AN INVESTIGATION OF FEATURE MODELS FOR MUSIC GENRE CLASSIFICATION USING THE SUPPORT VECTOR CLASSIFIER

Anders Meng
Informatics and Mathematical Modelling - B321
Technical University of Denmark
am@imm.dtu.dk

John Shawe-Taylor
University of Southampton
jst@ecs.soton.ac.uk

ABSTRACT

In music genre classification the decision time is typically of the order of several seconds, however, most automatic music genre classification systems focus on short time features derived from $10-50$ms. This work investigates two models, the *multivariate Gaussian model* and the *multivariate autoregressive model* for modelling short time features. Furthermore, it was investigated how these models can be integrated over a segment of short time features into a kernel such that a support vector machine can be applied. Two kernels with this property were considered, the *convolution kernel* and *product probability kernel*. In order to examine the different methods an 11 genre music setup was utilized. In this setup the *Mel Frequency Cepstral Coefficients* were used as short time features. The accuracy of the best performing model on this data set was $\sim 44\%$ compared to a human performance of $\sim 52\%$ on the same data set.

Keywords: Feature Integration, Product Probability Kernel, Convolution Kernel, Support Vector Machine, Music Genre

1 INTRODUCTION

The field of audio mining covering areas such as audio classification, retrieval, fingerprinting etc. has received quite a lot of attention lately both from academic and commercial groups. Some of this interest stems from an increased availability of large online music stores and growing access to live radio-programs, music stations, news on the internet etc. The big task for the academic world is to find methods for effectively searching and navigating these large amounts of data.

The genre is probably the most important descriptor of music in everyday life, however, it is not an intrinsic property of music such as e.g. tempo, which makes it more difficult to grasp with computational methods. Still, for a limited amount of data and for coherent music databases there seem to be a link between computational methods and human assessment, see e.g. [1, 2].

It is a well established fact that the success of a pattern recognition system is closely related to the task of finding descriptive features. There exist a large amount of descriptive audio features, each designed for a specific audio mining task. The various features can be grouped as perceptual features such as pitch, loudness, beat or as non-perceptual features as the Mel Frequency Cepstral Coefficients (MFCC). The MFCCs have been applied in a range of audio mining tasks, and have shown good performance compared to other features at a similar time scale.

In music genre classification the typical time horizon for a human to classify a piece of music as belonging to a specific genre is of the order of a quarter of a second up to several seconds, see [3]. Typically for automatic music genre classification systems whole pieces of music are available, so the decision time is generally longer than just a few seconds.

Short time features such as the MFCCs are typically derived at time horizons around $10-50ms$ depending on the stationarity of the audio signal. A few authors [4, 5, 1] have looked at methods for integrating (modelling) the short time features to classify at longer time horizons. Integration of short time features (*feature integration*) is also known as early information fusion. Late information fusion is another way of classifying at larger time horizons. The idea of late information fusion is to combine the sequence of outputs from a classifier, like e.g. majority voting. Some techniques of information fusion (both early and late) have been considered in more detail in [4, 2].

The focus of this work was to extend the model of [2] for modelling the temporal structure of short time features and secondly to investigate different methods for handling audio data using kernel methods such as the *Support Vector Machine (SVM)*. The support vector machine is known for its good generalization performance in high-dimensional spaces, furthermore, its ability to work implicitly in a possible high-dimensional feature space makes it possible to investigate non-linear relations in the data.

The paper is structured as follows. An overview of the investigated features as well as a description of the two feature integration models the *multivariate Gaussian*

model (GM) and the *multivariate autoregressive model (MAR)* are given in section 2. Section 3 briefly explains the classifiers applied to a music genre setup and furthermore explains the idea of information fusion. Section 4 presents the results of an 11 genre music genre setup. Last, but not least a conclusion in section 5.

2 FEATURES

The work presented in this paper will focus on constructing descriptive features at larger time scales by modelling short time features. Earlier work by [2, 1, 5] suggested to work with an intermediate time scale around 1 second. Here three time scales have been considered, a *short time scale* of 30ms where short time features are extracted, a *medium time scale* at 2 seconds (selected from the data set, see section 4) and a *long time scale* of 30 seconds, limited by the length of the music snippets. The long time scale contains information such as the "mood" of the song as well as long-structural correlations.

