VOCAL SEGMENT CLASSIFICATION IN POPULAR MUSIC

Ling Feng, Andreas Brinch Nielsen, Lars Kai Hansen
Technical University of Denmark
Department of Informatics and Mathematical Modelling
{lf, abn, lkh}@imm.dtu.dk

ABSTRACT

This paper explores the vocal and non-vocal music classification problem within popular songs. A newly built labeled database covering 147 popular songs is announced. It is designed for classifying signals from $1sec$ time windows. Features are selected for this particular task, in order to capture both the temporal correlations and the dependencies among the feature dimensions. We systematically study the performance of a set of classifiers, including linear regression, generalized linear model, Gaussian mixture model, reduced kernel orthonormalized partial least squares and K-means on cross-validated training and test setup. The database is divided in two different ways: with/without artist overlap between training and test sets, so as to study the so called 'artist effect'. The performance and results are analyzed in depth: from error rates to sample-to-sample error correlation. A voting scheme is proposed to enhance the performance under certain conditions.

1 INTRODUCTION

The wide availability of digital music has increased the interest in music information retrieval, and in particular in features of music and of music meta-data, that could be used for better indexing and search. High-level musical features aimed at better indexing comprise, e.g., music instrument detection and separation [13], automatic transcription of music [8], melody detection [2], musical genre classification [10], sound source separation [18], singer recognition [16], and vocal detection [4]. While the latter obviously is of interest for music indexing, it has shown to be a surprisingly hard problem. In this paper we will pursue two objectives in relation to vocal/non-vocal music classification. We will investigate a multi-classifier system, and we will publish a new labeled database that can hopefully stimulate further research in the area.

While almost all musical genres are represented in digital forms, naturally popular music is most widely distributed, and in this paper we focus solely on popular music. It is not clear that the classification problem can be generalized between genres, but this is a problem we will investigate in later work.

Singing voice segmentation research started less than a decade ago. Berenzweig and Ellis attempted to locate the vocal line from music using a multi-layer perceptron speech model, trained to discriminate 54 phone classes, as the first step for lyric recognition [4]. However, even though singing and speech share certain similarities, the singing process involves the rapid acoustic variation, which makes it statistically different from normal speech. Such differences may lie in the phonetic and timing modification to follow the tune of the background music, and the usage of words or phrases in lyrics and their sequences. Their work was inspired by [15] and [19], where the task was to distinguish speech and music signals within the "music-speech" corpus: 240 $15s$ extracts collected 'at random' from the radio. A set of features have been designed specifically for speech/music discrimination, and they are capable of measuring the conceptually distinct properties of both classes.

Lyrics recognition can be one of a variety of uses for vocal segmentation. By matching the word transcriptions, it is applicable to search for different versions of the same song. Moreover, accurate singing detection could be potential for online lyrics display by automatically aligning the singing pieces with the known lyrics available on the Internet. Singer recognition of music recordings has later received more attention, and has become one of the popular research topics within MIR. In early work of singer recognition, techniques were borrowed from speaker recognition. A Gaussian Mixture Model (GMM) was applied based on Mel-frequency Cepstral Coefficients (MFCC) to detect singer identity [20]. As briefly introduced, singing voices are different from the conventional speech in terms of time-frequency features; and vocal and non-vocal features have differences w.r.t. spectral distribution. Hence the performance of a singer recognition system has been investigated using the unsegmented music piece, the vocal segments, and the non-vocal ones in [5]. 15% improvement has been achieved by only using the vocal segments, compared to the baseline of the system trained on the unsegmented music signals; and the performance became 23% worse when only non-vocal segments were used. It demonstrated that the vocal segments are the primary source for recognizing singers. Later, work on automatic singer recognition took vocal segmentation as the first step to enhance

the system performance, e.g. [16].

Loosely speaking, vocal segmentation has two forms. One is to deal with a continuous music stream, and the locations of the singing voice have to be detected as well as classified, one example is [4]. The second one is to pre-segment the signals into windows, and the task is only to classify these segments into two classes. Our work follows the second line, in order to build models based on our in-house Pop music database. A detailed description of the database will be presented in section 4. The voice is only segmented in the time domain, instead of the frequency domain, meaning the resulting vocal segments will still be a mixture of singing voices and instrumental background. Here we will cast the vocal segments detection in its simplest form, i.e. as a binary classification problem: one class represents signals with singing voices (with or without background music); the other purely instrumental segments, which we call accompaniment.

