Discrete Wavelet Packet Transform and Ensembles of Lazy and Eager Learners for Music Genre Classification

Marco Grimaldi*, Pádraig Cunningham†, Anil Kokaram‡
*Department of Computer Science, University College Dublin
Marco.Grimaldi@ucd.ie
†Department of Computer Science, Trinity College Dublin
Padraig.Cunningham@tcd.ie
‡Electronic and Electrical Eng. Department, Trinity College Dublin
Anil.Kokaram@tcd.ie

Abstract

This paper presents a process for determining the music genre of an item using a new set of descriptors. A discrete wavelet packet transform is applied to obtain the signal representation at two different resolutions, a frequency resolution and a time resolution tuned to encode music notes and their onset and offset. These features are tested on a number of datasets as descriptors for music genre classification. Lazy learning classifiers (k-nearest neighbor) and eager learners (neural networks and support vector machines) are applied in order to assess the classification power of the proposed features. Different feature selection techniques and ensemble methods are explored to maximize the accuracy of the classifiers and stabilize their behavior. Our evaluation shows that these frequency descriptors perform better than a standard approach based on Mel-Frequency Cepstral Coefficients and on the Short Time Fourier Transform in music genre classification. Moreover, our work confirms that a parameterization of the music rythm based on the beat-histogram provides some meaningfull information in the context of music classification by genre. Finally, our evaluation suggests that multiclass support vector machines with a linear kernel and round-robin binarization are the simplest and more effective process for music genre classification.

1 Introduction

In music information retrieval (MIR) there is a need to annotate music items with descriptors in order to facilitate retrieval. MIR has traditionally been divided in 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. Information on; attack, duration, volume, velocity and instrument type of every single note in a piece is directly available. Therefore, it is possible to access statistical measures such as tempo and mean key for each music item [24]. Moreover, it is possible to attach to each item high-level descriptors such

as instrument type. On the other hand, audio MIR deals with real world signals where such information is not directly available. 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 [19, 20]. However, recent studies show that it is possible to apply signal processing techniques to extract features from audio files and derive reasonably accurate classification by genre [11, 16, 21, 29, 30, 34]. Other important examples of signal processing techniques applied to the audio domain include discrimination between speech and music [28], tempo and beat estimation [27, 30] and audio retrieval by example [9].

In this work we present a new set of descriptors extracted from the input signal (audio file) using the discrete wavelet packet transform (DWPT). Taking into account the characteristics of music, we propose a set of features that are able to well represent harmony and tempo in music. The proposed features are tested on a number of datasets as descriptors for music genre classification. Lazy learning algorithms (k-nearest neighbor classifiers - k-NN) and eager learning algorithms (artificial neural networks - ANN and support vector machines - SVM) are applied in order to better estimate the classification power of the proposed features. Different feature selection techniques and ensemble methods are explored in order to maximize the accuracy of the classifiers and stabilize their behavior. This work presents an evaluation of feature selection using forward and backward hill-climbing search applied on simple k-NN classifiers and round-robin, one-against-all and feature sub-space ensembles of k-NN. Round-robin and feature sub-space ensembles of ANN are evaluated in order to assess the classification power of the descriptors. Moreover, multi-class SVM (round-robin and one-against-all binarization) are also evaluated on the same datasets as an alternative classifier. The analysis shows that support vector machines are the most effective classifier for this problem. Among the different configurations explored, multi-class support vector machines (round-robin configuration and linear kernel) appear to be the simplest and most effective. While the k-NN-based solutions do not score as well as the SVM and ANN alternatives the fact that the k-NN performance is still good is important because of its role in query-by-example systems.

The work presented here builds on earlier work [10]. The set of descriptors used here has been modified from that used previously. The evaluation methodology has been improved, a new set of classifiers has been included and the evaluation has been extended to more datasets.

The description of this work begins with an introduction to the discrete wavelet packet transform and its application to music files for frequency and time features extraction. Once the time-feature characterization of the signal is presented, the different machine learning techniques adopted are presented. k-NN, ANN and support vector machines are briefely introduced and the ideas of ensemble learning and feature selection are illustrated. In the evaluation, we discuss how to bound the number of time and frequency features adopted for the characterization of the signal. Finally we identify the best classification setup by evaluating different ensembles of lazy and eager learners.

2 The DWPT

A common way to implement the wavelet transform is the discrete octave band decomposition (DWT) [18]. This involves recursively decomposing the input signal by applying a low pass (h) and a high pass (g) filter (*quadrature mirror filters* [18]). At each step of the decomposition, only the signal approximation obtained from the low pass is recursively decomposed. The octave band algorithm provides a series of signal approximations where the frequency resolution is fixed and *a priori* defined by the decomposition rule.

An alternative way to perform wavelet decomposition is by constructing a wavelet packet tree [18] applying a discrete wavelet packet transform (DWPT). 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 as described in [18]. The DWPT permits us to tile the frequency space in a discrete number of intervals. For music analysis, this has an enormous advantage, it allows us to define a grid of Heisenberg boxes [18] matching musical octaves and musical notes.

In this work we adopt Daubechies [5] wavelets with 8 taps (time support equal to 8 points) as quadrature mirror filters. The Daubechies wavelet family has been adopted successfully by other researchers [16, 30] for music genre classification. Moreover, Daubechies filters with enough vanishing moments [18] may be considered as good elementary functions for music parameterization [13]. The time envelope of any "musical" sound can be characterized with few parameters such as attack, decay, sustain and release. The energy of this kind of sound appears to be concentrated near the on-set point, according to the attack value. The time spread depends on its values of decay, sustain and release. Similarly, Daubechies wavelets are very asymmetric: the energy is concentrated at the beginning of the time support and the width of the time support depends on the number of vanishing moments.

2.1 DWPT and shift invariance

It is widely known that efficient wavelet representation using orthogonal bases results in shift-variant transforms. In such situations, a shift in the signal in time, can cause drastic changes in the amplitude of the wavelet coefficients. The shift-variant property is a problem when item recognition is needed (e.g. watermarking for copyright or security) or when signal manipulation is required. In both cases repeatability of the effect of the transform analysis is important, and the manipulation and analysis tends to be local in nature. However this is not an issue in the context of music genre classification. In fact we are only interested in extracting parameters that could be useful for representing the concept of music genre. As we present in the following sections, the features we propose for classification of music can be considered global, in the sense that the whole input file (song) is processed and its mean beat and harmony parameterized. The beat of the song is estimated through the extraction of the beat-histogram [30] from the whole file. The harmony is parameterized by extracting the main peaks present in the spectrum of the song. The Wavelet Packet Transform is used to meet the required time and frequency resolution defined by the nature of the signal. Therefore, as the features are global, signal

shifts are unlikely to affect their nature. Hence for this work, the data efficiency of orthogonal expansions makes them more attractive than the redundant shift-invariant transforms (e.g. the *à trou* algorithm [18]).

3 Feature extraction

One disadvantage of using DWPT for music parameterization 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 finite input response (FIR) filters, e.g. 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 work overcomes these problems by proposing two different decomposition levels, one for time-feature and one frequency-feature extraction.

3.1 Frequency features

The frequency features we propose are directly calculated from the frequency spectrum achieved via the DWPT. Generally speaking, in a song, each sound should be harmonically in tune with the next. For this reason the characteristic frequencies, as well as the component harmonics of the sound, must match musical notes. This simple empirical fact can be used to drive the characterization of the spectrum: the spectrum can be described with a resolution allowing the recognition of notes (the frequency intervals where fundamental and high and low harmonics lie). Considering input signals sampled at 44100 Hz, it is possible to demonstrate [13] that 16 levels of decomposition are necessary in order to have a frequency resolution matching music notes.

Once the desired frequency resolution is achieved, a set of intervals matching exactly the musical octaves is calculated (table 1). Because of the nature of the source under investigation, we decide to parameterize the frequency spectrum extracting the total number of peaks, the position and the intensity of the most intensive peaks in each interval (or musical octave). In this way the risk of an underestimation of the harmonic complexity of the sound is somewhat minimized: for each musical octave the fundamental and the high and the low harmonics of the sound can be characterized. The process of harmony characterization of the input signal is addressed considering the whole file.

Table 1 shows the defined frequency intervals matching the musical scales. The last four columns show the frequency range and the notes range lying in each interval.

In this work we do not define an *a priori* number of time features (namely the number of peaks we extract from each bin). As discussed in section 5, we evaluate different sets of frequency features in order to to minimize *overfitting problems* [22] and achieve better accuracy in predicting the music genres.

The different feature sets are composed of an initial 12 features:

1. Mean energy of the frequency-spectrum
2. Max energy of the frequency-spectrum
3. Number of peaks in the first octave

Table 1: Frequency intervals matching musical scales

bin N.	***freq. interval [Hz]***		***note interval***	
1	0.33	32.63	<C0	B0
2	32.63	64.26	C1	B1
3	64.26	128.19	C2	B2
4	128.19	256.04	C3	B3
5	256.04	512.42	C4	B4
6	512.42	1024.51	C5	B5
7	1024.51	2048.01	C6	B6
8	2048.01	4096.36	C7	B7
9	4096.36	8192.37	C8	B8
10	8192.37	16348.41	C9	B9

4. Number of peaks in the second octave
5. Number of peaks in the third octave
6. Number of peaks in the forth octave
7. Number of peaks in the fifth octave
8. Number of peaks in the sixth octave
9. Number of peaks in the seventh octave
10. Number of peaks in the eighth octave
11. Number of peaks in the ninth octave
12. Number of peaks in the tenth octave

A total of 32 features is obtained by adding to the above information the position and intensity of the most intensive peak in each bin of table 1:

13. Position of the first most intensive peak in the first octave
14. Energy of the first most intensive peak in the first octave
15. Position of the first most intensive peak in the second octave
16. Energy of the first most intensive peak in the second octave
17. Position of the first most intensive peak in the third octave
18. Energy of the first most intensive peak in the third octave
19. ...

