

Towards Intelligent Music Information Retrieval

Tao Li and Mitsunori Ogihara, *Member, IEEE*

Abstract—Efficient and intelligent music information retrieval is a very important topic of the 21st century. With the ultimate goal of building personal music information retrieval systems, this paper studies the problem of intelligent music information retrieval. Huron [10] points out that since the preeminent functions of music are social and psychological, the most useful characterization would be based on four types of information: *genre, emotion, style, and similarity*.

This paper introduces Daubechies Wavelet Coefficient Histograms (DWCH) for music feature extraction for music information retrieval. The histograms are computed from the coefficients of the db_8 Daubechies wavelet filter applied to three seconds of music. A comparative study of sound features and classification algorithms on a dataset compiled by Tzanetakis shows that combining DWCH with timbral features (MFCC and FFT), with the use of multi-class extensions of Support Vector Machine, achieves approximately 80% of accuracy, which is a significant improvement over the previously known result on this dataset. On another dataset the combination achieves 75% of accuracy.

The paper also studies the issue of detecting emotion in music. Rating of two subjects in the three bipolar adjective pairs are used. The accuracy of around 70% was achieved in predicting emotional labeling in these adjective pairs.

The paper also studies the problem of identifying groups of artists based on their lyrics and sound using a semi-supervised classification algorithm. Identification of artist groups based on the Similar Artist lists at All Music Guide is attempted. The semi-supervised learning algorithm resulted in non-trivial increases in the accuracy to more than 70%.

Finally, the paper conducts a proof-of-concept experiment on similarity search using the feature set.

Index Terms—Music information retrieval, machine learning, clustering, wavelet, FFT.

I. INTRODUCTION

The rapid growth of the Internet and the advancements of Internet technologies have made it possible for music listeners to have access to a large amount of on-line music data, including music sound signals, lyrics, biographies, and discographies. Music artists in the 21st century are promoted through various kinds of websites that are managed by themselves, by their fans, or by their record companies. Also, they are subjects of discussions in Internet newsgroups and bulletin boards.

This raises the question of whether computer programs can enrich the experience of music listeners by enabling the listeners to have access to such a large volume of on-line music data. Multimedia conferences, e.g. ISMIR (International Conference on Music Information Retrieval) and WEDELMUSIC (Web Delivery of Music), have a focus on the development of computational techniques for analyzing, summarizing, indexing, and classifying music data.

This work is supported in part by NSF grants EIA-0080124, DUE-9980943, and EIA-0205061, and in part by NIH grants RO1-AG18231 (5-25589) and P30-AG18254. T. Li is with School of Computer Science, Florida International University. M. Ogihara is with Department of Computer Science, University of Rochester.

We believe that music information retrieval should be tailored to fit the tastes and needs of individual listeners at the very moment. There are a number of ways for specifying the needs of a listener. In [10] Huron points out that since the preeminent functions of music are social and psychological, the most useful characterization would be based on four types of information: *genre, emotion, style, and similarity*.

This paper studies the question of computationally recognizing these characteristics of music, where parts of the work presented in this paper have appeared in [14], [15], [16], [18], [19]. The four types of characteristics are strongly related to each other. Certain emotional labels prominently apply to music in particular genres, e.g., “angry” for punk music, “depressed” for slow blues, and “happy” for children music. A style is often defined within a genre, e.g., “hard-bop jazz” and “American rock.” Similar music pieces are likely to be those in the same genre, of the same style, and with the same emotional labeling. However, there are traits that distinguish them from the rest. Emotional labeling is transient, in the sense that the labels can be dependent on the state of mind of the listener, and popular music styles are perhaps defined not just in terms of sound signals but in terms of the way the lyrics are written, which is likely beyond the reach of sound feature extraction algorithms.

An important step in studying the problem of recognizing the above four features is specifying the input-output relation. Of particular importance is the determination of the features extracted from audio signals. The features have to be *comprehensive* in the sense that they represent the music very well, *compact* in the sense that they require much smaller storage space than the raw acoustic data, and *efficient* in the sense that computation can be carried out efficiently. We propose here a new feature extraction method, *DWCH* (Daubechies Wavelet Coefficient Histograms), which are based on wavelet coefficients histogram. By computing histograms of Daubechies wavelet coefficients at different frequency sub-bands, DWCH is expected to extract both local and global information of input audio signals. This representation is readily very compact and efficient. Its comprehensiveness is tested here on the subject of *music genre classification*. We find that the DWCH features become the most comprehensive combined with timbral features and that the accuracy of the combined features significantly improve previous known best results.

This discovery encourages us to study the usefulness of the combined feature set in recognizing the emotion aroused in the listeners and in identifying similarities between music pieces. Our experiments show that the feature set does a very reasonable job in both tasks.

We then tackle the problems of style recognition. Here we choose to study the problem of recognizing the style of singer-

song-writers, i.e., those who write and perform songs, because the music style, however subtly it is defined, would be more richly represented in the music of singer-song-writers than would in the music of singers that sing someone else's tunes. We study the problem of distinguishing a group of artists deemed "similar" by a reliable music database (All Music Guide) by other artists based on the combined acoustic (DWCH + timbral) features and features extracted from lyrics, and obtain reasonably good results. We also conduct a proof-of-concept experiment on music similarity search.

The rest of the paper is organized as follows: Sections II and III describe the acoustic feature extraction algorithms we use. Sections IV, V, VI, and VII present the study of the four problems. Section VIII presents our conclusions and discussions of feature problems.

II. ACOUSTIC FEATURES OF MUSIC

Much work on extraction of features from music has been devoted to timbral texture features, rhythmic content features, and pitch (melody) content features. The MARSYAS system of Tzanetakis and Cook [35]¹, is a software package that allows one to use all of these features.

a) Rhythmic Content Features (Beat): Rhythmic content features are those that characterize the regularity of the rhythm, the beat, the tempo, and the time signature. The feature is calculated by extracting periodic changes from the beat histogram. Computation of the beat histogram goes through an elaborate process of identifying peaks in autocorrelation. We select two highest peaks and compute: their relative amplitudes to the overall average, the ratio between the relative amplitudes, and the period length of each. By adding the overall average of the amplitude, a total of six features are calculated.

b) Pitch Content Features (Pitch): The pitch content features describe the distribution of pitches. Here the features are calculated from the pitch histograms, laid out in the circle of fifths. As in rhythmic content features, we select the two highest peaks and then compute the distance between the two, the ID of the highest peak, the amplitude of the highest peak, and the period of the highest peak in the unfolded histogram. Thus, there are a total of four features.

c) Timbral Textural Feature: Timbral textural features are those used to differentiate mixtures of sounds based on their instrumental compositions when the melody and the pitch components are similar. The use of timbral textural features originates from speech recognition [29]. Extracting timbral features requires preprocessing of the sound signals. The signals are divided into statistically stationary frames, usually by applying a window function at fixed intervals. The application of a window function removes the so-called "edge effects." Popular window functions including the Hamming window function and the Blackman window function.

- **Mel-Frequency Cepstral Coefficients (MFCC)** MFCC is a feature set popular in speech processing and music modeling [20]. This feature is obtained as follows: We first compute, for each frame, the logarithm of the amplitude spectrum based on short-term Fourier transform,

where the frequencies are divided into thirteen bins using the Mel-frequency scaling. (The "cepstrum" is the FFT of this logarithm.) Then we apply discrete cosine transform to de-correlate the Mel-spectral vectors. In this study, we use the first five bins, and compute the mean and variance of each over the frames.

