An Evaluation of Alternative Feature Selection Strategies and Ensemble Techniques for Classifying Music

Marco Grimaldi[1], Pádraig Cunningham[1], Anil Kokaram[2]

[1]Computer Science Department, Trinity College Dublin, Ireland
[2]Electronic Engineering Department, Trinity College Dublin, Ireland
{Marco.Grimaldi, Padraig.Cunningham, Anil.Kokaram}@tcd.ie

Abstract. Automatic classification of music files is a key problem in multimedia information retrieval. In this paper we present a solution to this problem that addresses the issues of feature extraction, feature selection and design of classifier. We outline a process for feature extraction based on the discrete wavelet packet transform and we evaluate a variety of wrapper-based feature subset selection strategies that use feature ranking based on *information gain*, *gain ratio* and *principle components analysis*. We evaluate four alternative classifiers; simple *k*-nearest neighbour and o*ne-against-all*, *round-robin* and *feature-subspace* based ensembles of nearest neighbour classifiers. The best classification accuracy is achieved by the feature subspace-based ensemble with the round-robin ensemble also showing considerable promise.

1 Introduction

A key issue in multimedia information retrieval is the need to annotate assets with semantic descriptors that will facilitate retrieval [1]. An example of this is the need to annotate music files with descriptors such as genre. Such a characterization becomes indispensable in scenarios where enhanced browsing systems [2] allow users to inspect and select items from a huge database. One way to automate this process is to label a subset of assets by hand and train a classifier to automatically label the remainder. This is a very challenging machine learning problem because it is a multi-class problem with unresolved questions about how to represent the music files for classification.

In this paper we present a process based on the discrete wavelet packet transform (DWPT) [4] that allows us to represent music files as a set of 143 features. We evaluate a variety of feature selection techniques to reduce this set to a manageable size. We also evaluate four different nearest neighbour classifier techniques:

- Simple *k*-Nearest Neighbour
- One-Against -All Ensemble
- Round-Robin Ensemble
- Feature-Subspace-based Ensemble

We focus on nearest neighbour techniques because of their ease of interpretability, as we will present in section 3. An important objective of this research is to gain some insight into what measurable features predict users tastes in music [2].

When evaluated on a five-class problem with a data-set of 200 music files, we find that the best classification accuracy (84%) is achieved by the feature subspace-based ensemble with the round-robin ensemble also showing considerable promise. While similar music classification tasks have been tackled by other researchers [1, 3, 5] it is difficult to compare results because of the unavailability of benchmark datasets. This will continue to be a problem due to the copyright issues associated with sharing music files.

The paper proceeds with an overview of the music classification problem and a very brief description of the wavelet based feature extraction process in the next section. The different ensemble-based classifiers that are evaluated are described in section 3 and the feature selection process is described in section 4. The details of the evaluation are presented in section 5.

2 The Music Classification Problem

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 easily 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 and number. 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, 3] and derive reasonably sensible classification by genre.

In this work we apply a wavelet packed decomposition to the audio signal in order to decompose the signal spectrogram at two different resolutions; one suitable for frequency-feature extraction, one for time-feature extraction.

2.1 Feature Extraction

The discrete wavelet transform (DWT) is a well-known signal analysis methodology able to approximate a real signal at different scales in time and frequency. Taking into account the non-stationary nature of the input signal, the DWT provides an approximation with excellent time and frequency resolution [4]. The discrete wavelet packet transform (DWPT) [4] is a variant of the DWT. It 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 WPDT decomposes both bands at each level. This procedure defines a grid of Heisenberg Boxes [4] corresponding to musical notes and octaves. Our analysis demonstrates that 9 levels of decomposition are necessary to build a spectrogram suitable for time-feature extraction [6].

Time-features are extracted from the beat histogram. The beat-histogram represents the most intense periodicities found in the signal [5]. The time-features we take into account are: the intensity, the position and the width of the 20 most intensive peaks. The position of a peak is the frequency of a *dominant* beat, the intensity refers to the number of times a beat frequency is found in the song and the width corresponds to the accuracy in the extraction procedure. Additional time-features are: the total number of peaks present in the histogram, its max and mean energy and the length in seconds of the song. A total of 64 time-features are extracted.