2.1 Short Time Features (30ms)

The short time feature extraction stage is really important in all audio processing applications, since it is the first level of feature integration performed[1]. Earlier results [4, 5] indicate good performance in music genre classification using the MFCCs and therefore these will be the preferred choice in this investigation. These features were originally developed for classification of speech, however, they have been applied in various audio mining tasks, see e.g. [6] where they were used in a timbre similarity experiment. The low order MFCCs contain information of the slowly changing spectral envelope while the higher order MFCCs explains the fast variations of the envelope. Several authors report success using only the first $6-10$ MFCCs. In the music genre classification setup, see section 4, we found that the first seven MFCCs were adequate. Furthermore, a hop- and frame-size of 10ms and 30ms, respectively, were used. The larger overlap results in more smooth transitions between consecutive feature vectors.

2.2 Feature Integration (>30ms)

Feature integration is a method for capturing the temporal information in the features. With a good model the most salient structural information remains and the noisy part is suppressed. The idea of using feature integration in audio classification is not new, but has been investigated in earlier work by e.g. [1, 5, 2] where a performance increase was observed. The idea of feature integration can be stated more strict by observing a sequence of consecutive features

$$\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+L} \rightarrow \mathbf{f}(\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+L}) = \mathbf{z}, \quad (1)$$

where the sequence $\{\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+L}\} \in \mathcal{R}^{D\times L}$ are integrated into a new feature vector denoted as $\mathbf{z} \in \mathcal{R}^M$ where typically $M << D\cdot L$ and L indicates the number of short time features used in the integration step. A commonly used feature integration technique is the *mean-variance* of features, which provides a performance increase, but generally does not capture the temporal structure of the short time features. An improvement to this is the *filter-bank* approach considered in [5] to capture the frequency contents of the temporal structure in the short time features. This improvement indicated a performance increase compared to the mean-variance model, see [2]. Recently an autoregressive model [2] was suggested for feature integration and provided a performance increase compared to the mean-variance and filter-bank approach.

Figure 1 shows the first seven normalized MFCCs of a 10 second excerpt of the music piece *Master of Revenge* by the heavy metal group *Body Count*. As observed from the coefficients there is both temporal correlations as well as correlations among features dimensions.

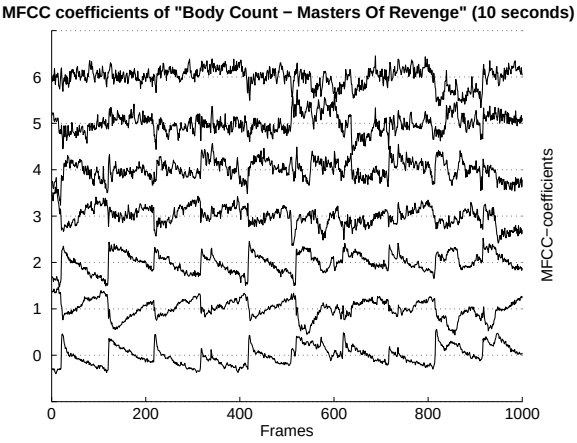

Figure 1: The first seven normalized MFCCs of a 10 second snippet of "Body Count - Masters of Revenge". The temporal correlation and correlations among feature dimensions are very clear from this piece of music.

2.2.1 *Multivariate autoregressive model (*MAR*)*

The *multivariate autoregressive* model handles both temporal and correlations among feature dimensions, which makes it a good candidate for feature integration. In [2] a simple autoregressive model was suggested where simple refers to considering each feature dimension independently. The MAR model is popular in time-series modelling and prediction being both simple and well understood, see e.g. [7]. For a stationary time series of state vectors $\mathbf{x}_n \in \mathcal{R}^D$ the MAR model is defined by

$$\mathbf{x}_n = \sum_{p=1}^{K} \mathbf{A}_p \mathbf{x}_{n-I(p)} + \boldsymbol{\mu} + \mathbf{u}_n, \quad (2)$$

where the noise term $\mathbf{u}_n$ (error-term) is assumed to be zero mean Gaussian distributed, hence $\mathbf{u}_n \sim \mathcal{N}(\mathbf{u}_n; \mathbf{0}, \mathbf{C})$.