- **Short-Term Fourier Transform Features (FFT)** This is a set of features related to timbral textures and is not captured using MFCC. It consists of the following five types. More detailed descriptions can be found in [35]. *Spectral Centroid* is the centroid of the amplitude spectrum of short-term Fourier transform and it is a measure of spectral brightness. Formally, the spectral centroid C_t is defined as

$$C_t = \frac{\sum_{n=1}^N M_t[n] * n}{\sum_{n=1}^N M_t[n]},$$

where $M_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n .

Spectral Rolloff is the frequency below which 85% of the amplitude distribution is concentrated. It measures the spectral shape. Formally, for spectral rolloff R_t , we have

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^N M_t[n].$$

Spectral Flux is the squared difference between the normalized amplitudes of successive spectral distributions. It measures the amount of local spectral change. Formally, the spectral flux F_t is defined as

$$F_t = \sum_{n=1}^N (N_t[n] - N_{t-1}[n])^2,$$

where $N_t[n]$ and $N_{t-1}[n]$ are the normalized magnitude of the Fourier transform at the current frame t , and the previous frame $t - 1$, respectively.

Low Energy is the percentage of frames that have energy less than the average energy over the whole signal. It measures the amplitude distribution of the signal.

Zero Crossings is the number of time domain zero crossings of the signal. It measures the noisiness of the signal. Formally, the time domain zero crossings Z_t is defined as

$$Z_t = \frac{1}{2} \sum_{n=1}^N |\text{sgn}(x[n]) - \text{sgn}(x[n-1])|,$$

where $x[n]$ is the time domain signal for frame t and sgn is the sign function (e.g., it takes the value of 1 for positive arguments and 0 otherwise.)

We compute the mean for all five and the variance for all but zero crossings. So, there are a total of nine features.

III. DWCH

A. The Wavelet Transform

The wavelet transform is a synthesis of ideas that have emerged over many years in such different fields as mathematics and image/signal processing. It has been widely used

¹Available at <http://marsyas.sourceforge.net/>.

in information retrieval and data mining [13], [21], [22] (see also [13] for a survey). In general, the wavelet transform provides good time and frequency resolution. Like Fourier transforms, a wavelet transform is viewed as a tool for dividing data, functions, or operators into different frequency components and then analyzing each component with a resolution matched to its scale [4]. The contribution of each component is represented as a coefficient. While each component of a Fourier transform is the wave of a fixed frequency, each component of a wavelet transform is the wave of a time-dependent frequency function. A wavelet coefficients histogram is the histogram of the (rounded) wavelet coefficients obtained by convolving a wavelet filter with an input music signal (details on wavelet histograms and on wavelet filters and analysis can be found in [33] and in [4], respectively).

Generally speaking, wavelets are designed to give good time resolution at high frequencies and good frequency resolution at low frequencies. They have several favorable properties, including compact support, vanishing moments and de-correlated coefficients and have been successfully applied in signal representation and transformation [7]. Compact support guarantees the localization of wavelet, vanishing moment property allows wavelet focusing on most important information and discarding noisy signal, and de-correlated coefficients property enables wavelet to reduce temporal correlation so that the correlation of wavelet coefficients are much smaller than that of the corresponding temporal process [7]. Hence, after wavelet transform, the complex signal in the time domain can be reduced into a much simpler process in the wavelet domain. By computing the histograms of wavelet coefficients, we could obtain a good estimation of the probability distribution over time. The good probability estimation thus leads to a good feature representation.

B. The DWCH Feature Extraction Method

A sound file is a kind of oscillation waveform in the time domain. It can be considered as a two-dimensional entity of the amplitude over time, in the form of $M(t) = D(A, t)$, where A is the amplitude, generally ranging from $[-1, 1]$. This variation in amplitudes makes wavelets useful in distinguishing a sound signal from others. Since identifying amplitude variations is a key in signal-based music analysis, one can expect wavelets to be useful for music classification.

The histogram technique is an efficient means for estimating a distribution. However, the raw signal in the time domain is not a good representation, particularly for the purpose of content-based categorization because the most distinguished characteristics are hidden in the frequency domain. For sound signals, the frequency domain is generally (and naturally) divided into octaves where each octave has a unique quality. An octave is an interval of frequencies in which the ratio of the highest and the lowest is 2, so it is a unit interval in the frequency domain in the logarithmic scale. The wavelet decomposition scheme matches the models of sound octave division for perceptual scales, and thus, provides good time and frequency resolutions [12]. In other words, the decomposition of audio signal using wavelets produces a

set of subband signals at different frequencies corresponding different characteristics.

Formally, a mother wavelet is a function $\psi(x)$ such that $\{\psi(2^j x - k), i, k \in Z\}$ is an orthogonal basis of $L^2(R)$. The basis functions are usually referred to as wavelets. There are many kinds of wavelet filters, including Daubechies wavelet filters and Gabor filter. Daubechies wavelet filters are the ones commonly used in image retrieval. In general, db_n represents the family of Daubechies Wavelets and n is the order. The family includes Haar wavelet since Haar wavelet represents the same wavelet as db_1 . To find wavelets, start with a scaling function $\phi(x)$ that is made up of a smaller version of itself

$$\phi(x) = \sum_{k=-\infty}^{\infty} a_k \phi(2x - k). \quad (1)$$

Here a_k 's are called filter coefficients or masks. Under certain conditions [4],

$$\begin{aligned} \psi(x) &= \sum_{k=-\infty}^{\infty} (-1)^k b_k \phi(2x - k) \\ &= \sum_{k=-\infty}^{\infty} (-1)^k \bar{a}_{1-k} \phi(2x - k) \end{aligned} \quad (2)$$

gives a wavelet, where \bar{a} is the conjugate of a .

Generally it can be shown that [13]

- The support for db_n is on the interval $[0, 2n - 1]$.
- The wavelet db_n has n vanishing moments [4].
- The regularity increases with the order. db_n has rn continuous derivatives (r is approximately 0.2).

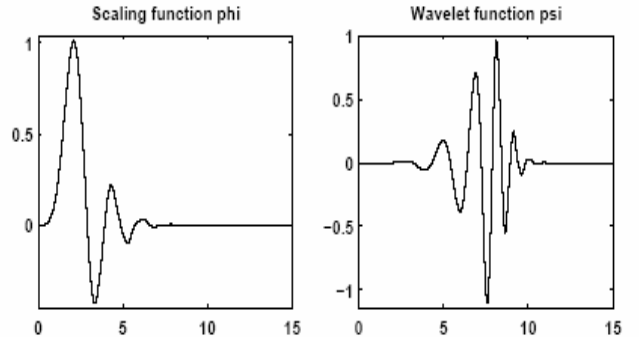


Fig. 1. Daubechies-8(db_8) Wavelet

The db_8 Daubechies wavelet filter with seven levels of decomposition, which is shown in Figure 1, is used in our experiments. After the decomposition, we construct the histogram of the wavelet coefficients at each subband. We use the subband energy, defined as the mean of the absolute value of coefficients, and the first three moments, i.e., the average, the variance, and the skewness. We calculate these four quantities for each subband. Therefore, there are 28 features.

