A Wavelet Packet Representation of Audio Signals for Music Genre Classification Using Different Ensemble and Feature Selection Techniques

Marco Grimaldi
Computer Science Department
Trinity College Dublin
Dublin 2, Dublin Ireland
+353 1 608 3688
Marco.Grimaldi@tcd.ie

Pádraig Cunningham
Computer Science Department
Trinity College Dublin
Dublin 2, Dublin Ireland
+353 1 608 3688
Pagraig.Cunningham@tcd.ie

Anil Kokaram
Electronic and Electrical Eng. Dep.
Trinity College Dublin
Dublin 2, Dublin Ireland
+353 1 608 3412
Anil.Kokaram@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 a new set of descriptors. A Wavelet Packet Transform is applied to obtain the signal representation at different levels. Time and frequency features are extracted from these levels taking into account the nature of music. Using *round-robin* and *one-against-all* ensembles of simple classifiers, together with feature selection methods, we evaluate the best signal representation for music genre classification. Ensembles based on different feature sub-spaces are explored as well in order to overcome over-fitting issues. Our evaluation shows that Wavelet Packet analysis together with ensemble methods achieves very good classification accuracy.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering, search process.*

General Terms

Algorithms, Design, Experimentation.

Keywords

Music information retrieval, Wavelet analysis, ensemble techniques, features selection.

1. INTRODUCTION

In recent years, the interest of the research community in indexing multimedia data for retrieval purposes has grown considerably [1]. The requirement is to enable access to multimedia data with the same ease as textual information. An example of this is the need to annotate music files with descriptors such as genre. This kind of characterization becomes indispensable in scenarios where enhanced browsing systems [2] allow users to inspect and select items from huge databases. In this domain, musical-genres are descriptors commonly used to catalog the increasing amounts of music available [3] and are important for music information retrieval.

Music information retrieval (MIR), as a research field, has two main branches: symbolic MIR and audio MIR. A symbolic representation of music such as MIDI describes items in a similar way to a musical score. Attack, duration, volume, velocity and instrument type of every single note are available information. Therefore, it is possible to access statistical measures such as tempo and mean key for each music item. Moreover, it is possible to attach to each item high-level descriptors such as instrument kind. On the other hand, audio MIR deals with real world signals and any features need to be extracted through signal analysis. In fact, extracting a symbolic representation from an arbitrary audio signal (polyphonic transcription) is an open research problem, solved only for simple examples. However, recent research shows that it is possible to apply signal processing techniques to extract features from audio files [1, 2] and derive reasonably sensible classification by genre [5, 6]. Other important examples of signal processing techniques applied to the audio domain involve discrimination between speech and music [16]; tempo and beat estimation [17]; audio retrieval by example [18].

This work presents a new approach for music genre classification. A new set of features 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. Different ensembles of simple classifiers (round-robin, one-against-all and feature sub-space based ensemble) together with different feature selection techniques are explored. Gain ratio and principle component analysis based ranking techniques are explored in order to overcome over-fitting issues. In section 5 we present an evaluation of different signal representations with the objective of determining the best descriptors for music genre classification.

MIR '03, November 7, 2003, Berkeley, California, USA.

2. WAVELET PACKET DECOMPOSITION

The discrete wavelet transform (DWT) is a well-known and powerful methodology that expresses a signal at different scales in time and frequency [4]. Taking into account the non-stationary characteristic of real signals, the DWT provides good time and frequency resolution. The discrete wavelet packet transform (DWPT) [4] is a variant of the DWT technique. DWPT permits to tile 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. DWPT is achieved by recursively convolving the input signal with a pair of *quadrature* mirror filters *g (low pass)* and *h (high pass)*. Unlike the DWT that recursively decomposes only the low-pass sub-band, the DWPT 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. This is done using a pyramidal algorithm [4].

3. FEATURE EXTRACTION

One disadvantage of using DWPT 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 loosing almost all the details about onset and offset of notes. Being able to recognize a note's onset entails loosing details about its frequency. 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.

