CO-OCCURRENCE MODELS IN MUSIC GENRE CLASSIFICATION

Peter Ahrendt[*], *Jan Larsen*

Informatics and Mathematical Modelling
Technical University of Denmark
2800 Kongens Lyngby, Denmark
pa,jl@imm.dtu.dk

Cyril Goutte

Xerox Research Centre Europe
6 ch. de Maupertuis
F-38240 Meylan, France
cyril.goutte@xrce.xerox.com

ABSTRACT

Music genre classification has been investigated using many different methods, but most of them build on probabilistic models of feature vectors x_r which only represent the short time segment with index r of the song. Here, three different co-occurrence models are proposed which instead consider the whole song as an integrated part of the probabilistic model. This was achieved by considering a song as a set of independent co-occurrences (s, x_r) (s is the song index) instead of just a set of independent (x_r)'s. The models were tested against two baseline classification methods on a difficult 11 genre data set with a variety of modern music. The basis was a so-called AR feature representation of the music. Besides the benefit of having proper probabilistic models of the whole song, the lowest classification test errors were found using one of the proposed models.

1. INTRODUCTION

In these years, the growth in digital music on the Internet is tremendous. Several companies now offer music for on-line sale, such as iTunes with more than 800,000 song tracks available. Besides, radio channels and TV broadcasting companies have started offering their services and the demand for efficient information retrieval in these streams is obvious. An important part in this is *music genre classification*, which will be addressed here. The general idea is to extract features from (most often) short frames of the digitized sound signal. A classifier then use this time sequence of features to classify the song[1] into genres such as jazz, pop and blues. Several researchers have contributed to this field, such as [1], [2], and [3].

In the current work, music genre classification will be addressed with a *co-occurrence model*. In this novel view, a song is seen as a set of co-occurrences between a song and its constituent *sound elements* which represent segments in time of the song. The inspiration to use this model came from the area of information retrieval, where the method of Probabilistic Latent Semantic Analysis (PLSA) [4] has shown to be very powerful in e.g. automated document indexing. In PLSA, the co-occurrences are between a document and words in the document where the words are elements of a discrete, finite vocabulary.

Analogies between music and textual language can be found on many levels and both can be seen to contain a notion of grammar, syntax and semantics [5]. [6] shows that the frequencies of the usage of different notes in musical compositions follow Zipf's law, which is also known to apply to word frequencies in documents. Zipf's law is said to apply if $f = k \cdot r^{-b}$, where f is the frequency, k is some constant, r is the rank (of the frequencies) and b is a constant that should be close to 1. These previous findings support the usage of the co-occurrence model in music genre classification, but extracting the note and instrument composition directly and correctly from general digitized music is still an open problem.

For this reason, experiments have been made with several different approaches to represent the equivalents of words in music (the sound elements). Section 2 discuss the so-called *AR features*, which give the basic (30 dimensional) feature space representation of the music. This feature space is seen as the ground on which to build different word equivalents. Section 3 first gives the formalism and theory of the co-occurrence model and PLSA. Afterwards, discrete and continuous vocabulary models are described. Section 4 presents the results using these models and section 5 concludes and outlines future perspectives.

2. MUSIC FEATURES

Many different features have been proposed to represent music, however, this work only use the so-called AR features due to the good results in music genre classification as reported in [7], where they were first proposed. Calcu-

[*] The author performed the work while at Xerox Research Centre Europe.

[1] The quantity to classify is often a whole song, but could be a sound clip of varying length. In the following, the quantity will simply be called a song.

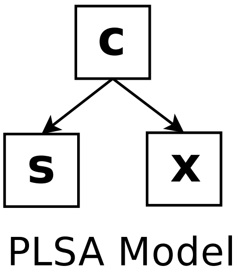

Fig. 1. The graphical model used in Probabilistic Latent Semantic Analysis (PLSA). This is also called the Aspect Model. Squares represent discrete variables.

lating the AR features is a 2-step procedure. First, the mel-frequency cepstral coefficients (MFCC) are calculated on small sound segments (here 30 ms). The MFCC features are very well-known in both speech and music processing, however, they represent only very short sound segments. Thus, the next step is to model the time sequence of the MFCC features individually as AR (auto-regressive) processes and use the AR coefficients as features. Together with the residuals from the AR models and the mean of the MFCCs, this gives the AR features which can now represent much larger sound segments than the MFCCs (here 760 ms).