We choose to obtain these features from the sound signals over three consecutive seconds in a given music file. Based

on an intuition that music is somewhat similar to itself, such a short duration is picked. In summary, the DWCH algorithm works as follows:

- 1) Obtain the db_8 wavelet decomposition of the input sound file.
- 2) Construct the histogram of each of the seven subbands.
- 3) Compute the first three moments of each subband.
- 4) Compute the subband energy of each subband.

The algorithm is very easy to implement in Matlab, which contains a complete wavelet package.

IV. GENRE CLASSIFICATION: A COMPARATIVE STUDY

We study the automatic music genre classification problem. We define this as the problem of classifying a given piece of music into a unique class solely based on its audio contents. There are two major issues in dealing with this problem: the features extracted from a given piece of music and the classification algorithm to be used. Here we conduct comparison of the five acoustic feature sets, Beat, FFT, MFCC, Pitch and DWCH, and of various multi-class classification algorithms.

A. The Datasets

We use two datasets for our experiments. The first dataset, Dataset A of Tzanetakis and Cook [35], consists of 1,000 30-second-long sound files covering ten genres with 100 files per genre. The ten genres are Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae, and Rock. The files are collected from radio and CD's. The second dataset, Dataset B, consists of 756 sound files covering five genres: Ambient, Classical, Fusion, Jazz, and Rock. The dataset was constructed by the authors from the CD collection of the second author. The files are collected from 189 music albums as follows: From each album the first four music tracks were chosen (three tracks from albums with only three music tracks); then from each music track the sound signals over a period of 30 seconds after the initial 30 seconds were extracted in the MP3 format. The distribution of the files in the five genres is: Ambient (109 files), Classical (164 files), Fusion (136 files), Jazz (251 files) and Rock (96 files). For both datasets, each sound file is converted to a 22,050Hz, 16-bit, mono audio file.

B. Experiment Setup

To extract the Beat, FFT, MFCC and Pitch feature sets we use the MARSYAS system. To extract the DWCH feature set we use our implementation with Matlab. The DWCH set consists of four quantities for each of the seven frequency subbands. A few trials reveal that of the seven subbands of db_8 (1: 11025–22050 Hz, 2: 5513–11025Hz, 3: 2756–5513Hz, 4: 1378–2756Hz, 5: 689–1378Hz, 6: 334–689Hz, 7: 0–334Hz), subbands 1, 2, and 4 show little variation. We thus choose to use only the remaining four subbands, 3, 5, 6, and 7, for our experiments. In fact, The subbands match the models of sound octave-division for perceptual scales [12].

We test various classification algorithms for the actual classification: GMM (Gaussian Mixture Models) with three Gaussians, KNN (k-Nearest Neighbors) with $k = 5$, LDA

(Linear Discriminant Analysis), and multi-class extensions of support vector machines (SVM). Support vector machines (SVM) [36] is a method that has shown superb performance in binary classification problems. Intuitively, it aims at searching for a hyperplane that separates the positive data points and the negative data points with maximum margin. The method was originally designed as a binary classification algorithm. Several binary decomposition techniques are known. We use one-against-the-rest (denoted by S1) and pairwise (denoted by S2), which assemble judgments respectively of the classifiers for distinguishing one class from the rest and of the classifiers for distinguishing one class from another. We also use a multi-class objective function version of SVM, MPSVM [8] (we use short-hand MPS to refer to this algorithm), which can directly deal with multi-class problems. For S1 and S2, our SVM implementation is based on the LIBSVM [3], a library for support vector classification and regression. For experiments involving SVM, we test linear, polynomial, and radius-based kernels. The results we show are the best of the three kernel functions.

K-Nearest Neighbors (KNN) is a non-parametric classifier. Theoretical results show that its error is asymptotically at most twice as large as the Bayesian error rate. KNN has been applied to various music sound analysis problems. Given K as a parameter, it finds the K nearest neighbors among training data and uses the categories of the K neighbors to determine the class of a given input. We use the parameter K to 5.

Gaussian Mixture Models (GMM) is a method that has been widely used in music information retrieval. The probability density function (pdf) for each class is assumed to consist of a mixture of a number of multidimensional Gaussian distributions. The iterative expectation-minimization (EM) algorithm is then used to estimate the parameters of each Gaussian component and the mixture weights.

Linear Discriminant Analysis (LDA) works by finding a linear transformation that best discriminates among classes. The classification is then performed in the transformed space using some metric such as Euclidean distances.

C. Results and Analysis

Table I shows the accuracy of the various classification algorithms on Dataset A. This table is based on the data for all combinations of the standard four feature sets. The results confirm that each of the four standard feature sets contains useful information characterizing music signals. The classification accuracy on any single feature set is significantly better than random guessing, which is 10% for this dataset. However, the information extracted in the four sets is insufficient for accomplishing highly accurate genre classification because even with all the four combined (the row B/F/M/P) the accuracy is never more than 72% and mostly in the 60% range. Note that in each of the classification algorithms tested, the performance on features set combinations including at least one of FFT and MFCC is significantly higher than the performance on combinations with the FFT and MFCC replaced by Beat and/or Pitch. We can conclude thus that FFT and MFCC are each better than Beat and Pitch combined. Also, note that if both

Features	Methods					
	S1	S2	MPS	GMM	LDA	KNN
D/F	74.9	78.5	68.3	63.5	71.3	62.1
/M	(4.97)	(4.07)	(4.34)	(4.72)	(6.10)	(4.54)
B/F	70.8	71.9	66.2	61.4	69.4	61.3
/M/P	(5.39)	(5.09)	(5.23)	(3.87)	(6.93)	(4.85)
B/F	71.2	72.1	64.6	60.8	70.2	62.3
/M	(4.98)	(4.68)	(4.16)	(3.25)	(6.61)	(4.03)
B/F	65.1	67.2	56.0	53.3	61.1	51.8
/P	(4.27)	(3.79)	(4.67)	(3.82)	(6.53)	(2.94)
B/M	64.3	63.7	57.8	50.4	61.7	54.0
/P	(4.24)	(4.27)	(3.82)	(2.22)	(5.23)	(3.30)
F/M/	70.9	72.2	64.9	59.6	69.9	61.0
/P	(6.22)	(3.90)	(5.06)	(3.22)	(6.76)	(5.40)
B/F	61.7	62.6	50.8	48.3	56.0	48.8
	(5.12)	(4.83)	(5.16)	(3.82)	(6.73)	(5.07)
B/M	60.4	60.2	53.5	47.7	59.6	50.5
	(3.19)	(4.84)	(4.45)	(2.24)	(4.03)	(4.53)
B/P	42.7	41.1	35.6	34.0	36.9	35.7
	(5.37)	(4.68)	(4.27)	(2.69)	(4.38)	(3.59)
F/M	70.5	71.8	63.6	59.1	66.8	61.2
	(5.98)	(4.83)	(4.71)	(3.20)	(6.77)	(7.12)
F/P	64.0	68.2	55.1	53.7	60.0	53.8
	(5.16)	(3.79)	(5.82)	(3.15)	(6.68)	(4.73)
M/P	60.6	64.4	53.3	48.2	59.4	54.7
	(4.54)	(4.37)	(2.95)	(2.71)	(4.50)	(3.50)
B	26.5	21.5	22.1	22.1	24.9	22.8
	(3.30)	(2.71)	(3.04)	(1.91)	(2.99)	(5.12)
F	61.2	61.8	50.6	47.9	56.5	52.6
	(6.74)	(3.39)	(5.76)	(4.91)	(6.90)	(3.81)
M	58.4	58.1	49.4	46.4	55.5	53.7
	(3.31)	(4.72)	(2.27)	(3.09)	(3.57)	(4.11)
P	36.6	33.6	29.9	25.8	30.7	33.3
	(2.95)	(3.23)	(3.76)	(3.02)	(2.79)	(3.20)

TABLE I

THE CLASSIFICATION ACCURACY (IN %) OF THE LEARNING METHODS TESTED ON DATASET A USING VARIOUS COMBINATIONS OF FEATURES. B, D, F, M, AND P RESPECTIVELY STAND FOR BEAT, DWCH, FFT, MFCC, AND PITCH. THE ACCURACY IS CALCULATED VIA TEN-FOLD CROSS VALIDATION. WITHIN PARENTHESES ARE STANDARD DEVIATIONS.