Table 1. Frequency bin definition for time-feature extraction

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

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 have to take into account the properties of the data and of the DWPT. Since the wavelets at any level *j* are obtained by stretching and dilating the mother wavelet by a factor 2^j [4], 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 DWPT.

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 (1),(2); 9 levels of decomposition are necessary. `

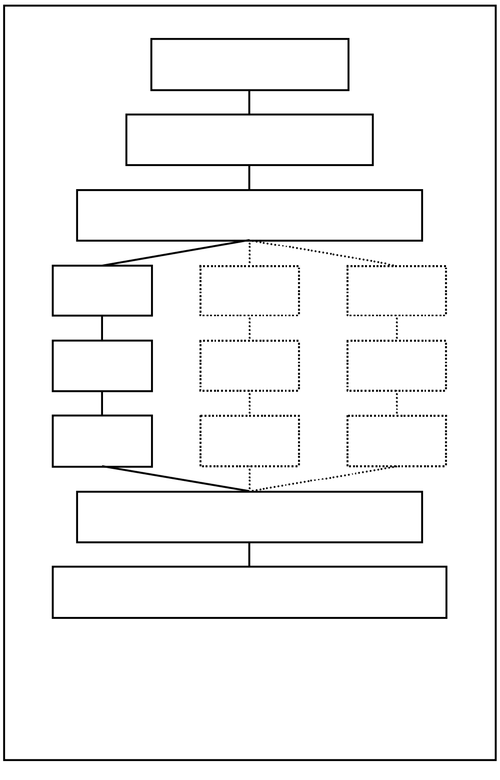

Figure 1. Time-features extraction

The time-features are therefore extracted directly from the beat-histogram [5] of the signal. It is calculated adding all the periodicities found in each sub-band to the same graph. The features are: the intensity, the position and the width of the 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 derivate 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.

The idea of the beat-histogram was proposed by G. Tzanetakis et al. [5]. In their work, they demonstrate the usefulness of such a characterization in music classification. The algorithm presented here uses a different analysis methodology (DWPT) and few little differences in the beat-histogram calculation. In this work we do not define an *a priori* number of time features (namely the number of peaks we extract from the beat-histogram). In section 5.1 we present the analysis we performed to determine the best number of peaks needed for beat description.

3.2 Frequency Feature

The feature set we propose is directly calculated from the frequency spectrum achieved via the DWPT. Given an input signal sampled at 44100 Hz, the DWPT divides the frequency axis between 0 Hz and 44100 Hz in 2^j intervals. It is possible to demonstrate that 16 levels of decomposition are necessary in order to have frequency bins matching music notes.

Table 2. Frequency bins definition for freq.-feature extraction

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

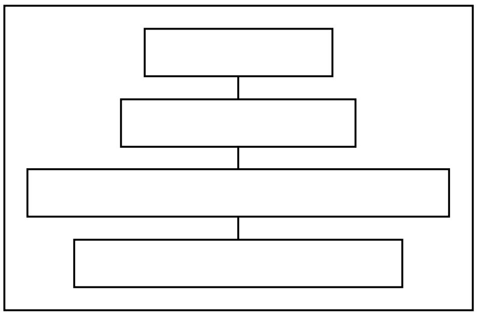

Figure 2. Frequency-feature extraction

With such a resolution, we can propose a new set of frequency features that takes into account some characteristics of music. Being able to tell which notes are 'dominant' means having a way to characterize the music harmony. Moreover, recording the note intensity and position at every octave means estimating implicitly the typology of playing instruments. The spectrum characterization is performed considering frequency intervals matching music octaves (table 2). Figure 2 shows the algorithm for frequency-feature extraction.

Section 5.2 shows the analysis we performed to determine the best number of frequency features (namely the number of peaks extracted from each bin in table 2) in order characterize the frequency spectrum.

