A Study on Music Genre Classification Based on Universal Acoustic Models

Jeremy Reed
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
jeremy.reed@gatech.edu

Chin-Hui Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
chl@ece.gatech.edu

Abstract

Classification of musical genres gives a useful measure of similarity and is often the most useful descriptor of a musical piece. Previous techniques to use hidden Markov models (HMMs) for automatic genre classification have used a single HMM to model an entire song or genre. This paper provides a framework to give finer segmentation of HMMs through acoustic segment modeling. Modeling each of these acoustic segments with an HMM builds a timbral dictionary in the same fashion that one would create a phonetic dictionary for speech. A symbolic transcription is created by finding the most likely sequence of symbols. These transcriptions then serve as inputs into an efficient text classifier utilized to provide a solution to the genre classification problem. This paper demonstrates that language-ignorant approaches provide results that are consistent with the current state-of-the-art for the genre classification problem. However, the finer segmentation potentially allows for "musical language"-based syntactic rules to enhance performance.

Keywords: musical genres, acoustic segment models, hidden Markov models, latent-semantic indexing

1. Introduction

With the advent of MP3 and other audio coding schemes, music content analysis has become a growing research area. Genre provides a very useful description of a musical piece. However, a lack of label consistency exists in the music community [1]. This leads to difficulties in comparing the performance of genre classification algorithms across databases.

Music genre classification is composed of two basic steps: feature extraction and classification. In the first stage, various features are extracted from the waveform. In the second stage, a classifier is built using the features extracted from the training data. Li and Sleep [2] vector quantized Mel-frequency cepstral coefficients (MFCC), and then used the codebook assignments for each frame as a textual representation of the song. A Lempel-Ziv-type coding algorithm was then utilized to build a modified support vector machine (SVM). In [3], spectral features are extracted and classification is performed using a binary classification tree with each node containing a linear discriminant function (LDF) or single Gaussian classifier. Meng and Shawe-Taylor [4] integrated MFCCs into an autoregressive model to build long-term features, which were placed into a linear neural network and a SVM classifier.

It has been suggested that music genre classification parallels the spoken language identification problem [5]. Just as language governs the syntax of phonemes and words, a musical genre's theoretical structure governs the syntactic order of sounds. For example, the basic 12-bar blues form specifies an ordering of I, IV, and V chords. In other words, music genre imposes syntactic constraints that influence transition probabilities between fundamental acoustic units (notes and chords), which is similar to how language imposes probabilistic constraints on phone and word transitions. In addition, these fundamental units vary in both observational feature values and in duration. In speech, variable-length acoustic units are modeled using hidden Markov models (HMMs) [6]. The variable-length segments is the key difference between this proposed approach and the one found in [2]. In addition, these variable-length segments are treated as fundamental acoustic units in this study, upon which higher-cognitive type approaches may be built to improve results. There have been attempts to incorporate HMMs into the design of genre classifiers. Scaringella and Zoia [7] modeled each genre by a single 4-state HMM, with each state characterized by a Gaussian mixture model (GMM) with 3 mixture components. Aucouturier and Pachet [8] use a single HMM for each song in the training database and associate each test song to the genre model that scores highest. The equivalence to speech would be to model speech at the language (genre) or utterance (song) level. Almost all speech HMMs model much smaller units, e.g. phonemes and words, which parallel musical notes and chords.

A problem in modeling at the note level in music lies in the fact that there are no transcribed databases for music and automatic transcription is not yet a solved problem. Many difficulties exist in creating these corpora, including

time, money, and copyright regulations. Even should transcribed corpora exist one day, many additional problems need to be addressed. For instance, music is not monophonic, but is often composed of multiple instruments playing many notes simultaneously. In order to have a single HMM for each note, source separation would have to become a realizable possibility. If HMMs are constructed for observations of multiple notes being played at once, studies need to be conducted to determine whether the number of states in a note is instrument dependent, if different HMMs need to be constructed for every possible note in a chord or whether key notes such as the root note could serve as an anchor note in identifying the HMM to use for the chord, etc.

