and weights to optimize the margin between classifier boundary and training examples. Since training examples are directly employed in classification, using entire songs as these examples aligns nicely with the problem of song classification.

2.3 Distance Measurements

In this paper, we compare three different distance measurements, all of which are classified using a radial basis function kernel. The MFCC statistics are the unwrapped mean and covariance of the MFCCs of an entire song. The distance between two such vectors is measured using the Mahalanobis distance,

$$D_M(\mathbf{u}, \mathbf{v}) = (\mathbf{u} - \mathbf{v})^T \Sigma^{-1} (\mathbf{u} - \mathbf{v}), \tag{1}$$

where Σ is the covariance matrix of the features across all songs, approximated as a diagonal matrix of the individual feature's variances.

The same means and covariances, when reinterpreted as a single Gaussian model, can be compared to one another using the Kullback Leibler divergence (KL divergence). For two distributions, $p(x)$ and $q(x)$, the KL divergences is defined as,

$$KL(p \mid\mid q) \equiv \int p(x) \log \frac{p(x)}{q(x)} \, dx \tag{2}$$

$$= E_p \left\{ \log \frac{p(X)}{q(X)} \right\}. \tag{3}$$

For single Gaussians, $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mu_p, \Sigma_p)$ and $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mu_q, \Sigma_q)$, there is a closed form for the KL divergence (Penny, 2001),

$$2KL(p \mid\mid q) = 2KL_{\mathcal{N}}(\mu_p, \Sigma_p; \mu_q, \Sigma_q) \tag{4}$$

$$= \log \frac{|\Sigma_q|}{|\Sigma_p|} + Tr(\Sigma_q^{-1} \Sigma_p) + (\mu_p - \mu_q)^T \Sigma_q^{-1} (\mu_p - \mu_q) - d. \tag{5}$$

Unfortunately, there is no closed form solution for the KL divergence between two GMMs, it must be approximated using Monte Carlo methods. An 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

$$E_p\left\{\log \frac{p(X)}{q(X)}\right\} \approx \frac{1}{n}\sum_{i=1}^{n} \log \frac{p(X_i)}{q(X_i)}. \quad (6)$$

We used the Kernel Density Estimation toolbox from Ihler (2005) for these calculations.

Also, note the relationship between the above Monte Carlo estimate of the KL divergence and maximum likelihood classification. Instead of drawing samples from a distribution modeling a collection of MFCC frames, the maximum likelihood classifier uses the MFCC frames directly as evaluation points. If $M_1, \ldots, M_n$ are MFCC frames from a song, drawn from some distribution $p(m)$, the KL divergence between the song and an artist model $q(m)$ can be approximated as

$$E_p\left\{\log \frac{p(M)}{q(M)}\right\} = E_p\left\{\log p(M)\right\} - E_p\left\{\log q(M)\right\} \quad (7)$$

$$\approx H_p - \frac{1}{n}\sum_{i=1}^{n} q(M_i), \quad (8)$$

where H_p, the entropy of $p(m)$, and n are constant for a given song and thus do not affect the optimization. For a given song, then, choosing the artist model with the smallest KL divergence is equivalent to choosing the artist model under which the song's frames have the maximum likelihood.

Since the KL divergence is neither symmetric nor positive definite, we must modify it to satisfy the Mercer conditions in order to use it as an SVM kernel. To symmetrize it, we add the two divergences together,

$$D_{KL}(p, q) = KL(p \mid\mid q) + KL(q \mid\mid p). \quad (9)$$

Exponentiating the elements of this matrix will create a positive definite matrix, so our final gram matrix has elements

$$K(X_i, X_j) = e^{-\gamma D_{KL}(X_i, X_j)}, \quad (10)$$

where γ is a parameter that can be tuned to maximize classification accuracy. Calculating these inner products is relatively costly and happens repeatedly, so we precompute $D_{KL}(X_i, X_j)$ off line and only perform lookups on line.

3 EVALUATION

3.1 Dataset

We ran our experiments on a subset of the *uspop2002* collection (Berenzweig et al., 2003; Ellis et al., 2005). To avoid the so called "producer effect" or "album effect" (Whitman et al., 2001) in which songs from the same album share overall spectral characteristics much more than songs from the same artist's other albums, we designated entire albums as training, testing, or validation. The training set was used for building classifiers, the validation set was used to tune model parameters, and final results were reported for songs in the test set.