The D-dimensional parameter vector $\boldsymbol{\mu}$ is a vector of intercept terms that is included to allow for a non-zero mean of the time-series, see [8]. The matrices $\mathbf{A}_p \in \mathcal{R}^{D\times D}$ for $p = 1\ldots K$ are the coefficient matrices of the K'th order multivariate autoregressive model. They

[1] Basically this first step is denoted as feature extraction and not feature integration.

encode how much of the previous information given in $\mathbf{x}_{n-I(1)}, \mathbf{x}_{n-I(2)}, .., \mathbf{x}_{n-I(K)}$ is present in $\mathbf{x}_n$. The above formulation is quite general as I refers to a general set. For a model order of $K = 4$, the set could be selected as $I = \{1, 2, 3, 4\}$ or $I = \{1, 2, 4, 8\}$ indicating that $\mathbf{x}_n$ is predicted from these previous state vectors. In this paper we focus on the standard multivariate autoregressive model where $I = \{1, 2, 3, \ldots, K\}$. When estimating the parameters of the model there is several methods available, see e.g. [7]. The authors have used the *ARFIT* package, a regularized ordinary least squares approach, described in [8]. This package ensures the uniqueness of the estimated parameters of the model.

*2.2.2 Multivariate Gaussian model (*GM*)*

Neglecting the temporal correlations in the data, hence setting the $\mathbf{A}_p$ matrices for $p = 1, \ldots, K$ in equation (2) to zero leads to the much simpler model

$$\mathbf{x}_n = \boldsymbol{\mu} + \mathbf{u}_n, \tag{3}$$

where $\boldsymbol{\mu}$ encode the mean value of the time series and $\mathbf{u}_n \sim \mathcal{N}(\mathbf{u_n}; \mathbf{0}, \mathbf{C})$ is denoted the multivariate Gaussian model. The previous mentioned *mean-variance* model is the mean value $\boldsymbol{\mu}$ and the variance components given from the diagonal of the covariance matrix $\mathbf{v} = \mathrm{diag}\{\mathbf{C}\}$. If the full covariance matrix is used, only the upper (or lower) triangular coefficients are needed due to the symmetry. The multivariate Gaussian model will be considered as the "base-line" against the MAR model in the experimental section since it performs better than the typical *mean-variance* model.

The two feature integration techniques described above can be used to derive features at the *medium time scale* or used directly to derive features at the *long time scale*. The model order for the MAR model can be selected from e.g. Schwarz's Bayesian Criterion (SBC) [8], which is implemented in the *ARFIT* package or as in our experimental setup, where a separate validation set was used to determine the optimal model order across data examples (music snippets).

2.3 Unique Solutions

Performing feature integration the model parameters are typically used as new feature vectors at the new time scale. If the model does not have a unique solution, two similar audio pieces could risk being classified as dissimilar. Consider using a *mixture of Gaussian (MoG)*, given as

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} p(k)p(\mathbf{x}|k, \boldsymbol{\theta}),$$

where $p(k)$ (and $\sum_{k=1}^{K} p(k) = 1$) are the mixing proportions and $p(\mathbf{x}|k, \hat{\boldsymbol{\theta}}) \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{C}_k)$, as a feature integration model. Optimizing the model parameters from the likelihood function using e.g. the *EM-algorithm* does not necessarily provide a global maximum since the likelihood function has many local maximums. So using these model parameters (mixing proportions, means and covariances) directly in a classifier[2] would make no sense. Recent studies in kernels indicate that it is possible to integrate this type of complicated models in a kernel, see e.g. [9, 10]. The mixture of Gaussian model was considered as modelling music snippets in [6] and will be investigated as a feature integration model in section 4.

[2] Stacked in a vector.

3 CLASSIFIERS

Earlier work in the field of music information retrieval (*MIR*) considered simple yet efficient classifiers such as K-nearest neighbors, however, lately more computationally demanding algorithms have been investigated. Only a few researchers within the field of *MIR* have considered support vector machines (*SVM*), see e.g. [11, 12]. In the following subsections the support vector classifier (SVC) and the linear neural network classifier (LNN) will be briefly discussed.