Similarly 52 features are obtained by adding information about position and energy of the second most intensive peak. 72 features are obtained by recording the energy and position of the 3 most intensive peaks lying in each bin.

Figure 1 shows the frequency-feature extraction process.

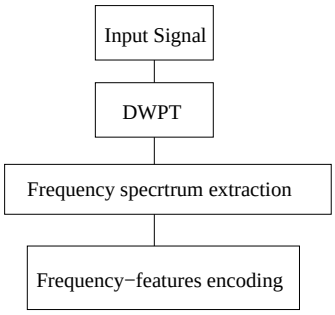

Figure 1: Frequency-features extraction diagram

3.2 Time features

The time-features we take into account for the classification task are directly derived from the work of Tzanetakis et al. [30]. The idea is to define a decomposition level j suitable for music tempo analysis and to extract the time envelope of different bins defined on the frequency axis. Calculating the autocorrelation function of each sub-band time envelope allows us to determine the main periodicities of the audio file being analyzed in different frequency intervals. Considering different frequency bins with the same time resolution, allows us to take into account rhythmical differences in different *instruments*. This process is addressed using a sliding-window of a few seconds, eventually analyzing the whole file.

As in the case of the frequency-features, the DWPT is used to obtain a signal approximation with time resolution suitable for tempo characterization. To achieve this tempo resolution we have to consider the wavelet decomposition algorithm. The decomposition algorithm consists of convolving the signal with the FIR filter defined by the wavelet basis. The result is down-sampled by 2. This process is repeated iteratively j times to calculate the wavelet coefficients at level j. Supposing an input signal sampled at 44100 Hz (F_s), the resolution on the time axis is given by:

$$T_{res} = \frac{1}{F_s} \cong 2.27 \cdot 10^{-5} sec. \tag{1}$$

When we convolve the input signal with the wavelet basis the time resolution changes and it must be corrected by the filter width (filter support- $F_{support}$). Hence at level $j = 1$ the temporal resolution is:

$$T^1_{res} = T_{res} \cdot F_{support} \tag{2}$$

Since the wavelets at any level j are obtained by stretching and dilating the mother wavelet by a factor 2^j, the temporal resolution at level j will be given by:

$$T^j_{res} = T_{res} \cdot F_{support} \cdot 2^j \tag{3}$$

The resolution in bpm (beat per minute) can be obtained as follows:

$$T_{res}[bpm] = \frac{1}{T_{res}[sec.]} \cdot \frac{60}{2} \tag{4}$$

The factor 2 has been introduced in order to take into account the sampling (Nyquist) theorem.

In the case of signals sampled at 44100 Hz, using a filter with 8 taps (Daubechies4 - Daubechies wavelet with 4 vanishing moments) and requiring a tempo resolution of about 300 bpm, 9 levels of decomposition are necessary.

The beat-histogram of the song is calculated by defining a set of frequency sub-bands (table 2) roughly matching musical octaves. Table 2 shows the defined frequency sub-bands roughly matching the musical scales. The last four columns show the frequency range and the notes range lying in each sub-band.

Table 2: Sub-band definition

bin N.	***freq. interval [Hz]***		***note interval***	
1	0.00	86.13	<C0	E2
2	86.13	172.27	F2	E3
3	172.27	344.53	F3	E4
4	344.53	689.06	F4	E5
5	689.06	1378.13	F5	E6
6	1378.13	2756.25	F6	E7
7	2756.25	5512.50	F7	E8
8	5512.50	11111.13	F8	E9
9	11111.13	22050.00	F9	>C10

The beat-histogram is hence calculated by adding all the periodicities found in each sub-band to the same graph, as presented in [30] and described in figure 2.

As in the case of the frequency features, 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 we present the analysis we performed to determine the best feature set for beat/tempo description.

The time-features we take into account are directly derived from the beat-histogram; the feature set is composed of an initial three features as follows:

1. Mean energy of the beat-histogram
2. Max energy of the beat-histogram
3. Total number of peaks in the beat-histogram

Additional features are derived by considering the position, intensity and width of successively less significant peaks. Hence, a total of six features are obtained by adding the following features to the above:

4. Position of the first most intensive peak
5. Intensity of the first most intensive peak
6. Width of the first most intensive peak

Similarly nine features are obtained by adding the same information about the second most intensive peak as follows:

7. Position of the second most intensive peak

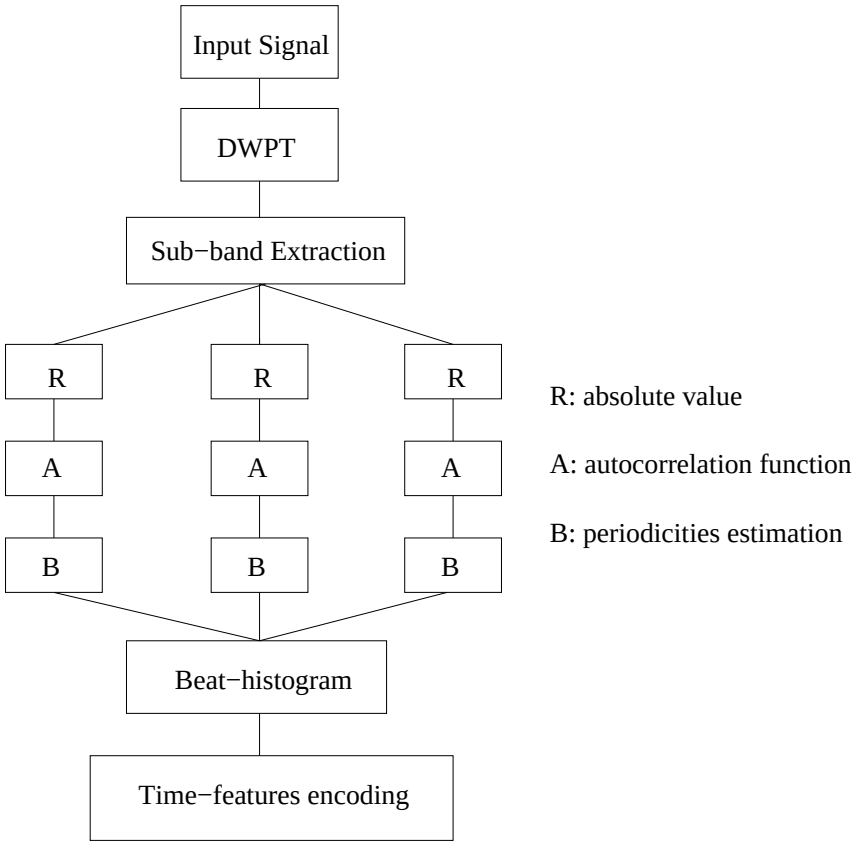

Figure 2: Time-features extraction diagram

8. Intensity of the second most intensive peak

9. Width of the second most intensive peak

A total of 12 features is obtained by adding the same features about the third most intensive peak.

10. Position of the third most intensive peak

11. Intensity of the third most intensive peak

12. Width of the third most intensive peak

Different feature sets can be derived from the beat-histogram by varying the number of peaks taken into account. The numbers of features in each set is a multiple of 3.

4 Classification of audio signals

In this work different classifiers are explored in order to determine the quality of the features proposed. Both eager and lazy learners are used; k-nearest

neighbors (k-NN), multilayer perceptrons (ANN) and support vector machines (SVMs). Different ensemble solutions are tested in order to maximize and stabilize the accuracy of each single predictor: round-robin, one-against-all and feature sub-space ensembles of k-NN are presented and discussed. Similarly round-robin and feature sub-space ensembles of ANN are tested. In order to handle multi-class problems, support vector machines [4] are evaluated in round-robin (aka one-versus-one) and one-against-all (aka one-versus-the-rest) configuration. Moreover both Gaussian and linear kernels are tested. Finally two different schema for feature selection in conjunction with k-NN classifiers are tested in order to minimize issues related to the *feature relevance problem* [15] and maximize the accuracy of the component classifier. In the next sections we briefly introduce the different component classifiers, the ensemble techniques and feature selection methods implemented.

4.1 k-NN classifier

The k-nearest neighbors classifier is one of the simplest and most effective predictors and has been studied for almost four decades. It is an instance-based algorithm taking a conceptually straightforward approach to approximate real or discrete valued target functions [22]. Since the k-NN is a lazy predictor, 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 usually defined in terms of the standard Euclidean distance [22]. The predicted class is inferred from the classes of the k nearest cases.

4.2 Multilayer perceptron

Artificial neural networks (ANN) - developed as generalizations of mathematical models of human cognition and neural biology [8] - are based on the following assumptions:

- information processing occurs through many simple elements called neurons (units or nodes);
- signals are passed through neurons over connection links;
- each link has an associated weight which multiplies the transmitted signal;
- each neuron applies an activation function to its input to determine its output.

In general, artificial neural networks are characterized by the pattern of connection between the units (its *architecture*), its method of calculating the weights on the connections - the training algorithm, and its activation function [8].

In this work we consider a standard feed-forward two-layer perceptrons as the base predictor. These multilayer perceptrons are trained applying the back-propagation algorithm and using the standard sigmoid function as the activation function.

4.3 Support vector machine

Support vector machines (SVMs) [31] perform classification by creating a maximum-margin hyper-plane that lies in a transformed input space. Given training examples (the training set) belonging to 2 different classes, a maximum-margin

hyper-plane splits the training set in its constituent classes, such that the distance from the closest examples (the margin) to the hyper-plane is maximized. The parameters of the maximum-margin hyper-plane are derived by solving a *quadratic programming* optimization problem. Two main enhancements to the basic algorithm have been introduced in order to handle non-linearly separable problems and to allow mis-labelled examples in the training set. The *soft margin* method tolerates mis-labelled examples by choosing a hyper-plane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. Non-linearly separable problems can be handled by introducing a *kernel function*. This causes the linear algorithm to operate in a different feature space. The new feature space is a non-linear map from the original input space, usually of much higher dimensionality than the original input space. In this way, non-linear classifiers can be created.