Table 1: Artists from *uspop2002* included in dataset

Aerosmith	Beatles	Bryan Adams
Creedence Clearwater Revival	Dave Matthews Band	Depeche Mode
Fleetwood Mac	Garth Brooks	Genesis
Green Day	Madonna	Metallica
Pink Floyd	Queen	Rolling Stones
Roxette	Tina Turner	U2

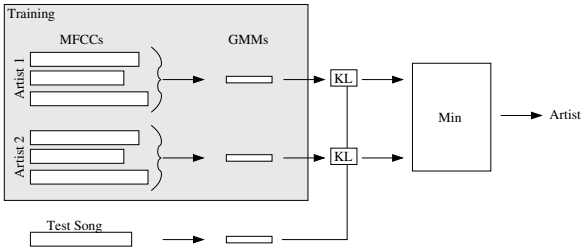

Figure 1: Classification of artist level features without using an SVM. The shaded region indicates calculations performed during training.

In order to have a meaningful artist identification task, we selected artists who had enough albums in *uspop2002* to partition in this way, namely three albums for training and two for testing. The validation set was made up of any albums the selected artists had in *uspop2002* in addition to those five. 18 artists (out of 400) met these criteria, see Table 1 for a complete list of the artists included in our experiments. In total, we used 90 albums by these 18 artists which contained a total of 1210 songs divided into 656 training, 451 testing, and 103 validation songs.

In addition to this fixed grouping of albums, we also evaluated our classifiers with three-fold cross-validation. Each song was randomly assigned to one of three groups and the classifier was trained on two groups and then tested on the third. All three sets were tested in this way and the final classification accuracy used the cumulative statistics over all rounds. We repeated these cross-validation experiments for five different divisions of the data and averaged the accuracy across all repetitions. This cross-validation setup divides songs, not albums, into groups, so the "album effect" is readily apparent in its results.

3.2 Experiments

In our experiments, we compared all four combinations of song-level versus artist-level features, and SVM versus non-SVM classifiers. We also investigated the effect of different distance measures on SVM and kNN classification. See Figures 1 and 2 for a graphical depiction of the feature extraction and classification processes for artist

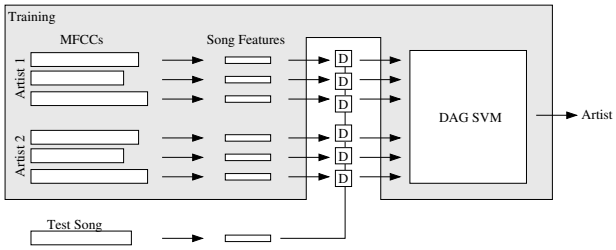

Figure 2: Classification of song level features with a DAG-SVM. The shaded region indicates calculations performed during training. Note that the song-level features could be GMMs and the distance function could be the KL divergence, but it is not required.

and song level features, respectively.

The first experiment used neither song-level features nor SVMs, training a single GMM on the MFCC frames from all of an artist's songs at once. The likelihood of each song's frames was evaluated under each artist model and a song was predicted to come from the model with the maximum likelihood of generating its frames. We used 50 Gaussians in each artist GMM, trained on 10% of the frames from all of the artist's training songs, for approximately 12000 frames per artist.

The second experiment used SVMs, but not song-level features. By training an 18-way DAG-SVM (Platt et al., 2000) on a subset of the frames used in the first experiment, we attempted to learn to classify MFCC frames by artist. To classify a song, we first classified all of its frames and then predicted the song's class to be the most frequently predicted frame class. Unfortunately, we were only able to train on 500 frames per artist, not enough to achieve a classification accuracy significantly above chance levels.

Experiments with song level features compared the effectiveness of three different distance measures and song models. The Mahalanobis distance and KL divergence between single Gaussians shared an underlying representation for songs, the mean and covariance of their MFCC frames. These two models were fixed by the songs themselves, except for the SVM's γ parameter. The KL divergence between GMMs, however, had a number of additional parameters that needed to be tuned. In order to make the calculations tractable, we trained our GMMs on 3000 MFCC frames from each song, roughly 10-20% of the total. We decided on 20 Gaussian components based on estimates of the number of samples needed per Gaussian given the previous constraint and the advice of Aucouturier and Pachet (2004). We also selected the number of Monte Carlo samples used to approximate the KL divergence. In this case 500 seemed to be high enough to give fairly consistent results, while still being fast enough to calculate for 1.4 million pairs of songs.

The third experiment used song-level features, but a simple k-nearest neighbors classifier. For all three song-level features and corresponding distance measures, we used a kNN classifier to label test songs with the label most prevalent among the k training songs the smallest distance away. For these experiments k was varied from 1 to 10, with $k = 1$ performing either the best or competitively.

