CLASSIFYING MUSIC BY GENRE USING THE WAVELET PACKET TRANSFORM AND A ROUND-ROBIN ENSEMBLE

Marco Grimaldi, Anil Kokaram, Pádraig Cunningham

Computer Science Dept.; Electronic and Electrical Engineering Dept.,
Trinity College Dublin, Ireland
grimaldm@cs.tcd.ie, anil.kokaram@tcd.ie, padraig.cunningham@tcd.ie

ABSTRACT

The vast amount of music available electronically presents considerable challenges for information retrieval. There is a need to annotate music items with descriptors in order to facilitate retrieval. In this paper we present a process for determining the music genre of an item using the Discrete Wavelet Transform and a round-robin classification technique. The wavelet transform is used to extract time and frequency features that are used to classify items by genre. Rather than use a single multi-class classifier we use an ensemble of binary classifiers with each classifier trained on a pair of genres. Our evaluation shows that this approach achieves very high classification accuracy.

1. INTRODUCTION

In recent years, the interest of the research community in indexing multimedia data for retrieval purposes has grown considerably [1,10,11]. The requirement is to enable access to multimedia data with the same ease as textual information. For music information retrieval, a direct way to compare music tracks would allow the construction of better music browsing systems [6] or improved recommendation systems [3]. In this domain, musical-genres are descriptors commonly used to catalog the increasing amounts of music available [6] and are important for music information retrieval.

This work presents a new system for music genre classification. A new feature set is accessed through a Wavelet Packet Decomposition transform, a process that has not been fully explored in the music domain (section 3). These new features are used within the framework of a supervised classifier for identifying genre. The paper discusses the performances of these features within that system. A round-robin ensemble of simple classifiers (k-NN) is trained for the musical-genre classification task (section 4). Our results show that this approach achieves very high classification accuracy.

2. WAVELET PACKET DECOMPOSITON

The discrete wavelet transform (DWT) is a well-known and powerful methodology that expresses a signal at different scales in time and frequency [2]. The DWT provides high time resolution and low frequency resolution for high frequencies. Vice versa, it provides high time and low frequency resolution for low frequencies.

The discrete wavelet packet transform (DWPT) [2] is a variant of the DWT technique. DWPT tiles the frequency space in a discrete number of intervals. For music analysis, this possibility has an enormous advantage: it allows us to define a grid of Heisenberg boxes matching musical octaves and musical notes. Considering just the frequencies corresponding to the musical notes, the spectrum characterization becomes a relatively easy task. Moreover defining a set of "virtual instruments" matching the musical octave tilling of the frequency axis permits to characterize in a meaningful way the time envelope of the song.

WPDT is achieved by recursively convolving the input signal with a pair of lo and high pass quadrature-mirror filters. Unlike the DWT that recursively decomposes only the low-pass sub-band, the WPDT decomposes both sub-bands at each level. It is possible to construct a tree (a wavelet packet tree) containing the signal approximated at different resolutions using a pyramidal algorithm [2].

3. FEATURE EXTRACTION

One disadvantage of using WPDT in this domain is that it is impossible to define a unique decomposition level suitable for time-feature and frequency-feature extraction. That depends on the properties of FIR filters (like Haar or Daubechies wavelets). Being able to recognize musical notes in the frequency domain implies losing almost all the details about on-set and off-set of notes. Vice versa being able to recognize note on-set, means losing details about the notes that are played. This paper overcomes these problems by proposing two different decomposition levels, one for time-feature and one frequency-feature extraction.

3.1. Time-feature

In order to characterize the beat of a song, we define a set of *virtual instruments* in the frequency domain. These

virtual instruments (frequency bins) correspond to different frequency sub-bands (table 1) extracted with the DWPT. Table 1 also shows in brackets the rough musical note range that corresponds to each frequency span.

Frequency Interval		*Bin Numb.*
0 HZ (C0)	86 Hz (E2)	0
86 Hz (F2)	172 Hz (E3)	1
172 Hz (F3)	345 Hz (E4)	2
345 Hz (F4)	689 Hz (E5)	3
689 Hz (F5)	1378 Hz (E6)	4
1378 Hz (F6)	2756 Hz (E7)	5
2756 Hz (F7)	5513 Hz (E8)	6
5513 Hz (F8)	11111 Hz (E9)	7
11111 Hz (F9)	22050 Hz (>C10)	8
22050 Hz (-)	44100 Hz (-)	9

Table 1: frequency bin definition for time-feature extraction

Using the DWPT the input music signal can be decomposed into these sub-bands. Each sub-band is then characterized in the time domain by measuring the range of beats that are found. The overall algorithm is shown in figure 1.