4. CLASSIFICATION OF AUDIO SIGNALS

In this work different classifiers are explored in order to determine the quality of the feature we propose. This is due to the fact that the classification accuracy depends strongly on the technique used. It is well known that different predictors behave differently changing the nature of the problem being explored and the kind of features taken as input. In this work we consider the *k*-NN classifier as base predictor. Implementing a variety of ensemble methods we consider the fact that different numbers and kinds of features are tested. Moreover, in order to boost the accuracy of each classifier (simple or ensemble) different feature selection strategies are taken into account. In particular, we consider a situation where we select the first *n* features based on one of the ranking criteria. We also consider a wrapper-like [7, 11, 12] forward sequential search that takes a ranked set of feature as starting point. Since the wrapper approach is essentially a greedy search in the feature space for the best feature mask, a key issue in a forward sequential search is the order in which to test the attributes. The ranking criteria we propose in this work are *gain ratio* [8] and *PCA* [9].

4.1 *k*-NN based classifiers

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 [8]. The probability of a query *q* belonging to a class *c* can be calculated as follows:

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

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

K is the set of nearest neighbors, *kc* the class of *k* and *d*(*k,q*) the Euclidean distance of *k* from *q*.

4.2 Round-robin ensemble

A RR ensemble converts a *c*-class problem into a series of two-class problems by creating one classifier for each pair of classes [10, 14]. New items are classified by submitting them to the *c*(*c*-*1*)*/2* binary predictors. The final prediction is achieved by majority voting. The probability of a query q belonging to a class c can be calculated as follows:

$$P(c \mid q) = \frac{\sum_{m \in M} p_m(c \mid q) \cdot 1_{(mc=c)}}{\sum_{m \in M} p_m(c \mid q)} \qquad (4)$$

M is the set of ensemble members, mc is the class predicted by m and $p_m(c|q)$ is the probability given by ensemble predictor m according to equation (3).

4.3 One-against-all ensemble

An OAA ensemble performs problem-space decomposition with each ensemble member trained on a re-labelled version of the same data-set. Each component classifier is trained to distinguishing between one single class and its complement in the class space. Thus the number of members in the ensemble is equal to the number of classes in the problem. The probability of a query q belonging to a class c can be calculated as follows:

$$P(c \mid q) = \arg\max_{m \in M}[p_m(c \mid q)] \qquad (5)$$

M is the set of ensemble members and $p_m(c|q)$ is the probability given by ensemble predictor m according to equation (3).

4.4 Feature sub-space ensemble

Sub-sampling the feature space and training a simple classifier for each sub-space is an alternative methodology for building an ensemble. This strategy differs completely from the OAA and RR approaches. It does not decompose the decision space based on the classification task. Instead, the strength of FSS depends on having a variety of simple classifiers trained on different feature sub-sets sampled form the original space. This approach is very similar to a bagging technique [13] where the ensemble is built using different subsets of the instances in the training data. In this work, each ensemble member is trained on different feature-subsets of predefined dimension. Each feature-subset is drawn randomly from the original set. The probability of a query q belonging to a class c can be calculated according to equation (4).

5. EVALUATION AND DISCUSSION

In this section we present a comparative analysis of the performance of the different classifiers we presented in section 4. As anticipated in section 3, the number of feature we extract from the wavelet packet approximation of the signal is not *a priori* defined. The signal analysis has been performed n times in order to obtain different representations of the same database. Each representation differs in the number of peaks taken in to account. Through a comparative analysis of the classifier performances we bound the number of time and frequency features.

All the classifiers are trained on the same dataset composed of 200 instances divided in 5 different musical genres (Jazz, Classical, Rock, Heavy Metal and Techno), with 40 items in each genre. Each item is sampled at 44100 Hz, mono. The songs have been labeled manually using [3] as musical-genre reference. The accuracy scores are obtained by running a stratified 10 fold cross validation experiment. The number of k nearest neighbours is 5.

5.1 Bounding the number of time features

Figure 3 shows the accuracy behavior of a simple k-NN classifier trained using only time-features. As the number of time features used for representing the audio signal (namely the number of peaks) increases, the accuracy decreases. Applying forward sequential search based on the gain ratio measure, the score stabilizes around 47% regardless of the number of time-feature.