The final experiment used song-level features and an SVM classifier. Again, for all three song-level features and Gram matrices of distances, we learned an 18-way DAG-SVM classifier for artists. We tuned the γ parameter of the SVMs to maximize classification accuracy. In contrast to the first two experiments, which were only performed for the fixed training and testing sets separated by album, the third and fourth experiments were also performed on cross-validation datasets.

3.3 Results

See Table 2 for the best performance of each of our classifiers and Figure 3 for a graph of the results for separate training and testing albums. These results clearly show the advantage of using both song-level features and SVM classifiers, a 15 percentage point gain in 18-way classification accuracy.

It should also be noted that training times for the two classifiers using low-level features were considerably higher than for those using song-level features. While song-level features involve an initial investment in extracting features and measuring distances between pairs of songs, the classifiers themselves can be trained quickly on any particular subset of songs. Fast training makes these methods useful for relevance feedback and active learning tasks, such as those described in Mandel et al. (2005).

In contrast, artist level classifiers spend little time extracting features from songs, but must train directly on a large quantity of data up front, making retraining just as costly as the initial computational expense. In addition, classifying each song is also relatively slow, as frames must be classified individually and the results aggregated into the final classification. For both of these reasons, it was difficult to obtain cross-validation data for the artist-level feature classifiers.

See Table 3 for the performance of the three distance measures used for song-level features. The Mahalanobis distance and KL divergence for single Gaussians performed comparably, since for the 451 test points, a difference of 0.039 is not statistically significant. Surprisingly, however, the KL divergence between single Gaussians greatly surpassed the Mahalanobis distance when trained and tested on songs from the same albums.

All of the SVM results in Table 3 were collected for optimal values of γ, which differed between distance measures, but not between groups of songs. Since training SVMs and changing γ took so little time after calculating the Gram matrix, it was easy to find the best performing γ by searching the one-dimensional parameter space.

4 DISCUSSION

Modeling songs instead of directly modeling artists makes intuitive sense. Models like GMMs assume stationarity or uniformity of the features they are trained on. This assumption is much more likely to hold over individual songs than over an artist's entire catalog. Individual songs might even be too varied, as in the case of extended-form

Table 2: Classification accuracy on 18-way artist identification reported for training and testing on separate albums (Sep) and training and testing on different songs from the same albums (Same). For separate albums ($N = 451$) statistical significance is achieved for a difference of around .06. For songs from the same album ($N = 2255$) statistical significance is achieved for a difference of around .02.

Classifier	Song-Level?	SVM?	Sep	Same
Artist GMM	No	No	.541	—
Artist SVM	No	Yes	.067	—
Song KNN	Yes	No	.524	.722
Song SVM	Yes	Yes	.687	.839

Table 3: Classification accuracy for different song-level distance measures.

Classifier	Distance	Sep	Same
KNN	Mahalanobis	.481	.594
KNN	KL-Div 1G	.524	.722
KNN	KL-Div 20G	.365	.515
SVM	Mahalanobis	.687	.792
SVM	KL-Div 1G	.648	.839
SVM	KL-Div 20G	.431	.365

compositions in which the overall timbre changes dramatically between sections. Such songs call for summaries over shorter intervals, perhaps at the level of seconds instead of minutes, so that there is enough data to support a rich model, but not so much data that the model averages out interesting detail.

Table 3 also clearly shows the "album effect" in which almost every classifier performs significantly better when trained and tested on songs from the same albums. Depending on the situation, one evaluation might be more useful than the other. For example, if a person hears a song on the radio that he or she likes, it would make sense to look for similar songs that could come from the same album. On the other hand, if a shopper is looking for new albums to buy based on his or her current collection, a recommendation system would want to avoid the training albums.

One reason the KL divergence on GMMs performed so badly might be the number of samples we used in our Monte Carlo estimates of KL divergence. 500 samples is just barely enough to get a reasonable estimate of the KL divergence, but apparently this estimate is too noisy to help SVM or kNN classification. A very good approximation would probably have taken thousands of samples, an therefore ten times as long to compute the 1.4 million element Gram matrix, already pushing the limits of our computational power.

We have shown that audio-based music classification is aided by computing features at the song level and by classifying the features with support vector machines instead of simple k-nearest neighbors classifiers.

