Building Ensembles of Audio and Lyrics Features to Improve Musical Genre Classification

Rudolf Mayer and Andreas Rauber
Institute of Software Technology and Interactive Systems
Vienna University of Technology, Vienna, Austria
Email: mayer@ifs.tuwien.ac.at, rauber@ifs.tuwien.ac.at

***Abstract*—Digital audio has become an almost ubiquitously spread medium, and for many consumers, digital audio is the major distribution and storage form of music. Numerous on-line music stores account for a growing share of record sales. The widespread adoption of digital audio on home computers and especially mobile devices, and numerous on-line music stores show the size of this market. Handling the ever growing size of both private and commercial collections however becomes increasingly difficult. Computer algorithms that can understand and interpret characteristics of music, and organise and recommend them for and to their users can be of great assistance. Music is an inherently multi-modal type of data, and the lyrics associated with the music are as essential to the reception and the message of a song as is the audio. Album covers are carefully designed by artists to convey a message consistent with the music and image of a band. Music videos, fan sites and other sources of information add to that in a usually coherent manner. In this paper, we focus on exploring the lyrics domain of music, and how this information can be combined with the acoustic domain. We evaluate our approach by means of a common task in music information retrieval, musical genre classification. Advancing over previous work that showed improvements with simple feature fusion, were we successfully demonstrated simple approaches of combining different representations of music, we apply a more sophisticated machine learning technique, ensemble classification. 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.**

I. Introduction and Related Work

Music incorporates multiple types of content: the audio itself, song lyrics, album covers, social and cultural data, and music videos. All those modalities contribute to the perception of a song, and an artist in general. However, in the music information retrieval community, often a strong focus is put on the audio content only, disregarding many other opportunities and exploitable modalities. Even though music perception itself is based on sonic characteristics to a large extent, and acoustic content makes it possible to differentiate between acoustic styles. However, a great share of the overall perception of a song can be only explained when considering other modalities. Often, consumers relate to a song mainly due to the topic of its lyrics. Some categories of songs, such as 'love songs' or 'Christmas' songs, are almost exclusively defined by their textual domain. Many traditional 'Christmas' songs were interpreted by modern artists and are heavily influenced by their style; 'Punk Rock' variations are recorded as well as 'Hip-Hop' or 'Rap' versions. They all of share a common set of topics to be sung about.

These simple examples show that there is a whole level of semantics inherent in song lyrics, that can not be detected by audio based techniques alone. We thus assume that a song's text content can help in better understanding its perception. In this paper, we thus evaluate a new approach for combining descriptors extracted from the audio domain of music with descriptors derived from the textual content of lyrics. Our new 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.

Music information retrieval is a sub-area of information retrieval concerned with adequately accessing (digital) audio. Important research directions include, but are not limited to similarity retrieval, musical genre classification, or music analysis and knowledge representation. A comprehensive overviews of the research field is given in [11]. The prevalent technique of processing audio files in information retrieval is to analyse the audio signal. Popular feature sets computed thereof include MFCCs, Chroma, the MPEG-7 audio descriptors, and Rhythm Patterns.

Previous studies reported about a glass ceiling being reached using timbral audio features for music classification [1]. Thus, several research teams have been working on analysing textual information, predominantly in the form of song lyrics and an abstract vector representation of the term information contained in other text documents. A semantic and structural analysis of song lyrics is conducted in [6]. An evaluation of artist similarity via song lyrics is given in [5]. It is pointed out that acoustic similarity is superior to textual similarity yet a combination of both approaches might lead to better results.

In our evaluation in this paper, we employ a set of style features derived from the lyrics content. These features capture rhyme structures, part-of-speech of the employed words, and a set of statistical style features, such as diversification of the words used, sentence complexity, and punctuation. These feature sets are described in detail in [8], which reports on first results for genre classification with these feature subspaces; the results particularly showed that simple lyrics features may well be worthwhile. This approach has further been extended on two bigger test collections, and to combining and comparing the lyrics features with audio features in [7]. The approach employed in [7] is however simplistic, as it works on simple feature fusion (early fusion), i.e. concatenating all feature

TABLE I: Summary of combination rules.