Figure 4 shows how a round-robin ensemble performs on the same experiment. While the accuracy behavior obtained without feature selection is comparable with the one showed by the simple k-NN, the ensemble outperforms the simple k-NN classifier once the feature selection is applied. The round-robin ensemble score keeps almost constant around 65%.

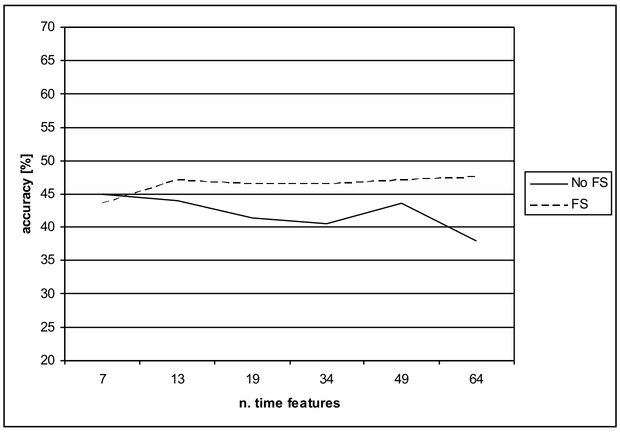

Figure 3. Simple *k*-NN performance using time-features

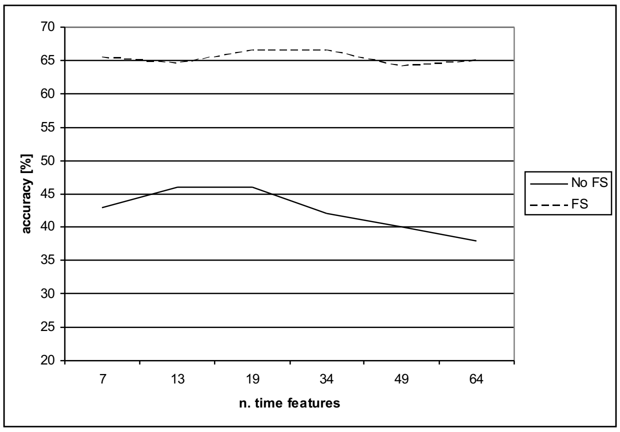

Figure 4. RR ensemble performance using time-features

In figure 5 we present the results of an identical experiment conducted using a one-against-all ensemble as classifier. The accuracy behaviour of such an ensemble reflects almost exactly the one showed by the simple k-NN when no feature selection is applied. The small differences between the two behaviours depend on how the ties are broken inside the ensemble. Applying the feature selection, the accuracy jumps to 59.5% for 7 features. It is interesting noting that the accuracy deteriorates abruptly when more than 19 features are used for describing the beat-histogram. This kind of behaviour depends on over-fitting due to lack of diversity in the ensemble [15].

According to figure 3, 4 and 5 we choose the minimum number of peaks necessary to characterize successfully the beat-histogram. The 13 time features taken into account are: time length; mean and max energy of the beat-histogram; total number of peaks; position, intensity and width of the 3 most intensive peaks.

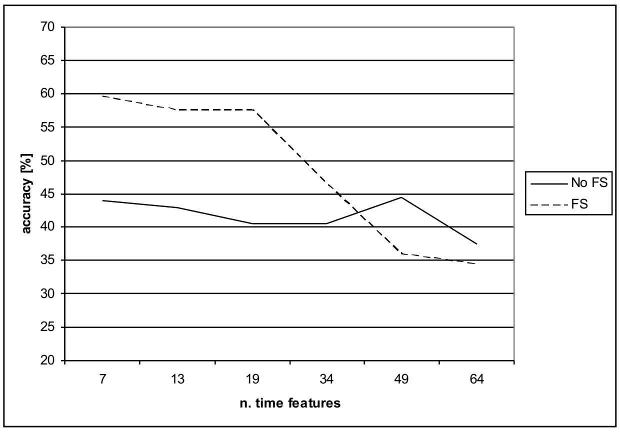

Figure 5. OAA ensemble performance using time-features

5.2 Bounding the number of freq. features

In order to restrict the number of features used to characterize the signal, we perform a slightly different experiment. As figure 5 shows, a big problem arises when the number of features increases (i.e. over-fitting). This problem is very common in classification and regression tasks when the number of features and the number of items becomes comparable. Since the number of frequency feature we evaluate varies between 35 and 123, we applied a principle component analysis and produced a simple *k*-NN classifier using the new set of features.