In this paper we study this problem from a different angle. Several classifiers are invoked, and individual performance (errors and error rates) is inspected. To enhance performance, we study the possibility of sample-to-sample cross-classifier voting, where the outputs of several classifiers are merged to give a single prediction. The paper is organized as follows. Section 2 explains the selection of features. Classification frameworks are covered by section 3. With the purpose of announcing the Pop music database, we introduce the database design in section 4. In section 5, the experiments are described in depth, and the performance characteristics are presented. At last, section 6 concludes the current work.

2 ACOUSTIC FEATURES

2.1 Mel-Frequency Cepstral Coefficients

MFCCs are well-known in the speech and speaker recognition society. They are designed as perceptually weighted cepstral coefficients, since the mel-frequency warping emulates human sound perception. MFCCs share two aspects with the human auditory system: A logarithmic dependence on signal power and a simple bandwidth-to-center frequency scaling so that the frequency resolution is better at lower frequencies. MFCCs have recently shown their applicability in music signal processing realm, e.g. [1] for music genre classification, [16] and [5] for singer recognition, and [14] for vocal segmentation, and many more exist.

Features are extracted from short time scales, e.g. $20ms$, due to the stationarity of music signals. To process windows at longer time scales, temporal feature integration is needed. Features at different time scales may contain different information. A small frame size may result in a noisy estimation; and a long frame size may cover multiple sounds (phonemes) and fail to capture appropriate information.

2.2 Multivariate AR

During the course of searching for appropriate features, researchers have realized that system performance can be improved by combining short-time frame-level features into clip-level features. Feature integration is one of the methods to form a long-time feature, in order to capture the discriminative information and characterize how frame-level features change over longer time periods for a certain task. Often the mean and variance of several short-time features are extracted as the clip-level features [17], using multivariate Gaussian model or a mixture of them. However, both the mean-variance and mean-covariance model fail to capture the temporal correlations. A frequency band approach has been proposed in [9], and the energy of the features was summarized into 4 frequency bands. Even though this method can represent temporal development, it does not model the feature correlations.

The multivariate autoregressive model (MAR) was recently introduced to music genre classification [11], and a detailed comparison of different temporal feature integration methods was reported. MAR being able to capture both the temporal correlations and the dependencies among the feature dimensions, has shown its superiority for representing music. We adapt this model in the feature extraction phase on top of short-time MFCCs. Here, a brief description of MAR will be given, for detail, see [11].

Assume the short-time MFCC at time t is denoted as $\mathbf{x}_t$, which is extracted from a short period of stationary signals. The MAR can be stated as,

$$\mathbf{x}_t = \sum_{p=1}^{P} \mathbf{A}_p \mathbf{x}_{t-p} + \mathbf{u}_t, \tag{1}$$

where $\mathbf{u}_t$ is the Gaussian noise $\mathcal{N}(\mathbf{v}, \mathbf{\Sigma})$, assumed i.i.d. $\mathbf{A}_p$ is the coefficients matrix for order p; and if it is defined as a diagonal matrix, dependencies among dimensions will not be considered. P indicates the order of the multivariate auto regressive model, meaning that $\mathbf{x}_t$ is predicted from the previous P short-time features. It is worth to mention that the mean of MFFCs $\mathbf{m}$ is related to the mean of the noise $\mathbf{v}$ in the following way (note: $\mathbf{I}$ is an identity matrix),

$$\mathbf{m} = (\mathbf{I} - \sum_{p=1}^{P} \mathbf{A}_p)^{-1} \mathbf{v}. \tag{2}$$

3 CLASSIFICATION FRAMEWORKS

We have examined a number of classifiers: linear regression model (LR), generalized linear model (GLM), Gaussian mixture model (GMM), reduced kernel orthonormalized partial least squares (rKOPLS) and K-means.