FFT and MFCC are included then adding Beat or Pitch does not significantly increase the accuracy. Thus, in the presence of FFT and MFCC, we can dispense with Beat and Pitch.

We then test the accuracy of adding DWCH to the combination of FFT and MFCC. The addition increases accuracy with respect to each classification method. The increase is significant for S1 and S2. The accuracy for the latter is 78.5% on the average in the ten-fold cross validation, where the accuracy higher than 80% is achieved for some trials. This is a remarkable improvement from 61% which was achieved by Tzanetakis and Cook [35]. The superiority of DWCH+FFT+MFCC can be seen from the graph shown in Figure 2.

The average accuracy of the one-versus-the-rest classifiers over a ten-fold cross-validation test is very high for all ten

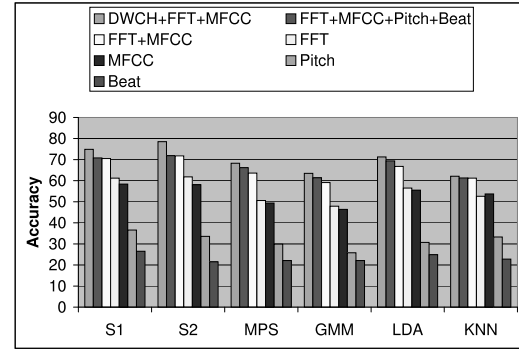


Fig. 2. The classification accuracy of the learning methods tested on Dataset A using various combinations of features. The accuracy values are calculated via ten-fold cross validation.

classes (ranging from 91 to 99 %). The most accurate of the ten classes is Classical. The Classical sound files of this dataset seem to have significant timbral difference from other sound files: bowed string instruments are prominent and percussions are rarely heard. We suspect that this timbral difference has made it very easy to distinguish the Classical pieces from the others. Similarly, in the Jazz pieces brass instruments are frequently heard and there is strong presence of rhythm. That could have made it easy to distinguish this class. On the other hand, Rock, Reggae, and Disco have accuracy lower than the others (but it is still above 90%). This can be attributed to the fact that the three classes seem to be close to each other in terms of timbral features (the use of drums, electric bass, electronic guitar, and electric keyboard).

Perrot and Gjerdingen [26] report a human subject study in which college students were trained to learn a music company's genre classification on a ten-genre data collection, where the trained students achieved about 70% accuracy. Our results cannot be directly compared against their results because the datasets are different, but one can clearly say that the precision of the classification achieved here is satisfyingly high.

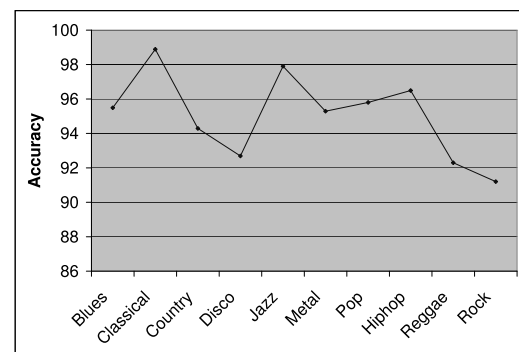


Fig. 3. The genre specific accuracy of the S1 method with DWCH, FFT and MFCC. The results are calculated via ten-fold cross validation.

Some reports better accuracy than ours in automatic music genre recognition of smaller datasets. Pye [28] reports 90%

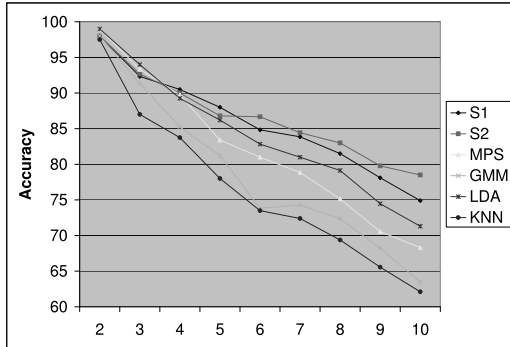


Fig. 4. The accuracy of the S1 method with DWCH, FFT and MFCC on various subsets of Dataset A. The accuracy values are calculated via ten-fold cross validation. The X-axis represents the number of genres.

Classes	Methods					
	S1	S2	MPS	GMM	LDA	KNN
1,2	98.0 (3.50)	98.0 (2.58)	99.0 (2.11)	98.0 (3.22)	99.0 (2.11)	97.5 (2.64)
1-3	92.3 (5.46)	92.6 (4.92)	93.3 (3.51)	91.3 (3.91)	94.0 (4.10)	87.0 (5.54)
1-4	90.5 (4.53)	90.0 (4.25)	89.7 (3.99)	85.2 (5.20)	89.2 (3.92)	83.7 (5.92)
1-5	88.0 (3.89)	86.8 (4.54)	83.4 (5.42)	81.2 (4.92)	86.2 (5.03)	78.0 (5.89)
1-6	84.8 (4.81)	86.6 (5.27)	81.0 (6.05)	73.8 (5.78)	82.8 (6.37)	73.5 (6.01)
1-7	83.8 (4.26)	84.4 (3.53)	78.8 (3.67)	74.2 (6.90)	81.0 (5.87)	73.2 (5.88)
1-8	81.5 (4.56)	83.0 (3.64)	75.1 (4.84)	72.3 (6.22)	79.1 (6.07)	69.3 (5.47)
1-9	78.1 (4.83)	79.7 (2.76)	70.5 (4.30)	68.2 (7.26)	74.4 (6.22)	65.5 (4.66)

TABLE II

THE ACCURACY (IN %) OF THE S1 METHOD WITH DWCH, FFT AND MFCC ON VARIOUS SUBSETS OF DATASET A. THE ACCURACY IS CALCULATED VIA TEN-FOLD CROSS VALIDATION. WITHIN PARENTHESES ARE STANDARD DEVIATIONS.

of accuracy on a dataset of 175 music tracks covering six genres (Blues, Easy Listening, Classical, Opera, Dance, and Indie Rock). Soltau, Schultz, and Westphal [31] report 80% of accuracy on a dataset of four classes (Rock, Pop, Techno, and Classical). Just for the sake of comparison, Figure 3 shows the accuracy of the best classifier (DWCH, FFT and MFCC with S1) on each of the ten music genres of Dataset A. The accuracy is extremely high. Table II and Figure 4 show the accuracy of the multi-class classification for distinguishing among smaller numbers of classes. The classes one through ten respectively correspond to: Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae and Rock. The accuracy gradually decreases as the number of classes increases.