3.1 Support Vector Classifier

The challenge of machine learning is to provide the learner with as broad a range of functions as possible while still ensuring that accurate learning can be achieved. Using high-dimensional feature spaces satisfies the first constraint of ensuring high flexibility, but appears to be at odds with the second since it is undermined by the curse of dimensionality. As a result we would expect that a good fit on the training data could still leave the generalization very poor. Support vector machines [13] manage to avoid this difficulty by optimizing a bound on the generalization error in terms of quantities that do not depend on the dimension of the feature space [14], hence enabling good performance unaffected by the curse of dimensionality. In the present work, the C-library *LIBSVM* [15] was used. This library implements the one-against-one voting terminology to handle more than two classes.

3.1.1 Kernels

A typical applied kernel for the support vector classifier is the *linear kernel*, which is defined as $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$, hence an inner product between the input vectors. Another well known kernel is the Gaussian kernel (or *RBF-kernel*) with width parameter σ defined as $\kappa(\mathbf{x}, \mathbf{x}') = \exp(- \parallel \mathbf{x} - \mathbf{x}' \parallel^2 /2\sigma^2)$. Using this kernel the support vector classifier is basically finding discriminating dimensions in an infinite feature space.

The linear and RBF kernel can be used in comparing vector data, however, when handling audio we are typically forced to calculate the distance between two audio snippets of varying lengths, which for two pieces of audio is presented by the sequence of short time features: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L] \in \mathcal{R}^{D\times L}$ and $\mathbf{X}' = [\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_{L'}] \in \mathcal{R}^{D\times L'}$. The two audio files are not required to be of same length, though in the present investigation they are ($L = L'$). Two different kernels have been investigated, which calculate a similarity between sequences of data, the *convolution kernel* [16] and the *product probability kernel* [9]. These kernels naturally incorporate feature integration.

Convolution Kernel - CK

The convolution kernel [16] handles all kinds of discrete

structures such as strings, trees and graphs. In this work the convolution kernel measures the distance (correlation) between two audio pieces (between their feature vectors). The kernel is defined as

$$\kappa(\mathbf{X}, \mathbf{X}') = \frac{1}{L^2} \sum_{v=1}^{L} \sum_{v'=1}^{L} \kappa_I \left(\mathbf{x}_v, \mathbf{x'}_{v'} \right), \qquad (4)$$

where $\kappa_I(\mathbf{x}, \mathbf{z})$ must be a valid kernel. It is interesting to note that if a linear kernel is used a fast calculation can be obtained.

Product Probability Kernel - PPK
The *product probability kernel* introduced in [9] measures the distance between probability models of the feature vectors. Other divergence based kernels have been suggested, see e.g. [10], for measuring a similar distance. In [6] the Kullback-Leibler similarity measure was applied to measure the distance between timbre models of music snippets modelled by a mixture of Gaussian, however, no closed form solution could be found using this divergence measure. With the *product probability kernel*, a closed form solution can be determined for e.g. a mixture of Gaussian, furthermore, the *PPK* fulfills the requirement for a kernel to be positive semi-definite. From [9] the *PPK* is given as

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = \int p(\mathbf{x}|\boldsymbol{\theta})^{\rho} p(\mathbf{x}|\boldsymbol{\theta}')^{\rho} d\mathbf{x}, \qquad (5)$$

where $\boldsymbol{\theta}(\boldsymbol{\theta}')$ are the parameters from modelling $\mathbf{X}(\mathbf{X}')$, $\rho > 0$ and $p(\mathbf{x}|\boldsymbol{\theta})$ is the probabilistic model of the short time features of a music piece. ρ controls the weighting of low or high density areas of the probability distribution. Selecting $\rho = 1/2$ the *Bhattacharyya* affinity between distributions is found. A nice bi-product of selecting $\rho = 1/2$ is a normalized kernel structure, since $\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}) = \int p(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x} = 1$. This kernel can directly compute the distance between the models suggested in section 2.2, and thus incorporates feature integration. As mentioned in section 2.3 the problem of uniqueness is alleviated for this kernel, since probabilistic models are compared instead of model parameters.

