

A CARTESIAN ENSEMBLE OF FEATURE SUBSPACE CLASSIFIERS FOR MUSIC CATEGORIZATION

T. Lidy, R. Mayer, A. Rauber
Vienna University of Technology, Austria
Department of Software Technology
and Interactive Systems

P. J. Ponce de León, A. Pertusa, J. M. Iñesta
University of Alicante, Spain
Departamento de Lenguajes y
Sistemas Informáticos

ABSTRACT

We present a cartesian ensemble classification system that is based on the principle of late fusion and feature subspaces. These feature subspaces describe different aspects of the same data set. The framework is built on the Weka machine learning toolkit and able to combine arbitrary feature sets and learning schemes. In our scenario, we use it for the ensemble classification of multiple feature sets from the audio and symbolic domains. We present an extensive set of experiments in the context of music genre classification, based on numerous Music IR benchmark datasets, and evaluate a set of combination/voting rules. The results show that the approach is superior to the best choice of a single algorithm on a single feature set. Moreover, it also releases the user from making this choice explicitly.

1. INTRODUCTION AND RELATED WORK

Classification of music into different categories is an important task for retrieval and organization of music libraries. Previous studies reported about a glass ceiling reached using timbral audio features for music classification [1]. Our approach is based on the assumption that a diversity of music descriptors and a diversity of machine learning algorithms are able to make further improvements. We created an ensemble learning system with these two dimensions (feature sets, learning schemes) as input and train models for each combination of those two input dimensions. We call our approach a *cartesian ensemble system*.

Our original motivation has been to combine multiple approaches from the music information retrieval (MIR) domain in order to improve (the reliability of) genre classification results based on the assumption that the various music descriptors are complementary [12]. In our previous work we combined spectrum-based audio features that cover timbral and rhythmic aspects of the sound with symbolic descriptors, based on note and chord sequence statistics. A polyphonic transcription system has been presented as the “missing link” that transcribes audio data into a sym-

bolic notation. In this approach the combination of multiple features from the audio and symbolic domains was performed by a concatenation of feature vectors, jointly used as input to a classification algorithm (*early fusion*). In a previous comparison of employing MIR algorithms on Western vs. ethnic music [10] we included a *time decomposition* approach, which was already a first ensemble-like approach, applying one learning scheme on multiple input features from different segments of a piece of music and using four different combination (voting) rules to make the final prediction.

The Autonomous Classification Engine ACE [13], by contrast, is a general framework for *model selection*. In machine learning, model selection is the task of selecting one classification model from a pool of models. ACE trains a range of classifiers, with different parameters, and feature selection methods, and then selects the most fitting ones for the current task at hand. ACE is built on top of Weka [20] and thus provides the ensemble techniques implemented in the toolkit, most prominently boosting and bagging, but is not capable of handling *feature subspaces*, or weighted methods as the ones described in Section 3.

The combination of different segments extracted from the same song is studied in [2]. The approach is based on grouping and aggregating non-overlapping blocks of consecutive frames into segments. The segments are then classified individually and the results are aggregated for a song by majority voting. Three different ensemble methods and their applicability to music are investigated in [7]. The first method is based on a *one against all* scheme, i.e. for each class, a classifier is trained on the class and its complement. A second method is based on building a classifier for each pairwise combination of classes. The third method investigates in training different classifiers on different subsets of the feature space. In all methods, the final class label is determined by the probabilities of the individual classifiers.

The approach presented in this paper is a *cartesian ensemble classification* system, which trains a matrix of models built from the combination of a range of individual feature sets and a number of classification algorithms. Our system builds on the Weka machine learning toolkit [20] in an open and flexible way. In contrast to ACE no preselection of classification algorithms has been made – any classification algorithm available can be used with arbitrary parameters in the ensemble. Further, an arbitrary number of feature files can be used. We provide a number of com-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© 2010 International Society for Music Information Retrieval.