Figure 6 shows the accuracy behaviour training the classifier with 5 different representations of the same database. Each point on the graphs is obtained by running the classification algorithm considering a pre-defined number of features. I.e. 13 features, means that the classification is accomplished considering only the 13 best ranked features.

The accuracy curves obtained for the 5 different signal representations show similar behaviors in the whole range of ranked features taken into account. Considering the 5 best ranked features the difference between the poorest description (35 features) and the richest (123 features) is 3%, scoring respectively 71% and 74%. Augmenting the number of ranked features, the curves show similar noisy behavior.

In order to keep the frequency representation as simple as possible, without loosing much information (Occam's razor) we characterize the frequency spectrum of the audio signal with 35 frequency features. They are: mean and max energy of the spectrum and, for each frequency bin in table 2, position and intensity of the most prominent peak plus the total number of peaks in each bin.

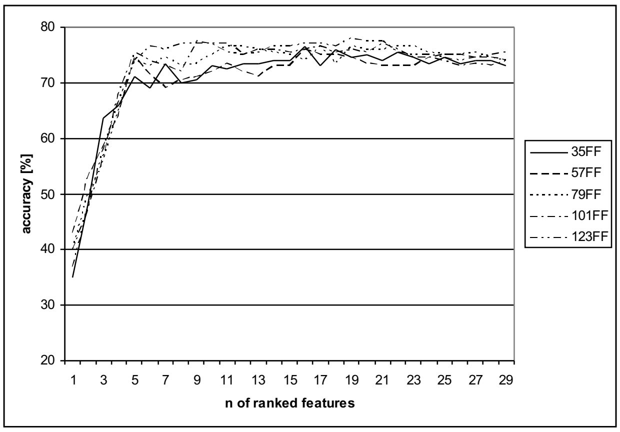

Figure 6. Accuracy curves of a simple *k*-NN for 5 different signal representation. Only frequency features have been taken into account.

5.3 Combining time and frequency features

In table 3 we show the accuracy of a simple *k*-NN classifier, a round-robin ensemble and a one-against-all ensemble trained using 48 time-frequency features. Columns 3 and 4 of table 3 show the classifier accuracies applying the full greedy search in the feature space ranking the features according to gain ratio and PCA.

Table 3. Prediction accuracies achieved through feature selection, varying ranking procedure and classifier.

	No FS	FS (GR)	FS (PCA)
Simple *k*-NN	78.5%	70.5%	65.5%
RR Ensemble	78.0%	78.5%	72.5%
OAA Ensemble	78.0%	77.0%	64.0%

Table 3 shows that the 3 classifiers tend to overfit. Without applying feature selection the accuracy score is higher than after the greedy search. The only exception is the score of a round-robin ensemble (column 3), but the achieved accuracy equals the simple *k*-NN accuracy obtained without feature selection. This behavior must be ascribed to a failure of the search for the best feature subset.

In order to avoid over-fitting we performed an experiment similar to the one presented in section 5.2. Each classifier has been trained using a pre-defined number of features and the accuracy score achieved trough a 10 fold-cross validation. The graph in figure 7 shows the accuracy curves obtained with the 3 different classifiers ranking the features according to gain ratio or PCA (the ranking strategy applied is given in brackets).

Considering the two ensembles (round-robin and one-against all), the same number of features is selected in each ensemble member. Selecting 10 features implies selecting the first 10 best ranked features in a each simple predictor. The ranking procedure is accomplished independently in each ensemble member. The graph clearly demonstrates that a round-robin ensemble outperforms the OAA ensemble and the simple *k*-NN classifier. The RR ensemble scores 77.5% with 3 features, 80.0% with 9 features and 81.0% with 16 features.