Closed form solutions of the kernel for the multivariate Gaussian and mixture of Gaussian can be found in [9]. Additionally, we have calculated a closed form solution of the MAR model, but the details have been omitted through lack of space [3].

3.2 Linear Neural Network classifier (LNN)

The linear Neural Network has c outputs and is trained using a squared loss function [17]. This classifier has previously been applied with success in music genre classification, see e.g. [2, 4].

3.3 Fusion Techniques

The early information fusion (feature integration) was discussed in section 2.2. Late information fusion is the problem of combining the results from the classifier. There exist several ways of performing late information fusion, see [18]. In the present work, the majority voting rule was applied due to the SVM classifier. In the majority vote rule, the votes received from the classifier are counted and the class with the largest amount of votes is selected, hereby performing consensus decision.

[3] Regarding computational complexity the methods ranked after numerical complexity are (top: least computational intensive): GM, MAR, MoG. The GM and MAR are closer related in complexity than the MAR and MoG.

4 EXPERIMENTS

To evaluate the different feature integration techniques an 11 genre music setup was investigated. As discussed in the introduction, decisions can be made at different time scales. In the present work, the best achievable performance at 30 seconds will be pursued, using the above feature integration techniques, voting technique and combinations of the two.

4.1 Data set

The data set consists of 11 music genres distributed evenly among the following categories: *Alternative, Country, Easy Listening, Electronica, Jazz, Latin, Pop&Dance, Rap&Hiphop, R&B and Soul, Reggae and Rock*. The data set consists of a training set of 1098 music snippets, 100 from each genre except for latin, of each 30 seconds and a separate test set of 220 music snippets each of 30 seconds in length. The music snippets were *MP3* encoded music with a bit-rate $\geq 128kB$ down-sampled with a factor two to 22050Hz.

4.1.1 Human evaluation

To test the integrity of the data set a human evaluation was performed on the music snippets (at a 30 second time scale) of the test set. Each test person out of 9 was asked to classify each music snippet into one of the 11 genres on a forced choice basis. Each person evaluated 33 music snippets out of the 220 music pieces. No information except for the genre of the music pieces was given prior to the test. The average accuracy of the human evaluation across people and across genre was 51.8% as opposed to random guessing, which is $\sim 9.1\%$. The lower/upper 95% confidence limits were $46.0\%/57.7\%$ (results shown in figure 2, upper figure). The human evaluation shows that the common genre definition is less consistent for this data set, however, it is still interesting to observe how an automatic genre system works in this setup.

4.1.2 Results & Discussion

In each genre 90 out of the 100 music snippets from the training set were randomly selected 10 times to assess the variations in the data. In each of these runs the remaining music pieces (10 in each genre, except *latin*) was used as a validation set for tuning parameters such as C in the support vector classifier and σ in the RBF kernel. Optimal model order selection for the MAR models were determined across music samples and evaluated on the validation set. A model order of $K = 3$ at both 2 and 30 seconds was found optimal.

The medium time scale was selected by evaluating the performance at 30 seconds using both the *GMMV* and the

Table 1: Description of the different combinations investigated. All investigations with the product probability kernel, $\rho = 1/2$ was used.

Scheme	Description
MOG,PPK	Mixture of Gaussian applied to each 30 second music snippet. A PPK kernel was generated (dimension 990×990).
GM,PPK	A multivariate Gaussian is fitted for each 30 second music snippet. A PPK kernel was generated.
GM,PPK,MV	A multivariate Gaussian is fitted for each 2 seconds of music data. A PPK kernel is generated (sampling applied using only 3 samples from each music piece resulting in a kernel of 2970×2970). After classification with SVM, majority voting is applied.
GM,CONV	A multivariate Gaussian is fitted for each 2 seconds of music data and a linear convolution kernel is applied (taking mean of the parameters).
GM,MV	A multivariate Gaussian is fitted for each 2 seconds of music data and majority voting is applied to the outputs of the classifiers. For the SVM a RBF-kernel was applied.
GM	A multivariate Gaussian is fitted for each 30 second music snippet. For the SVM a RBF-kernel was applied.
MAR,PPK	Same as above (see GMPPK), just with a multivariate AR process.
MAR,PPK,MV	Same as above, just with a multivariate AR process.
MAR,PPK,CONV	Same as above, just with a multivariate AR process.
MAR,CONV	Same as above, just with a multivariate AR process.
MAR,MV	Same as above, just with a multivariate AR process.
MAR	Same as above, just with a multivariate AR process.