In Dataset A multiple segments are extracted from a single piece of recording. This may raise a question of whether

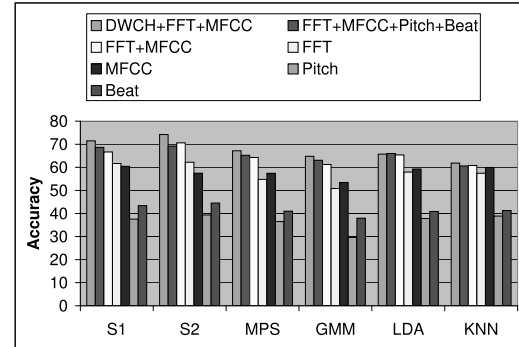


Fig. 5. The classification accuracy of the learning methods tested on Dataset B using various combinations of features. The accuracy values are calculated via ten-fold cross validation.

the high accuracy achieved by the classifier is very narrowly trained to recognize only the artists and/or albums presented in the dataset. That motivates us to create our own dataset (Dataset B) in which a large number of artists and albums are represented so as to avoid artist/album specific training of the classifier (Dataset B consists of 189 albums over five genres, covering many different styles and artists).

Figure 5 shows the results on Dataset B. Recall that this dataset was generated with little control (from each piece the 30 seconds after the initial 30 seconds are used) and it covers a large number of albums. So, the accuracy can be expected to be lower than that on Dataset A. Also, the inclusion of intermediate genres, Ambient and Fusion, which respectively sit between Jazz and Classical and between Jazz and Rock, may reduce the accuracy. The difficulty in classifying such borderline cases is somewhat compensated for by the reduction in the number of classes. In reality, the difference is 4.0–5.0%.

V. EMOTION DETECTION

Relations between musical sounds and their impact on the emotion of the listeners have been studied for decades. The celebrated paper of Hevner [9] studied this relation through experiments in which the listeners are asked to write adjectives that came to their minds as the most descriptive of the music played. The experiments confirmed a hypothesis that music inherently carries emotional meaning. Hevner discovered the existence of clusters of descriptive adjectives and laid them out (there were eight of them) in a circle. She also discovered that the labeling is consistent within a group having a similar cultural background. The Hevner adjectives were refined and regrouped into ten adjective groups by Farnsworth [6].

The hypothesis that musical sounds arouse emotion is also substantiated by a recent paper by Peretz, Gagnon, and Bouchard [25], which shows that distinction between sad and happy music sounds is unaffected in listeners with brain damage, implying that emotional reaction to music is firmly grounded in our brain. These discoveries make us to hypothesize that emotion detection in music can be made by analyzing music signals. Our goal is to treat the emotion detection problem as a multi-label classification problem.

We cast the emotion detection problem as a *multi-label classification problem*, where the music sounds are classified into multiple classes simultaneously. That is a single music sound may be characterized by more than one label, e.g., both “dreamy” and “cheerful.”

We resort to the scarcity of literature in multi-label classification by decomposing the problem into a set of binary classification problems. In this approach, for each binary problem a classifier is developed using the projection of the training data to the binary problem. To determine labels of a test data, the binary classifiers thus developed are run individually on the data and every label for which the output of the classifier exceeds a predetermined threshold is selected as a label of the data. See [30] for similar treatments in the text classification domain. To build classifiers we used Support Vector Machines [36].

A. The Dataset and Emotional Labeling

A subset consisting of 235 instrumental jazz tracks from Dataset B is used for the experiment. The files are labeled independently by two subjects: a 39 year old male (subject 1) and a 25 year old male (subject 2). Each track is labeled using a scale ranging from -4 to $+4$ on each of three bipolar adjective pairs: (Cheerful versus Depressing), (Relaxing versus Exciting), and (Comforting versus Disturbing), where 0 is thought of as neutral. Our early work on emotion labeling [14] uses binary labels (existence versus non-existence) based on the adjective groups of Farnsworth. The classification accuracy is not very high (around 60%). The low accuracy can be attributed to the presence of many labels to choose from. The recent experiments conducted by Leman *et al.* [24] using scales on ten bipolar adjective pairs suggest that variations in emotional labeling can be approximated using only spanned three major principal components, which are hard to name. With these results in mind we decided to generate three bipolar adjective pairs based on the eight adjective groups of Hevner.

B. Experiments

The accuracy of the performance is presented in Table III. Here the accuracy measure is the Hamming accuracy, that is, the ratio of the number of True Positives and True Negative against the total number of inputs. In each measure, the tracks labeled 0 are altogether put on either the positive side or the negative side. It is clear that the accuracy of detection was always at least 70% and sometimes more than 80%. Also, there is a large gap in the performance between the two subjects on the first two measures. We observe that this difference is coming from the difference in the cultural background of the subjects. To deal with labeling of a much larger group of listeners one should cluster them into groups depending on their labeling and train the emotion detection system for each group.

VI. STYLE IDENTIFICATION

This section addresses the issue of identifying the artist style. Ellis *et al.* [27] point out that similarity between artists

Subject	Cheerful vs. Depressing	Relaxing vs. Exciting	Comforting vs. Disturbing
1	83.3 (8.0)	70.4 (9.9)	72.4 (5.1)
2	69.6 (10.0)	83.7 (7.3)	70.9 (9.1)

TABLE III

THE ACCURACY (IN %) OF EMOTION DETECTION. WITHIN PARENTHESES ARE STANDARD DEVIATIONS.

reflects personal tastes and suggest that different measures have to be combined together so as to achieve reasonable results in similar artist discovery. We focus our attention to singer-song-writers, i.e., those who sing their own compositions. We take the standpoint that the artistic style of a singer-song-writer is reflected both in the acoustic sounds and in the lyrics. We therefore hypothesize that the artistic styles of an artist can be captured better by combining acoustic features and linguistic features of songs than by using only one type of features. We study this question by examining the accuracy of search for similar artists. Although we believe that the degree at which a listener finds a piece of music similar to another is influenced by the listener’s cultural and music backgrounds and by the listener’s state of mind, to make our investigation more plausible we choose to use the Similar Artist lists available at All Music Guide (www.allmusic.com). In our experiments two artists are thought of as similar if this guide asserts one to be an artist similar to the other on the Similar Artist lists. We take the standpoint that the artist similarity information in this guide summarizes the opinions of one or more listeners.

A. Lyrics-Based Feature Sets

Previous study on stylometric analysis has shown that statistical analysis on text properties could be used for text genre identification and authorship attribution [1], [11], [32] and over one thousand stylometric features (style makers) have been proposed in variety research disciplines [34]. To choose features for analyzing lyrics, one should be aware of some characteristics of popular song lyrics. For example, song lyrics are usually brief and are often built from a very small vocabulary. In song lyrics, words are pronounced with melody, so the construction of melody lines and that of words are closely tied to each other. Also, the stemming technique, though useful in reducing the number of words to be examined, may have a negative effect. Furthermore, in song lyrics, word orders are often different from those in conversational sentences, and song lyrics are often presented without punctuation.

To accommodate the characteristics of the lyrics, our text-based feature extraction consists of four components: bag-of-words features, Part-of-Speech statistics, lexical features and orthographic features. The features are listed in Table IV.