Unweighted rules	
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 $P(L_k \| \mathbf{x}_i)$
Weighted rules	
SWV	Simple Weighted Vote
RSWV	Rescaled Simple Weighted Vote
BWWV	Best-Worst Weighted Vote
QBWWV	Quadratic Best-Worst Weighted Vote
WMV	Weighted Majority Vote

subspaces. The approach presented in this paper is rather based on **late fusion**, combining classifier outcomes rather than features. We create a two-dimensional ensemble system, combining different feature subspaces from different domains, and different classification algorithms. We thus call the approach a **Cartesian classifier**, which automatically predicts labels of a song, such as the genre, an important categorisation of music.

This paper is structured as follows. We describe our ensemble approach in Section II. We then evaluate and analyse its results on two corpora in Section IV. Finally, we conclude, and give a short outlook on future research in Section V.

II. Cartesian Ensemble

The approach is called *Cartesian ensemble* as the set of models used as base classifiers is the Cartesian product of D feature subspaces/sets by C classification schemes. A model is build 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.

Model diversity is a key design factor for building effective classifier ensembles [3]. This has been empirically shown to improve the accuracy of an ensemble over its base models when they are numerous 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.

The 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 [3]. 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

$$\kappa_{ij} = \frac{\sum_k m_{kk} - \sum_r (\sum_s m_{r,s})(\sum_s m_{s,r})}{1 - \sum_r (\sum_s m_{r,s})(\sum_s m_{s,r})} \tag{1}$$

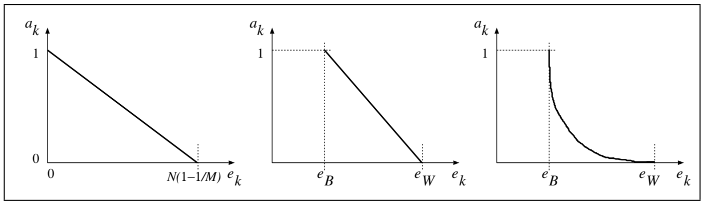

Fig. 1: Model weight computation: RSWV, BWWV, QBWWV, giving the authority a_k as a function of the estimated number of errors e_k made by model h_k on a validation set. N is the number of instances in the set, M is the number of class labels.

The second one is the pair average error, computed by

$$e_{ij} = 1 - \frac{\alpha_i + \alpha_j}{2} \tag{2}$$

where α_i and α_j are the estimated accuracy of the two models, computed as described below. 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.

The **combination rules** implemented in the system are both weighted and unweighted majority voting rules; a summary is presented in Table I, 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 [2].

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 [10] have been used in this work. SVW computes weights as described. Weight functions for rules RSWV, BWWV and QBWWV are shown in Figure 1. 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 [3], where model weights are set proportionally to $\log(\alpha_i/(1-\alpha_i))$.

The classification results presented in Section IV 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 of the system.

III. Feature Spaces

In this section, we present the audio and lyrics feature subspaces that we subsequently employed in our evaluation.

A. Audio Feature Subspaces

The audio 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) from features of multiple segments. These audio feature sets were chosen as they have shown to yield the best results on the databases used in our experimental evaluation [9]

1) Rhythm Patterns: Rhythm Patterns (RP) are a feature set for handling audio data based on analysis of the spectral audio data and psycho-acoustic transformations [12]. In a pre-processing stage, multiple channels are averaged to one, and the audio is split into segments of six seconds, possibly leaving out lead-in and fade-out segments, and further skipping other segments, e.g. out of the remaining segments every third may be processed.

The feature extraction process for a Rhythm Pattern is composed of two stages. For each segment, the spectrogram of the audio is computed using a Fast Fourier Transform. The window size is set to 1024 samples, applying a Hanning window of 50% overlap. The Bark scale groups frequencies to critical bands according to perceptive pitch regions, is applied to the spectrogram, aggregating it to 24 frequency bands. Then, the spectrogram is transformed into the decibel scale, and further psycho-acoustic transformations are applied. Subsequently, the values are transformed into the unit Sone, on which a doubling on the scale sounds to the human ear like a doubling of the loudness. This results in a psycho-acoustically modified representation reflecting human loudness sensation.

In the second stage, a discrete Fourier transform is applied to the Sonogram, resulting in a (time-invariant) spectrum of loudness amplitude modulation per modulation frequency for each individual critical band. After additional weighting and smoothing steps, a Rhythm Pattern exhibits magnitudes of modulation for 60 modulation frequencies on 24 bands, and has thus 1440 dimensions. Finally, the feature vectors of a songs segments are simply averaged by computing the median.