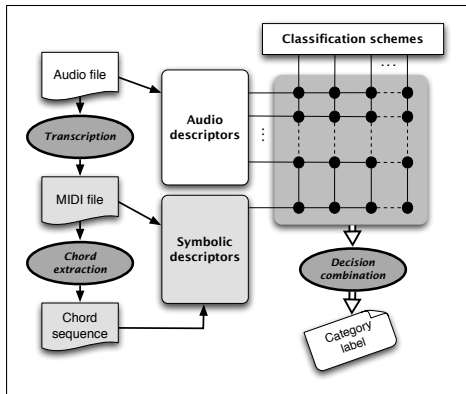


Figure 1. Framework of the cartesian ensemble system

combination and voting rules, which are employed to obtain the final prediction of the classifier ensemble. Our framework is not limited to MIR applications. With regard to our original motivation and our research background, however, we focus on the scenario of music classification into genre categories, in order to show the applicability of the system and the progress in our domain.

The overall scheme of our proposed ensemble classification system is shown in Figure 1. It includes our scenario of a music classification system that processes different descriptors from the audio and symbolic domains (c.f. Section 2). Audio feature extraction algorithms are applied directly to the audio signal data. There is an intermediate step for the symbolic descriptors: A polyphonic transcription system converts the audio information into a symbolic notation (i.e. MIDI files). A chord inference algorithm is applied to provide information about the polyphonic structure of the note stream. Finally, a symbolic feature extractor is applied on the resulting representation. The feature extraction stage provides multiple viewpoints on music objects, called *feature subspaces*. There are several ways of combining them for building a music classification system. *Early fusion* concatenates all feature subspaces to produce so called *superinstances*, including all features at hand. Then a suitable classification scheme is used to learn categories from such data. This approach was used in our previous work [12]. *Late fusion* combines classifier outcomes rather than features. This is the approach employed in our proposed framework.

Section 3 describes the general architecture of our ensemble framework. In Section 4, we evaluate our approach on numerous well-known reference music datasets and show the applicability of the approach. It includes also preliminary research on the use of audio segmentation for generating extended feature subspaces. Finally, Section 5 provides conclusions and an outlook on future work.

2. MUSIC DESCRIPTION

We use two sources of input to our ensemble music classification approach: audio features extracted from audio files and symbolic music descriptors derived from MIDI files that are generated from audio files through a transcription

system. We employ features that proved well in our previous works [5, 10–12], also in order to be able to compare progress and results of the new ensemble approach with previous findings. We emphasize, however, that arbitrary feature sets can be used with our classifier ensemble approach presented in Section 3.

2.1 Audio Features

All the following descriptors are extracted from a spectral representation of an audio signal, partitioned into segments of 6 sec. Features are extracted segment-wise, and then aggregated for a piece of music computing the median (RP, RH) or mean (SSD, MVD) from features of multiple segments. We describe the feature extraction algorithms very briefly, please refer to the references for further details.

Rhythm Pattern (RP) The feature extraction process for a Rhythm Pattern is composed of two stages. First, the specific loudness sensation on 24 critical frequency bands is computed through a Short Time FFT, grouping the resulting frequency bands to the Bark scale, and successive transformation into the Decibel, Phon and Sone scales. This results in a psycho-acoustically modified Sonogram representation that reflects human loudness sensation. In the second step, a discrete Fourier transform is applied to this Sonogram, resulting in a spectrum of loudness amplitude modulation per modulation frequency for each critical band. After additional weighting and smoothing steps, a Rhythm Pattern exhibits magnitude of modulation for 60 modulation frequencies on the 24 critical bands [11].

Rhythm Histogram (RH) A Rhythm Histogram (RH) aggregates the modulation amplitude values of the critical bands computed in a Rhythm Pattern and is a descriptor for general rhythmic characteristics in a piece of audio [11].

Statistical Spectrum Descriptor (SSD) The first part of the algorithm, the computation of specific loudness sensation, is equal to the Rhythm Pattern algorithm. Subsequently at set of statistical values¹ are calculated for each individual critical band. SSDs describe fluctuations on the critical bands and capture both timbral and rhythmic information very well [11].

Modulation Frequency Variance Descriptor (MVD) This descriptor measures variations in the critical bands for a specific modulation frequency of the Rhythm Pattern matrix, representing the amplitudes of 60 modulation frequencies on 24 critical bands. The MVD vector is computed by taking statistics¹ for each modulation frequency over the 24 bands [10, 12].

Temporal Features (TRH, TSSD) Feature sets are frequently computed on a per segment basis and do not incorporate time series aspects. We introduced therefore TRH and TSSD features that include a temporal dimension describing variations over time.