MARMV method explained in table 1 varying the frame-/hop size of the medium time scale[4]. No big performance fluctuation was observed in this investigation, however, a small favor of a frame-/hop size of $2/1$ second was observed. The various combinations investigated have been described in more detail in table 1. For the mixture of Gaussian model incorporated in a product probability kernel (MOG,PPK) the optimal model order for each music snippet of 30 seconds were selected by varying the model order between $2-6$ mixtures, and selecting the optimal order from the Bayesian Information Criterion (BIC).

The average accuracy over the ten runs of the various combinations illustrated in table 1 have been plotted in figure 2 (upper figure) with a 95% binomial confidence applied to the average values. From the accuracy plot there is a clear indication that the MAR model is performing better than the GM for both the *SVM* and *LNN* classifier. Performing a McNemar test, see e.g. [19], on the mixture of Gaussian model (MOGPPK) and the Gaussian model in a product probability kernel (GMPPK) the probability that

[4]The investigated frame-/hop sizes were:{1s/0.5s, 1.5s/0.75s, 2s/1s, 2.5s/1.25s, 3s/1.5s, 3.5s/1.75s}.

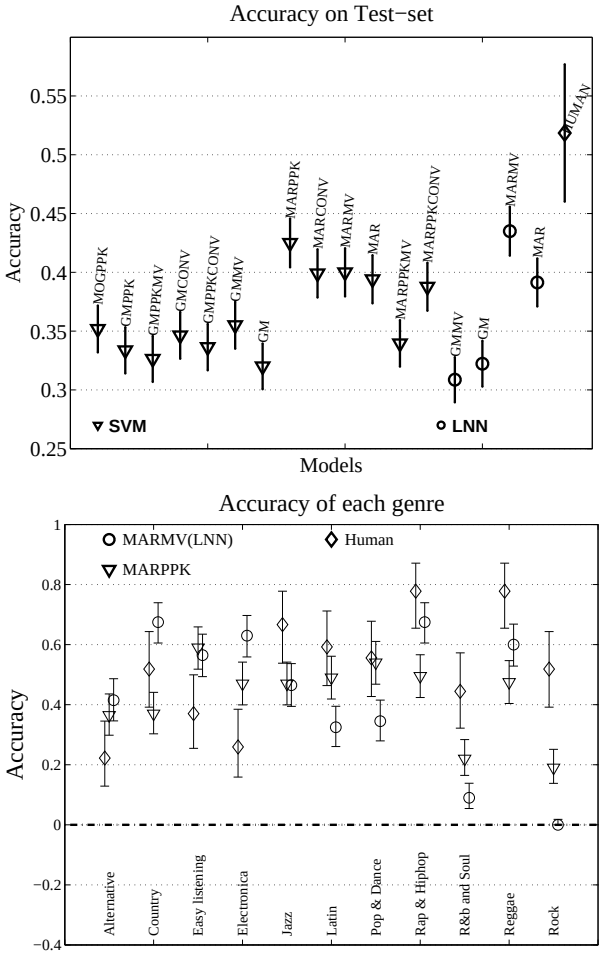

Figure 2: **Upper:** Average accuracy at 30 seconds shown with a 95% binomial confidence interval for all investigated combinations. The larger confidence interval for humans is due to only nine persons evaluating a part of the test-data. **Lower:** Average accuracy with 95% confidential interval of each genre at a time scale of 30 seconds using the two best performing combinations, *MARMV* and *MARPPK*. The average human accuracy in each genre is also shown with a 75% confidence interval.

the two models are equal is 76%, hence the hypothesis that the models are equal cannot be rejected on a 5% significance level. This observation, together with the good performance of the MAR model illustrate the importance of the temporal information in the short time features. Even with the various techniques applied in this setup we are still around $\sim 8\%$ from the average human accuracy of $\sim 52\%$ on this data set, but it is interesting to notice that reasonable performance is achieved with fairly simple feature integration models and fusion techniques using only the first seven MFCCs. The two best performing models are the MAR model in a product probability kernel (MARPPK) and the MAR model modelled at 2 seconds, after which majority voting is applied on the LNN outputs (MARMV), see figure 2 (upper). The McNemar test on these two models showed a 43% significance level thus it can not be rejected that the two models are similar.