In this work we take into account two different kernels, Gaussian (radial basis) and linear kernels. Moreover, in order to handle multi-class problems, two different binarization procedures are evaluated, round-robin and one-against-all. As proposed in [14] and discussed in [4] the different kernel parameters are estimated in a wrapper-like [26] fashion (section 4.4) using a grid search.

4.4 Feature selection

A lazy learner, such as k-NN, uses the whole set of features describing the instances in order to predict the class of the unseen query q. This simplicity has an important drawback in that both relevant and irrelevant features are used for the classification. Thus, the classifier is very sensitive to irrelevant and redundant features. In order to overcome this problem we implement a hill-climbing search based on the *wrapper approach* [26]. As might be inferred from the name, the wrapper approach uses the induction algorithm as the evaluation function in the feature selection search process. At each stage in the search process the evaluation function is an estimation of the generalization accuracy of the classifier using the feature subset under evaluation. Once the *best* feature subset is identified, the resulting classifier is tested against a separated set not used during the search.

The strategy used to evaluate the different feature sub-sets is based on an *inner* 10-fold cross-validation. The training-set is divided into 10 folds of which nine are used to train the algorithm using the selected feature set, and the remaining one to test the accuracy of the classifier. The estimation of the accuracy is achieved by running the training/test process on the 10 different combinations of training-set/test-set.

A key issue in searching the feature space for the best sub-set is the order in which the attributes are tested – the wrapper approach is essentially a greedy search in the feature space for the best feature mask. In order to better drive such a search the *information gain ratio* [25] is used in this work to rank the features. Information gain ratio is a measure based on the notion of entropy [22] that estimates the effectiveness of an attribute (feature) in classifying the training data. It measures the expected reduction in entropy caused by partitioning the examples according to a given feature.

In this work both forward and backward hill-climbing are evaluated. The forward hill-climbing search is implemented as described by the following schema:

- at the beginning of the search the features are ranked according to the information gain ratio measure.
- the accuracy of the classifier is evaluated through a 10-fold cross-validation using only the feature that scores the best according to the given measure (information gain ratio).
- the digit in the feature mask corresponding to the next best feature is flipped (0 to 1) and the new accuracy evaluated through the 10-fold cross-validation.
- if the accuracy has improved the new feature mask is kept, otherwise the feature mask corresponding to the previous state is selected.
- sequentially the process is repeated until all the features are evaluated. from the most important to the least important.

Similarly to the forward case, the backward hill-climbing search is implemented as follows:

- at the beginning in the search the features are ranked according to the information gain ratio measure.
- the accuracy of the classifier is evaluated through a 10-fold cross-validation using all the feature.
- the digit in the feature mask corresponding to the worst feature is flipped (1 to 0) and the new accuracy evaluated through the 10-fold cross-validation.
- if the accuracy has improved the new feature mask is kept, otherwise the feature mask corresponding to the previous state is selected.
- sequentially the process is repeated until all the features are evaluated, from the least important to the most important

4.5 Ensemble methods

The idea behind the definition of an ensemble of predictors comes from a simple observation: it is commonly the case that a group of experts in a given domain can make better decisions than a single one. The obvious fact that a single expert may not posses the wide knowledge necessary to cover all the possible aspects of the domain is the intuitive motivation leading to the research of ensemble methods [6]. Different solutions in order to build a committee of expert have been proposed; manipulating the training examples, manipulating the input features, manipulating the output targets, injecting randomness.

We adopt 3 different approaches to build an ensemble of classifiers; round-robin binarization, one-against-all binarization and ensembles based on different sub-spaces of the original feature space. In the next sections we introduce the ensemble strategies adopted.

4.5.1 Round-robin ensemble

A round-robin ensemble converts a c-class problem into a series of *two*-class problems by creating one classifier for each pair of classes [12]. New items are classified by submitting them to the $c(c-1)/2$ binary predictors. The final prediction is achieved by weighted majority voting. The weights correspond to the probability estimated by each component classifier for the given query q.

4.5.2 One-against-all ensemble

An one-against-all 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 final prediction is achieved by weighted majority voting. The weights correspond to the probability estimated by each component classifier for the given query q.

4.5.3 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 one-against-all and round-robin approaches. It does not decompose the decision space based on the classification task. Instead, the strength of feature sub-space ensembles 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 [6] where the ensemble is built using different subsets of the instances in the training data. Each ensemble member is trained on different feature-subsets of predefined dimension. Each feature-subset is drawn randomly from the original set. The final prediction is achieved by weighted majority voting. The weights correspond to the probability estimated by each component classifier for the given query q.

4.5.4 Ensembles and Feature Selection

The effectiveness of the three ensemble strategies in conjunction with k-NN classifiers is evaluated with and without the application of the feature selection strategies proposed in section 4.4. Feature selection is applied to each member of the ensemble. Partitioning the solution space according to a given rule or sub-sampling the feature space causes the gain ratio to provide different feature ranking; hence the feature selection process increases the diversity among the constituent classifiers. However, we will not take into account an explicit measure for the diversity, as proposed in [33]. Diversity arises naturally in the different ensembles. In section 5, in the case of feature sub-space based ensembles, we will show how the ensemble accuracy changes depenting on the number of ensemble member and the dimension of the feature sub-space: how the intrinsic diversity among ensemble members affects the generalization accuracy.

5 Evaluation and discussion

5.1 Evaluation methodology

The evaluation methodology is based on a randomized 10-fold cross-validation repeated 10 times. The dataset is divided in to 10 folds of which nine are used to train the algorithm and the remaining one to test the accuracy of the classifier. The estimation of the accuracy is achieved by running the training/test process on the 10 different combinations of training-set/test-set. At the beginning of each 10-fold cross-validation the instances are shuffled randomly. This is done in

order to assess the generalization accuracy of the predictor. The generalization score is calculated as the mean of 10 independent 10-fold cross-validations. The error of the measure is calculated as the standard deviation. As described in section 4.4, the feature selection process is based on a second (*inner*) 10-fold cross-validation:the training-set is divided in to 10 folds of which nine are used to train the algorithm using the selected feature set, and the remaining one to test the accuracy of the classifier. The estimation of the accuracy is achieved by running the training/test process on the 10 different combinations of training-set/test-set.

5.2 Datasets

In this section we present the datasets used in the evaluation (section 5) of the different predictors used for music genre classification.

Table 3: The different datasets used for classification

dataset	***n. of classes***	***n. of instances***
4classes	4	160
5classes	5	200
6classes	6	240
7classes	7	280
8classes	8	320

The music files are sampled at 44100 Hz, mono. The music genres in the datasets of table 3 are as follows:

- *4classes*: Classical, Jazz, Rock, Techno;
- *5classes*: Classical, Jazz, Rock, Techno, Heavy-Metal;
- *6classes*: Classical, Jazz, Rock, Techno, Heavy-Metal, Acoustic;
- *7classes*: Classical, Jazz, Rock, Techno, Heavy-Metal, Acoustic, Celtic.
- *8classes*: Classical, Jazz, Rock, Techno, Heavy-Metal, Acoustic, Celtic, Country

Each class is equally represented in the datasets: 40 instances per class. The songs belonging to *Acoustic* have been manually chosen and are defined as folk/country songs where only acoustic guitar(s) and voice(s) are present. Similarly The songs belonging to *Celtic* are folk songs typical of Irish music. The music genre of each instance (song) in table 3 is obtained using [1] as reference. Allmusic [1] is a website desicated to the world of music. They claim that: "All genres and styles of music are covered here, ranging from the most commercially popular to the most obscure." [1].

5.3 Evaluation of the optimal item description

The number of features that best suit the classification problem is estimated using a unique dataset. This approach tries to minimize possible overfitting issues arising by using too many features to encode the classification problem. All the classifiers are trained on a dataset composed of 200 instances divided into 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 labelled manually using [1] as the musical-genre reference. The predictor chosen for this purpose is the k-NN classifier and the 3 ensemble methods build upon k-NNs as described in section 4.5.

In order to assess the impact of the different types of features on the classification process, we will begin by looking at the time and frequency features separately. Secondly, the optimal setup for the different classifiers is estimated. Hence, the ability of the best predictor to generalize is evaluated. This is done on different datasets (section 5.2), presenting slightly different aspects of the same problem. Varying the number of classes and their typology (namely the music genre), the accuracy of the classifier is evaluated.

5.3.1 Bounding the number of time features

Figure 3 shows the accuracy behavior of a simple k-NN (k=5) classifier trained using only time-features. The x-axis in figure 3 (and similarly in the rest of the figures in this section) indicates the total number of time features used to parameterize the beat of the signal. In that figure the acronym SKNN-FS Fw indicates the accuracy obtained applying forward hill-climbing search. Similarly SKNN-FS Bk indicates the accuracy score obtained applying backward hill-climbing search. In the subsequent figures in this section, FS Fw indicates that the accuracy score is obtained applying the forward hill-climbing search on the classifier. Similarly FS Bk indicates accuracy score obtained applying backward hill-climbing search.

Figure 3 shows that applying feature selection, the score decreases significantly: the classifier (SKNN) performs worse, showing problems due to the over-fitting phenomenon. Applying backward hill-climbing search the score does not improve either.