For TRH, statistical measures¹ are computed over the individual Rhythm Histograms extracted from the individual 6-second segments in a piece of audio. Thus, change and variation of rhythmic aspects in time are captured.

¹ mean, median, variance, skewness, kurtosis, min and max

TSSD analogously capture timbral variations and changes over time in the spectrum on the critical frequency bands. Hence, a change of rhythmic, instruments, voices, etc. over time is reflected by this feature set [10].

2.2 Transcription from Audio to MIDI

A multiple fundamental frequency (f_0) estimation method is used to convert the audio files to MIDI files. This is a joint estimation approach, which experimentally obtained a high accuracy with a low computational cost. It extends a previous work [16] by adding information about neighboring frames to get a smooth temporal estimation. It does not separate instruments, therefore producing single track MIDI files without any timbral information.

2.3 Symbolic Features

A set of statistical descriptors is extracted directly from transcribed notes. This set is based on the features described in [5], well suited for monophonic classical/jazz classification, and on features described in [17], used for melody track selection in MIDI files. Overall statistics, such as the average number of notes per beat, the occupation rate (non-silence periods with respect to song length) and polyphony rate (proportion of sounding note periods with more than one note active simultaneously) are computed. Further, note pitches, pitch intervals, note durations, silence durations, Inter Onset Intervals (IOI) and non-diatonic notes are analyzed; each property is described by min and max values, range, average, standard deviation, and a normality distribution estimator. Other features include the number of distinct pitch intervals, pitch interval mode, and an estimation of the number of syncopations in the song.

Most of these features are somewhat 'melody-oriented' (e.g., interval-based features). In order to capture relevant information about the polyphonic structure of the transcription, a chord sequence is extracted from it, using the algorithm from Pardo and Birmingham [14], and subsequently analyzed. The different kinds of chord extracted are: major triad, major 7th, dominant 7th, dominant suspended 7th, dominant 7th (sharp 5th), dominant 7th (flat 5th), minor 7th, half diminished and fully diminished chords. The relative frequencies of these chords in a chord sequence are computed as symbolic features. A total of 61 statistical descriptors are therefore provided to the system as a symbolic feature subspace.

3. CARTESIAN ENSEMBLE SYSTEM

Our approach is named a *cartesian ensemble* because the set of models used as base classifiers is the cartesian product of D feature subspaces by C classification schemes. A model is built by training classification scheme c_i on feature subspace d_j . This produces a total of $D \times C$ base models as the ensemble. The aim of this approach is to obtain a sufficiently *diverse* ensemble of models that will guarantee, up to a certain degree, an improvement of the ensemble accuracy over the best single model trained. Moreover, the

ensemble abstracts from the selection of a particular classifier and feature set to use for a particular problem. Selecting sufficiently different schemes (different classification paradigms, methods,...) the ensemble provide results that are at least comparable to the best single scheme.

Model diversity is a key design factor for building effective classifier ensembles [9]. This has been empirically shown to improve the accuracy of an ensemble over its base models when they are numerous enough. For selecting the most diverse models within the ensemble the *Pareto-optimal* selection strategy is applied in order to discard models not diverse or not accurate enough.

When a new music instance is presented to the trained ensemble, predictions are made by selected models, which are then combined to produce a single category prediction outcome. A number of decision *combination* (or label fusion) *rules*, can be used for this final prediction.

The cartesian ensemble system is built on the Weka toolkit [20]. The ensemble is a Weka classifier itself, so it can be plugged into any system using this toolkit.

3.1 Pareto-optimal Classifier Selection

This strategy for selecting the best set of models is based on finding the Pareto-optimal set of models by rating them in pairs, according to two measures [9]. The first one is the *inter-rater agreement* diversity measure κ , defined on the coincidence matrix M of the two models. The entry $m_{r,s}$ is the proportion of the dataset, which model h_i labels as L_r and model h_j labels as L_s . The agreement between both classifiers is given by

$$\kappa_{ij} = \frac{\sum_k m_{kk} - ABC}{1 - ABC} \quad (1)$$

where ABC is *agreement-by-chance*

$$ABC = \sum_r \left(\sum_s m_{r,s} \right) \left(\sum_s m_{s,r} \right) \quad (2)$$

The second one is the pair average error, computed by

$$e_{ij} = 1 - \frac{\alpha_i + \alpha_j}{2} \quad (3)$$

where α_i and α_j are the estimated accuracy of the two models, computed as described in Section 3.3. The Pareto-optimal set contains all non-dominated pairs. A pair of classifiers is non-dominated iff there is no other pair that is better than it on both criteria.