This paper argues that a smaller representation based on acoustic segment models (ASMs) [9] is a possible solution until transcribed databases become a realistic alternative. In fact, much of this paper is based on a language identification approach by Ma, *et al*. [10]. A textual transcription of each song is created by finding the most probable sequence of ASMs. Therefore, each song can serve as the musical equivalent of a document that is composed of a vocabulary of symbolic units. These transcriptions allow for more robust text retrieval algorithms, which this paper accomplishes through latent semantic indexing (LSI). This algorithm transforms word counts into multidimensional vectors, which are then used to build the final classifiers. The SVM [11] was used for this study. This approach provides an initial foundation for future improvements through the use of syntax and "musical language"-based rules.

The rest of this paper is organized in accordance with the flow of the algorithm. In Section 2, ASMs are discussed for a musical framework. A musical text-based SVM classifier design is discussed in Section 3. The results, including an analysis across new standard databases is given in Section 4. Finally, our conclusions are given in Section 5.

2. Universal Acoustic Models

Universal acoustic models are based on the idea that an acoustic utterance can be described by a sequence of smaller units, e.g. phones build up to words and sentences. Real signal observations can be considered as noisy representations of these basic units. The models corresponding to these units form a standard set capable of representing every possible combination of sounds. If a labeled training corpus exists, it can be used to train HMMs, as is done in speech. However, no such corpus exists for music. Therefore, an unsupervised approach is utilized.

Assuming that the features extracted from the audio signal accurately describe the various sounds encountered, one would expect that the real, noisy observations of the same fundamental unit to be close by some metric and observations of different units to be far apart. Therefore, the basic units can be found by vector quantizing (VQ) [12] the acoustic space. The resulting clusters are then represented with symbols that serve as entries in an acoustic codebook. Each song is then represented as a sequence of symbolic observations based on distance measures between song segments and codebook entries. This idea of breaking an acoustic utterance into segments and assigning each segment to an entry of a global acoustic codebook is known as tokenization [9].

2.1 Initial Segmentation and transcription

As described in [6], training HMMs requires labeled training data, but since no such data exists for music currently, the ASM approach is used to build initial transcripts. The individual ASMs are a global set that is found by finding clusters of observations in the training data. Because the HMM training process is an iterative process, only a rough initial transcription is needed. Potentially, a better segmentation scheme based on musical analysis and theory can provide better results. However, the focus of this paper is to demonstrate that the ASM approach to segmentation provides results consistent with current solutions. More advanced domain specific-knowledge principles will be investigated in later research.

To find an initial set of ASMs and transcripts, each audio file is first divided into 25 ms, non-overlapping frames that are weighted by a Hamming window. The windows are chosen to be non-overlapping to decrease computation time as this is the most time consuming step in the algorithm. Because later HMM training stages will redefine better boundary locations, it was felt that sacrificing a finer segmentation at this stage for speed was a fair tradeoff. For each audio frame, 8 MFCCs are extracted. This number was chosen empirically to balance between segments that were too short, e.g. every individual frame being labeled as a segment, and segments that were too long, e.g. multiple notes being grouped to form a segment. Intuitively, this makes sense because these low-order MFCCs describe the slowly changing spectral shape [4]. For each song, cepstral mean subtraction [13] and variance normalization [14] have been applied, such that the mean and variance of each coefficient are zero and one, respectively. Successive frames are then grouped into clusters such that they minimize the following distortion function

$$D(O,Q)=\sum_{q=1}^{Q}\sum_{t=b_{q-1}+1}^{b_q} d\left(o_t,\mu_q\right) \qquad (1)$$

where $O = (o_1, o_2, \ldots, o_T)$ are the observation vectors, μ_q is the centroid of the *q-th* segment which ends at b_q ($b_0=0$), and $d(o_t, \mu_q)$ is a distortion metric between o_t and μ_q. This paper uses a simple Euclidean distance metric. The distortion is taken across the Q segments for each song.

The segmentation that minimizes this distortion function can be found using the dynamic time-warping procedure described in [15]. An example of segmented audio for two notes from a song in the RWC Classical Music Database [16] is given in Figure 1.

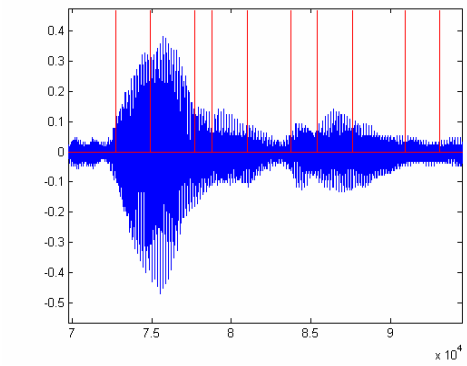

Figure 1. Example of segmentation algorithm output.