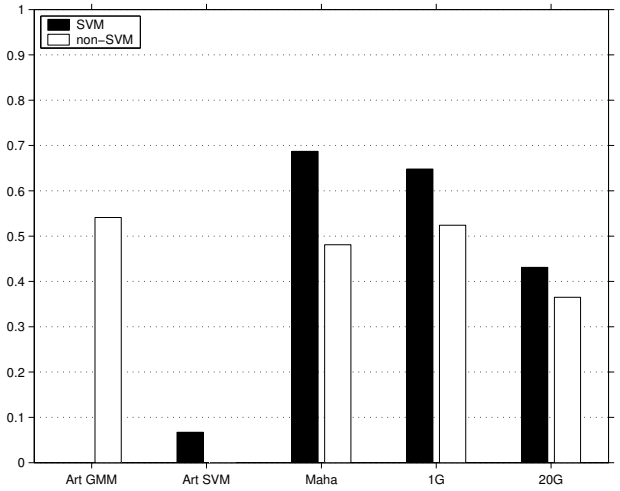

Figure 3: Classification accuracy on separate training and testing albums. From left to right, the columns are: GMMs trained on artist-level features, SVMs trained on artist-level features, and then kNN and SVMs using the Mahalanobis distance, the KL divergence between single Gaussians, and the KL divergence between mixtures of 20 Gaussians.

4.1 Future Work

As a simple extension to this work, we could use a feature mid-way between the song and frame levels. By dividing a song into dozens of pieces, extracting the features of those pieces and classifying them individually, we would get many of the advantages of both approaches. There would be a relatively small number of feature vectors per song, making training and testing fast, and the smaller pieces would be more likely to be timbrally uniform. This division could also allow a classifier to consider a song's temporal structure, employing, for example, a hidden Markov model. Other authors have used hidden Markov models for music classification and description, but the input to those models has been individual MFCCs or spectral slices, not larger structures.

ACKNOWLEDGEMENTS

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

Jean-Julien Aucouturier and Francois Pachet. Improving timbre similarity : How high's the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.

Adam Berenzweig, Beth Logan, Dan Ellis, and Brian Whitman. A large-scale evaluation of acoustic and

subjective music similarity measures. In *International Symposium on Music Information Retrieval*, October 2003.

Nello Cristianini and John Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-78019-5.

Dan Ellis, Adam Berenzweig, and Brian Whitman. The "uspop2002" pop music data set, 2005. http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html.

Jonathan T. Foote. Content-based retrieval of music and audio. In C.-C. J. Kuo, Shih-Fu Chang, and Venkat N. Gudivada, editors, *Proc. SPIE Vol. 3229, p. 138-147, Multimedia Storage and Archiving Systems II*, pages 138–147, October 1997.

Alex Ihler. Kernel density estimation toolbox for matlab, 2005. http://ssg.mit.edu/ ihler/code/.

Beth Logan. Mel frequency cepstral coefficients for music modelling. In *International Symposium on Music Information Retrieval*, 2000.

Beth Logan and Ariel Salomon. A music similarity function based on signal analysis. In *ICME 2001*, Tokyo, Japan, 2001.

Michael I. Mandel, Graham E. Poliner, and Daniel P. W. Ellis. Support vector machine active learning for music retrieval. *ACM Multimedia Systems Journal*, 2005. Submitted for review.

Pedro J. Moreno, Purdy P. Ho, and Nuno Vasconcelos. A kullback-leibler divergence based kernel for SVM classification in multimedia applications. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

Alan V. Oppenheim. A speech analysis-synthesis system based on homomorphic filtering. *Journal of the Acostical Society of America*, 45:458–465, February 1969.

William D. Penny. Kullback-liebler divergences of normal, gamma, dirichlet and wishart densities. Technical report, Wellcome Department of Cognitive Neurology, 2001.

John C. Platt, Nello Cristianini, and John Shawe-Taylor. Large margin dags for multiclass classification. In S.A. Solla, T.K. Leen, and K.-R. Mueller, editors, *Advances in Neural Information Processing Systems 12*, pages 547–553, 2000.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

Kristopher West and Stephen Cox. Features and classifiers for the automatic classification of musical audio signals. In *International Symposium on Music Information Retrieval*, 2004.

Brian Whitman, Gary Flake, and Steve Lawrence. Artist detection in music with minnowmatch. In *IEEE Workshop on Neural Networks for Signal Processing*, pages 559–568, Falmouth, Massachusetts, September 10–12 2001.

Changsheng Xu, Namunu C Maddage, Xi Shao, Fang Cao, and Qi Tian. Musical genre classification using support vector machines. In *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2003.