Frequency-features are extracted from the spectrum obtained by applying 16 levels of decomposition [6]. Dividing the frequency axis in intervals matching musical octaves, it is possible to characterize the spectrum in a relatively simple way. For every single frequency interval, we calculate the intensity and position of the first 3 most intensive peaks. We record the total number of peaks in each interval, the max and mean energy of the spectrogram as well – 79 frequency-features in total. The total number of features extracted for each song is 143.

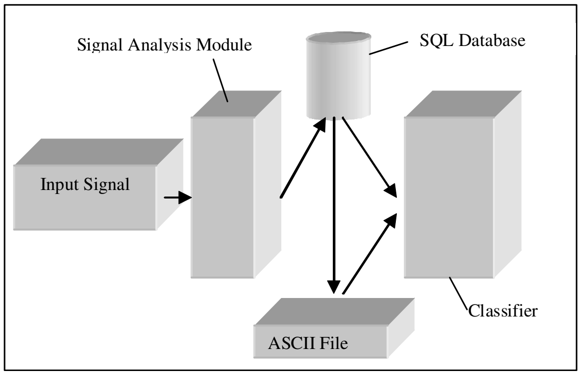

Fig. 1. System Architecture

Characterizing each item with 143 features has advantages and disadvantages. From a knowledge acquisition point of view it is useful to describe an item with the maximum amount of information we can obtain. This is important, because an *a priori* domain description is not available and every feature is potentially useful for classification. Moreover, the music genre classification problem has an implicit difficulty: music genres are not easily definable in terms of low or high level features. On the other hand, dealing with a high dimension feature space brings it own set of problems – perhaps the most important being the increased risk of overfitting. In Section 3, 4 and 5 we show how it is possible to overcome some of these problems.

2.2 Overall System Architecture

The system we have developed has two main units; the first one responsible for signal analysis, the second one of classification. Each audio file is decomposed through the wavelet packet decomposition software and the features are stored into a SQL database.

The classification module can connect to the database to retrieve the item characterization or load the whole dataset as ASCII file. Figure 1 shows the overall system architecture. The module responsible for classification has been designed so that it is possible to choose between different kinds of *k*-NN based predictors (Section 3) and different feature ranking techniques (Section 4).

3 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 [7]. The probability of a query *q* belonging to a class *c* can be calculated as follows:

$$\begin{aligned} p(c \mid q) &= \frac{\sum_{k \in K} w_k \cdot 1_{(kc=c)}}{\sum_{k \in K} w_k} \\ w_k &= 1/d(k,q) \end{aligned} \qquad \textbf{(1)}$$

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

Despite their simplicity, *k*-NN classifiers suffer a serious drawback. The distance between items is calculated based on all the attributes. That implies that any features that are in fact irrelevant for classification have the same impact at relevant features. This sensitivity to noise leads to miss-classification problems and to a degradation in the system accuracy. Such a behavior is well known in the literature and is usually referred to as *curse of dimensionality* [7]. In Section 5, we will show that this problem affects heavily the classification accuracy of the system. In fact, not only noisy features affect the classifier accuracy but correlated features may also cause problems.

However, *k*-NN classifiers used in conjunction with effective feature subset selection techniques are readily interpretable and can provide important insight into a *weak theory* domain. Black-box classifiers (e.g. neural nets) do not offer the same insight.

In order to overcome the problem of the high dimensional feature space it is possible to use different strategies. In the next section we present a set of ensemble methods we applied in order to simplify the decision surface the *k*-NN deals with. This simplification is obtained in the ensemble members by reducing the number of classes used to train the k-NN (section 3.1.1 and 3.1.2) or by reducing the feature space dimensionality (section 3.1.3).

3.1 Ensemble Alternatives

An ensemble of classifiers is a set of classifiers whose individual decisions are combined to classify a new item. The final prediction can be derived by weighted or unweighted voting. Research has shown that ensembles can improve on the accuracy of a single classifier, depending on the quality and the diversity of the ensemble members [8]. Ensembles can be implemented in a variety of different ways. In this work we present a comparison of three different ensemble strategies: *one-against-all* (OAA), *round-robin* (RR) [10] and *feature-sub-space* [9] based ensembles.