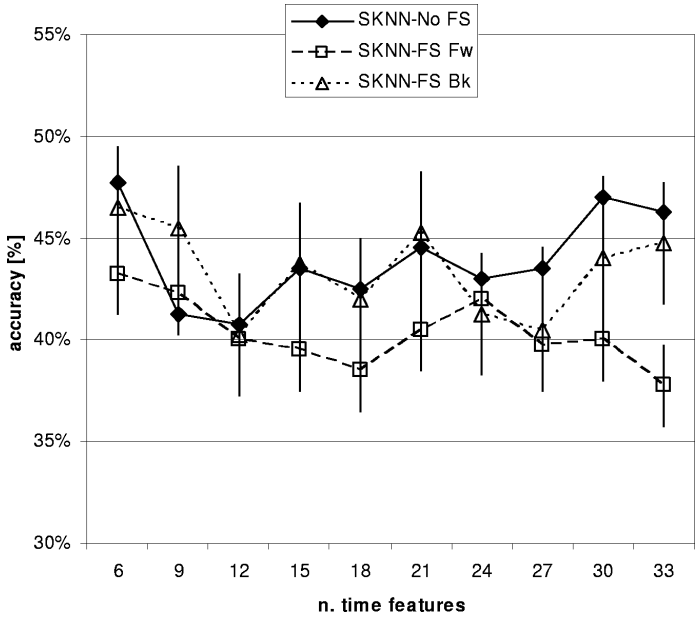

Figure 3: Simple k-NN (SKNN) performance using time-features

Figure 4 shows the performance of a round-robin ensemble (RRE) on the same experiment; each ensemble member considers 5 nearest neighbors (k=5) for classification. While the accuracy behaviour obtained without feature selection is comparable with the one shown by the simple k-NN, the ensemble outperforms the simple k-NN classifier once the forward hill-climbing search is applied. In this case, the round-robin ensemble accuracy keeps almost stable within the error bars (60%). It is interesting to note how the accuracy decreases dramatically when applying backward hill-climbing search. This behaviour seems to suggest that this kind of ensemble works better when its members highly over-fit the problem [23].

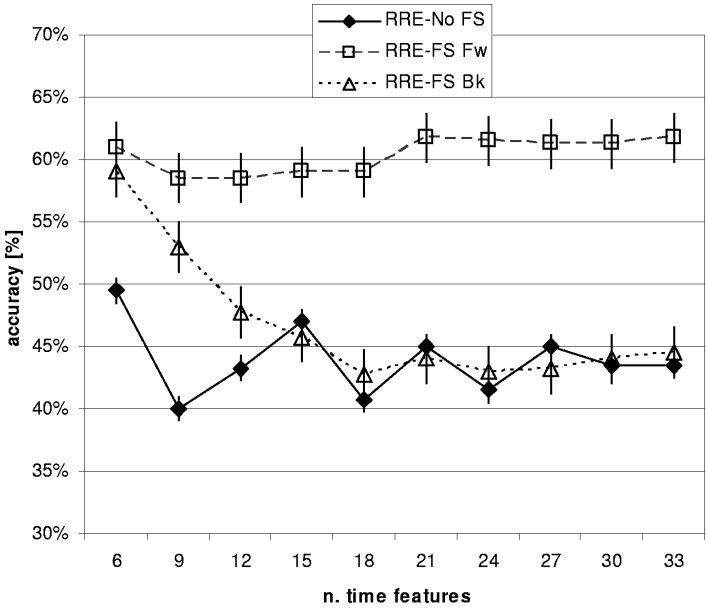

Figure 4: Round-robin ensemble (RRE) performance using time-features

In figure 5 we present the results of an identical experiment conducted using a one-against-all ensemble (OAE) as classifier. As in the previous experiments, each ensemble member considers the first 5 nearest neighbors (k=5) for classification. Applying the search, the ensemble scores $(57 \pm 3)\%$, using 9 and 12 features. Increasing the number of features, the accuracy stabilises around 55%. Just as in the round-robin case, the accuracy drops when backward search is applied.

Figure 6 shows the behavior of a feature sub space based ensemble (FSSE) on the same problem. The number of members in the ensemble is 50; the dimension of the subspaces is 4 and k=5. The accuracy tends to increase with the number of features until a suitable number (9-15) is used. Any further increase in the number of time features causes the accuracy to drop. The higher number of ensemble members guarantees more stability and over-fitting issues are less pronounced when increasing the number of features. Moreover, the backward search seems to be more efficient than the forward search. These differences reflect the different nature of the classifiers.

According to figures 4, 5 and 6 we choose the minimum number of peaks

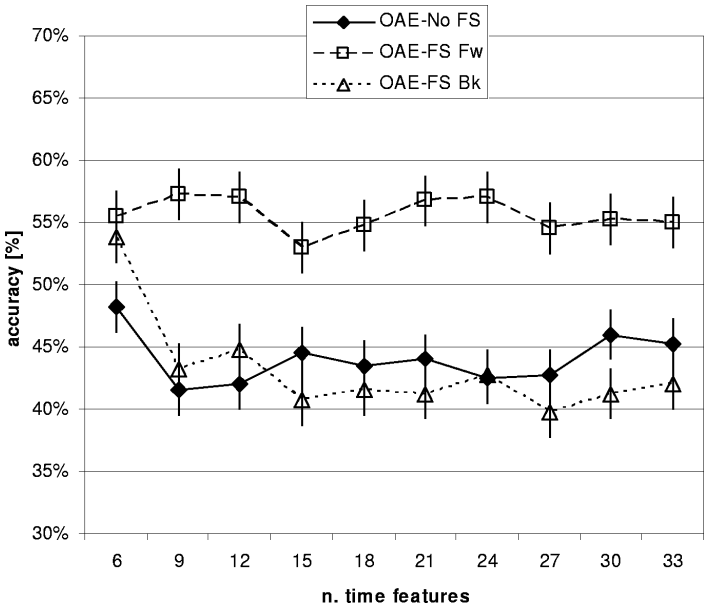

Figure 5: One-against-all ensemble (OAE) performance using time-features

necessary to characterize successfully the beat-histogram. In fact: the accuracy of a RRE-FS Fw (figure 4) is stable (within the error bars) about 60%; the curve relative to OAE-No FS (figure 5) has a knee point at 12 features; comparing the 3 curves relatives to FSSE (figure 6), 9, 12 and 15 features can be considered their relative knee points (FSSE-No FS, FSSE-FS Bk, FSSE-FS Fw). In order to assure a good representation of the problem, keeping a sufficient parameterization of the signal, 12 time features are chosen as descriptors of the beat of the signal. Table 4 summarizes the accuracy score of each classifier using 12 time features.

Table 4: Classifiers accuracy using 12 time features

classifier	***accuracy [%]***	***error [%]***
SKNN	41	2
SKNN-FS Fw	40	3
SKNN-FS Bk	40	2
RRE	43	1
RRE-FS Fw	59	2
RRE-FS Bk	48	2
OAE	42	1
OAE-FS Fw	57	3
OAE-FS Bk	45	2
FSSE50	50	2
FSSE50-FS Fw	52	3
FSSE50-FS Bk	53	2

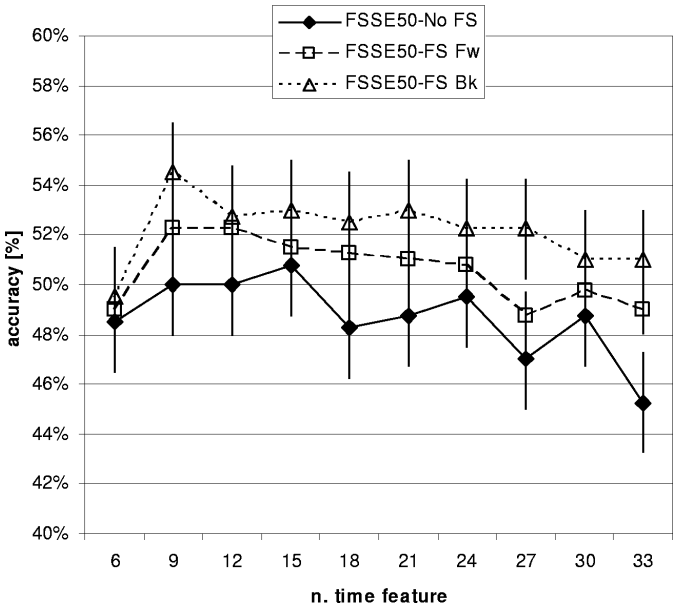

Figure 6: Feature sub space based ensemble (FSSE) performance using time-features

5.3.2 Bounding the number of frequency features

In order to restrict the number of frequency features used to characterize the signal, we performed a similar experiment. Using 4 different signal representations, we evaluated the accuracy behavior of the classifiers in order to estimate the best number of frequency features. As explained in section 3.1, 32 features are obtained by considering position and energy of the most intense peak lying in each bin of table 1; 52 features are obtained by adding information about the position and the energy of the second most intensive peak; 72 features are obtained recording energy and position of the 3 most intensive peaks lying in each bin; 92 features are obtained recording energy and position of the 4 most intensive peaks lying in each bin. The evaluation methodology is the same as described in section 5.1.

Figure 7 (error bars are omitted for clarity) shows the accuracy curves obtained for the 4 different signal representations, using a simple k-NN (SKNN), a round-robin ensemble (RRE) and a one-against-all ensemble (OAE). The single classifier and the ensemble members consider the first 5 nearest neighbors (k=5) for classification. the round-robin ensemble with forward search gets the best score among the other classifiers shown, considering both 32 - about $(80 \pm 1)\%$ - and 52 - about $(80 \pm 2)\%$ - frequency features.