3.2 Combination Rules

The combination rules implemented in the system are both weighted and unweighted majority voting rules. A summary of weighted and unweighted combination rules is presented in Table 1, where $P(L_k | \mathbf{x}_i)$ is the posterior probability of instance \mathbf{x} to belong to category L_k , given by model h_i . \mathbf{x}_i is what h_i knows about \mathbf{x} , i. e., feature values that correspond to the feature subspace h_i was trained on. Unweighted combination rules are described in [8], and used through their implementation in Weka.

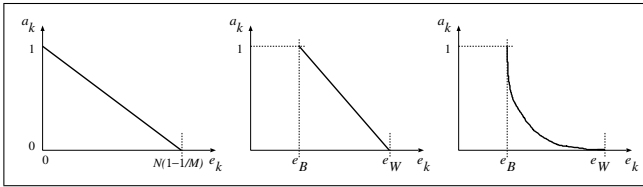


Figure 2. Model weight computation: RSWV (left), BWWV (center), QBWWV (right), giving the model authority a_k as a function of the estimated number of errors e_k made by model h_k on a validation set. N is the number of instances in the set, M is the number of class labels.

All weighted rules multiply model decisions by weights and select the label L_k that gets the maximum score. Model weights are based on the estimated accuracy α_i of the trained models. The *authority* a_i of each model h_i is established as a function of α_i , normalized, and used as its weight ω_i . Weighted methods discussed in [6] have been used in this work. SVW computes weights as described. Weight functions for rules RSWV, BWWV and QBWWV are shown in Figure 2. There, e_B is the lowest estimated number of errors made by any model in the ensemble on a given validation dataset, and e_W is the highest estimated number of errors made by any of those classifiers. WMV is a theoretically optimal weighted vote rule described in [9], where model weights are set proportionally to $\log(\alpha_i/(1 - \alpha_i))$.

Table 1. Summary of combination rules.

Rule mnemonic	Description
<i>Unweighted rules</i>	
MAJ	Majority vote rule
AVG	Average of $P(L_k \mathbf{x}_i)$
MAX	Maximum of $P(L_k \mathbf{x}_i)$
MED	Median of $P(L_k \mathbf{x}_i)$
<i>Weighted rules</i>	
SWV	Simple Weighted Vote
RSWV	Rescaled Simple Weighted Vote
BWWV	Best-Worst Weighted Vote
QBWWV	Quadratic Best-Worst Weighted Vote
WMV	Weighted Majority Vote

3.3 Inner/Outer Cross Validation

The classification results presented below are estimated by cross-validating the ensemble. The accuracy of individual ensemble models (α_i), used to compute model weights for combining their outputs, is also estimated through cross-validation. In order to avoid using test data for the ensemble for single model accuracy estimation, an *inner cross-validation*, relying only on ensemble training data, is performed. The number of folds for the ensemble (outer) and the single models (inner) cross-validation are parameters.

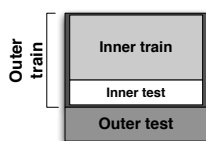


Figure 3. Inner and outer cross-validation scheme.

4. EVALUATION

We performed an extensive evaluation of our ensemble approach on a range of well-known MIR benchmark datasets in order to show both the feasibility and generality of our approach. Classification results are presented as accuracy values with standard deviations.

4.1 Datasets

A dataset overview is given in Table 2. Either full songs or 30 second excerpts were available. 9GDB is originally a MIDI collection, but was synthesized to wav for our experiments and re-transcribed to MIDI to obtain symbolic features. For all other collections audio files were transcribed to MIDI. The GTZAN collection was assembled and used in experiments by G. Tzanetakis [19]. The ISMIRgenre and ISMIRrhythm collections were compiled for the genre and rhythm classification tasks, respectively, of the ISMIR 2004 Audio Description contest [3] and used frequently thereafter by Music IR researchers. ISMIRgenre consists of 6 popular music genres and ISMIRrhythm comprises 8 Latin and Ballroom dances. The Latin Music Database comprises 10 Latin music genres [18]. The African collection is a sub-set of 1024 instances of the audio archive of the Royal Museum of Central-African Belgium, digitized in the course of the DEKKMMA project [4]. Various meta-data categories are available for this set, including 27 different *functions*, 11 different *instrument families*, 11 different *countries* and 40 *ethnic groups* [10]. The number of files varies according to number of meta-data available in each category.