The segmentation is very efficient in not only describing the starting and endpoints of the audio, but also is able to describe the rough locations of the transitional parts, such as the attack, sustain, and release.

Every segment is then summarized by the means of the frames that compose the segment. The means from each segment in every training file is used to build a global VQ codebook. A transcript for each training file is then built by identifying the closest codebook entry for each segment. An example of how these transcripts might look is given in Figure 2.

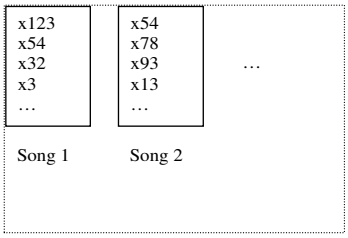

Figure 2. Initial segment model transcripts.

Each line in a file represents a symbolic codebook entry. For example, "x123" is the first "word" in Song 1, "x54" is the second, etc. In this way, each song is sequence of symbols in the same way that a text document or speech transcription is a sequence of words.

2.2 ASM/HMM Training

The transcripts obtained in the previous step provide a starting point for an iterative HMM training process. While the first 8 MFCCs accomplish the task of finding the initial segmentation, it has been found that using higher-order coefficients, energy, and their derivatives and accelerations yield better results for audio and speech content. With this idea in mind, each training and testing audio file is divided using a sliding window of 25 ms taken every 10 ms. Each frame is weighted by a Hamming window to limit edge effects. For each frame, the first 12 MFCCs energy, derivatives, and acceleration coefficients are found to build a 39-dimension feature vector for each frame. Again, cepstral mean subtraction and variance normalization is applied to both the training and testing data. The training data and associated initial transcripts are used to train a set of HMMs (equal to the number of ASMs), with each HMM having 3 states. This number was chosen based on current trends in speech research, but may not be the most ideal choice. More experimentation will be necessary to determine an appropriate number and whether this number is dependent on instruments, style of play, etc. Each state is characterized by a Gaussian mixture model, with the number of mixtures found by increasing the number until no noticeable improvement is found in the performance. For this study, a total of 16 mixtures per state was adequate. For a detailed description of the HMM training process the reader is referred to [6]. After training the HMMs, they are used to re-estimate the transcription of the training files. These transcripts will be different from the original transcripts and are used to further train the HMMs. This process is repeated until only a small amount of improvement is noticed with the training data.

3. Text-Based Classification

The ASM transcription process creates a string of HMM symbols for each training song. These final transcripts can be thought of as a timbral score (as in a "score of music"). This symbolic format allows for the use of proven text classification techniques commonly used in the information retrieval community.

3.1 Acoustic Language

As stated in Section 1, music is structured in its creation, with deviations being used to incite senses of novelty and to prevent boredom. The ASMs produced in Section 2 can be viewed as terms or even as an alphabet of an acoustic language. Their co-occurrences could be seen as syntax, even if on a rough level. While the authors want to caution against the belief that the brain processes music and language in the same fashion, there does seem to be some similarity. This can be seen with music theory, which dictates syntactical usage. One common phenomena in language processing is Zipf's Law [17], which says if one ranks the terms in order of their frequency, *f*, in a large corpus of any language, then the relationship between *f* and the rank, *r*, will be

$$f \propto \frac{1}{r} \qquad (2)$$

A surprising result was found when this applied to the Magnatune[1] database. While the unigrams (appearance of a single symbol) did not show this result, the bigrams (appearance of two symbols in specific order) did, as demonstrated in Figure 3.

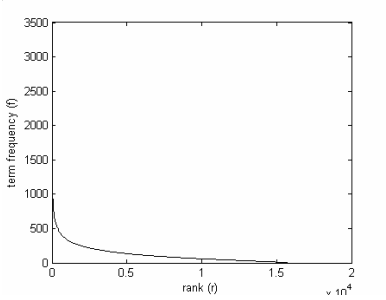

Figure 3. Bigram frequency versus rank for the songs in the Magnatunes training set.