OAA and RR strategies are used especially with multi-class problems and both work by performing problem-space decomposition. Each ensemble member is a classifier specializing on a two-class problem. On the other hand, each member of the FSS ensemble covers the whole problem space. Each ensemble member is a *k*-NN classifier trained on the same multi-class problem. The improvement due to the ensemble is attributable to aggregation rather than problem decomposition.

3.1.1 One-Against-All Ensemble

As already mentioned, 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:

$$P(c \mid q) = \arg\max_{m \in M}[p_m(c \mid q)] \qquad (\mathbf{2})$$

M is the set of ensemble members and $p_m(c|q)$ is the probability given by ensemble predictor *m* according to equation (1). The big drawback of the OAA technique is that there are no benefits of aggregation; the classification of a given class depends heavily on the member responsible for that class (Even if that member does get to *specialize* on that class).

3.1.2 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]. 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:

$$P(c \mid q) = \frac{\sum_{m \in M} p_m(c \mid q) \cdot 1_{(mc=c)}}{\sum_{m \in M} p_m(c \mid q)} \qquad (\mathbf{3})$$

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 (1). Clearly, RR is a problem decomposition technique. However there are some aggregation benefits as each class is focused on by *c*-1 classifiers.

3.1.3 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 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 with replacement from the original set. The probability of a query q belonging to a class c can be calculated according to equation (3).

4 Feature Selection and Ranking Techniques

It is well known that implementing feature selection improves the accuracy of a classifier. The degree of improvement will depend on many factors; the type of classifier, the effectiveness of the feature selection and the quality of the features. In the case of simple k-NN classifier, the feature selection deletes noisy features and reduces the feature-space dimension. Moreover, for an ensemble of classifiers, the feature selection can promote diversity among the ensemble members and can improve their local specialization. The potential for an ensemble to be more accurate than its constituent members depends on the diversity among its members [8].

In this work we consider two approaches to feature selection. We consider a situation where we select the first n features based on one of the ranking criteria. We also consider a wrapper-like [11] 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. It is important to start with the more promising attributes. This is in the spirit of Filter/Wrapper algorithms as discussed by Seban and Nock [14]. In the following, we present the ranking algorithms we applied.

4.1 Information Gain

Given entropy (E) as a measure of the impurity in a collection of items, it is possible to quantify the effectiveness of a feature in classifying the training data [7, 13].

$$\begin{aligned} IG(S,A) &= E(S) - \sum_{v\in V(A)} \frac{|S_v|}{|S|}\cdot E(S_v) \\ E(S) &= \sum_{c\in C} -\frac{|S_c|}{|S|}\cdot \log_2 \frac{|S_c|}{|S|} \end{aligned} \qquad (4)$$

Information gain (IG) measure the expected reduction of entropy caused by partitioning the examples according to attribute A.

In the above equation: *S* is the item collection, |*S*| its cardinality; *V(A)* is the set of all possible values for attribute *A*; S_v is the subset of *S* for which *A* has value *v*; *C* is the class collection S_c is the subset of S containing items belonging to class *c*.

It is possible to extend the discrete equation (4) in order to handle continuous-valued attribute. It is done by searching for candidate thresholds sorting the items according to the continuous feature and identifying adjacent items that differ in their target classification [7]. The IG of feature A is equal to the maximum IG value obtained for the various thresholds.

4.2 Gain Ratio

The information gain measure favors attributes with many values over those with few values [7]. *Gain ratio* (GR) overcomes this problem by introducing an extra term taking into account how the feature splits the data:

$$GR(S,A) = \frac{IG(S,A)}{SI(S,A)}$$
$$SI(S,A) = -\sum_{i=1}^{d} \frac{|S_i|}{|S|} \cdot \log_2 \frac{|S_i|}{|S|} \qquad \textbf{(5)}$$

S_i are *d* subsets of examples resulting from partitioning *S* by the *d*-valued feature A. Since the SI term can be zero in some special cases, we define: *GR*(*S*,*A*) = *IG*(*S*,*A*) if *SI(S,A) = 0* for feature *A*. For the most part, this improvement over IG proves significant in the evaluation presented here.