Table 2. Datasets used in experiments

dataset	files	genres	file length	ref.
9GDB	856	9	full	[15]
GTZAN	1000	10	30 sec	[19]
ISMIRgenre	1458	6	full	[3]
ISMIRrhythm	698	8	30 sec	[3]
LatinMusic	3225	10	full	[18]
Africa	1024	var.	full	[4]

4.2 Classification Schemes and System Parameters

For our experiments, we set the system to perform 10-fold outer cross-validation and 3-fold inner cross-validation. As for the classification schemes, a selection of classifiers from the *Weka* toolkit has been made, aiming at choosing schemes from different machine learning paradigms. We chose Naïve Bayes, Nearest Neighbor (IB1²) with Euclidean distance, 3-NN with Manhattan distance (IBk), the RIPPER rule learner (JRip), the C4.5 (J48) decision tree learner, the REPTree, a fast decision tree learner, Random Forest, a forest of random trees, and three Support Vector Machines, the first with a linear kernel, the second with a quadratic one and the third with the Puk kernel, a Pearson VII function-based universal kernel with parameter values $C = 4$, $\omega = 3.2$, $\sigma = 13$. Please consult [20] for further reference on these methods.

² Weka names for these classifiers in parenthesis.

4.3 Ensemble Classification Results

Table 3. Best results on individual classification of feature sets and classifiers on different datasets

Dataset	Classifier	Featureset	Accuracy
9GDB	SVM-Puk	TSSD	78.15
GTZAN	SVM-lin	SSD	72.60
ISMIRgenre	SVM-quad	TSSD	81.28
ISMIRrhythm	SVM-lin	RP	87.97
LatinMusic	SVM-Puk	TSSD	89.46
Africa/country	SMO-quad	SSD	86.29
Africa/ethnic group	SVM-lin	TSSD	81.10
Africa/function	1-NN	SSD	51.06
Africa/instrument	SVM-Puk	TSSD	69.90

To have a baseline for the cartesian ensemble, we trained all the classification schemes described in Section 4.2 on all the feature sets described in Section 2, i.e. one model for each cell in the cartesian set $D \times C$. Table 3 gives an extract of the accuracies achieved with these single models – due to space limitation, only the best combination of an algorithm and a feature set are given. It can be observed that there is no clear trend, neither for a classifier, nor a feature set. While SVMs clearly dominate the results, the choice of the kernel is not obvious, and results can vary by several percent points. Also the feature sets do not show a clear trend – in approximately half of the cases, TSSDs are the best set to use, while also SSD and RP features sometimes yield clearly better results. These results nourish the hypothesis that ensemble classifiers may provide means to release the user from the difficult choice of the proper feature set and classifier combination.

The accuracy results for the classifier ensembles are shown in Table 4, with the best single classifier as our assumed baseline to improve on. Note that achieving the baseline result would require to know the best combination of feature set and classifier in advance. On each of the datasets, we can observe higher classification accuracies with the ensembles than with the baseline. The improvements are three percent points on average. The highest gains are on the GTZAN dataset, with five percent points, while the improvements on the ISMIRrhythm dataset are of 1.14 percent point. However, the baseline on this dataset is already very high, at approx. 88%.

Out of the nine classification tasks, the QBWWV rule was five times the best, followed by WMV which is three times the best performing rule. AVG and BWV are both once the highest ranked combination rule. In the tasks where QBWWV is not the rule with the highest accuracy, the relative difference to the top rule is minimal – the largest margin is 0.7 percent points, or 0.86% relative difference.