In order to assure a time-resolution suitable for extracting periodicities in music we must take into account the properties of the data and of the WPDT. Since the wavelets at any level j are obtained by stretching and dilating the mother wavelet by a factor 2^j [2], the time resolution at level j is given by:

$$T_{\sec}^{\ j} = \frac{1}{S_{rate}} \cdot W_{\sup} \cdot 2^j \qquad (1)$$

where W_{sup} is the wavelet support and j is the decomposition level of the WPDT.
The resolution in beat per minute (b.p.m.) at level j is given by:

$$F_{bpm}^{\ j} = \frac{1}{T_{\sec}^{\ j}} \cdot \frac{60}{2} \qquad (2)$$

The factor 2 in the above formula has been introduced in order to take into account the sampling theorem.

Given music sampled at 44100 Hz, and using the Daubechies4 wavelet (W_{sup} = 8 taps), a maximum resolution of 300 b.p.m., and using equations (2),(3); 9 levels of decomposition are necessary.

The time-features are extracted from the beat-histogram [10] of the signal. The histogram used here is based on the accumulation of ALL of the periodicities found in each sub-band of the same graph. As indicated in Figure 1, the periodicities are found by locating peaks in the autocorrelation function of each subband. The features are: the intensity, the position and the width of the 20 first most intensive peaks. The position of a peak is the frequency of a `dominant' beat, the intensity refers to the number of times that beat frequency is found in the song, the width corresponds to the accuracy in the extraction procedure. The peak detection algorithm uses the first derivative of the signal. Additional features used are: the total number of peaks present in the histogram, the histogram max and mean energy and the length in seconds of the song.

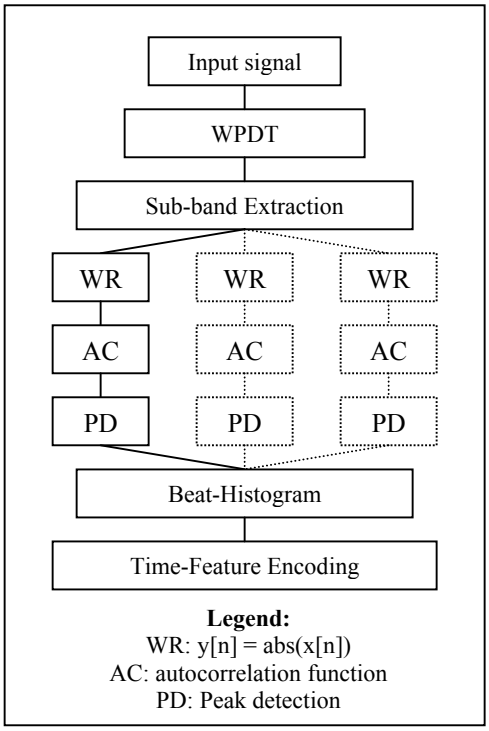

Figure 1: time-feature extraction

The idea of the beat-histogram was proposed by G. Tzanetakis et al. [10]. In their work, they demonstrate the usefulness of such a characterization in music classification. The algorithm presented here uses a different analysis methodology (DWPT), that we would argue is more capable of accurately expressing musical notes. Moreover, we take into account a higher number of time-features: 64 time-features and we analyze the input file completely by using a different number of decomposition levels.

3.2. Frequency feature

In order to take into account the peculiarity of music, we define a set of frequency intervals matching the music octaves (table 2). Considering that for a music file sampled at 44100 Hz the discrete wavelet transform divides the frequency axis between 0 Hz and 44100 Hz in 2^j intervals, it is possible to demonstrate that 13 levels of decomposition are necessary. In figure 2 we present the algorithm for the frequency-feature extraction process.

With the frequency resolution guaranteed by decomposition level *j=13*, only 2 notes out of 121 are difficult to recognize with a peak detection algorithm. That is because those two different notes (C0 and C#0) lie in adjacent bins. Since the characteristic frequencies of these two notes are below the frequency response of a standard audio CD player (20 Hz), this lack of resolution can be ignored.