2) Statistical Spectrum Descriptors: Computing Statistical Spectrum Descriptors (SSD) features [4] relies on the first part of the algorithm for computing RP features, specifically on the Bark-scale representation of the frequency spectrum. From this representation of perceived loudness, seven statistical measures are computed for each of the 24 critical band, to describe fluctuations within them. The statistical measures comprise mean, median, variance, skewness, kurtosis, min- and max-value. A Statistical Spectrum Descriptor is extracted for each segment, and the SSD feature vector of a song is then calculated as the median of its segments. In contrast to the Rhythm Patterns feature set, the dimensionality of the feature space is much lower – SSDs have 168 instead of 1440 dimensions, still at matching performance in terms of genre classification accuracies [4].

3) Rhythm Histograms: The Rhythm Histogram [4] features are a descriptor for the rhythmic characteristics in a song. Contrary to the RP and the SSD, information is not stored per critical band. Rather, the magnitudes of each modulation frequency bin (at the end of the second stage of the RP calculation process) of all 24 critical bands are summed up, to form a histogram of 'rhythmic energy' for each of the 60 modulation frequencies. The RH feature vector for a piece of music is calculated as the median of the histograms of each segment. The dimensionality of RH of 60 features is much lower than with the other sets.

TABLE II: Rhyme features for lyrics analysis

Feature Name	Description
Rhymes-AA	A sequence of two (or more) rhyming lines ('Couplet')
Rhymes-AABB	A block of two rhyming sequences of two lines ('Clerihew')
Rhymes-ABAB	A block of alternating rhymes
Rhymes-ABBA	A sequence of rhymes with a nested sequence ('Enclosing rhyme')
RhymePercent	The percentage of blocks that rhyme
UniqueRhymeWords	The fraction of unique terms used to build the rhymes

B. Lyrics Feature Subspace

In this section we describe the four types of lyrics features we use in the experiments throughout the remainder of the paper: a) bag-of-words features computed from tokens or terms occurring in documents, b) rhyme features taking into account the rhyming structure of lyrics, c) features considering the distribution of certain parts-of-speech, and d) text statistics features covering average numbers of words and particular characters.

The following feature subspaces are all based on song lyrics, and analyse rhyme and style of them. The feature sets were introduce in [7], [8], and are analysing (1) the rhyming structure of lyrics, (2) the distribution of certain parts-of-speech, and (3) text statistics features covering average numbers of words and particular characters in the different lyrics documents.

1) Rhyme Features: Rhyme denotes the consonance or similar sound of two or more syllables or whole words. This linguistic style is most commonly used in poetry and songs. The reason for considering rhyme as feature is that different genres of music should exhibit different styles of lyrics. We assume the rhyming characteristics of a song to be given by the degree and form of the rhymes used. 'Hip-Hop' or 'Rap' music, for instance, makes heavy use of rhymes, which (along with a dominant bass) leads to their characteristic sound. To identify such patterns we extract several descriptors from the song lyrics to represent different types of rhymes.

Our approach is based on a phoneme transcription of the lyrics. The words 'sky' and 'lie', for instance, both end with the same phoneme /ai/. The transcription is language dependent; however, our test collection is predominantly composed of English tracks. After transcribing the lyrics to a phoneme representation, we distinguish two elements of subsequent lines in a song text: *AA* and *AB*. The former represents two rhyming lines, while the latter denotes non-rhyming. Based on these, we extract the rhyme patterns described in Table II.

TABLE III: Overview of text statistic features

Feature Name	Description
exclamation_mark, colon, single_quote, comma, question_mark, dot, hyphen, semicolon	simple counts of occurrences
d0 - d9	occurrences of digits
WordsPerLine	words / number of lines
UniqueWordsPerLine	unique words / number of lines
UniqueWordsRatio	unique words / words
CharsPerWord	number of chars / number of words
WordsPerMinute	the number of words / length of the song

Subsequently, we compute the percentage of rhyming blocks, and define the unique rhyme words as the fraction of unique terms used to build rhymes, describing whether rhymes are frequently formed using the same word pairs. Experimental results indicate that more elaborate patterns based on assonance, semirhymes, or alliterations may well be worth studying.