4.4 Segmentation Ensemble Approach

A logical next step for ensemble classification is the use of individual features from different segments of an audio file as an input to classification. We conducted an experiment segmenting each audio file into 3 equal-sized segments, and extracting individual features from each of those segments. Note that for audio collections with 30 second ex-

cerpts we did not do this for TSSD and TRH features, as there would be no temporal variation within a segment, given the feature algorithm’s segment-window-length of 6 seconds (c.f. Sec. 2.1). In those cases we used TSSD and TRH features from the full song, as in the previous experiments. Also the symbolic features were used from full songs. Our hypothesis was that with more (detailed) information about the audio content, results would be improved in the ensemble setting. However, results of this segmentation approach were in general inferior compared to using features aggregated over entire songs, as seen from the bottom two lines of Table 4. As the performance decrease was independent of the combination rule applied, we included only the results of the two best combination rules (QBWWV and WMV) for space reasons.

Even though the results of this first experiment did not improve the ensemble approach, we will further pursue this strategy and refine it in multiple ways: First, we will extend the segmentation also to symbolic features. Then we will conduct research on different classifier model combination strategies. Instead of a combination of all classifier/feature set models into one ensemble, a two-tier approach is envisaged, where a decision is made by an ensemble of features from different segments first and then the decisions of multiple different feature sets and classifiers are combined on a second level. Further future work will be the experimentation with different degrees of segmentation of an audio file. Moreover, instead of using equally sized segments, a structural audio segmentation algorithm for segmentation into chorus, verse etc. could be used for semantic segmentation, aiming at an enhanced diversity of the features and the knowledge of the content.

5. CONCLUSIONS

In this paper, we presented a framework for automatic classification of music data. Our system builds ensembles of classifiers in two ways – first, several different algorithms (and parameter variations) are used, and secondly, a set of different features, describing different aspects of the same dataset. We have demonstrated the power of this approach on the classification task for six different datasets and achieved improvements on the classification accuracies in each single task. When comparing the results of the ensemble to the single feature sets, we could observe that there is no clear trend on which classification algorithm, and which feature set to use for a specific dataset. The advantage of the ensemble approach is that the user is released from this task. The ensemble approach delivers superior results through adding a reasonable amount of feature sets and classifiers. Even though we did not discover a combination rule that always outperforms all the others, relying on the QBWWV rule seems feasible.

Future work will include an even wider set of experiments on more datasets, also involving other modalities such as song lyrics. Another area is the above mentioned ensemble of different segments from the same song.

Table 4. Results of the ensemble classification on different datasets (Standard deviations are given in parentheses). The lower section of the table shows the results of the segmentation approach.

Rule	9GDB	GTZAN	ISMIR genre	ISMIR rhythm	Latin Music	Africa country	Africa ethnic group	Africa function	Africa instrument
Single best	78.15 (2.25)	72.60 (3.92)	81.28 (3.13)	87.97 (4.28)	89.46 (1.62)	86.29 (2.30)	81.10 (2.41)	51.06 (6.63)	69.90 (4.69)
MAJ	79.56 (4.78)	72.60 (3.31)	77.78 (2.15)	88.25 (5.08)	89.33 (1.55)	85.31 (4.04)	71.86 (3.41)	37.37 (7.36)	59.63 (5.79)
MAX	60.05 (6.67)	44.00 (6.60)	60.97 (6.71)	54.87 (8.95)	50.64 (2.06)	77.67 (9.16)	73.16 (6.40)	40.38 (7.10)	61.32 (5.88)
MED	74.30 (4.32)	55.90 (3.84)	72.02 (2.74)	77.79 (4.27)	73.64 (2.37)	83.84 (3.77)	70.71 (3.62)	39.49 (5.22)	60.34 (4.67)
AVG	81.66 (3.96)	68.40 (2.37)	79.70 (3.35)	86.82 (4.29)	86.85 (1.96)	87.66 (2.28)	78.21 (3.50)	53.73 (5.35)	70.60 (3.82)
SWV	81.31 (3.32)	77.10 (3.98)	78.33 (2.48)	88.97 (5.39)	92.00 (1.34)	86.97 (2.98)	75.47 (3.62)	46.83 (4.44)	67.09 (3.99)
RSWV	80.96 (3.26)	77.40 (4.22)	79.22 (2.38)	88.97 (4.94)	92.25 (1.16)	87.17 (2.77)	75.47 (3.62)	48.39 (5.63)	68.35 (4.22)
BWWV	81.54 (3.17)	77.40 (4.22)	82.03 (1.83)	89.11 (4.62)	92.25 (1.16)	88.34 (2.22)	79.37 (3.95)	52.61 (5.76)	72.71 (3.47)
QBWWV	80.96 (2.94)	77.50 (4.30)	84.02 (1.50)	88.97 (3.86)	92.71 (0.99)	89.03 (1.63)	82.68 (3.18)	54.84 (6.29)	72.86 (3.52)
WMV	80.84 (2.90)	76.10 (4.20)	84.02 (2.02)	87.97 (3.92)	92.59 (1.29)	88.93 (1.76)	82.97 (3.30)	51.28 (6.93)	73.00 (4.25)
QBWWV	81.31 (2.78)	76.80 (3.33)	76.95 (3.28)	88.25 (4.39)	91.66 (1.17)	88.44 (2.75)	78.35 (4.08)	50.95 (6.62)	71.03 (3.99)
WMV	80.49 (2.40)	74.50 (4.53)	81.48 (3.01)	87.68 (3.74)	91.56 (1.29)	88.05 (2.12)	80.23 (3.35)	44.83 (4.54)	72.29 (4.45)