Frequency Interval		*Bin Numb.*
0 HZ (C0)	33 Hz (B0)	0
33 Hz (C1)	64 Hz (B1)	1
64 Hz (C2)	128 Hz (B2)	2
128 Hz (C3)	256 Hz (B3)	3
256 Hz (C4)	512 Hz (B4)	4
512 Hz (C5)	1025 Hz (B5)	5
1025 Hz (C6)	2048 Hz (B6)	6
2048 Hz (C7)	4096 Hz (B7)	7
4096 Hz (C8)	8192 Hz (B8)	8
8192 Hz (C9)	16348 Hz (B9)	9
16348 Hz (C10)	32769 HZ (>C10)	10

Table 2: frequency bins definition for frequency-feature extraction

The spectrum characterization is performed as follows: for every single frequency bin, we calculate the intensity and position of the first 3 most intensive peaks. Moreover, we consider a characteristic parameter the total number of peaks in each bin. This value can be interpreted as a measure of the error in estimating the harmonical complexity of the sound using just the 3 peaks per octave. For completeness, we record the max and mean energy of the spectrum as well: 79 frequency-features in total.

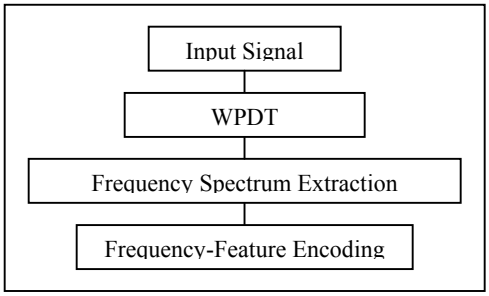

Figure 2: frequency-feature extraction

4. CLASSIFICATION

In order to evaluate the feature-set we use a round-robin ensemble of simple classifiers, i.e. k-NN classifiers. Our dataset consists of 182 songs belonging to 7 different genres. Each song has been labeled manually using [6] as musical-genre reference.

4.1. k-NN classifier

k-NN classifiers are instance-based algorithms taking a conceptually straightforward approach to approximating real or discrete valued target functions. The learning process consists in simply storing the presented data. All instances correspond to points in an *n*-dimensional space and the nearest neighbors of a given query are defined in terms of the standard Euclidean distance [4]. The probability of a query *q* belonging to a class *c* can be calculated as follows:

$$p(c \mid q) = \frac{\sum_{k \in K} w_k \cdot 1(kc = c)}{\sum_{k \in K} w_k} \qquad (3)$$

$$w_k = 1/d(k,q)$$

where *K* is the set of nearest neighbors, *kc* the class of *k* and *d(k,q)* the Euclidean distance of *k* from *q*. In this work, we define K as the set of the first 5 nearest neighbors.

4.2. Feature weighting

k-NN classifiers are particularly sensitive to noisy features [4]. The distance between instances is calculated based on all the attributes. That implies that features meaningful for the classification have the same weight as features less important for that purpose. This fact leads to miss-classification problems and to degradation in the system accuracy. This behavior is well known in the literature and is usually referred to as the *curse of dimensionality* [4].

We address the problem of high dimensionality (143-dimensional feature space) implementing a feature weighting strategy based on the concept of *information gain*. It is possible to evaluate the importance of every single feature by performing a leave-one-out classification. This classification procedure is repeated 143 times selecting at every run a different feature. Their importance is hence calculated in terms of relative accuracy. The feature weights are defined equal to the relative accuracies.

4.3. Ensemble of classifiers

An ensemble of classifiers is a set of classifiers whose predictions are combined to classify a query. Typically, the predictions are combined by weighted or unweighted voting. Ensembles of predictors can improve the accuracy of a single classifier, depending on the diversity of the ensemble members [5,8]. We constructed the ensemble of classifier using a relatively new approach: pair-wise or round-robin binarization [9]. This methodology converts a *c*-class problem into a series of two-class problems, using as training set only the appropriate classes and ignoring the others. The query is classified by submitting it to the *c(c-1)/2* binary predictors. The ensemble prediction can be determined via weighted or unweighted voting. The classes we consider are the musical-genre of each song. The ensemble prediction is achieved by weighting the prediction of every single classifier by its probability (3).

5. RESULTS

Figure 3 shows a comparison between performance of a random classifier, a simple k-NN classifier and the round-robin ensemble. For the simple classifier and the ensemble of classifiers, the feature weighting procedure described in section 4.2 has been applied. The round-robin ensemble outperforms the random and the simple k-NN classifier achieving a score of 84.64%. These results compare favorably with those reported in [10].