Figure 8 shows the accuracy curve of a feature sub-space based ensemble (FSSE) varying the number of ensemble members and the number of frequency features. It presents the accuracy behavior considering 3 different FSSE ensembles of 10, 30 and 50 members. The dimension of the subspaces is 4. Error bars are omitted for clarity. The graph clearly demonstrates that there is an accuracy improvement by increasing the number of ensemble members. Moreover, the higher the number of ensemble members, the higher the stability of

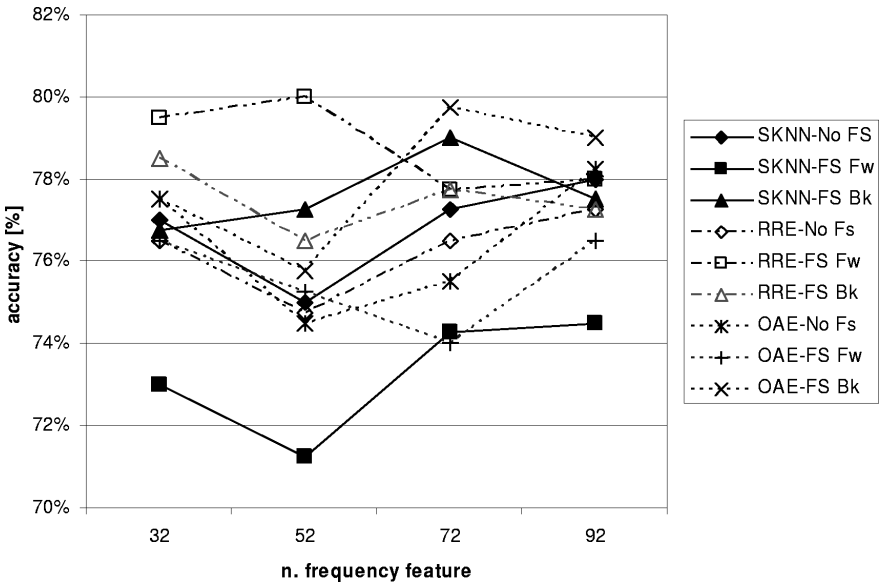

Figure 7: Accuracy curves of a simple k-NN, RRE ensemble and OAE ensemble using frequency features.

the system. Using 10 ensemble members, the accuracy curves range between 73% and 77%. With 50 ensemble members, the score ranges between 81% and 83%. Feature selection tends to outperform the standard classifier, resulting in accuracy gain when a small number of features is used.

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 32 frequency features: mean and max energy of the spectrum and, for each frequency bin in table 1, the position and intensity of the most prominent peak plus the total number of peaks in each bin. Table 5 summarizes the accuracy score of each classifier using 32 frequency features.

Table 5: Classifiers accuracy using 32 frequency features

classifier	***accuracy [%]***	***error [%]***
SKNN	77	1
SKNN-FS Fw	73	1
SKNN-FS Bk	77	2
RRE	77	1
RRE-FS Fw	80	1
RRE-FS Bk	79	1
OAE	77	1
OAE-FS Fw	78	2
OAE-FS Bk	77	1
FSSE50	81	2
FSSE50-FS Fw	81	1
FSSE50-FS Bk	82	1

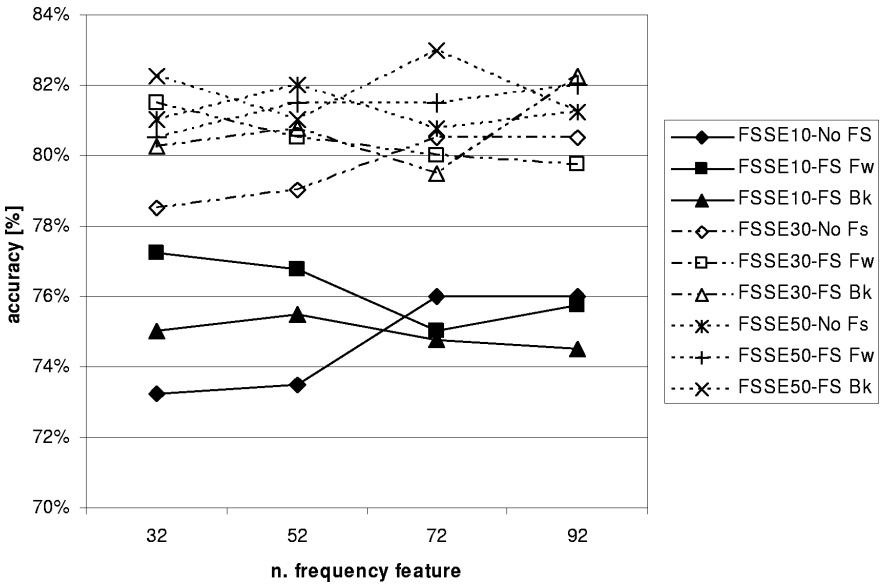

Figure 8: Accuracy curves of a FSSE ensemble varying the number of ensemble members.

In conclusion, we parameterize the input signal using a total of 44 time-frequency features. The 12 time-features are obtained from the beat-histogram of the sound: its mean energy, its max energy, the total number of peaks and position, intensity and width of the 3 most intensive peaks. The frequency features are extracted from the spectrum of the input signal as described in section 3.1 and are: mean and max energy of the spectrum and, for each frequency bin in table 1, the position and intensity of the most intensive peak plus the total number of peaks in each bin.

Comparison against alternative descriptors In this section we present a comparison of the features presented here against alternatives presented in the research literature [16, 17, 30] on this music classification task. Because of the absence of benchmark datasets in this area the comparison is not straightforward. We used a public software framework for computer audition applications (Marsyas [2]) to extract two sets of new descriptors:

- Mel-Frequency Cesptral Coefficients (MFCC)
- Features based on Short Time Fourier analysis (STFT) such as:
  - Spectral Centroid
  - Spectral Roll-off
  - Spectral Flux
  - Zero Crossing

The very same software has been used in [30] for feature extraction. For a comprehensive description of the features listed above, we refer the reader to [17, 30, 16] and to the user manual of Marsyas [2].

Table 6 shows the results obtained on the *5classes* dataset using frequency features based on MFCC and STFT respectively. The evaluation methodology is the same as presented in section 5.1. The number of nearest neighbors used for classification is 5; the subspace dimension for the FSSE classifier is equal to 4 and the total number of ensemble members is 50.

Table 6: Classifiers accuracy using MFCC and STFT features on the *5classes* database

	MFCC		***STFT***	
classifier	***a. [%]***	***e. [%]***	***a. [%]***	***e. [%]***
SKNN	65	2	54	2
RRE	66	1	53	2
OAE	64	2	55	1
FSSE50	69	2	59	2

An examination of table 5 and table 6 shows that the set of frequency-features proposed in section 3.1 (32 frequency features derived from the DWPT) provides a good characterization of the spectrum of the signal and is appropriate for the problem of music genre classification. The set of features extracted through the DWPT guarantees a gain in generalization accuracy of more than 10% compared to MFCC features and more than 20% compared to STFT features.

In table 7 and table 8, we present the results obtained on the *4classes* and *6classes* databases.

Table 7: Classifier accuracy using different frequency features on the *4classes* database

classifier	***accuracy [%]***	***error [%]***
DWPT features		
SKNN	78	2
RRE	78	1
OAE	78	1
FSSE50	80	1
MFCC features		
SKNN	70	2
RRE	72	1
OAE	70	1
FSSE50	74	2
STFT features		
SKNN	62	2
RRE	62	2
OAE	63	2
FSSE50	67	2

The results reported in the two tables confirm that our approach is consistently more effective than a parameterization of the audio signal based on STFT or MFCC. Both increasing and decreasing the number of classes in the dataset,

Table 8: Classifiers accuracy using different frequency features on the *6classes* database

classifier	***accuracy [%]***	***error [%]***
DWPT features		
SKNN	73	1
RRE	74	1
OAE	73	1
FSSE50	78	1
MFCC features		
SKNN	66	1
RRE	65	1
OAE	66	1
FSSE50	67	1
STFT features		
SKNN	54	1
RRE	52	2
OAE	54	1
FSSE50	57	1

the gain in accuracy provided by DWPT frequency features ranges between 10% and 20%.

5.4 Combining time and frequency features

5.4.1 Applying *k*-NN classifiers

Figure 9 shows the accuracy curves for 4 different classifiers; simple k-NN (SKNN), round-robin ensemble (RRE), one-against-all ensemble (OAE) and feature sub-space ensemble (FSSE) trained on the 5 class problem datased (*5classes* dataset).

The feature sub-space ensemble (FSSE) has 50 ensemble members and 8 features per member (sub-space dimension). The FSSE ensemble performs better than any other predictor, in the range of k explored. The ensemble shows a mean accuracy of 83%, and scores about $(84 \pm 1)\%$ for k=4,5,6.

Figure 10 illustrates the accuracy behavior of a feature sub-space ensemble (FSSE) varying the sub-space dimension on the *5classes* dataset The number of neighbors (5) and the number of ensemble members (50) have been kept fixed. The graph shows that for sub-space dimension 6-10 the accuracy reach its maximum: $(84 \pm 1)\%$.

Figure 11 shows the accuracy curve of a feature sub-space ensemble varying the number of members. The number of neighbors (5) and the sub-space dimension (8) has been kept fixed. The graph demonstrates that by increasing the number of members in the ensemble the accuracy increases. The curve suggests that after a certain value (50) the accuracy tends to stabilize around 84%.

Table 9, table 10 and table 11 summarize the results obtained applying SKNN, RRE, OAE and FSSE to the datasets presented in section 5.2. The tables clearly demonstrate that in case of k-NN classifiers feature sub-space based ensemble is a winning strategy. Moreover, applying the feature selection

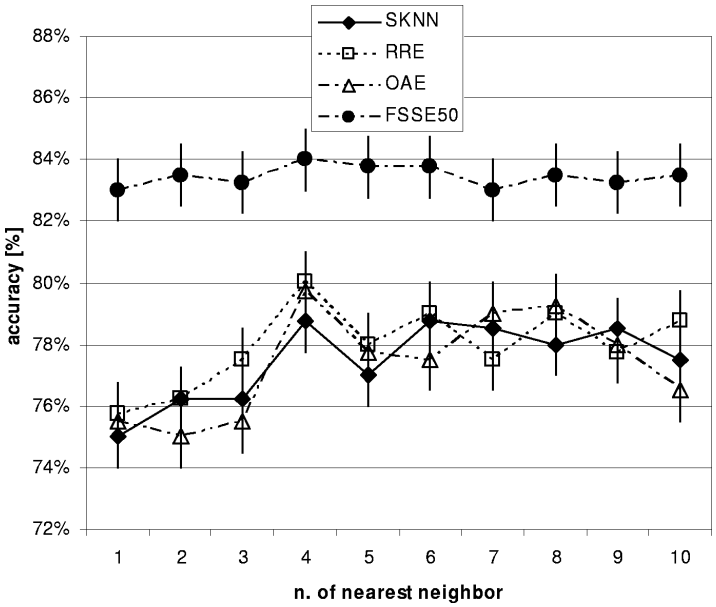

Figure 9: Accuracy curves of different classifiers varying the number of nearest neighbors.

strategies presented in section 4.4 provide an improvement of the performance of the ensemble.