4.3 PCA

Principal component analysis (PCA) is a standard technique used to handle linear dependence among variables. A PCA of a set of *m* variables generates *m* new variables (the principal components), $PC_1 \ldots PC_m$. Each component is obtained by linear combination of the original variables [12], that is:

$$PC_i = \sum_{j=1}^{m} b_{i,j} \cdot X_j$$
$$\overrightarrow{PC} = B^T \overrightarrow{X} \qquad \textbf{(6)}$$

Where X_j is the j^{th} original variable, $b_{i,j}$ the linear factor. The coefficients for PC_i are chosen so as to make its variance as large as possible. Mathematically, the variation of the original *m* variables is expressed by the covariance matrix. The transformation matrix *B*, containing the $b_{i,j}$ coefficients, corresponds to the covariance eigenvector matrix. Sorting the eigenvectors by their eigenvalues, the resulting principal components will be sorted by variance. In fact, the size of an eigenvalue defines how far a feature vector projected onto the eigenspace will be scaled along the correspondent eigenvector direction. Thus this new feature set is naturally ranked by variance which is useful if variance is a reasonable proxy for predictivness. This PCA approach to feature selection has two drawbacks. The first is that it is based on variance of the features only and does not take the class labels into account. The second is that the new features are not readily interpretable.

5 Evaluation and Discussion

In this section we present an evaluation of the different classification techniques presented previously (Section 3). In Section 5.1, we compare the ranking strategies described in section 4 with regard to the kind of classifier (simple *k*-NN, OAA and RR ensemble). In Section 5.2 we evaluate the feature selection applied to the different ensemble strategies. All the classifiers are trained on the same dataset composed of 200 instances divided in 5 different musical genres; with 40 items in each genre. Each accuracy score is obtained by running a stratified 10 fold cross validation experiment. The musical genres we consider are: classical, jazz, techno, rock and heavy metal. The OAA ensemble has 5 members, the RR 10 and FSS ensemble 100. The fact that the FSS ensemble has so many members might not be considered 'fair' and we return to this issue in the Conclusions. The number of *k* nearest neighbours is 5.

5.1 Ranking the Features

The graph presented in Figure 2 show the increase in accuracy of a simple *k*-NN classifiers as features are added based on the three ranking techniques. 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 the 13 best ranked features with respect to the ranking schema selected.

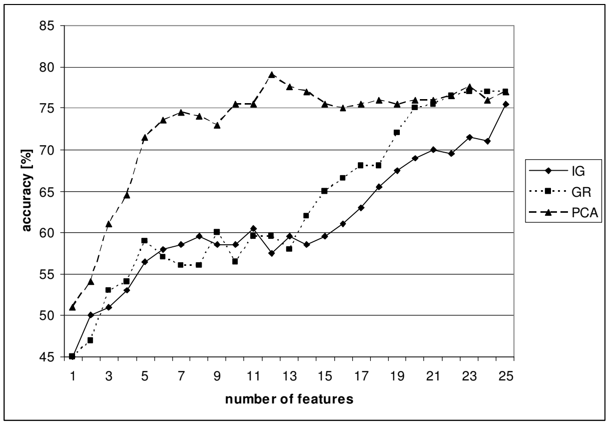

Fig. 2. Comparison of the accuracy score achieved by a simple *k*-NN ranking the features according to information gain, gain ratio and principle component analysis.

Comparing the accuracy behaviour obtained by ranking the features according to IG and GR, it is interesting to note how the system accuracy improves gradually as the number of feature increases. In both cases the classification accuracy doesn't catch up with PCA until a significant number of features are selected (13-20). This kind of behaviour has to be ascribed to correlation among the first 13-20 high ranking features. In fact, the graph shows clearly how PCA improves accuracy by reducing this correlation. Using the first 5 features, the system accuracy increases from 51% to 72%. After a steady state, the accuracy jumps to a value of 79% (12 features).