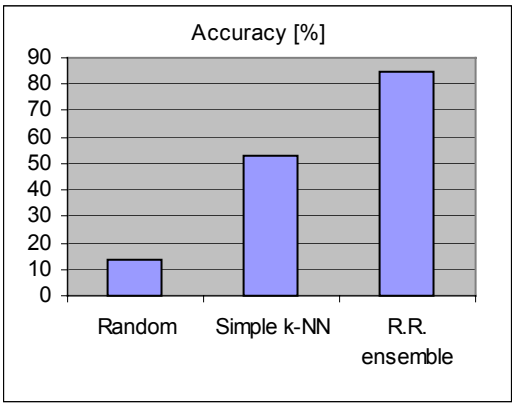

Figure 3: classifier accuracy comparison

	C1	C2	C3	C4	C5	C6	C7
C1	**25**	0	0	0	0	1	0
C2	1	**22**	1	0	1	0	1
C3	4	1	**20**	0	0	0	1
C4	0	2	2	**17**	0	4	1
C5	0	0	0	0	**26**	0	0
C6	1	0	1	3	0	**20**	1
C7	0	0	1	0	0	1	**24**

Legend:	C1 : Classical
C2 : Jazz	C3 : Electronic
C4 : Rock	C5 : Hard Rock
C6 : Alternative Rock	C7 : Heavy Metal

Table 3: genre classification confusion matrix

Table 3 shows the confusion matrix for the round-robin ensemble classification. In similar evaluations on two problems with 5 classes and 205 instances and 11 classes and 421 instances, round-robin classification had an accuracy of 91% and 77% respectively: again significantly outperforming the simple k-NN classifier.

6. CONCLUSIONS

This work demonstrates the usefulness of a DWPT applied to signal analysis in the music domain. The new set of music descriptors proposed in this paper, captures some important characteristics of music. Even the result achieved with a simple k-NN classifier (52.75%) is a good index about the usefulness of such a characterization. Adding the ability to manage multi-class problems (round-robin ensemble) to the predictor, we achieved impressive results. In fact, the result could be even better if we look closely at the classes we chose. Classes C5, C6 and C7 (Table 3) are sub-classes of C4 (3 different styles of the same genre [6]). Merging these classes in one unique "super-class" would increase the total score of our system. Finally the feature weighting strategy proposed permits the ensemble to overcome problems due to the hierarchical structure of the data [9].

7. REFERENCES

[1] Y. Wang, Z. Liu, J.C. Huang, "Multimedia Content Analysis Using Both Audio and Visual Clues", IEEE Signal Processing Magazine, 12-36, November 2000.

[2] S.G. Mallat, "A Wavelet Tour of Signal Processing", Academic Press 1999.

[3] C. Hayes, P. Cunningham, P. Clerkin, M. Grimaldi, "Programme-driven music radio", Proceedings of the 15th European Conference on Artificial Intelligence 2002, Lyons France. ECAI'02, F. van Harmelen (Ed.): IOS Press, Amsterdam, 2002

[4] T. M. Mitchell, ``Machine Learning", McGraw-Hill International Edition, Computer Science Series, 1997

[5] T. G. Dietterich, "Ensemble Methods in Machine Learning", First International Workshop on Multiple Classifier System, Lecture Notes in Computer Science, J. Kittler & F. Roli (Ed.), 1-15. New York: Springer Verlag.

[6] http://www.allmusic.com.

[7] L.K. Hansen, P. Salamon, "Neural Network Ensemble", IEEE Trans. Pattern Analysis and Machine Learning, vol. 12, 993-1001, 1990.

[8] G. Zenobi, P. Cunningham, "Using Diversity in Preparing Ensemble of Classifiers Based on Different Subsets to Minimize Generalization Error", 12th European Conference on Machine Learning (ECML 2001), L. De Readt & P. Flach (Ed.), LNAI2167, 576-587, Springer Verlag.

[9] J. Fürnkranz, "Round Robin Rule Learning", Proc. 18th International Conference on Machine Learning (ICML-01), C.E. Brodley & A.P. Danyluk (Ed.), 146-153, Williamstown, MA, 2001.

[10] G. Tzanetakis, G. Essl, P. Cook, "Automatic Musical Genre Classification of Audio Signals", In. Proc. Int. Symposium on Music Information Retrieval (ISMIR), Bloomington, Indiana, 2001.

[11] J. Pinquier, C. Senac, R. Andre-Obrecht, "Speech and Music Classification in Audio Documents", ICASSP 2002, Orlando, Florida, May 2002.