The authors suggest that a possible reason that bigrams exhibit this behavior and why unigrams do not is that music's information is not carried in the individual tones, but in the difference between pairs of tones. That is, it is not the individual sounds which are the basic building blocks, but the pairs of tones which develop the concept of melody. For instance, transposing a melody to another key changes the note names and individual sound, but the sense of melody remains the same.

3.2 Latent Semantic Indexing

One such approach that has proven successful is latent semantic indexing (LSI) [18], which represents a training corpus by a term-document matrix with the rows corresponding to the individual terms and the columns representing the documents. In addition to the unigram counts, bigram counts can also be considered for each document. Therefore, if there are *J=128* terms, then each column (in this case, song) is a vector of size *M = J+J*J=16512*, with *J* unigrams, which are the ASMs described in Section 2, and *J*J* bigrams. Specifically, each element in the matrix, *W*, is given by

$$w_{i,j} = \left(1 - \varepsilon_i \frac{c_{i,j}}{n_j}\right) \tag{2}$$

where $c_{i,j}$ is the number of times that word *i* appears in document *j* and n_j is the number of words in document *j*. The term ε_i is the normalized entropy of word *i* and is given by

$$\varepsilon_i = -\frac{1}{\log T}\sum_{j=1}^{N}\frac{c_{i,j}}{t_i}\log t_i \tag{3}$$

where t_i is the total number of times that word *i* appears in the training documents and *T* is the number of training documents. The word entropy gives a measure of the indexing power. Specifically, values close to zero indicate that the word has more indexing power than words with values close to one because the former appears in fewer documents. Typical examples of maximal indexing words are proper names, while values very close to one are often in function words (e.g. "*the*" and "*a*").

Even for large databases, many bigrams will never appear in the training data. Leaving these values as zero can lead to undesirable results. Therefore, they are often assigned a very small constant or smoothed by some other methods. In general, one implements feature reduction through singular value decomposition [18], however, because current music research databases do not contain a large number of songs, this was seen as unnecessary for this study.

3.3 Evaluation Measures

Three measures of performance are often used: precision, recall, and accuracy. Precision and recall are useful measures for comparing individual genres because they are independent to the number of examples that may exist in each genre. The formulas for these two performance measures are

$$precision = \Pr ec = \frac{tp}{tp + fp} \tag{4}$$

$$recall = \mathrm{Re}\, c = \frac{tp}{tp + fn} \tag{5}$$

$$accuracy = Acc = \frac{tp}{N_t} \tag{6}$$

where *tp*, *fp*, *fn*, and N_t are the true positives, false positives, false negatives, and total number of test queries, respectively.

3.4 Support Vector Machines

Support vector machines (SVM) [11] have proven to be effective in a variety of classification problems. The idea behind SVMs is to project data onto a higher dimensional space in order to separate classes with a LDF, which maximizes the margin between competing classes. The software package SVM^{light} [19] was used for training and classification.

The inputs into the SVMs are the LSI vectors created in the previous step (one vector per song). The output of this classifier is a score, with a positive value indicating one class and a negative value indicating the other class. However, the song genre problem is multi-category problem where each song is assigned to the most likely

[1] http://www.magnatune.com

genre. In order to incorporate SVM into a multi-category problem, the "one-against-one" voting scheme [20] was employed.

4. Experimental Results

4.1 Dataset

For this study, the training and testing files were obtained from Magnatunes, which was used for the 2004 ISMIR Contest. However, HMMs require a lot of data during training; therefore, the RWC [16] and Dortmund [21] databases were added to train the global HMMs. However, different people will label the same song differently. To prevent such labeling inaccuracies from influencing the results, only the final transcripts arising from Magnatunes files were utilized in the creation of the term-document matrix. To create this matrix, each training song was divided into 30-second, non-overlapping segments, and these segments served as the documents with their symbolic unigram and bigram counts serving as the terms. The test songs were divided in a similar fashion to create the test queries. The authors realize that ideally, each song in its entirety would be represented by a single document. However, with the size of current research databases, some genres are underrepresented. In addition, diversity in artists is also needed to adequately describe a genre. Therefore, the files were divided because of the large data demands of SVM classifiers. An artist filter [22] was used so that songs from a single artist were used for either training or testing, but not both. The breakdown is demonstrated in Table 1 and is given in full detail on the first author's website[2] For each genre, the number of individual songs and the total number of 30-second segments for each genre is illustrated.