Figure 3 and 4 shows how the prediction accuracy changes when the ranking techniques are applied to OAA and RR ensembles.

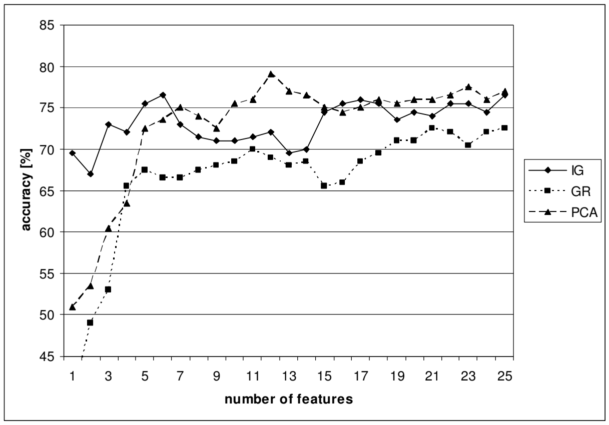

Fig. 3. Accuracy achieved by an OAA ensemble ranking the features using, IG, GR and PCA.

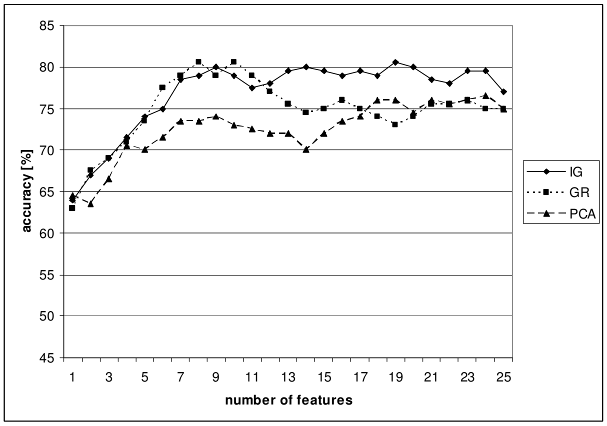

Fig. 4. Accuracy achieved by an RR ensemble ranking the features using, IG, GR and PCA

It is important to point out that the same number of features is selected in each ensemble member. Selecting 10 features implies selecting the first 10 best ranked features in each simple predictor. In this work, the ranking procedure is accomplished

independently in each ensemble member. In this way each simple predictor is locally sensitive to the classification problem it is dealing with.
It is interesting to note that the accuracy obtained by applying PCA matches *exactly* the one presented in figure 2. This fact is due to the lack of diversity in the ensemble: The OAA ensemble is formed by simply re-labelling the instances. Applying PCA in case of simple *k*-NN or OAA ensemble does not change the rank of the features.

Figure 4 shows the results obtained by running the same experiment on a RR ensemble. The RR ensemble seems to be more effective in the classification problem than the other two techniques. Ranking the features according to IG and GR, the ensemble achieves an accuracy of 81%. On the other hand, RR ensemble fails to take advantage of PCA analysis to boost the classification score. This fact is probably due to the poor statistical accuracy of the covariance matrix: the total number of items taken into account decreases from 180 to 72.

5.2 Applying Feature Selection

The forward sequential search algorithm is based on a 10-fold cross validation. In Table 1 we show the system accuracy varying the feature ranking technique and the kind of classifier.

Table 1. Comparison of the prediction accuracies achieved through feature selection, varying the ranking procedure and the kind of classifier.

	IG	GR	PCA
Simple *k*-NN	75%	77%	69%
OAA ensemble	78%	73%	76%
RR ensemble	78%	77%	66%

If we compare these results with those presented in the previous section it is clear that the forward sequential search suffers from over-fitting (simply selecting the top *n* features has better generalisation accuracy than the greedy search). This is due to the small number of example given the large number of features. In this scenario, considering small feature sub-spaces would allow the feature selection to be more effective. In figure 5 we present the results achieved applying the feature selection on a feature sub-space based ensemble. 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).