2) Part-of-Speech Features: Part-of-speech (POS) tagging is a lexical categorisation or grammatical tagging of words. Different POS categories are e.g. nouns, verbs, articles or adjectives. We presume that different genres will differ also in the category of words they are using; thus, we extract several POS descriptors from the lyrics. We count the numbers of: *nouns*, *verbs*, *pronouns*, *relational pronouns* (such as ‘that’ or ‘which’), *prepositions*, *adverbs*, *articles*, *modals*, and *adjectives*. To account for different document lengths, all of these values are normalised by the number of words of the respective lyrics document.

3) Text Statistic Features: Text documents can also be described by simple statistical measures based on word or character frequencies. Measures such as the average length of words or the ratio of unique words in the vocabulary might give an indication of the complexity of the texts, and are expected to vary over different genres. Further, the usage of punctuation marks such as exclamation or question marks may be specific for some genres, and some genres might make increased use of apostrophes when omitting the correct spelling of word endings. The list of extracted features is given in Table III.

All features that simply count character occurrences are normalised by the number of words of the song text to accommodate for different lyrics lengths. ‘WordsPerLine’ and ‘UniqueWordsPerLine’ describe the words per line and the unique number of words per line. The ‘UniqueWordsRatio’ is the ratio of the number of unique words and the total number of words. ‘CharsPerWord’ denotes the simple average number of characters per word. ‘WordsPerMinute’ is computed analogously to the well-known beats-per-minute (BPM) value.

IV. Evaluation

In this section, we first present the feature subspaces and datasets employed in our evaluation, followed by a detailed analysis of the classification accurarcies achieved.

TABLE IV: Composition of the small test collection)

Genre	Artists	Albums	Songs
Country	6	13	60
Folk	5	7	60
Grunge	8	14	60
Hip-Hop	15	18	60
Metal	22	37	60
Pop	24	37	60
Punk Rock	32	38	60
R&B	14	19	60
Reggae	12	24	60
Slow Rock	21	35	60
Total	159	241	600

TABLE V: Composition of the large test collection

Genre	Artists	Albums	Songs
Country	9	23	227
Folk	11	16	179
Grunge	9	17	181
Hip-Hop	21	34	381
Metal	25	46	371
Pop	26	53	371
Punk Rock	30	68	374
R&B	18	31	373
Reggae	16	36	181
Slow Rock	23	47	372
Total	188	370	3010

A. Datasets

The data sets used in this evaluation were previously used in [7]. Both datasets were randomly drawn from a personal collection that was manually pre-categorised into ten popular genres. The smaller dataset contains 600 songs, 60 from each of the genres listed in Table IV. When composing these collections, it was aimed at having a high number of different artists, represented by songs from different albums, in order to prevent biased results by too many songs from the same artist. This collection thus comprises songs from 159 different artists, stemming from 241 different albums. The larger dataset, given in Table V, consists of 3010 songs, and can be seen as prototypical for a private collection. It is not equaly distributed, containing between 180 and 380 songs per genre.

For these collections, the lyrics were then automatically fetched from the Internet using scripts extracting the lyrics content from popular lyrics portals. To obtain all lyrics, scripts were subsequently ran on different portals, until all lyrics were available, regardless of the quality of the texts with respect to content or structure. While this approach implies that some lyrics will not be properly fetched, experiments in [7] have shown that this automatic fetching provides sufficiently good quality.

B. Classification Schemes and Parameters

We used a 10-fold outer and 3-fold inner cross-validation. We aimed at selecting classification schemes from different machine learning paradigms, and thus chose Naïve Bayes, k-Nearest Neighbour (varying k of 1, 5, and 10), RIPPER rule learner, C4.5 decision tree, Random Forest, and Support Vector Machines with different kernels (linear and quadratic kernel).

TABLE VI: Results of the single classification on the small dataset (600 songs); standard deviations are given in parentheses