Table 9: Accuracy of k-NN classifiers on the *4classes* and *5classes* dataset.

	4classes		***5classes***	
classifier	***a. [%]***	***e. [%]***	***a. [%]***	***e. [%]***
SKNN	83	1	78	1
SKNN-FS Fw	79	3	76	2
SKNN-FS Bk	84	2	78	1
RRE	82	1	78	1
RRE-FS Fw	82	1	82	2
RRE-FS Bk	85	2	80	3
OAE	84	1	78	1
OAE-FS Fw	84	2	80	1
OAE-FS Bk	86	2	81	1
FSSE50	86	2	84	1
FSSE50-FS Fw	86	1	85	1
FSSE50-FS Bk	86	1	85	1

5.4.2 Applying ANN classifiers

In the following we present the evaluation of a simple ANN (SANN), a round-robin ensemble of ANNs (RRANN) and a feature sub-space ensemble of ANNs (FSSANN). The FSSANN has 50 ensemble members and sub-space dimension

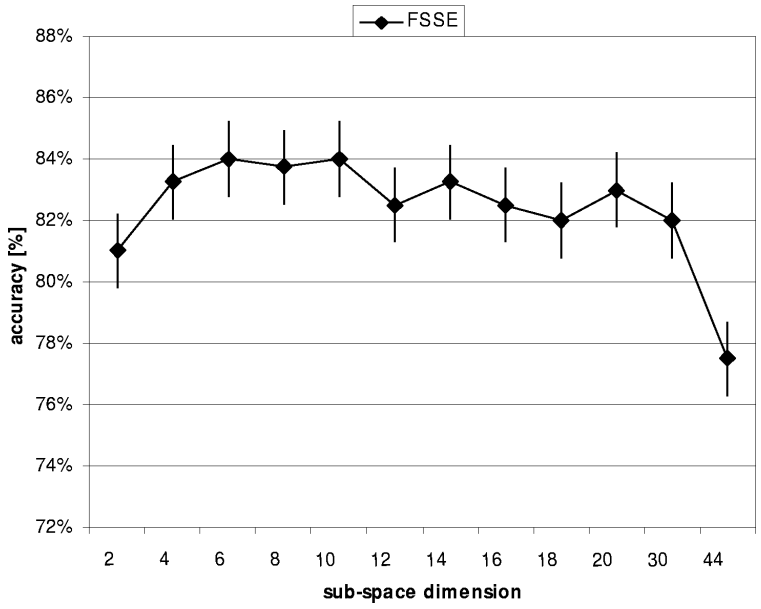

Figure 10: Accuracy curves of FSSE varying the subspace dimension.

Table 10: Accuracy of k-NN classifiers on the *6classes*, *7classes* and dataset.

	6classes		***7classes***	
classifier	***a.** [%]*	***e.** [%]*	***a.** [%]*	***e.** [%]*
SKNN	75	1	64	1
SKNN-FS Fw	71	3	64	1
SKNN-FS Bk	75	2	64	2
RRE	75	1	65	1
RRE-FS Fw	78	2	73	2
RRE-FS Bk	76	2	68	2
OAE	75	1	64	1
OAE-FS Fw	71	2	65	3
OAE-FS Bk	76	1	68	1
FSSE50	81	1	74	2
FSSE50-FS Fw	82	2	77	1
FSSE50-FS Bk	83	1	76	1

equal to 8.

The signal representation (aka the number of time and frequency features) is the same as evaluated in section 5.3: 12 time-features and 32 frequency features (for a total of 44 time-frequency features).

The back-propagation algorithm is used for training the predictors on the datasets in section 5.2. The parameters used in the back-propagation are as follows: learning rate equal to 0.3 and the momentum equal to 0.2. The number of units in the inner layer is calculated at runtime as the square root of $n_{inputs} \cdot n_{outputs}$. The number of input units (n_{inputs}) is given by the number of features (44) used for characterization of the audio signal. The number of output units

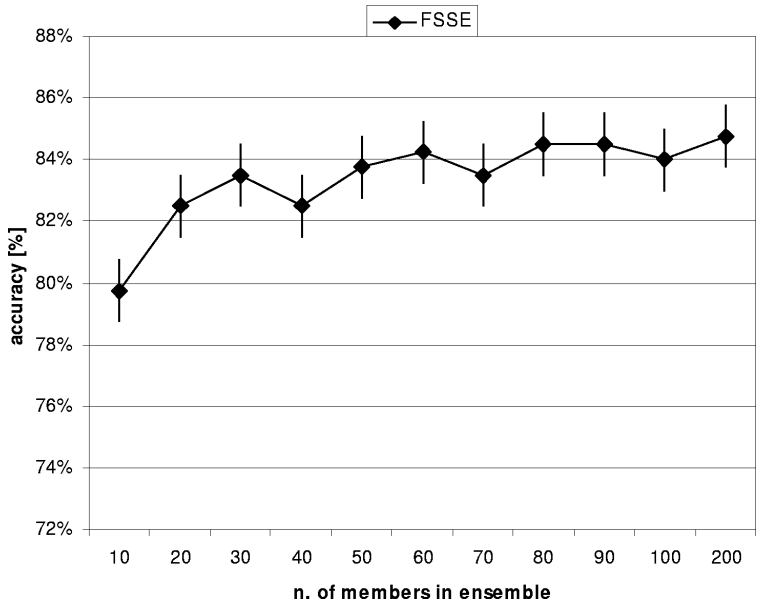

Figure 11: Accuracy curves of FSSE ensemble varying the number of members.

Table 11: Accuracy of k-NN classifiers on the *8classes* dataset.

classifier	***accuracy [%]***	***error [%]***
SKNN	58	1
SKNN-FS Fw	59	2
SKNN-FS Bk	57	2
RRE	59	1
RRE-FS Fw	65	2
RRE-FS Bk	61	2
OAE	58	1
OAE-FS Fw	54	2
OAE-FS Bk	61	1
FSSE50	67	1
FSSE50-FS Fw	69	2
FSSE50-FS Bk	68	1

($n_{outputs}$) is equal to the number of classes in the dataset: each output unit represent one single output class. For a 2 class - A B - problem, the output unit configuration (1, 0) represents instances belonging to class A; the output unit configuration (0, 1) represents instances belonging to class B. The evaluation methodology is the same as described in section 5.1: 10 randomized 10-fold cross-validations, the generalization accuracy is calculated as mean of the 10 runs; the error as standard deviation.

Figure 12 shows the learning curves of the three predictors trained on the *4classes* dataset, varying the training time (epochs).

The graphs show that a simple neural network (SANN) is able to model efficiently the given 4 class problem. After 500 epochs the SANN scores $(89 \pm 1)\%$.

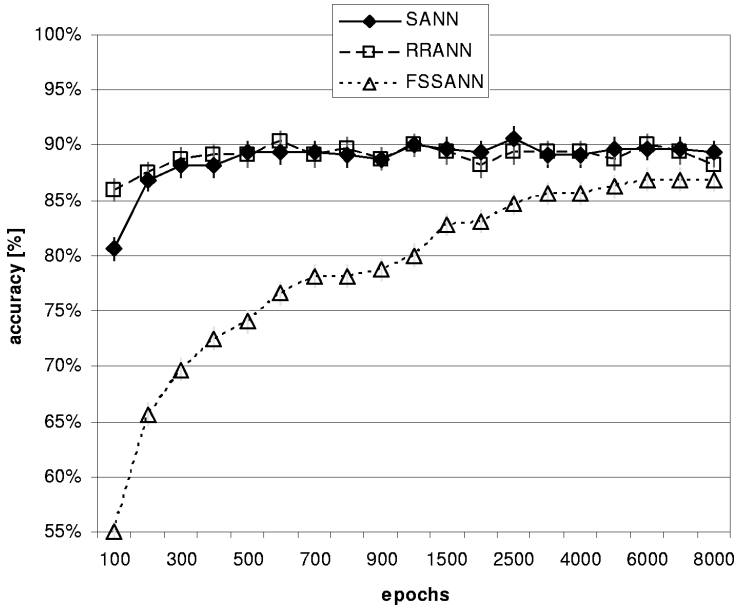

Figure 12: SANN, RRANN and FSSANN on the *4classes* dataset

The round-robin ensemble (RRANN) matches the overall score of the SANN, showing, however, a tendency in providing better results in the early stage of the training process. The accuracy graph of such an ensemble suggests that the RRANN benefits from the aggregation mechanism in this early stage. The accuracy curve for the FSSANN shows that this kind of ensemble technique is not a good methodology if ANNs are used as base predictors. In fact, the two-layer perceptron (SANN) is very stable within the range of epochs explored. In this scenario, the FSSANN ensemble lacks diversity as its members are not locally specialized. The accuracy curve shows that after large training times the FSSANN eventually catches up with the SANN.

Figure 13, 14 and 15 present the same evaluation performed on other 3 datasets (*5classes*, *6classes* and *8classes*). The figures confirm that a simple two-layer perceptron is able to well model the problem of music genre recognition, given the set of features proposed. The ensemble techniques tested do not generally improve the overall score of the SANN, mainly because the high stability of the base predictor in this domain. However, by increasing the number of classes (figure 14), the RRANN show slightly better results than the SANN. In this case, the solution space decomposition performed by the ensemble has its benefit. The FSSANN (figure 15) shows again a characteristic learning curve that suffers due to lack in diversity. By applying large training times, the accuracy increase slowly, showing a tendency to catch up with the accuracy scores of the other two predictors.

Table 12 summarizes the results obtained applying artificial neural networks as base predictors on the datasets presented in section 5.2.