Without implementing a feature selection, an ensemble based on sub-spaces of dimension 4 achieves an accuracy of 83%. Applying the forward sequential search, the ensemble accuracy stabilise around 83%, 84% for a large number dimensions (4-10). It is interesting noting that the gain ratio based feature selection gets the best score for different sub-space dimensions: 84%. Given that we are using one small dataset to test the effectiveness of the ensemble methods and the ranking schemas proposed, we cannot claim a generalisation accuracy of 84%. We can say that we would expect an FSS based ensemble with 8 features in each member to produce a very good classifier.

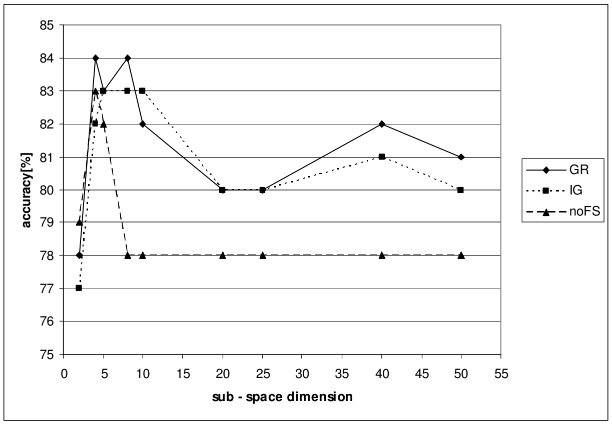

Fig. 5. Accuracy of 3 different feature sub-space ensembles varying the sub-space dimension.

Since a FSS ensemble with 8 features in each classifier is the most promising classifier, we present the confusion matrix in Table 2. This confusion matrix was produced using stratified 10-fold cross validation. The results are even better than the figure of 84% suggests, given that 4% of the error comes from misclassifications between Rock and Heavy Metal. The accuracy for Classical music is 100% although 4 Jazz and 2 Rock tracks are classified as Classical.

Table 2. The confusion matrix for the best classifier developed (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

In this work we have evaluated alternative approaches to the problem of classifying music audio files by genre. Since this is a multi-class problem we have considered the ensemble techniques that are specialised for multi-class – viz, OAA and RR. Because the DWPT process we use for feature extraction produces 143 features we have examined feature ranking and forward sequential search as mechanisms for feature selection. We found that FSS was inclined to overfit the feature selection process but ranking based on GR or IG worked well. In case of simple *k*-NN classifiers, PCA analysis proves to be the most effective feature selection technique, achieving a score of 79%.

The One-against-all ensemble does not appear to be a powerful strategy. The poor diversity between ensemble members is emphasised in figure 3. When PCA is applied, the ensemble technique presents an accuracy behaviour matching exactly that of simple *k*-NN. The Round Robin ensemble scores 81% with both IG and GR, showing to be an effective ensemble technique. However, these classifiers show over-fitting due to the feature selection process. When we apply feature selection to boost the accuracy of the component classifiers, the performance deteriorates. The small number of training examples compared to the number of features is clearly a problem. The best results are achieved with a feature sub-space based ensemble. When we apply a forward sequential search based on the GR ranking, the ensemble scores 84%.

Our evaluation shows that benefits accrue from the problem decomposition that occurs in the RR ensemble but the FSS ensemble wins out because of the aggregation benefits of the large ensemble. The focus of our current work is to bring these two benefits together in a RR ensemble with more than one ensemble member per class pair. This is also identified as a promising avenue of research by Fürnkranz [10].

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] 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] G.Tzanetakis, A.Ermolinskyi, P.Cook: Pitch Histograms in Audio and Symbolic Music Information Retrieval. Proceedings of 3rd International Conference on Music Information Retrieval. ISMIR 2002, Paris, October 2002.

[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] M. Grimaldi, P. Cunningham, A. Kokaram, "Classifying Music by Genre Using the Wavelet Packet Transform and a Round-Robin Ensemble", Trinity College Dublin, CS Dep., Technical Report, TCD-CS-2002-64, November 2002.
(http://www.cs.tcd.ie/publications/tech-reports/tr-index.02.html)

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

[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.), 576-587, Springer Verlag, 2001.

[9] 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.

[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] 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.

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

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

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