Feature set	NB	1-NN	5-NN	10-NN	SVMLin	SVMPol	RF
Rhyme	15.67 (3.06)	12.83 (4.58)	13.33 (3.04)	14.17 (4.25)	13.17 (5.00)	11.17 (4.01)	15.67 (2.96)
POS	19.67 (4.43)	14.50 (6.14)	18.00 (3.50)	18.50 (4.04)	20.33 (4.07)	20.17 (4.34)	17.83 (3.85)
Stat	21.50 (2.66)	20.50 (4.72)	22.00 (6.42)	24.33 (4.73)	30.00 (4.01)	28.17 (3.09)	25.50 (4.72)
RH	32.00 (3.50)	30.00 (5.03)	31.17 (4.72)	30.67 (5.45)	37.17 (3.52)	37.33 (4.32)	30.17 (3.37)
RP	38.67 (4.89)	33.17 (6.16)	32.67 (5.68)	29.83 (4.74)	49.17 (6.72)	46.33 (5.60)	32.67 (5.68)
SSD (audio baseline)	45.50 (5.16)	52.17 (7.46)	50.17 (6.91)	51.50 (6.91)	**59.00** (4.66)	58.67 (6.23)	48.67 (9.09)
SSD/Stat (comb. baseline)	47.17 (6.67)	55.33 (6.56)	53.00 (6.37)	52.33 (4.53)	**65.83** (3.17)	61.33 (6.02)	45.00 (6.43)
SSD/Stat/Rhyme	47.33 (6.49)	54.17 (5.94)	52.67 (5.68)	54.00 (5.04)	63.50 (5.00)	62.17 (5.88)	48.67 (8.08)
SSD/Stat/POS	46.67 (6.24)	51.50 (5.12)	50.33 (5.14)	52.67 (4.79)	64.00 (3.16)	60.50 (5.56)	50.67 (6.68)
SSD/Stat/POS/Rhyme	47.17 (6.14)	52.17 (5.33)	50.67 (5.40)	53.50 (3.46)	64.00 (4.79)	60.33 (7.02)	48.00 (7.36)

C. *Ensemble Classification Results*

In Table VI, the classification accuracies of the single classifiers on single feature sets on the **small datase**t are given, as well as our previous approach of concatenating the features (early fusion). It can be noted that the SSD features are the best performing single feature set, and the linear SVM the best classifier. With the previous approach, when combining SSD and lyrics style statistics features, we could significantly improve the result, by almost 7% points.

Table VII in turn shows the results of a number of selected combination rules. These rules have been selected as they showed to be the most performing rules over a series of experiments. We can see from that results that we were still able to improve on the SSD audio baseline by 4.5% point. While we failed to beat the best linear SVM classifier on the early fusion approach of SSD and text statistics features, we obtained a better result than the concatenation approach on the SVM with a quadratic kernel. It has to be noted, however, that finding this best early fusion result requires testing a number of different feature combinations, as well as testing a lot of different algorithms. This is a time-consuming and labour-intensive task. More importantly, we can select the best model only **a posterio**.

We further tested whether the improvements achieved with the ensemble approach are not only due to the ensemble of classifier methods itself, but are really due to the *Cartesian ensemble* that combines both different machine learning methods, as well as different feature subspaces. Thus, as a baseline to the ensembles of multiple features, an ensemble of SSD features only is given in Table VII; it can be seen that the gain in accuracy is indeed not due to only to combining different classificiation schemes, but due to the Cartesian ensemble of both feature subspaces and algorithms.

Classification accuracies for the single classifiers on the **large dataset** are given in Table VIII. We can see that SSD was once again clearly the best audio feature set. However, another very interesting observation is that on this very dataset, the SVM classifier with quadratic kernel performed significantly better than the SVM with a linear kernel. We could significantly improve the results of the SVM with the linear kernel on SSD features by concatenating them with the lyrics features, this time by combining them with both of text statistic and part-of-speech features. In turn, the improvements for the SVM with quadratic kernel are only minor.

TABLE VII: Results of the ensemble classification on the small dataset (600 songs); standard deviations are given in parentheses

Rule	Cartesian Ensemble	SSD Ensemble
SWV	61.50 (3.19)	59.00 (7.67)
RSWV	63.67 (3.31)	59.00 (7.67)
BWWV	63.67 (3.31)	59.33 (7.29)
QBWWV	63.17 (3.55)	60.17 (6.01)
WMV	59.67 (4.57)	58.50 (4.12)

However, when using the result fusion approach shown in Table IX, one can see that most of the results of the vote combination methods are between 2 and 4% points better than with the best feature concatenation approach, the latter ones being statistically significantly better. Again, we compare these results against a baseline of an ensemble of the same classification schemes on SSD features only, and can make the same observations as above.

V. Conclusions

In this paper, we presented an approach for multi-modal classification of music. Contrary to earlier work which focused on fusion of the feature subspaces, so called early fusion, this approach is built on classifier ensemble techniques, i.e. fusion of the labels assigned by each single classifier. We evaluated the method by musical genre classification, a popular task in music information retrieval, on two different, previously used datasets. We have shown that the new approach is capable of achieving better results than the best single audio feature set alone. Further, for the large dataset, we achieved also better results than with the best concatenation (early fusion) approach.