3. CO-OCCURRENCE MODELS

Co-occurrence models regard a song as a set of co-occurrences (s, x_r) where s denotes the song label and x_r is some feature x at index r in the song. This implies that the song can be modelled directly into the probabilistic model as opposed to previous music genre classification methods.

One advantage of this framework is that a probabilistic measure of $p(c|s)$ can be found, where c denotes the genre label and s is the song index of the new song to be classified. Traditional approaches ([2], [8]) only model $p(c|x_r)$ or $p(x_r|c)$ and combine this information to take a decision for the entire song s. Combination techniques include Majority Voting and the Sum-rule method. With Majority Voting the quantity of interest would be the vote

$$\Delta_r = \arg\max_c p(c|x_r) \qquad r = 1, \ldots, N_r \qquad (1)$$

for each of the N_r time frames in the new song and the genre label of the whole song is chosen as the genre with the most votes. The Sum-rule use the quantity

$$\hat{C} = \arg\max_c \sum_r p(c|x_r) \qquad r = 1, \ldots, N_r \qquad (2)$$

directly as the estimate of the genre label.

3.1. PLSA and Folding-in

The graphical model used in Probabilistic Latent Semantic Analysis (PLSA) is illustrated in fig. 1 and the original formulation will be described in the following. This model is also called the *Aspect Model*. The idea is that a topic c is first chosen with probability $p(c)$. Then a word x_r is generated with probability $p(x_r|c)$ and the document with index s is generated with probability $p(s|c)$. Note that all the variables are discrete and finite and the topic c is seen as a hidden variable. Assuming that co-occurrences are independent, the log-likelihood function for a given training set then becomes :

$$\begin{aligned} L &= \sum_r \log p(x_r, s_{n(r)}) & (3) \\ &= \sum_r \log \sum_c p(s_{n(r)}|c) p(c) p(x_r|c) & (4) \end{aligned}$$

where r runs over all samples/words in all documents and $n(r)$ is a function that assigns the words to the document which they belong to. In the supervised version where the topics of the training set are known, this simply becomes :

$$L = \sum_r \log p(s_{n(r)}|c_{n(r)}) p(c_{n(r)}) p(x_r|c_{n(r)}) \qquad (5)$$

Note that the document index s is in the range $1, \ldots, N_s$, where N_s is the total number of training documents. Thus, to predict the topic of a new document, a new index $N_s + 1$ is used and $p(c|\tilde{s}) \equiv p(c|s = N_s + 1)$ is found by the so-called *Folding-in method*[2] as described in [4]. The idea is to consider $\tilde{s}$ as a hidden variable, which results in the following log-likelihood function :

$$L(\tilde{s}) = \sum_{r=1}^{N_r} \log \left(\sum_{c=1}^{N_c} p(\tilde{s}|c) p(c) p(x_r|c) \right) \qquad (6)$$

where N_r is the number of words in the new document and N_c is the number of topics. All probabilities apart from $p(\tilde{s}|c)$ were estimated in the training phase and are now kept constant. Using the EM algorithm to infer $p(\tilde{s}|c)$, as in [9], results in the following update equations :

$$\begin{aligned} p^{(t)}(c|x_r, \tilde{s}) &= \frac{p^{(t)}(\tilde{s}|c)\ p(c)\ p(x_r|c)}{\sum_{c=1}^{N_c} p^{(t)}(\tilde{s}|c)\ p(c)\ p(x_r|c)} & (7) \\ p^{(t+1)}(\tilde{s}|c) &= \frac{\sum_{r=1}^{N_r}\ p^{(t)}(c|x_r, \tilde{s})}{C_c + \sum_{r=1}^{N_r}\ p^{(t)}(c|x_r, \tilde{s})} & (8) \end{aligned}$$

where C_c is the total number of words in all documents from class c. The quantity $p(c|\tilde{s})$ can now be found using Bayes' rule.

[2] Folding-in refers to folding in the new document into the existing collection of documents

3.2. Discrete vocabulary model

In the discrete word model, a vector quantization was first performed on the AR feature space. This is a method that has been quite successful in e.g. speech recognition together with (discrete) hidden Markov models. Using the training set, a finite code book of code vectors was obtained in analogy to the vocabulary of words for a set of documents. A standard vector quantization method was used, where the code vectors were initially chosen randomly from the training set. Then, iteratively, each vector in the training set was assigned to the cluster with the closest (in Euclidean distance) code vector and the new code vectors were found as the means in each cluster. The stopping criteria was a sufficiently small change in the total MSE distortion measure. Finally, each vector in the test set was given the label (word) of the closest code vector in the code book. Now, having mapped the original multi-dimensional, continuous AR feature space into a finite, discrete vocabulary of sound elements, the supervised version of PLSA model can be applied.