As the problem is a binary task, only a single dimension is needed for linear regression, and the labels are coded as

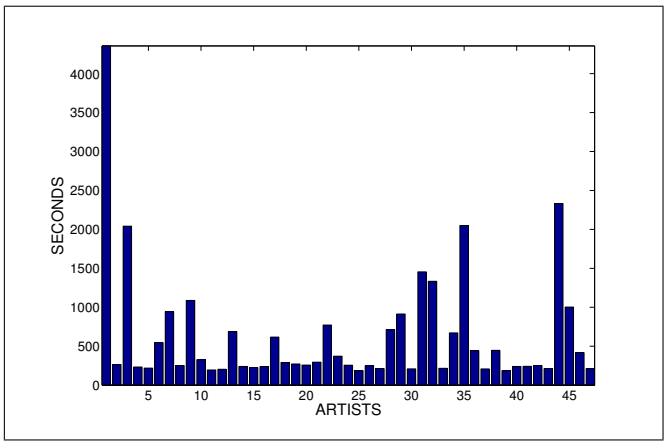

Figure 1. Distribution of Pop music among artists

± 1. The model is $l_n = \mathbf{w^T y}$. A 1 is added to the feature vector to model offset. Least squares is used as the cost function for training, and the minimum solution is the pseudo inverse. The prediction is made based on the *sign* of the output: we tag the sample as a vocal segment if the output is greater than zero; and as a non-vocal segment otherwise.

Generalized linear model relates a linear function of the inputs, through a link function to the mean of an exponential family function, $\mu = g(\mathbf{w}^T \mathbf{x}^n)$, where $\mathbf{w}$ is a weight vector of the model and $\mathbf{x}^n$ is the n'th feature vector. In our case we use the *softmax* link function, $\mu_i = \frac{e^{\mathbf{w}_i^T \mathbf{x}_i^n}}{\sum_j e^{\mathbf{w}_j^T \mathbf{x}_j^n}}$. $\mathbf{w}$ is found using iterative reweighted least squares [12].

GMM as one of the Bayesian classifiers, assumes a known probabilistic density distribution for each class. Hence we model data from each class as a group of Gaussian clusters. The parameters are estimated from training sets via the standard Expectation-Maximization (EM) algorithm. For simplicity, we assume the covariance matrices to be diagonal. Note that although features are independent within each mixture component due to the diagonal covariance matrix, the mixture model does not factorize over features. The diagonal covariance constraint posits the axes of the resulting Gaussian clusters parallel to the axes of the feature space. Observations are assigned to the class having the maximum *posterior* probability.

Any classification problem is solvable by a linear classifier if the data is projected into a high enough dimensional space (possibly infinite). To work in an infinite dimensional space is impossible, and kernel methods solve the problem by using inner products, which can be computed in the original space. Relevant features are found using orthonormalized partial least squares in kernel space. Then a linear classifier is trained and used for prediction. In the reduced form, rKOPLS [3] is able to handle large data sets, by only using a selection of the input samples to compute the relevant features, however all dimensions are used for the linear classifier, so this is not equal to a reduction of the training set.

K-means uses K clusters to model the distribution of each class. The optimization is done by assigning data points to the closest cluster centroid, and then updating the cluster centroid as the mean of the assigned data points. This is done iteratively, and minimizes the overall distances to cluster centroids. Optimization is very dependent on the initial centroids, and training should be repeated a number of times. Prediction is done by assigning a data point to the class of the closest cluster centroid.

4 DATABASE

The database used in the experiments is our recently built in-house database for vocal and non-vocal segments classification purpose. Due to the complexity of music signals and the dramatic variations of music, in the preliminary stage of the research, we focus only on one music genre: the popular music. Even within one music genre, Berenzweig et al. have pointed out the 'Album Effect'. That is songs from one album tend to have similarities w.r.t. audio production techniques, stylistic themes and instrumentation, etc. [5].

This database contains 147 Pop mp3s: with 141 singing songs and 6 pure accompaniment songs. The 6 accompaniment songs are not the accompaniment of any of the other singing songs. The music in total lasts $8h\ 40min\ 2sec$. All songs are sampled at 44.1 kHz. Two channels are averaged, and segmentation is based on the mean. Songs are manually segmented into $1sec$ segments without overlap, and are annotated second-by-second. The labeling is based on the following strategy: if the major part of this $1sec$ music piece is singing voice, it is tagged as vocal segment; otherwise non-vocal segment. We believe that the long-term acoustic features are more capable of differentiating singing voice, and $1sec$ seems to be a reasonable choice based on [14]. Furthermore labeling signals at this time scale is not only more accurate, but also less expensive.