We could further observe that the best performing classifier, and best concatenation of feature sets can vary a lot for different datasets – there seems to be no rule that can determine the best classifier model and concatenation approach *a priori*, and we are only able to select the best approach *a posterio*. However, when using the ensemble approach, we can release the user from this choice. We have determined over our series of experiments that the QBWWV combination method is the most promising one.

TABLE VIII: Results of the single classification on the large dataset (3.010 songs); standard deviations are given in parentheses

Feature set	NB	1-NN	5-NN	10-NN	SVMLin	SVMPol	RF
Rhyme	16.62(2.00)	16.92(2.91)	16.58(1.57)	18.11(1.99)	16.08(1.86)	15.65(1.38)	19.91(2.89)
POS	23.53(1.72)	20.94(2.30)	21.64(1.79)	22.60(1.93)	23.66(2.06)	24.53(1.71)	24.59(2.86)
TextStat	17.91(1.60)	23.40(1.56)	25.09(2.48)	25.86(3.19)	28.38(3.39)	25.49(2.93)	34.30(2.92)
SSD (audio baseline)	42.11(2.39)	62.58(2.27)	62.21(3.06)	62.78(2.87)	66.37(2.21)	**69.43**(3.20)	55.07(2.53)
SSD/Stat (comb. baseline)	43.87 (2.51)	63.88 (2.63)	63.01 (2.73)	62.12 (3.60)	68.60 (2.51)	**69.99** (2.42)	57.06 (3.41)
SSD/Stat/POS	44.50 (2.26)	62.51 (2.99)	63.18 (4.07)	62.48 (4.06)	68.86 (2.51)	69.46 (2.94)	55.90 (3.58)
SSD/Stat/POS/Rhyme	44.80 (2.05)	62.74 (2.56)	62.41 (3.52)	61.78 (3.09)	67.83 (2.33)	69.69 (3.07)	57.63 (2.22)

TABLE IX: Results of the ensemble classification on the large dataset (3.010 songs); standard deviations are given in parentheses

Rule	Cartesian Ensemble	SSD Ensemble
SWV	70.22 (3.90)	69.09 (2.29)
RSWV	71.55 (3.84)	69.33 (2.25)
BWWV	72.12 (3.34)	69.69 (2.13)
QBWWV	73.35 (2.30)	70.62 (2.14)
WMV	72.32 (2.25)	70.75 (2.59)

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. Kittler, M. Hatef, R. P. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[3] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[4] T. Lidy and A. Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, UK, September 11-15 2005.

[5] B. Logan, A. Kositsky, and P. Moreno. Semantic analysis of song lyrics. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 827–830, Taipei, Taiwan, June 27-30 2004.

[6] J. P. G. Mahedero, Á. Martínez, P. Cano, M. Koppenberger, and F. Gouyon. Natural language processing of lyrics. In *Proceedings of the ACM 13th International Conference on Multimedia*, pages 475–478, New York, NY, USA, 2005.

[7] R. Mayer, R. Neumayer, and A. Rauber. Combination of audio and lyrics features for genre classification in digital audio collections. In *Proceedings of the ACM Multimedia 2008*, pages 159–168. ACM New York, NY, USA, October 27-31 2008.

[8] R. Mayer, R. Neumayer, and A. Rauber. Rhyme and style features for musical genre classification by song lyrics. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'08)*, Philadelphia, PA, USA, September 14-18 2008.

[9] R. Mayer and A. Rauber. Multimodal aspects of music retrieval: Audio, song lyrics - and beyond? In *Advances in Music Information Retrieval*, pages 333–363. 2010.

[10] F. Moreno-Seco, J. M. I. nesta, P. J. P. de León, and L. Mico. Comparison of classifier fusion methods for classification in pattern recognition tasks. In *Proceedings of the International Workshop on Structural and Syntactic Pattern Recognition*, Hong Kong, China, August 17-19, 2006 2006.

[11] N. Orio. Music retrieval: A tutorial and review. *Foundations and Trends in Information Retrieval*, 1(1):1–90, September 2006.

[12] A. Rauber, E. Pampalk, and D. Merkl. Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR'02)*, pages 71–80, Paris, France, October 13-17 2002.