The motivation for the discretisation of the feature space was the analogies between music and language. However, the vocabulary of sound elements has a very different distribution from the distribution of words in documents, which usually follows Zipf's law. This is illustrated in figure 2. Several explanations for this could of course be hypothesized, such as the tendency of vector quantization to cluster vectors evenly, but note also that contrary to e.g. [6], the analyzed music spans a large range of genres. The right mapping of such multifaceted music to a finite vocabulary is a problem that is far from being solved. Adding the fact that the AR feature space is continuous in nature, motivated the development of *continuous vocabulary models*.

3.3. Continuous vocabulary models

These models can be seen as the natural generalization of discrete co-occurrence models like PLSA into the limit where the words become continuous, multidimensional feature vectors. Besides, they can be seen as extensions of well-known probabilistic models to include co-occurrence. Two generative, probabilistic models with considerable success in music genre classification, the Gaussian Classifier (GC) and the Gaussian Mixture Model (GMM), have been augmented to co-occurrence models, which will be named *Aspect Gaussian Classifier* (AGC) and *Aspect Gaussian Mixture Model* (AGMM), respectively[3]. Note that similar ideas are proposed in [11], where a so-called Aspect Hidden Markov Model was developed, and in [12] where an Aspect Bernoulli model was proposed.

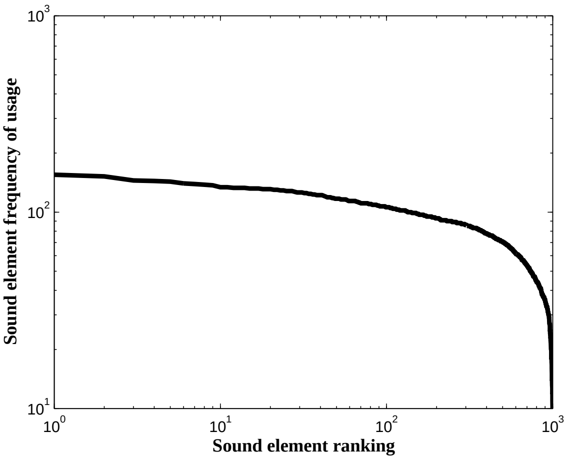

Fig. 2. Frequency of usage of sound elements in the music training set vs. rank of the sound elements (sorted in descending order). The vocabulary of sound elements was found by vector quantization of the AR feature space using 1000 code vectors. A log-log plot is used to test whether Zipf's law applies to the sound elements, in which case the graph should have resembled a straight line with slope approximately -1.

Aspect Gaussian Classifier (AGC)

In figure 3, the graphical models of both the GC and the AGC are illustrated. The log-likelihood function of the AGC becomes :

$$L = \sum_r \log p(s_{n(r)}|c_{n(r)})p(c_{n(r)})p(x_r|c_{n(r)})$$

which seems to be identical to the PLSA equation 5. Note, however, that x_r is now a continuous variable and $p(x|c)$ is a gaussian probability distribution $\mathrm{N}_x(\mu_c, \Sigma_c)$. x_r is the feature vector from time frame r which belongs to the song with index $n(r)$. Additionally, notice that the only difference to the log-likelihood function of the GC is the additional term $p(s_{n(r)}|c_{n(r)})$. Following the maximum likelihood paradigm of parameter inference, the log-likelihood can be maximized directly without resorting to methods like the EM algorithm and the parameter estimates are :

$$\hat{\mu_c} = \frac{1}{N_c}\sum_{r\in C} x_r \tag{9}$$

$$\hat{\Sigma}_c = \frac{1}{N_c}\sum_{r\in C}(x_r - \hat{\mu_c})(x_r - \hat{\mu_c})^T \tag{10}$$

$$\hat{p}(c) = \frac{1}{N_c} \tag{11}$$

$$\hat{p}(s|c) = \frac{N_s}{N_c}\ (\text{if} s \in C, 0 \text{ otherwise}) \tag{12}$$

[3] The word aspect is used with reference to [10], although only supervised training is considered here.