Usually the average partition of vocal/non-vocal in Pop music is about 70%/30%. Around 28% of the 141 singing songs is non-vocal music in the collection of this database. Forty-seven artists/groups are covered. By artists in Pop music we mean the performers (singers) or bands instead of composers. The distribution of songs among artists is not even, and Figure 1 gives the total number of windows (seconds) each artist contributes.

5 EXPERIMENTS AND RESULTS

We have used a set of features extracted from the music database. First, we extracted the first 6 original MFCCs over a $20ms$ frame hopped every $10ms$. The 0^{th} MFCC representing the log-energy was computed as well. The means were calculated on signals covering $1sec$ in time. MAR were afterwards computed on top of the first 6 MFCCs with $P = 3$, and we ended up with a 6-by-18 $\mathbf{A}_p$ matrix, a 1-by-6

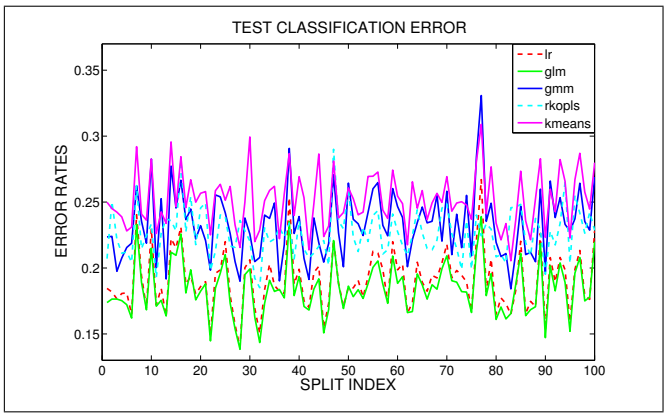

Figure 2. Classification error rates as a function of splits of five classifiers on test sets.

vector $\mathbf{v}$ and a 6-by-6 covariance matrix Σ. Since Σ is symmetric, repetitions were discarded. $\mathbf{A}_p$, $\mathbf{v}$ and Σ all together form a 135-dimensional feature set. The choice for 6 MFCC is on one hand empirical, and on the other hand to reduce the computational complexity. All in all, for $1sec$ music signal we concatenated 135-d MAR, the means of both 0^{th} and 6 original MFCCs to form a 142-d feature vector.

5.1 Data Dependency and Song Variation

We used one type of cross-validation, namely holdout validation, to evaluate the performance of the classification frameworks. To represent the breadth of available signals in the database, we kept 117 songs with the 6 accompaniment songs to train the models, and the remaining 30 to test. We randomly split the database 100 times and evaluated each classifier based on the aggregate average. In this way we eliminated the data set dependencies, due to the possible similarities between certain songs. The random splitting regarded a song as one unit, therefore there was no overlap song-wise in the training and test set. On the other hand artist overlap did exist. The models were trained and test set errors were calculated for each split. The GLM model from the Netlab toolbox was used with *softmax* activation function on outputs, and the model was trained using iterative reweighted least squares. As to GMM, we used the generalizable gaussian mixture model introduced in [7], where the mean and variance of GMM are updated with separate subsets of data. Music components have earlier been considered as 'noise' and modeled by a simpler model [16], thus we employed a more flexible model for the vocal than non-vocal parts: 8 mixtures for the vocal model, and 4 for the non-vocal model. For rKOPLS, we randomly chose 1000 windows from the training set to calculate the feature projections. The average error rates of the five classification algorithms are summarized in the left column of Table 1.

A bit surprisingly the performance is significantly better for the linear models. We show the performance of the chosen classifiers as a function of splits in Figure 2. Each curve represents one classifier, and the trial-by-trial difference is quite striking. It proved our assumption that the classification performance depends heavily on the data sets, and the misclassification varies between 13.8% and 23.9% for the best model (GLM). We envision that there is significant variation in the data set, and the characteristics of some songs may be distinguishing to the others. To test the hypothesis, we studied the performance on individual songs. Figure 3 presents the average classification errors of each song predicted by the best model: GLM, and the inter-song variation is obviously revealed: for some songs it is easy to distinguish the voice and music segments; and some songs are hard to classify.