- *Bag-of-words*: We compute the TF-IDF measure for each word and select top 200 words as our features. Stemming operations are not applied.
- *Part-of-Speech statistics*: We use the output of the part-of-speech (POS) tagger by Brill [2] as the basis for

feature extraction. The POS statistics usually reflect the characteristics of writing. There are 36 POS features extracted from each document, one for each POS tag expressed as a percentage of the total number of words for the document.

- *Lexical Features*: By “lexical features” we mean the features of individual word-tokens in the text. The most basic lexical features are lists of 303 generic function words taken from [23]², which generally serve as proxies for choice in syntactic (e.g., preposition phrase modifiers vs. adjectives or adverbs), semantic (e.g., usage of passive voice indicated by auxiliary verbs), and pragmatic (e.g., first-person pronouns indicating personalization of a text) planes. Function words have been shown to be effective style markers.
- *Orthographic features*: We also use orthographic features of lexical items, such as capitalization, word placement, word length distribution as our features. Word orders and lengths are very useful since the writing of lyrics usually follows certain melody.

Type	Size	Type	Size
Bag-Of-Words	200	Part-of-Speech	36
Function Words	303	Token Place	5
Start of . . .	9	Capitalization	10
Word Length	6	Line Length	6
Ave. Word Length	1	Ave. Sentence Length	1

TABLE IV

A SUMMARY OF THE FEATURE SETS FOR ANALYZING LYRICS.

B. Semi-supervised Learning

In the presence of sound-based and lyrics-based features, the identification problem of artistic styles based on sound and lyrics falls into the realm of learning from heterogeneous data. Here we take a semi-supervised learning approach, in which a classification algorithm is trained for each feature set but the target label is adjusted for input data so as to minimized disagreement between the classifiers.

1) *Minimizing Disagreement*: Suppose we have an instance space $X = (X_1, X_2)$ where X_1 and X_2 are from different observations. Let D be the distribution over X . If f is the target function over D , then for any example $x = (x_1, x_2)$ we would have $f(x_1, x_2) = f_1(x_1) = f_2(x_2)$ where f_1 and f_2 are the target functions over X_1 and X_2 respectively. It has been shown that minimizing the disagreement between two individual models could lead to the improvement of the classification accuracy of individual models [17].

Theorem 1: [17] Under a conditional independence assumption, the disagreement upper bounds the misclassification error for the nontrivial classifier.

2) *Co-updating*: Based on Theorem 1, we developed a co-updating approach to learn from both labeled and unlabeled data which aims to minimizing the disagreement on unlabeled

data. The approach is an iterative Expectation-Maximization (EM)-type procedure. Its basic idea is as follows: The labeled samples are first used to obtain weak classifiers, f_1^0 on X_1 and f_2^0 on X_2 . Then we repeat a two-step process: in the expectation step we use the current classifiers to predict the labels of unlabeled data, and then in the maximization step we rebuild the classifiers using the labeled samples and a random collection of unlabeled samples on which the classifiers agree (i.e., they have the same predictions). We repeat the two-step process until some termination condition holds. The intuition behind the co-updating algorithm is that we stochastically select the unlabeled samples on which the two component classifiers agree and confident, and then use them along with the labeled samples to train/update the classifiers. The approach iteratively updates classifier models by using current models to infer (a probability distribution on) labels for unlabeled data and then adjusting the models to fit the (distribution on) filled-in labels. For more detail of the algorithm see [17].

C. The Dataset

56 albums of a total of 43 artists are selected. The sound recordings and the lyrics from them are obtained. Similarity between artists is identified by examining their All Music Guide pages. If the name of an artist X appears on the “Similar Artist” of the web page of another artist Y, then X and Y are thought of as similar. Based on this relation, artists having a large number of neighbors are selected. There are three of them, Fleetwood Mac, Yes, and Utopia. These three artists form a triangle, so the neighbors of these three are chosen as a group. Of the remaining nodes two groups are identified in a similar manner. The clusters are shown in Figure 6. Our subjective evaluation does not completely agree with the artist groups or the similarity itself. Nonetheless we use it as the ground truth.

D. Experiments

Generally, building models when one class is rare is quite difficult because there are often many unstated assumptions [38]. A conventional wisdom is that classifiers built using all the data tend to perform worse on the minority class than on the majority class since the class *priors* in the natural distribution are biased strongly in favor of the majority class and the minority class has much fewer training and test samples [37]. Although the balanced distribution will not necessarily yield optimal distribution; it will generally lead to results which are no worse than, and often superior to, those that use the natural class distribution [37]. We sample roughly balanced datasets from the original dataset and the distributions of samples are shown in Table V.

We train a classifier that distinguishes each group from the rest of the artists. To build the classifiers we use support vector machines with linear kernels. The performance of the classifiers is measured using *accuracy*, *precision*, and *recall*. For each classifier, we compute a confusion matrix of four quantities, TP, FP, FN, and TN, which are the counts of, respectively, the *true positives* (the positives asserted to be positives), the *false positives* (the negatives asserted to be

²See <http://www.cse.unsw.edu.au/~min/ILLDATA/Function.word.htm>.

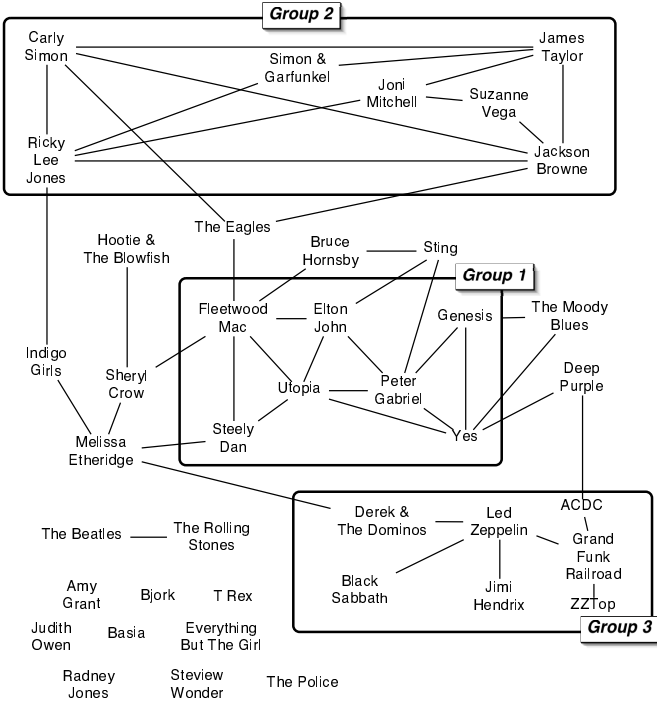


Fig. 6. The Artist Similarity Graph. Three groups of artists are identified in the graph.

Group	Total	Labeled		Unlabeled	
		Pos.	Neg.	Pos.	Neg.
1	217	26	29	77	85
2	196	19	31	56	90
3	200	20	30	60	90

TABLE V

THE DISTRIBUTION OF THE SAMPLES USED IN THE EXPERIMENTS.

positives), the *false negatives* (the positives asserted to be negatives), and the *true negatives* (the negatives asserted to be negatives). The accuracy, the precision, and the recall are respectively given by $(TP + TN)/(TP + FP + FN + TN)$, $TP/(TP + FP)$, and $TP/(TP + FN)$.