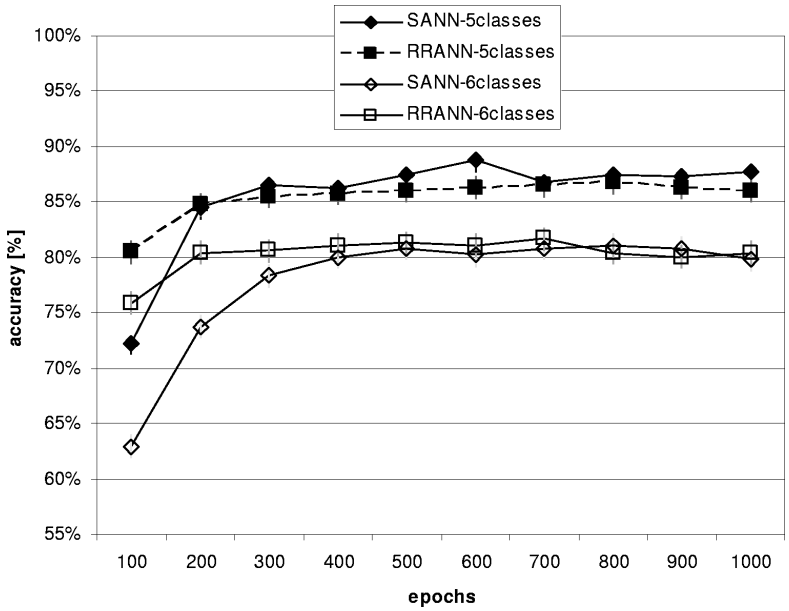

Figure 13: SANN and RRANN on the *5classes* and *6classes* dataset

Table 12: Accuracy of ANN classifiers on 4 different datasets

dataset	***SANN***	***RRANN***	***FSSANN***
4classes	$(89 \pm 1)\%$	$(89 \pm 1)\%$	$(87 \pm 1)\%$
5classes	$(89 \pm 1)\%$	$(87 \pm 1)\%$	$(86 \pm 1)\%$
6classes	$(81 \pm 1)\%$	$(81 \pm 1)\%$	$(79 \pm 1)\%$
7classes	$(74 \pm 1)\%$	$(77 \pm 1)\%$	$(72 \pm 1)\%$
8classes	$(67 \pm 1)\%$	$(68 \pm 1)\%$	$(64 \pm 1)\%$

5.4.3 Applying SVM classifiers

In this section we present the evaluation conducted on the datasets presented in 5.2 using support vector machines as classifiers. Gaussian kernels (radial basis) and linear kernels are evaluated together with round-robin and one-against-all binarization. The kernel (Gaussian and linear) parameters are evaluated using a grid search [14] in a wrapper-like configuration (section 4.4). The signal representation (aka the number of time and frequency features) is the same as evaluated in section 5.3: 12 time-features and 32 frequency features (44 time-frequency features).

In the following we will use the acronym SVM-RR to indicate a support vector machine that makes use of the round-robin binarization schema to handle multi-class problems. Similarly, SVM-OA indicates a support vector machines using the one-against-all binarization procedure to handle multi-class problems. We will use the acronym NOOP to indicate that no optimization has been performed on the kernel parameters. Similarly, OP indicates that the different kernel parameters have been optimize using a grid search in a wrapper-like configuration.

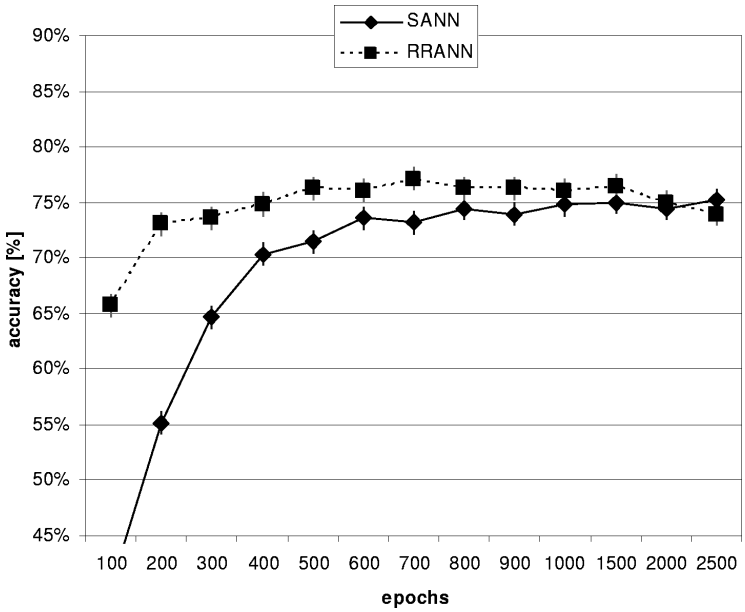

Figure 14: SANN and RRANN on the *7classes* dataset

Table 13: SVM accuracy on the *4classes* and *5classes*dataset.

classifier	***Gauss. Kernel***		***Liner Kernel***	
	a. [%]	***e. [%]***	***a. [%]***	***e. [%]***
	4classes			
SVM-RR-NOOP	83.2	0.5	91	1
SVM-RR-OP	90	2	89	1
SVM-OA-NOOP	66.3	0.5	85	2
SVM-OA-OP	86	1	85	2
	5classes			
SVM-RR-NOOP	75.8	0.8	88	1
SVM-RR-OP	88	1	87	1
SVM-OA-NOOP	49.0	0.5	74	2
SVM-OA-OP	78	1	76	1

Table 13 and 14 indicate that support vector machines handle the problem of music genre classification very well. It is interesting to note that the configuration SVM-RR-NOOP scores the best in the majority of the cases. This fact indicates that the round-robin binarization creates binary-classifiers with linearly separable classes. In fact, applying the grid search to further improve the accuracy of the classifier produces a decrease of the accuracy score, probably due to the occurrence of overfitting in the search. On the other hand, SVM classifiers based on a Gaussian kernel tend to perform worse if no parameter optimization is implemented. The behavior of SVM-OA classifiers (Gaussian kernel) is symptomatic: applying the grid search for the optimization of the kernel parameters, the generalization accuracy improves abruptly. It tends to catch up with the accuracy shown by the SVM-RR (Gaussian kernel) classifiers.

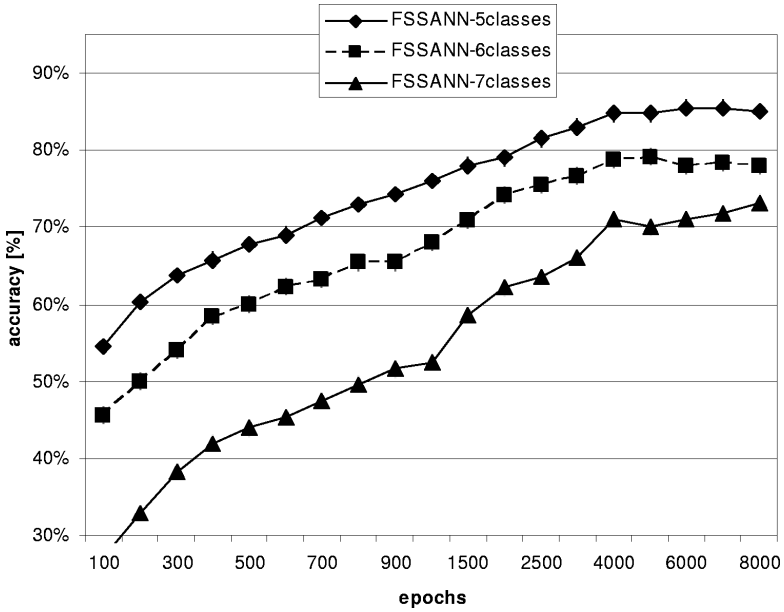

Figure 15: FSSANN on 3 different datasets (*5classes*, *6classes* and *7classes*)

5.5 Fuzziness of the class definition

It is interesting to note how the accuracy of the different classifiers (table 10, table 11, table 12, table 14) drops as the number of classes is increased. As suggested by other authors [21, 30], this is probably due to the inherent fuzziness in the class definition – a factor that becomes more significant as the number of classes is increased.

In table 15 we present a sample confusion matrix obtained using a round-robin SVM ensemble of the *8classes* dataset. C1 indicates songs belonging to class Jazz, C2 to class Rock, C3 to class Tecno, C4 to Acoustic, C5 to Heavy Metal, C6 to Classical, C7 to Country and C8 songs belonging to class Celtic. In table 15, e.g. 5 Rock songs (C2) have been classified as Heavy Metal (C5).

The confusion matrix shows that the fuzziness in the class definition is a significant factor affecting the classifier accuracy. 7 Celtic songs (mainly folk songs of the Irish tradition) are classified as Country. Similarly 6 Acoutsic songs (see section 5.2 for a definition) are classified as Celtic and 5 Acoustic songs as Country. It seems clear that a move to handle more classes will result in less separation between the classes and thus poorer accuracy.

6 Related work

Among the different studies in the literature [3, 7, 9, 21, 27, 28, 29, 30, 32], the work of Tzanetakis et al. [30] is particularly relevant. The authors proposed for the first time the concept of beat-histogram and its implementation using a discrete wavelet transform. Even if the octave band decomposition is not optimal in terms of time resolution, their work demonstrates the usefulness of such a characterization. By combining the rhythmic description with features like

Table 14: SVM accuracy on the *6classes*, *7classes* and *8classes* dataset.

classifier	***Gauss. Kernel***		***Liner Kernel***	
	a. [%]	***e. [%]***	***a. [%]***	***e. [%]***
	6classes			
SVM-RR-NOOP	72	1	82	1
SVM-RR-OP	83	2	81	2
SVM-OA-NOOP	28.5	0.7	68	1
SVM-OA-OP	72	2	68	2
	7classes			
SVM-RR-NOOP	64.0	0.8	76	1
SVM-RR-OP	77	2	76	1
SVM-OA-NOOP	22.5	0.3	61	1
SVM-OA-OP	64	2	62	1
	8classes			
SVM-RR-NOOP	58	1	68	1
SVM-RR-OP	67	1	68	1
SVM-OA-NOOP	20.0	0.3	51	1
SVM-OA-OP	57	2	51	1

Table 15: Sample confusion matrix obtained using a SVM-RR-NOOP on the *8classes* dataset; the accuracy is about 67%.