	Error Rates	
Artists	overlap	no overlap
LR	19.03±2.25%	20.52±3.5%
GLM	18.46±2.02%	19.82±2.81%
GMM	23.27±2.54%	24.50±2.99%
rKOPLS	22.62±1.85%	24.60±3.14%
K-means	25.13±2.11%	NA

Table 1. The average error rates (mean ± standard deviation) of 5 classifiers on test sets.

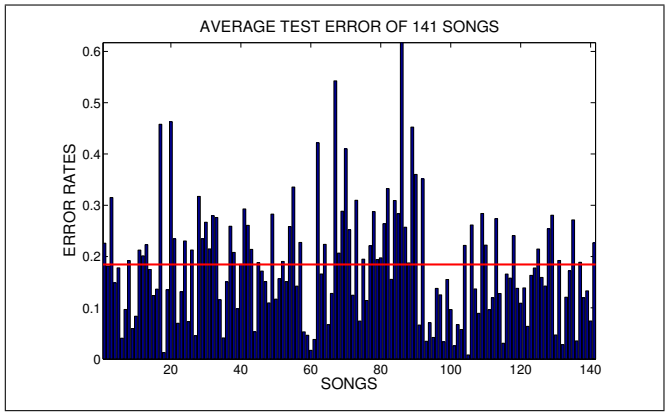

Figure 3. Test classification error rates for individual songs by GLM model. The dash line gives the average error rates of the 100-split cross-validation.

5.2 Correlation Between Classifiers and Voting

While observing the classification variation among data splits in Figure 2, we also noticed that even though classification performance is different from classifier to classifier, the tendency of these five curves does share some similarity. Here we first carefully studied the pair-to-pair performance correlation between the classification algorithms. In Table 2 the degree of matching is reported: 1 refers to perfect match; 0 to no match. It seems that the two linear classifiers have a very high degree of matching, which means that little will be gained by combining these two.

The simplest way of combining classification results is

	LR	GLM	GMM	rKOPLS	K-means
LR	1.0000	0.9603	0.8203	0.8040	0.8110
GLM	0.9603	1.0000	0.8141	0.8266	0.8091
GMM	0.8203	0.8141	1.0000	0.7309	0.7745
rKOPLS	0.8040	0.8266	0.7309	1.0000	0.7568
K-means	0.8110	0.8091	0.7745	0.7568	1.0000

Table 2. A matrix of the degree of matching.

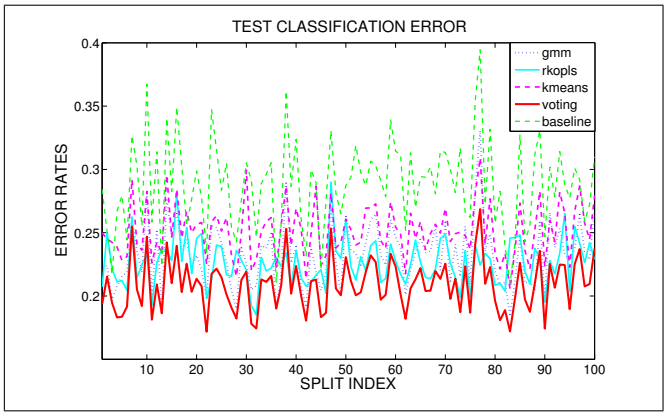

Figure 4. Voting results. It gives the voting performance among GMM, rKOPLS and K-means. The light dash line shows the baseline of random guessing for each data split.

by majority voting, meaning that the class with the most votes is chosen as the output. The voting has been done crossing all five classifiers, unfortunately the average voting results (error rates) on the test sets was 18.62%, which is slightly worse than the best individual classifier. The reason seems to be that even though the other classifiers are not so correlated with the linear ones, the miss classification rate is too high to improve performance.

However voting does help enhance the performance, if it performs among not so correlated classification results. Figure 4 demonstrates the sample-to-sample majority voting among three classifiers: GMM, rKOPLS and K-means. The similar tendency was preserved in the voting results, and there were only 10 splits out of 100, where the voting results were worse than the best ones among these three. The average performance of voting on test sets was $20.90 \pm 2.02\%$.