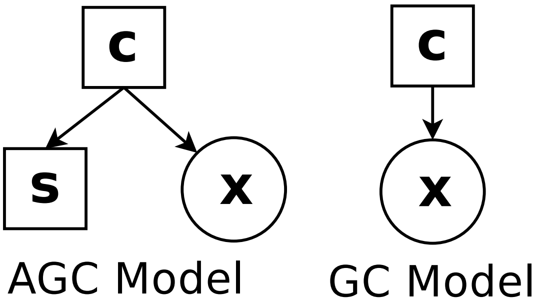

Fig. 3. The graphical models of the Gaussian Classifier (GC) and Aspect Gaussian Classifier (AGC). Round circles represent continuous variables, while squares represent discrete variables.

where N_s and N_c are the total number of time frames in song s and in class c, respectively, and C is the set of time frames from the songs in class c. These estimates are exactly the same as ordinary GC, with the addition of the song probability $p(s|c)$. Given a new song in the testing phase, now requires using the Folding-in method to estimate the probability $p(c|\tilde{s})$, where $\tilde{s}$ is the index of the new song to be folded in. This is done using the update equations in 7 and 8.

Aspect Gaussian Mixture Model (AGMM)

The graphical models of the GMM and the AGMM are shown in figure 4. Now, the log-likelihood function of the AGMM is again similar to the one of the GMM, but with an additional co-occurrence term :

$$L = \sum_r \log \left(p(s_{n(r)}|c_{n(r)}) \sum_{k=1}^{K} p(c_{n(r)}) p(x_r|k) p(k|c_{n(r)}) \right) \tag{13}$$

K denotes the number of components in the model. As for the AGC model, all the parameter estimation equations become the same as in the original GMM model where now the EM algorithm will be used due to the hidden variable k. The probability $p(s_{n(r)}|c)$ again becomes a count of the number of songs in each genre in the training set as in equation 12. The equivalent of equation 6 for the Folding-in procedure, now becomes :

$$L(\tilde{s}) = \sum_{r=1}^{N_r} \log \left(\sum_{c=1}^{N_c} p(\tilde{s}|c) \sum_{k=1}^{K} p(c) p(x_r|k) p(k|c) \right)$$

with update equations :

$$p^{(t)}(c|x_r,\tilde{s}) = \frac{p^{(t)}(\tilde{s}|c) \sum_{k=1}^{K} p(c)\, p(x_r|k) p(k|c)}{\sum_{c=1}^{C_c} p^{(t)}(\tilde{s}|c) \sum_{k=1}^{K} p(c) p(x_r|k) p(k|c)}$$

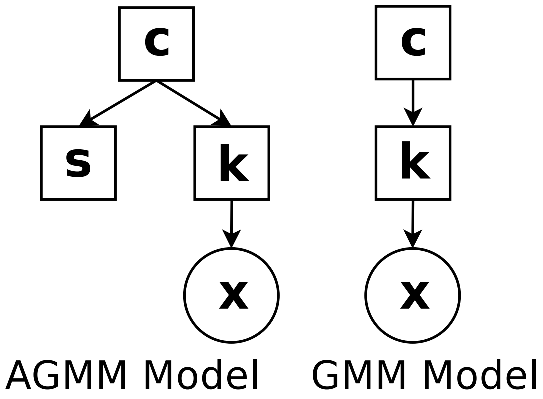

Fig. 4. The graphical models of the Gaussian Mixture Model (GMM) and Aspect Gaussian Mixture Model (AGMM). Round circles represent continuous variables, while squares represent discrete variables.

and

$$p^{(t+1)}(\tilde{s}|c) = \frac{\sum_{r=1}^{N_r} p^{(t)}(c|x_r,\tilde{s})}{C_c + \sum_{r=1}^{N_r} p^{(t)}(c|x_r,\tilde{s})}$$

Note that the only necessary quantity in the E-step is simply the estimate of $p(x_r, c)$ from the training phase for both the AGC and AGMM models. Thus, standard software packages can be used for training both GC and GMM and calculating the estimates of $p(x_r, c)$ for the new song. The Folding-in procedure then becomes a simple extension to this.

Comparing Folding-in and Sum-rule

Looking more carefully at the Folding-in method as described in the last part of section 3.1 reveals a relation to the Sum-rule method in equation 2. It is assumed that the initial guess of $p^{(0)}(\tilde{s}|c)$ in equation 7 is uniform over the classes c and that $p(c)$ is also uniform over classes. This is obviously often not the case, however, in the current music genre classification problem these are reasonable assumptions. It is now seen that the right side of equation 7 simply reduces to the probability $p(c|x_r)$ and the sums on the right side of equation 8 are seen to be simply equal to the sum used in the Sum-rule. Thus, with the mentioned assumptions *the decisions from the Sum-rule are the same as from the first iteration of the Folding-in method*. In this view, the Sum-rule may be seen as an approximation to the full probabilistic model with the Folding-in method.