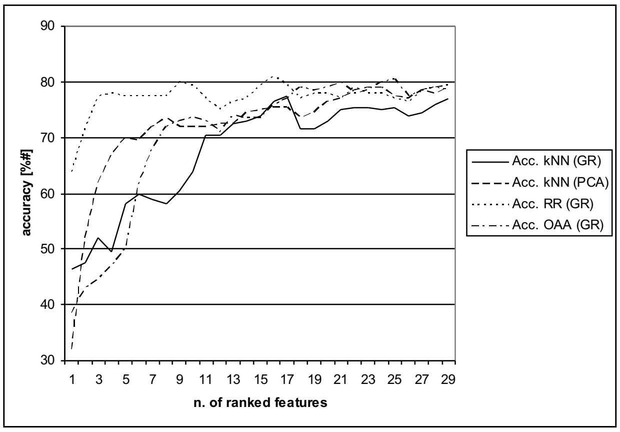

Figure 7. Accuracy achieved by 3 different classifiers ranking the features using gain ratio and PCA.

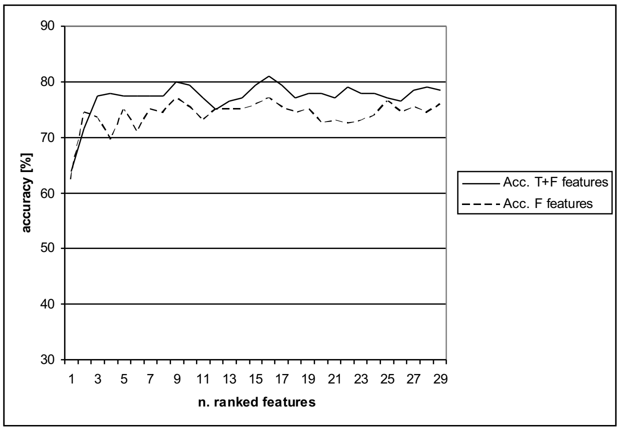

Figure 8. Accuracy achieved by a RR ensemble using frequency features and time and frequency features together.

In figure 8 we present the results of a similar experiment conducted using a round-robin ensemble trained with only frequency features and both time and frequency features. The graph demonstrates the usefulness of representing an audio signal with both time and frequency features in terms of improved accuracy.

5.4 An alternative ensemble method

In the previous subsections we demonstrated that representing a collection of audio signal through 48 time-frequency features is a successful approximation for music genre classification. However our analysis points out that a greedy search in the feature space for the best feature subset fails because of over-fitting problems. A way to overcome this issue is to consider an ensemble constituted by different small feature sub-spaces.

Figure 9 shows the accuracy curve of such an ensemble varying the dimension of the sub-spaces. Each point on the graph is obtained running 10 times a stratified 10 fold cross validation. The number of nearest neighbours for each run is 11. The error in the accuracy measure is ± 1% (standard deviation). The dashed curve represents the accuracy obtained implementing a forward sequential search based on gain ratio. Even if the difference between the two curves lies in the error bar, the feature selection assures better performance throughout the explored dimensions. In table 4 we present the confusion matrix obtained by the ensemble with feature sub-space of dimension 5 (83.5%).

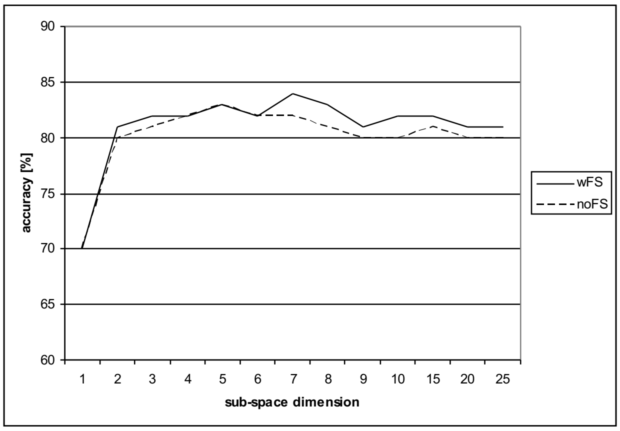

Figure 9. Accuracy achieved by a feature sub-space ensemble varying the sub-space dimension.