Here we will elaborate on the performance on individual songs, by looking at the predicted labels from each classifier and voting predictions. Figure 5 demonstrates how voting works, and how the prediction results correlate. Two songs: 'Do You Know What You Want' by M2M, and 'A Thousand Times' by Sophie Zelmani, have been chosen to illustrate the 'good' and 'bad' cases, i.e. when voting helps and fails. Vocal segments are tagged with '1', and '0' for non-vocal ones. The ground truth is given as a reference. The voting was carried out among GMM, rKOPLS and K-means, and their predictions are shown. If the classifiers make mistakes in a similar pattern, the voting cannot recover the wrong predictions, e.g. area B. If the predictions are not correlated to a high degree voting helps, e.g. area A.

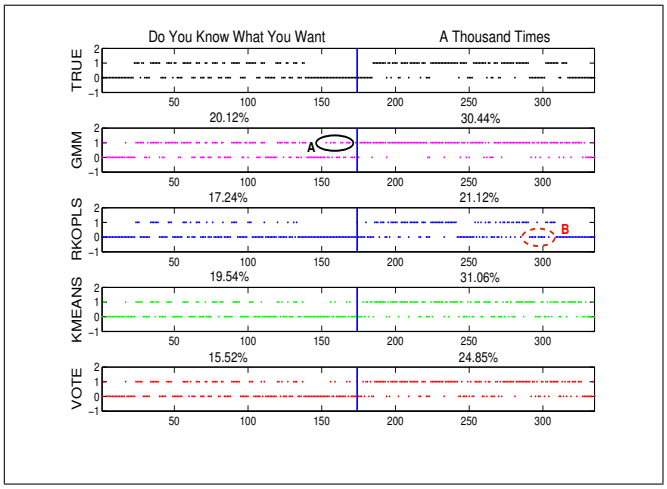

Figure 5. Sample-to-sample errors and voting results. Two songs represent the 'good' and 'bad' voting cases. Individual error rates for each classifier and voting results are given. Two areas marked A & B indicate the scenarios when voting helps and fails.

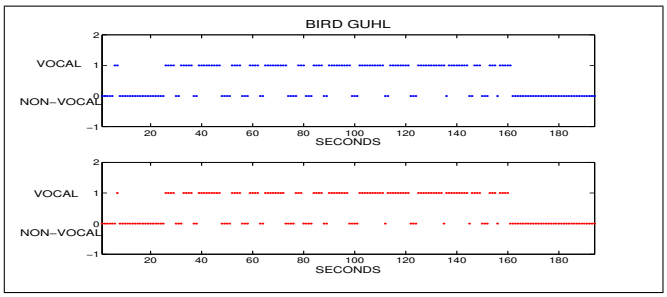

Figure 6. Two manual label results of the same song: 'Bird Guhl'. It is obvious that the disagreement only appears in the transition parts.

Moreover, we noticed that it is very likely for classifiers to make wrong predictions in the transition sections, meaning the changing from vocal to non-vocal parts, and vice versa. We found this is reasonable comparing with manual labels by different persons, shown in Figure 6. The song was labeled carefully by both people, the absence of mind or guessing should not be a concern. The mismatch indicates the perception or judging difference, and it only happens in the transition parts. The total mismatch is about 3% for this particular song: 'Bird Guhl' by Antony and the Johnsons.

5.3 'Artist Effect'

In previous experiments, we randomly selected songs to form training and test sets, hence the same artist may appear in both sets. Taking the previous results as a baseline, we studied the 'artist effect' in this classification problem. We tried to keep the size of test sets the same as before, and carefully selected around 30 songs in order to avoid artist overlap for each split, and formed 100 splits. The second column of Table 1 summarizes the average error rates for 4 classi-

fiers. The average results are a little worse than the previous ones, and they also have bigger variance along the splits. Therefore we speculate that artists do have some influence in vocal/non-vocal music classification, and the influence may be caused by different styles, and models trained on particular styles are hard to be generalized to other styles.

6 CONCLUSION AND DISCUSSION

We have investigated the vocal/non-vocal popular music classification. Experiments were carried out on our database, containing 147 popular songs. To be in line with the label set, the classifiers were trained based on features at $1sec$ time scale. We have employed 142-d acoustic features, consisting MFCCs and MAR, to measure the distinct properties of vocal and non-vocal music. Five classifiers have been invoked: LR, GLM, GMM, rKOPLS and K-means.