Table 1. Training and testing databases used for each genre.

	Training		Testing	
Genre	Full	Segment	Full	Segment
Classical	109	580	30	287
Electronic	115	580	30	316
Rock	92	560	30	223
Jazz/Blues	53	430	21	180
Ambient	50	571	28	297

4.2 Behavior of ASM Training

As stated in Section 2, the training of HMMs is an iterative process of finding the ASMs and then creating new transcripts. To view how the testing data responds to this process, the results for the first four iteration rates are shown in Table 2 using 128 ASMs.

Table 2. Accuracy versus iteration number.

Iteration	1	2	3	4
Acc (%)	67.87	69.32	72.14	72.86

[2] http://users.ece.gatech.edu/~jreed/

The accuracy rates increase each time a new set of transcripts for the training data are created and the HMMs are retrained with the new transcripts. There does appear to an asymptotic value close to 73%. This is consistent with previous solutions to this problem and is often cited as the "glass ceiling" of performance which cannot be surpassed without taking higher level cognitive processing into account [8].

To get an accurate view of the SVM training process, the number of support vectors (SV) for the 128-ASM classifiers are listed in Table 3.

Table 3. Number of support vectors for the 128 ASM classifiers.

Classifier type	Num. SV
Classical versus Electronica	488
Classical versus Rock	478
Classical versus Jazz	550
Classical versus Ambient	677
Electronica versus Rock	672
Electronica versus Jazz	558
Electronica versus Ambient	716
Rock verus Jazz	561
Rock versus Ambient	574
Jazz versus Ambient	641

4.3 Genre Confusion

The final confusion matrix is displayed in Table 4 for the SVM maximum vote classifier, where the rows represent the ground truth as labeled in the metadata from Magnatunes and the columns represent how the algorithm classified the test songs. Recall and precision rates are shown as defined in Section 3.3 as well.

Table 4. Final confusion matrix for SVM classifier with C = classical, E = electronic, R = rock, J/B = jazz and blues, and A = ambient

Genre	C	E	R	J/B	A	Rec
C	26	0	1	1	2	86.7
E	0	19	9	0	2	63.3
R	0	5	24	0	1	80.0
J/B	1	2	5	12	1	57.1
A	1	4	2	1	21	72.4
Prec	92.9	63.3	58.5	85.7	77.8	

Most errors occur in just one other class and can be explained by the fact that many songs are not necessarily "strictly jazz", "strictly electronic", etc. For instance, some of the files in the Magnatunes corpus are described as "electronic rock with a pop edge." This may indicate that many of the proposed genre classification schemes need to be extended to allow for multi-topic categorization. Additionally, heuristic clues based on perception and cognition may help in discrimination.

4.4 ASM Size Performance

An important variable is the number of ASMs that are used as unigram terms. If the number of ASMs is too small,

then there will not be enough acoustic coverage. However, too many ASMs will lead to a large dimensionality and requires more training data and computation time. Accuracies, as defined in Section 3.3, using 64 and 128 ASMs after 2 iterations are shown in Table 5.

Table 5. Genre accuracies vs. number of ASMs

64 ASM	128 ASM
55.41%	69.32%

A significant increase in performance (13.91%) can be seen as the number of ASMs increases from 64 to 128.

5. Conclusion

The algorithm we have presented provides comparable results to past solutions of the genre classification problem. However, efficient segmentation of HMM modeling is provided with this approach. Previous use of HMMs for this problem modeled an entire song or genre with a single HMM. If genre classification is comparable to language recognition, then modeling HMMs in this way would equate to having a single HMM for an entire spoken document or entire language. Most speech applications use HMMs on the phonetic level and are therefore able to use syntactic rules to improve classification performance. This study demonstrates that a similar approach may be possible for music, even though labeled training corpora are not in existence.

Using the acoustic segment model idea on music allows for a "timbre dictionary" to be created, which is then used to train HMMs that represent the entire acoustic space. The resulting transcriptions allow for a conversion into a textual transcript so that efficient text retrieval algorithms can then be utilized.

References

[1] F. Pachet and D. Cazaly. "A Taxonomy of Musical Genres," in *Proc. Of Content-Based Multimedia Conf. on Information Access Conf*, April 2000.