6. REFERENCES

- [1] J.-J. Aucouturier and F. Pachet. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.
- [2] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kegl. Aggregate features and Adaboost for music classification. *Machine Learning*, 65:473–484, 2006.
- [3] P. Cano, E. Gómez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack. ISMIR 2004 audio description contest. Technical Report MTG-TR-2006-02, Pompeu Fabra University, 2006.
- [4] O. Cornelis, R. De Caluwe, G. De Tré, A. Hallez, M. Leman, T. Matthé, D. Moelants, and J. Gansemans. Digitisation of the ethnomusicological sound archive of the royal museum for central africa (belgium). *International Association of Sound and Audio-visual Archives Journal*, 26:35–43, 2005.
- [5] P.J. Ponce de León and J. M. Iñesta. A pattern recognition approach for music style identification using shallow statistical descriptors. *IEEE Trans. on Systems Man and Cybernetics C*, 37(2):248–257, 2007.
- [6] F. Moreno-Seco; J. M. Iñesta; P. Ponce de León; L. Micó. Comparison of classifier fusion methods for classification in pattern recognition tasks. *Lecture Notes in Computer Science*, 4109:705–713, 2006.
- [7] M. Grimaldi, P. Cunningham, and A. Kokaram. An evaluation of alternative feature selection strategies and ensemble techniques for classifying music. In *Proc. Workshop on Multimedia Discovery and Mining*, 2003.
- [8] J. Kittler, M. Hatef, R. P.W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
- [9] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.
- [10] T. Lidy, C. N. Silla Jr., O. Cornelis, F. Gouyon, A. Rauber, C. A. A. Kaestner, and A. L. Koerich. On the suitability of state-of-the-art music information retrieval methods for analyzing, categorizing, structuring and accessing non-western and ethnic music collections. *Signal Processing*, 90(4):1032 – 1048, 2010.
- [11] T. Lidy and A. Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proc. ISMIR*, London, UK, 2005.
- [12] T. Lidy, A. Rauber, A. Pertusa, and J.M. Iñesta. Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In *Proc. ISMIR*, Vienna, Austria, 2007.
- [13] C. McKay, R. Fiebrink, D. McEnnis, B. Li, and I. Fujinaga. Ace: A framework for optimizing music classification. In *Proc. ISMIR*, London, UK, 2005.
- [14] B. Pardo and W. P. Birmingham. Algorithms for chordal analysis. *Comput. Music J.*, 26:27–49, 2002.
- [15] C. Perez-Sancho, D. Rizo, and J. M. Iñesta. Genre classification using chords and stochastic language models. *Connection Science*, 21(2 & 3):145–159, May 2009.
- [16] A. Pertusa and J. M. Iñesta. Multiple fundamental frequency estimation using Gaussian smoothness. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, USA, 2008.
- [17] D. Rizo, P.J. Ponce de León, C. Pérez-Sancho, A. Pertusa, and J.M. Iñesta. A pattern recognition approach for melody track selection in midi files. In *Proc. ISMIR*, Victoria, Canada, 2006.
- [18] C. N. Silla Jr., A. L. Koerich, and C. A. A. Kaestner. The latin music database. In *Proc. ISMIR*, Philadelphia, USA, 2008.
- [19] G. Tzanetakis. *Manipulation, Analysis and Retrieval Systems for Audio Signals*. PhD thesis, Computer Science Department, Princeton University, 2002.
- [20] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.