Q/A	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
C1	*33*	2	0	2	0	0	0	3
C2	3	*24*	0	2	5	0	4	2
C3	0	4	*32*	0	2	0	1	1
C4	4	2	0	*19*	0	4	5	6
C5	0	3	3	0	*33*	0	1	0
C6	0	0	0	1	0	*35*	2	2
C7	4	4	0	6	0	3	*21*	2
C8	3	2	0	7	0	4	7	*17*

spectral centroid, roll-off, flux, zero-crossing and Mel-Frequency Cepstral Coefficients, the author obtain interesting results on a number of different datasets: using a k-NN classifier, trained on a dataset counting 10 different genres, they obtain a generalization accuracy ranging between 56% and 60%. Our work confirms that a characterization of the song beat benefits the overall accuracy of the classifiers - a FSSE of k-NN scores $(81 \pm 2)\%$ considering only frequency features, as table 5 shows. By providing some information about the song beat, the classifier trained on the same database (*5classes* dataset) gets an accuracy score of about $(84 \pm 1)\%$, as table 9 points out. Moreover, our work shows that a better recognition rate can be obtained using frequancy features derived from the DWPT (section 5.3.2).

The study presented by McKinney et al. [21] is another example of research conducted in the area. Using a combination of static and temporal features, the authors performed several experiments in sounds discrimination (music, speech, noise) and music genre recognition. On a dataset counting 7 different gen-

res, using Gaussian-based quadratic discriminant analysis, the authors report a generalization accuracy ranging between $(61 \pm 11)\%$ and $(74 \pm 9)\%$. On a similar problem, the FSSE ensemble based on k-NN classifiers scores $(74 \pm 1)\%$, $(77 \pm 1)\%$ and $(76 \pm 1)\%$, when no feature selection, forward hill-climbing and backward hill-climbing are respectively applied. Artificial neural networks score $(74 \pm 1)\%$, $(77 \pm 1)\%$ and $(72 \pm 1)\%$ in simple ANN configuration (SANN), round-robin configuration (RRANN) and feature sub-space configuration (FASSANN) respectively. A support vector machine (round-robin configuration and linear kernel) scores $(76 \pm 1)\%$

Li et al. [16] showed that a discrete wavelet transform applied to the characterization of the spectrum of audio signals (music) has an enormous advantage compared to other techniques. Moreover, the authors pointed out that a simple ensemble of support vector machines (SVM) performs better than predictors like k-NN and Gaussian mixture models (GMM). Round-robin ensembles and one-against-all ensembles of SVMs outperform simple classifiers built upon k-NN and GMM. Using SVM classifiers and round-robin binarization, the authors reported an accuracy of $(90.50 \pm 4.53)\%$, $(88.00 \pm 3.89)\%$, $(84.83 \pm 4.81)\%$ and $(83.86 \pm 4.26)\%$ on datasets composed respectively of 4, 5, 6 and 7 different classes. Similar results are reported in this work using SVM classifiers and a linear kernel (table 13 and 14.

7 Summary

We have presented a new set of descriptors for music classification that uses a discrete wavelet packet transformation (DWPT) of the input signal. These features are extracted from the spectrogram of the signal taking advantage of the multi-resolution property of the DWPT and the mathematical characteristics of music. Features capturing the harmony of the input signal are obtained by defining a decomposition level suitable for recognizing notes and matching music octaves. Similarly, following the idea of the beat-histogram presented in [30], the tempo of the song is characterized using a decomposition level that can recognize the onset and offset of notes.

A comparison with alternative representations presented in the literature [17, 30] shows that our parameterization is very effective for music genre classification (section 5.3.2). On 4,5 and 6 class problems, the set of features extracted through the DWPT shows and increase in generalization accuracy of more than 10% over that produced using MFCC features and more than 20% compared to STFT features.

Our evaluation using k-NN classifiers (section 5.4.1) shows that feature subspace ensembles (section 4.5) perform better than the other kinds of ensembles tested. Moreover, the adoption of feature selection (section 4.4) delivers an improvement in the performance of the k-NN (table 9, 10 and 11). As emphasized in the introduction, this respectable performance of the k-NN ensemble is important because of its potential role in query-by-example systems.

Employing an ANN as the classifier shows a further increase in the generalization accuracy of classifiers built upon k-NN. However in the case of ANNs, feature sub-space based ensembles do not appear to be a good ensemble strategy. As reported in section 5.4.2 a simple ANN is capable of modelling the classification problem and the ensemble strategies tested do not provide a gain

in generalization accuracy. Interestingly, this gain tends to disappear when increasing the number of classes in the dataset (*7classes* and *6classes* datasets). This phenomenon may be due to the fuzziness in the class (music genre) definition (section 5.5).

Section 5.4.3 shows the results obtained using SVM classifiers. Our analysis indicates that the round-robin binarization creates binary-classifiers with linearly separable classes. In fact, applying a grid search to further improve the accuracy of the classifier produces a decrease in the accuracy score, probably due to the occurrence of overfitting in the parameter setting process. The comparison of the results obtained with SVM and ANN classifiers shows that both of the eager learners are effective at this music classification problem achieving similar accuracy results on the 4 datasets tested. We conclude that a support vector machines with a simple linear kernel and implementing a round-robin binarization of the multi-class problem appers to be the simplest and most effective classifier in this scenario.

References

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

[2] http://opihi.cs.uvic.ca/marsyas/.

[3] E. Allamanche, J. Herre, O. Hellmuth, B. Froeba, T. Kastner, and M. Kremer. Content-based identification of audio material using mpeg-7 low level descritpion. In *Proc. Int. Symposium on Music Information Retrieval (ISMIR 2001)*, Bloomington, IN, USA, 2001.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[5] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[6] T. G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000.

[7] Dixon, E. Pampalk, and G. Widmer. Classification of dance music by periodicity patterns. In *Proc. Int. Symposium on Music Information Retrieval (ISMIR 2003)*, Baltimore, MA, USA, 2003.

[8] L. Fausett. *Fundamentals of Neural Networks: Architecture, Algorithms and Applications*. Prentice-Hall, 1994.

[9] J. Foote. Arthur: Retrieving orchestral music by long term structure. In *Proc. Int. Symposium on Music Information Retrieval (ISMIR 2000)*, Plymouth, MA, USA, October 2000.

[10] J. Foote. A wavelet packet representation of audio signals for music genre classification using different ensemble and feature selection. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, Berkeley, CA, USA, November 2003.

[11] T. Foote. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, volume 3229, pages 138–147, 1997.

[12] J. Fürnkranz. Round robin rule learning. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 146–153, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[13] M. Grimaldi. *Learning to Annotate Music Files using Content Based Retrieval Systems and Wavelet Packet Approximations of the Input Signals*. PhD thesis, Trinity College Dublin, 2004.

[14] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transaction on Neural Networks*, 13(2):415–425, 2002.

[15] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, New Orleans, USA, 1994. AAAI Press.

[16] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 282–289, 2003.

[17] B. Logan. Mel frequency ceptral coefficients for music modeling. In *Proc. Int. Symposium on Music Information Retrieval (ISMIR 2000)*, Plymouth, MA, USA, 2000.

[18] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[19] K. Martin. Musical instrument identification: A pattern-recognition approach. In *The 136th meeting of the Acoustical Society of America.*, October 1998.

[20] K. Martin. Toward automatic sound source recognition: Identifying musical instruments. In *Proc. NATO Computational Hearing Advanced Institute*, Il Ciocco, Italy, July 1998.

[21] M. McKinney and J. Breebaart. Features for audio and music classification. In *Proc. Int. Symposium on Music Information Retrieval (ISMIR 2003)*, Baltimore, MA, USA, 2003.

[22] T. Mitchell. *Machine Learning*. McGraw Hills, 1997.

[23] P.Cunningham and J.Carney. Diversity versus quality in classification ensembles based on feature selection. In *Eleventh European Conference on Machine Learning (ECML 2000)*, pages 109–116. Springer Verlag, 2000.

[24] A. Pienimäki. Indexing music databases using automatic extraction of frequent phrases. In *Proc. Int. Symposium on Music Information Retrieval (ISMIR)*, Paris, France, October 2002.

[25] J. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[26] R.Kohavi and G.H.John. Wrappers for feature subset selection. *IEEE Transaction on Neural Networks*, 97(1-2):273–324, 1997.

[27] E. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of Acouststic Society of America*, 103(1):588–601, January 1998.

[28] E. Scheirer and M. Slaney. Construction and evaluation of arobust multi-features speech/music discriminator. In *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, volume 2, pages 1331–1334, Munich, Germany, April 1997.

[29] G. Tzanetakis, A. Ermolinskyi, and P. Cook. Pitch histograms in audio and symbolic music information retrieval. In *Proc. Int. Symposium on Music Information Retrieval (ISMIR 2002)*, Paris, France, October 2002.

[30] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. In *Proc. Int. Symposium on Music Information Retrieval. (ISMIR)*, Bloomington, IN, USA, October 2001.

[31] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[32] Y. Wang, Z. Liu, and J.C.Huang. Multimedia content analysis using both audio and visual clues. *IEEE Signal Processing Magazine*, pages 12–36, November 2000.

[33] G. Zenobi and P. Cunningham. Using diversity in preparing ensemble of classifiers based on different subsets to minimize generalization error. In *12th European Conference on Machine Leaming (ECML 2001)*. Springer Verlag, 2001.

[34] T. Zhang and C. Kuo. Hierarchical classification of audio data for archiving and retrieving. In *In IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Phoenix, AZ, USA, March 1999.