[2] M. Li and R. Sleep. "Genre Classification via an LZ78-Based String Kernel," in *ISMIR 2005 Sixth Conf. on Music Inf. Retr. Proc.,* 2005, pp. 252-259.

[3] K. West and S. Cox. "Features and Classifiers for the Automatic Classification of Musical Audio Signals," in *ISMIR 2004 Fifth Conf. on Music Inf. Retr. Proc.,* 2004.

[4] A. Meng and J. Shawe-Taylor. "An Investigation of Feature Models for Music Genre Classification Using the Support Vector Classifier," in *ISMIR 2005 Sixth Conf. on Music Inf. Retr. Proc.,* 2005, pp. 604-609.

[5] A.G. Krishna and T.V. Sreenivas. "Music Instrument Recognition: From Isolated Notes to Solo Phrases," in *ICASSP '04*, vol. 4, pp. 265-268, May 2004.

[6] L.R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition," in *Proc. Of IEEE*, Iss. 2, vol. 77, pp. 257-286, Feb. 1989.

[7] N. Scaringella and G. Zoia. "On the Modeling of Time Information for Automatic Genre Recognition Systems in Audio Signals," in *ISMIR 2005 Sixth Conf. on Music Inf. Retr. Proc.,* 2005, pp. 666-671.

[8] J.-J. Aucouturier and F. Pachet. "Improving Timbre Similarity: How High's the Sky?" in *J. of Negative Results in Speech and Audio Sciences*, vol. 1, 2004.

[9] C.-H. Lee, F.K. Soong, and B.-H. Juang. "A Segment Model Based Approach to Speech Recognition," in *ICASSP '88*, vol. 1, pp. 501-541, 1998.

[10] B. Ma, H. Li, and C.-H. Lee. "An Acoustic Segment Modeling Approach to Automatic Language Identification," in *Interspeech 2005 Eurospeech – 9th European Conf. on Speech Comm. and Technology*, September 4-8, 2005.

[11] C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition," in *Data Mining and Knowledge Discovery*, vol. 2, 121-167, 1998.

[12] Y. Linde, A. Buzo, and R. M. Gray. "An Algorithm for Vector Quantizer Design," in *IEEE Trans. On Comm.*, vol. 28, iss. 1, pp. 85-95, Jan. 1980.

[13] A. Anastasakos, *et. al.* "Adaptation to new microphones using tied-mixture normalization," in *ICASSP '94*, vol. I, pp. 19-22, April 1994.

[14] C.-P. Chen, J. Bilmes, and K. Kirchhoff. "Low-Resource Noise-Robust Feature Post-Processing on the Aurora 2.0/3.0 Databases," in *ICSLP '02*, Denver, Col., 2002.

[15] T. Svendsen and F. Soong. "On the Automatic Segmentation of Speech Signals," in *ICASSP '87*, vol. 12, pp. 77-80, April 1987.

[16] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. "RWC Music Database: Popular, Classical, and Jazz Music Databases", *ISMIR 2002*, pp.287-288, October 2002.

[17] G. K. Zipf. *Human Behavior and the Principle of Least Effort*. Cambridge, MA: Addison-Wesley.

[18] J.R. Bellegarda. "Exploiting Latent Semantic Information in Statisical Language Modeling," in *Proc. IEEE*, vol. 88, no. 8, pp. 1279-1296, 2000.

[19] T. Joachims. *Making large-Scale SVM Learning Practical. Advances in Kernal Methods – Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[20] C.-W. Hsu and C.-J. Lin. "A comparison of methods for multi-class support vector machines," in *IEEE Trans. On Neural Networks*, vol. 13, pp. 415-425, 2002.

[21] H. Helge, I. Mierswa, B. Möller, K. Morik, and M. Wurst. "A Benchmark Dataset for Audio Classification and Clustering," in *ISMIR 2005Sixth Conf. on Music Inf. Retr. Proc.*, pp. 528-531, Sept. 2005.

[22] E. Pampalk, A. Flexer, and G. Widner. "Improvements of Audio-Based Music Similarity and Genre Classification," in *ISMIR 2005Sixth Conf. on Music Inf. Retr. Proc.*, pp. 623-633, Sept. 2005.