Table 4. Confusion matrix for a feature sub-set ensemble (e.g. 7 Rock tracks have been classified as Heavy Metal).

Q\A	Jazz	Rock	Techno	Classical	H. Metal
Jazz	33	2	1	4	0
Rock	2	27	2	2	7
Techno	2	1	34	0	3
Classical	0	0	0	40	0
H Metal	0	1	5	0	34

6. CONCLUSION AND FUTURE WORK

In this work we demonstrated that a Wavelet Packet analysis combined with ensembles of simple predictors can successfully classify a set of audio signals representing different music genres. The experiment shows how to reduce the number of features extracted from signal analysis in order to minimize over-fitting issues. Combining the properties of different ensemble strategies and feature selection methods, we showed that a feature set counting 48 time-feature descriptors can be used to successfully accomplish the genre classification. The analysis we performed shows that a Round-robin ensemble outperforms a simple *k*-NN classifier and a one-against-all ensemble considering only time-features and considering both time and frequency features. Due to the lack of music items in our database, the full greedy search in the features space fails because of over-fitting. The feature sub-space ensemble strategy seems to be the more appropriate solution in this context, since the ratio between number of features and number of instances can be easily controlled. However, over-fitting issues will be only partially solved until a large database of music items is available to the research community.

In the future we plan to evaluate other kind of ensembles together with different basic classifiers. Moreover we plan to extend our database. Another interesting aspect will be the consideration of the hierarchical structure of music genres. Music genre and music style [3] classification can be separately addressed in order to simplify the representation of the problem space and hence enhance the system performance.

7. ACKNOWLEDGEMENT

The support of the MOUMIR (www.mee.tcd.ie/~moumir) project is gratefully acknowledged

8. 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] 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.

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

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

[5] 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.

[6] G. Tzanetakis, A. Ermolinskyi, P. Cook, "Pitch Histograms in Audio and Symbolic Music Information Retrieval", In. Proc. Int. Symposium on Music Information Retrieval (ISMIR), Paris, France, 2002.

[7] R. Kohavi, G.H. John, "The Wrapper Approach", in Feature Selection for Knowledge Discovery and Data Mining, H. Liu & H. Motoda (Ed.), Kluwer, 33-50, 1998.

[8] T. M. Mitchell, "Machine Learning", McGraw-Hill, 1997.

[9] R.J Harris, "A Primer of Multivariate Statistics", Academic Press, 1975.

[10] J. Fürnkranz, "Pairwise Classification as an Ensemble Technique", Proceedings of the 13th European Conference on Machine Learning, pp.97-110, Springer Verlag, 2002.

[11] J.R. Quinlan, "C4.5 Programs for Machine Learning", Morgan Kauffman, 1994.

[12] M. Sebban, R, Nock, "A Hybrid Filter/Wrapper Approach of Feature Selection Using Information Theory", Pattern Recognition (35), pp. 835-846, 2002.

[13] 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, 2000.

[14] 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

[15] M. Grimaldi, P. Cunningham, A. Kokaram, "An Evaluation of Alternative Feature Selection Strategies and Ensemble Techniques for Classifying Music", to appear in Workshop in Multimedia Discovery and Mining, ECML/PKDD03, Dubrovnik, Croatia, September 2003.

[16] E. Scheirer, M. Slaney, "Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator", In Proc. Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP), Munich, Germany, 1997.

[17] E. Scheirer, "Tempo and Beat Analysis of Acoustic Music Signals", Journal of the Acoustic Society of America, 103(1), 588-601, January 1998.

[18] J. Foote, "ARTHUR: Retrieving Orchestral Music by Long Term Structure", in Proc. Int. Symposium on Music Information Retrieval (ISMIR), Plymouth, MA, 2000.