We cross-validated the entire database, and measured the aggregate average to eliminate the data set dependency. GLM outperformed all the others, and provided us with 18.46% error rate on the baseline of 28%. The performance has great variation among data splits and songs, indicating the variability of popular songs. The correlations among classifiers have been investigated, and the proposed voting scheme did help among less correlated classifiers. Finally we looked into the 'artist effect', and it did degrade the classification accuracy a bit by separating artists in training and test sets. All in all vocal/non-vocal music classification was found to be a difficult problem, and it depends heavily on the music itself. Maybe classification within similar song styles can improve the performance.

7 REFERENCES

[1] Ahrendt, P., Meng, A. and Larsen, J. "Decision time horizon for music genre classification using short time features", *Proceedings of EUSIPCO*, pp. 1293-1296, 2004.

[2] Akeroyd, M. A., Moore, B. C. J. and Moore, G. A. "Melody recognition using three types of dichotic-pitch stimulus", *The Journal of the Acoustical Society of America*, vol. 110, Issue 3, pp. 1498-1504, 2001.

[3] Arenas-García, J., Petersen, K.B., Hansen, L.K. "Sparse Kernel Orthonormalized PLS for feature extraction in large data sets", *Proceedings of NIPS*, pp. 33-40, MIT Press, Cambridge, MA, 2007.

[4] Berenzweig, A. L., Ellis, D. P. W., and Lawrence, S. "Locating Singing Voice Segments Within Music Signals", *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Aoustics*, New Paltz, NY, 2001.

[5] Berenzweig, A. L., Ellis, D. P. W., and Lawrence, S. "Using Voice Segments to Improve Artist Classification of Music", *Proceedings of the International Conference on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, 2002.

[6] Eronen, A. "Automatic musical instrument recognition". Master Thesis, Tampere University of Technology, 2001.

[7] Hansen, L. K., Sigurdsson, S., Kolenda, T., Nielsen, F. ., Kjems, U., Larsen, J. "Modeling text with generalizable gaussian mixtures", *Proceedings of ICASSP*, vol. 4, pp. 3494-3497, 2000

[8] Heln, M. and Virtanen, T. "Separation of Drums From Polyphonic Music Using Non-Negative Matrix Factorization and Support Vector Machine", *Proceedings of EUSIPCO*, Antalya, Turkey, 2005.

[9] McKinney, M. F. and Breebart, J. "Features for audio and music classification", *Proceedings of ISMIR*, Baltimore, Maryland (USA), pp.151-158, 2003.

[10] Meng, A., Shawe-Taylor J., "An Investigation of Feature Models for Music Genre Classification using the Support Vector Classifier", *Proceedings of ISMIR*, London, UK, pp. 604-609, 2005.

[11] Meng, A., Ahrendt, P., Larsen, J. and Hansen, L. K. "Temporal Feature Integration for Music Genre Classification", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15(5), pp. 1654-1664, 2007.

[12] Nabney, I., Bishop, C. "Netlab neural network software", ver. 3.2, 2001

[13] Nielsen, A. B., Sigurdsson, S., Hansen, L. K., and Arenas-Garca, J. "On the relevance of spectral features for instrument classification", *Proceedings of ICASSP*, Honolulu, Hawaii, vol. 5, pp. 485-488, 2007.

[14] Nwe, T. L. and Wang, Y. "Automatic Detection of Vocal Segments in Popular Songs" *Proceedings of the ISMIR*, Barcelona, Spain, 2004.

[15] Scheirer, E. and Slaney, M. "Construction and Evaluation fo A Robust Multifeature Speech/Music Discriminator", *Proceedings of ICASSP*, Munich, 1997.

[16] Tsai W.-H. and Wang, H.-M. "Automatic singer recognition of popular music recordings via estimation and modeling of solo vocal signals", *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 330–341, 2006.

[17] Tzanetakis, G. "Manipulation, analysis and retrieval systems for audio signal", *Ph.D. dissertation* , Faculty of Princeton University, Department of Computer Science, 2002

[18] Virtanen, T. "Separation of Sound Sources by Convolutive Sparse Coding", *Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, SAPA*, 2004.

[19] Williams, G. and Ellis, D. P. W. "Speech/Music Discrimination Based on Posterior Probability Features", *Proceedings of Eurospeech*, Budapest, 1999.

[20] Zhang, T. "Automatic singer identification", *Proceedings of ICME*, Baltimore, 2003.