4. RESULTS AND DISCUSSION

A series of experiments were made to compare the three proposed models (the Discrete Model, the AGC and the

AGMM models) with the GC and GMM models. These two models combined with the Sum-rule method (equation 2) can be seen as good baseline methods [7]. The choice of using the Sum-rule instead of Majority Voting (equation 1), is based on experimental results which show that the Sum-rule consistently performs slightly better than Majority Voting. This is in agreement with the findings in [13].

Data set

The music data set that was used in the experiments consisted of $115 * 11 = 1265$ songs evenly distributed among 11 genres which were “Alternative”, “Country”, “Easy Listening”, “Electronica”, “Jazz”, “Latin”, “Pop and Dance”, “Rap and HipHop”, “R&B and Soul”, “Reggae” and “Rock”. The songs had a sampling frequency of 22050 Hz. From each song, 30 seconds were used from the middle part of the song. The data set is considered difficult to classify with overlap between genres, since a small-scale human evaluation involving 10 people gave a classification error rate with mean 48 % and standard deviation on the mean of 1.6 %. The evaluation involved each person classifying 30 of the sound clips (randomly chosen) on a forced-choice basis.

Feature extraction

The AR features were extracted from the data set along the lines described in section 2. 6 MFCC features were calculated from each frame of size 30 ms and the hopsize between frames was 10 ms. For each of the MFCC features, 3 AR coefficients were found along with the residual and the mean, thus resulting in $6 * 5 = 30$ dimensional AR features. The AR framesize was 760 ms and with a hopsize of 390 ms. Thus, each 30 second song was represented by 80 30-dimensional AR features.

Classification

At first, methods for preprocessing were examined such as whitening and dimension reduction by PCA. However, the classification performance was not significantly affected by the preprocessing. It was decided to normalize each feature dimension individually to avoid numerical problems in the covariance matrix calculations.

The results for all the examined models are shown in figure 5, calculated as described in section 3. The results were found by cross-validation using 80 songs in the training set and 20 in the testing set from each genre. Parameters in the model structure, such as the number of components in the GMM and AGMM models were also found by cross-validation as shown in figure 6. For the continuous models, experiments were made with both diagonal and full covariance matrices in $p(x_r|c)$ and $p(x_r|k)$. Best results were obtained with fairly small numbers of full covariance matrices.

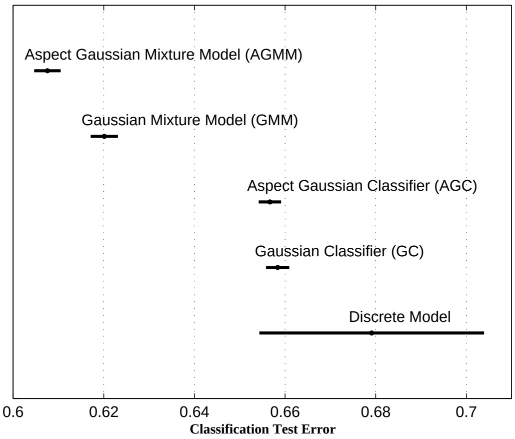

Fig. 5. Classification test error results for the Discrete Model, the Aspect Gaussian Classifier, the Aspect Gaussian Mixture Model and the two baseline methods Gaussian Classifier and Gaussian Mixture Model. The results are the mean values using cross-validation (5-fold for the Discrete Model and 50-fold for the rest) and the error bars are the standard deviations on the means. 7 components were used for the GMM and AGMM.

Note that only similar numbers of mixtures were chosen to represent each genre as seen in figure 6. Better results could possibly be obtained using different numbers of mixtures for the different genres, however, the main focus in the current work has been the comparisons between the baselines and their extensions more than optimizing for performance.

A practical complication was the choice of the vocabulary size in the Discrete Model, since the code book generation was computationally demanding in both space and time due to the large vocabulary size. Experiments were made with sizes in the range of 25 to 2000 code vectors and the test error minimum was found to be around 1000 code vectors.