The unlabeled data are used for testing. The results of the three experiments are shown in Table VI. Without co-updating (labeled data only) we have three choices for the data source: only lyrics, only acoustics, and both. Co-updating approaches use both types of data. Accuracy measures can be applied to the lyrics-based classifier in its final form (after co-updating), acoustic-based classifier in its final form, and the combination of the two³. So, the tables below have six rows each.

We observe that the accuracy of the classifier built using labeled lyrics data are almost equal to the accuracy of a classifier built using labeled acoustic data. We also observe that combining the two sources improves the accuracy of classifier in the case of labeled data. The use of co-updating significantly improves the accuracy for each of the three cases

³The combined classifier after co-updating is constructed by multiplying the probability outputs of the lyrics-based and content-based classifiers with the conditional Independence assumption.

of data sources, but there is a slight gap between the two classifiers at the end.

We can conclude from these experiments that artist similarity can be efficiently learned using a small number of labeled samples by combining multiple data sources. We looked at the performance of the classifiers for Group 1 in more detail. The core of the group consists of Fleetwood Mac, Yes, and Utopia. We examined for which music tracks the combined classifier made an error after co-updating. Of the 71 errors it made, 38 were from albums of Peter Gabriel, Genesis, Elton John, and Steely Dan, none of which are not in the core of the group. Using analytical similarity measures to obtain the ground truth about artist similarity, thereby improving upon the data provided by web information resources, will be our future goal.

	Classifier	Accuracy	Precision	Recall
G r o u p 1	Lyrics-based	50.7	50.0	38.4
	Sound-Based	51.2	51.0	33.8
	Combined	53.1	55.7	46.8
	CoUp/Lyrics	63.6	57.3	62.2
	CoUp/Sound	68.5	65.5	71.4
	CoUp/Combined	69.8	69.4	64.9

G r o u p 2	Lyrics-based	50.7	38.2	46.4
	Sound-Based	60.3	42.6	46.8
	Combined	63.0	51.7	55.4
	CoUp/Lyrics	66.4	56.4	55.4
	CoUp/Sound	68.5	58.3	62.5
	CoUp/Combined	69.9	61.1	58.9

G r o u p 3	Lyrics-based	64.4	54.1	61.7
	Sounb-based	66.4	53.8	59.7
	Combined	68.7	60.3	63.3
	CoUp/Lyrics	76.0	70.0	70.0
	CoUp/Sound	76.0	66.2	81.7
	CoUp/Combined	78.7	74.1	78.7

TABLE VI

THE ACCURACY (IN %) OF ARTIST GROUP CLASSIFICATION. COUP STANDS FOR CO-UPDATING.

VII. SIMILARITY SEARCH

The objective of similarity search is to find music sound files similar to a music sound file given as input. Music classification based on genre and style naturally forms a hierarchy. Similarity can be used to group sounds together at any node in the hierarchies. The use of sound signals for similarity is justified by an observation that audio signals (digital or analog) of music belonging to the same genre share certain characteristics, because they are composed of similar types of instruments, having similar rhythmic patterns, and similar pitch distributions [5]. The similarity search processes can be divided into *feature extraction* and *query processing*.

A. The Method

Again we use the DWCH+FFT+MFCC feature set. The 35-dimensional vector represents each music file. After fea-

ture extraction, we represent each music track M_i by a 35-dimensional vector $V_i = (V_{i1}, \dots, V_{i35})$. We normalize each dimension of the vector by subtracting the mean of that dimension across all the tracks and then dividing the standard deviation. The normalized 35-dimensional representation vector is

$$\hat{V}_i = (\hat{V}_{i1}, \dots, \hat{V}_{i35}),$$

where $\hat{V}_{ij} = \frac{V_{ij} - \text{mean}(V_{.j})}{\text{std}(V_{.j})}$, $1 \leq j \leq 35$. After normalization, we compute the Euclidean distance between the normalized representation and the distance serve as similarity (in fact, dissimilarity) measure for our purpose. We then return the tracks with shortest distances to the given query as our similarity search result.

B. Experiments

1) *Jazz Vocal Music*: A collection of 250 Jazz vocal sounds files is created, which covers 18 vocalists and 35 albums. The vocalists are: Chet Baker, Tony Bennett, Rosemary Clooney, Blossom Dearie, Ella Fitzgerald, Johnny Hartman, Billie Holiday, Sheila Jordan, Ricky Lee Jones, Karin Krog, Abbie Lincoln, Helen Merrill, Joni Mitchell, Dianne Reeves, Carly Simon, Mel Tormé, Sarah Vaughan, and Nancy Wilson. For each music file, its first 30 seconds of the music are profiled.

Of these 250 tracks 60 tracks are selected as queries. For each query the nine closest matches are found using the similarity measure, which are presented in the increasing order of the Euclidean distance to the input sounds. Of the 60 queries, 28 queries (46.7%) had a track from the same album as the closest match, 38 (28 plus additional 10) queries (63.3%) had at least one track from the same album in the top three matches, and 54 (38 plus addition 16) queries (90.0%) had at least one track from the same album in the top nine.

We think that the 63.3% accuracy in terms of selecting one track from the same album in the top three is encouraging, because tracks by different artists may sound similar. In fact, for each of the 22 queries for which no tracks from the same album appears in the top three closest, we observe that at least one of the top three choices sounds very close to the query. For example, the system selected a segment from a ballad with a low-range female voice (Sarah Vaughan) accompanied by a piano trio as the most similar to a ballad with a low-range male voice (Johnny Hartman) accompanied by a piano trio; the system found the husky voice of Blossom Dearie to be similar to the husky voice of Karin Krog.

2) *Classical Music*: The same type of experiment is conducted for classical music, using a collection of 288 sound files, covering 72 albums (fifteen orchestral albums, ten chamber music albums, six art song and aria collections, ten keyboard solo albums, ten string solo and ensemble albums, seven choral albums, six opera albums, and eight concerto albums). We select a track from each album to obtain a list of nine closest sound files in the entire collection. For 33 queries (45.3%) the top two selections contain a track from the same album, for 29 of the remaining 39 (41.3% of the total), at least three out of top five were of the same format and from the same period (one of baroque, classical-romantic, and contemporary). Thus, for a total of 62 out of 72 (86%),

the tracks identified were highly satisfactory. In comparison, accuracy of the similar search seems lower in classical music than in jazz vocal music.

VIII. CONCLUSIONS AND FUTURE WORK

This paper introduces Daubechies Wavelet Coefficient Histograms (DWCH) for music feature extraction for music information retrieval. DWCH represents both local and global information by computing histograms on db_8 Daubechies wavelet coefficients at different frequency subbands with different resolutions. A comparative study on a dataset of Tzanetakis shows that Beat and Pitch features have little effect in automatic music genre classification when short segments of sound signals are used and that combining DWCH with the timbral features (MFCC and FFT) achieves the highest accuracy of 80%. The highest accuracy achieved by the combination improves by a margin of approximately 20% the previously known best results. A test on another dataset, compiled by the authors, resulted in 75% of accuracy. It raises a question of what techniques are effective in recognizing music genres in much larger taxonomy.

The paper also studies the issue of detecting emotion in music. Rating of two subjects in the three bipolar adjective pairs are used. The accuracy of around 70% was achieved in predicting emotional labeling in these adjective pairs. Effective emotional labeling for music has been much studied. An interesting research issue will be whether the feature set used here is effective in identifying emotions with respect to other taxonomy.