The advantage of the MARPPK model is that we only need to store the model parameters at 30 seconds, while

for the MARMV model a sequence of model parameters need to be saved for each music snippet. The computational workload though, is a little larger for the MARPPK model when compared to the MARMV model.

Figure 2 (lower) shows the accuracy on each of the 11 genres of the two models MARMV and MARPPK. The MARPPK seem to be more robust in classifying all genres, whereas the MARMV is much better at specific genres such as *Rap & Hiphop* and *Reggae*. However, the MARMV does not capture any of the *Rock* pieces, but generally confuses them with *Alternative* (not shown here). Also illustrated in this figure is the human performance in the different classes. A confidence interval of 75% has been shown on the human performance, due to the few test persons involved in the test. The humans are much better at genres such as *Rap & Hiphop* and *Reggae* than, e.g. *Alternative*, which also corresponds to some of the behavior observed with the MARMV method.

5 CONCLUSION

The purpose of this work has partly been to illustrate the importance of modelling the temporal structure in the short time features, and secondly how models of short time features can be integrated into kernels, such that the support vector machine can be applied. In the music genre setup the best performance was achieved with the MAR model in a product probability kernel (MARPPK) used in combination with an SVM and with the MAR model used in combination with majority voting (MARMV) in a linear neural network. The average accuracy of these two methods were $\sim 43\%$ compared to a human average accuracy of $\sim 52\%$.

Even though the results presented in this article were a music genre setup, the general idea of feature integration and generating a kernel function, which efficiently evaluates the difference between audio-models can be generalized and used in other fields of *MIR*.

ACKNOWLEDGEMENTS

The work was supported by the European Commission through the sixth framework IST Network of Excellence: Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), contract no. 506778.

REFERENCES

[1] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5), July 2002.

[2] A. Meng, P. Ahrendt, and J. Larsen. Improving music genre classification by short-time feature integration. In *Proc. of ICASSP*, pages 1293–1296, 2005.

[3] D. Perrot and R. Gjerdigen. Scanning the dial: An exploration of factors in identification of musical style. In *Proc. of Soc. Music Perception Cognition*, 1999.

[4] P. Ahrendt, A. Meng, and J. Larsen. Decision time horizon for music genre classification using short-time features. In *Proc. of EUSIPCO*, pages 1293–1296, 2004.

[5] M. F. McKinney and J. Breebaart. Features for audio and music classification. In *Proc. of ISMIR*, pages 151–158, 2003.

[6] J-J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *Proc. of ISMIR*, 2002.

[7] Helmut Lütkepohl. *Introduction to Multiple Time Series Analysis*. Springer-Verlag, 1993.

[8] A. Neumaier and T. Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Transactions on Mathematical Software*, 27(1):27–57, March 2001.

[9] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research*, pages 819–844, July 2004.

[10] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A kullback-leibler divergence based kernel for svm classification in multimedia applications. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[11] L. Lu, S. Z. Li, and Zhang H.-J. Content-based audio segmentation using support vector machines. In *ACM Multimedia Systems Journal*, volume 8, pages 482–492, March 2003.

[12] S.-Z. Li and G. Guo. Content-based audio classification and retrieval using svm learning. In *First IEEE Pacific-Rim Conference on Multimedia, Invited Talk, Australia*, 2000.

[13] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[14] J. Shawe-Taylor and N. Cristianini. On the generalisation of soft margin algorithms. *IEEE Transactions on Information Theory*, 48(10):2721–2735, 2002.

[15] C-C. Chang and C-J. Lin. Libsvm : A library for support vector machines, 2001. (http://www.csie.ntu.edu.tw/ cjlin/libsvm).

[16] David Haussler. Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz, July 1999.

[17] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[18] J. Kittler, M. Hatef, Robert P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[19] T.G Diettereich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, (10):1895–1924, 1998.