Another problem studied here is that of identifying groups of artists based on their lyrics and sound. Identification of artist groups based on the Similar Artist lists at All Music Guide is attempted. A semi-supervised learning algorithm called co-updating, which incorporates unlabeled data, is used on the combination of a sound feature set and a POS statistics feature set. The semi-supervised learning algorithm resulted in non-trivial increases in the accuracy to more than 70%. How this algorithm will perform on much larger collections of artists is an interesting problem.

Finally, a proof-of-concept experiment on similarity search using the DWCH+MFCC+FFT sound feature set was done in this paper. A more detailed experiment will clarify the efficacy of the approach.

ACKNOWLEDGMENTS

The authors are grateful to Qi Li for his assistance in conducting the experiments described in Section IV. The authors thank George Tzanetakis for useful discussions and for generosity in sharing his data with us, Dirk Moelants for sharing with us results obtained by his group, and Sterling S. Stein for providing us the tools for extracting lexical and orthographic features. The authors are grateful to anonymous referees for their invaluable comments.

REFERENCES

- [1] Shlomo Argamon, Marin Saric, and Sterling S. Stein. Style mining of electronic messages for multiple authorship discrimination: first results. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 475–480. ACM Press, 2003.
- [2] Eric Bill. Some advances in transformation-based parts of speech tagging. In *Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pages 722–727. American Association for Artificial Intelligence, 1994.
- [3] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [4] Ingrid Daubechies. *Ten lectures on wavelets*. SIAM, Philadelphia, 1992.
- [5] W. Jay Dowling and Dane L. Harwood. *Music Cognition*. Academic Press, Inc, 1986.
- [6] Paul R. Farnsworth. *The social psychology of music*. The Dryden Press, 1958.
- [7] Patrick Flandrin. Wavelet analysis and synthesis of fractional Brownian motion. *IEEE Transactions on Information Theory*, 38(2):910–917, 1992.
- [8] Glenn Fung and Olvi L. Mangasarian. Multicategory proximal support vector machine classifiers. Technical Report 01-06, University of Wisconsin at Madison, 2001.
- [9] Kate Hevner. Experimental studies of the elements of expression in music. *American Journal of Psychology*, 48:246–268, 1936.
- [10] David Huron. Perceptual and cognitive applications in music information retrieval. In *Proceedings of International Symposium on Music Information Retrieval*, 2000.
- [11] Brett Kessler, Geoffrey Nunberg, and Hinrich Schütze. Automatic detection of text genre. In Philip R. Cohen and Wolfgang Wahlster, editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 32–38, Somerset, New Jersey, 1997. Association for Computational Linguistics.
- [12] Guohui Li and Ashfaq A. Khokhar. Content-based indexing and retrieval of audio data using wavelets. In *IEEE International Conference on Multimedia and Expo (II)*, pages 885–888, 2000.
- [13] Tao Li, Qi Li, Shenghuo Zhu, and Mitsunori Ogihara. A survey on wavelet applications in data mining. *SIGKDD Explorations*, 4(2):49–68, 2003.
- [14] Tao Li and Mitsunori Ogihara. Detecting emotion in music. In *Proceedings of the Fifth International Symposium on Music Information Retrieval (ISMIR2003)*, pages 239–240, 2003.
- [15] Tao Li and Mitsunori Ogihara. Content-based music similarity search and emotion detection. In *Proceedings of The 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages V705–V708, 2004.
- [16] Tao Li and Mitsunori Ogihara. Music artist style identification by semisupervised learning from both lyrics and content. In *Proceedings of the ACM Conference on Multimedia*, pages 364–367, 2004.
- [17] Tao Li and Mitsunori Ogihara. Semi-supervised learning from different information sources. *Knowledge and Information Systems Journal*, 7(3):289–309, 2005.
- [18] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In *Proceedings of 26th Annual ACM Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 282–289. ACM Press, 2003.
- [19] Tao Li and George Tzanetakis. Factors in automatic musical genre classification of audio signals. In *Proceedings of 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, pages 143–146. IEEE Computer Society, 2003.
- [20] Beth Logan. Mel Frequency Cepstral Coefficients for Music Modeling. In *Proceedings of International Symposium on Music Information Retrieval (ISMIR)*, 2000.
- [21] Mrinal K. Mandal, Tyseer Aboulnasr, and Securaman Panchanathan. Fast wavelet histogram techniques for image indexing. *Computer Vision and Image Understanding: CVIU*, 75(1–2):99–110, 1999.
- [22] Yossi Matias, Jeffrey Scott Vitter, and Min Wang. Wavelet-based histograms for selectivity estimation. In *Proceeding of the ACM SIGMOD Conference*, pages 448–459, 1998.
- [23] Roger Mitton. Spelling checkers, spelling correctors and the misspellings of poor spellers. *Information Processing and Management*, 23(5):103–209, 1987.
- [24] Marc Leman, Valery Vermeulen, Liesbeth De Voogdt, Johannes Taelman, and Dirk Moelants. Acoustical and computational modeling of musical affect perception. Manuscript, November, 2003.
- [25] Isabelle Peretz, Lise Gagnon, and Bernard Bouchard. Music and emotion: perceptual determinant, immediacy, and isolation after brain damage. *Cognition*, 68:111–141, 1998.
- [26] David Perrot and Robert O. Gjerdingen. Scanning the dial: an exploration of factors in the identification of musical style. In *Proceedings of the 1999 Society for Music Perception and Cognition*, page 88, 1999.
- [27] Daniel P. W. Ellis, Brian Whitman, Adam Berenzweig, and Steve Lawrence. The quest for ground truth in musical artist similarity. In *Proceedings of 3rd International Conference on Music Information Retrieval*, pages 170–177, 2002.
- [28] David Pye. Content-based methods for managing electronic music. In *Proceedings of the 2000 IEEE International Conference on Acoustic Speech and Signal Processing*, 2000.
- [29] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, NJ, 1993.
- [30] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.
- [31] Hagen Soltau, Tania Schultz, Martin Westphal, and Alex Waibel. Recognition of music types. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998.
- [32] Efsthathios Stamatatos, Nikos Fakotakis, and George Kokkinakis. Automatic text categorization in terms of genre and author. *Computational Linguistics*, 26(4):471–496, 2000.
- [33] Mark Swain and Dana H. Ballard. Color indexing. *Int. J. computer vision*, 7:11–32, 1991.
- [34] Fiona J. Tweedie and R. Harald Baayen. How variable may a constant be? Measure of lexical richness in perspective. *Computers and the Humanities*, 32:323–352, 1998.
- [35] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5), July 2002.
- [36] Vladimir N. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, 1998.
- [37] Gary Weiss and Foster Provost. The effect of class distribution on classifier learning: An empirical study. Technical Report ML-TR 44, Rutgers University, 2001.
- [38] Bianca Zadrozny and Charles Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge discovery and data mining (SIGKDD 2001)*, pages 204–213. ACM Press, 2001.



Tao Li is currently an assistant professor in the School of Computer Science at Florida International University. He received his Ph.D. degree in Computer Science from University of Rochester in 2004. He received an IBM Faculty Award in 2005. His primary research interests are: data mining, machine learning, bioinformatics, and music information retrieval.



Mitsunori Ogihara received a PhD in Information Sciences at Tokyo Institute of Technology in 1993. He is currently Professor and Chair of the Department of Computer Science at the University of Rochester. His primary research interests are data mining, computational complexity, and molecular computation.