Discussion

Figure 5 shows that the Discrete Model performs within the range of the GC/AGC models, but it has the added computational processing in the vocabulary creation and mapping parts in the training and test phases, respectively. The testing parts of the AGC and AGMM models are much less computationally demanding which makes them more useful in practical applications. Both of the proposed continuous vocabulary aspect models do better than their baseline counterparts, although it is almost negligible in the case of the AGC as compared to the GC.

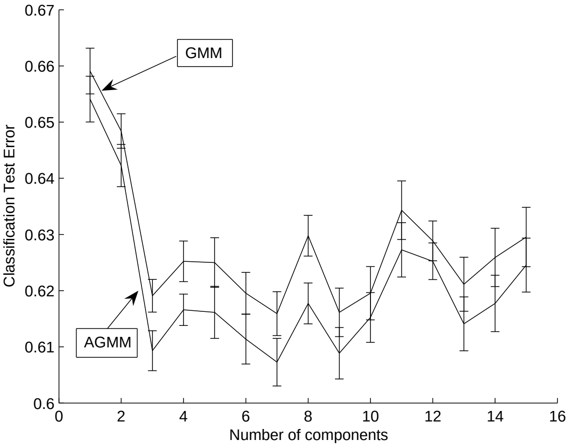

Fig. 6. Classification test error is shown as a function of the number of components in the Gaussian Mixture Model and the Aspect Gaussian Mixture Model. The line illustrates the mean value over 20-fold cross-validation and the error bars show the standard deviation on the mean.

5. CONCLUSION

Three co-occurrence models have been proposed and tested in this work. The first model was the Discrete Model, which was fully based on the PLSA model and used vector quantization to transform the continuous feature space into a finite, discrete vocabulary of sound elements. The two other models, the Aspect Gaussian Classifier and the Aspect Gaussian Mixture Model, were modifications of well-known probabilistic models into co-occurrence models.

The proposed models all have the benefit of modelling the class-conditional probability $p(\tilde{s}|c)$ of the whole song $\tilde{s}$ instead of just modelling short time frames $p(x_r|c)$ as is often the case. This feature of the models could be useful in e.g. music recommendation systems, where only the songs with the highest $p(c|\tilde{s})$ are recommended.

The Discrete Model gave classification test errors in a range comparable to the GC/AGC models, but suffers from the drawback of being demanding in computational time and space due to the vector quantization. The AGC and AGMM models performed slightly better than their baseline counterparts in combination with the Sum-rule method and with a fairly modest increase in computational time.

6. ACKNOWLEDGEMENTS

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

7. REFERENCES

[1] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, July 2002.

[2] M. F. McKinney and J. Breebaart, “Features for audio and music classification,” in *Proceedings of ISMIR*, 2003.

[3] C. Xu, N. C. Maddage, X. Shao, F. Cao, and Q. Tian, “Musical genre classification using support vector machines,” in *Proceedings of ICASSP*, Hong Kong, China, Apr. 2003, pp. 429–432.

[4] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of SIGIR*, Berkeley, CA, 1999, pp. 35–44.

[5] A. D. Patel, “Language, music, syntax and the brain,” *Nature Neuroscience*, vol. 6, no. 7, pp. 674–681, July 2003.

[6] D. H. Zanette, “Zipf’s law and the creation of musical context,” *Musicae Scientiae*, 2005, In Press.

[7] A. Meng, P. Ahrendt, and J. Larsen, “Improving music genre classification using short-time feature integration,” in *Proceedings of ICASSP*, 2005.

[8] P. Ahrendt, A. Meng, and J. Larsen, “Decision time horizon for music genre classification using short-time features,” in *Proceedings of EUSIPCO*, 2004.

[9] E. Gaussier, C. Goutte, K. Popat, and F. Chen, “Hierarchical model for clustering and categorising documents,” in *Proceedings of the 24th BCS-IRSG European Colloqium on IR Research (ECIR-02)*, 2002.

[10] T. Hofmann and J. Puzicha, “Unsupervised learning from dyadic data,” Tech. Rep. TR-98-042, International Computer Science Institute, Berkeley, CA, December 1998.

[11] D. Blei and P. Moreno, “Topic segmentation with an aspect hidden markov model.,” in *Proceedings of the 24th international ACM SIGIR conference.*, 2001, pp. 343–348.

[12] A. Kaban, E. Bingham, and T. Hirsimki, “Learning to read between the lines: The aspect bernoulli model,” in *Proceedings of the 4th SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, April 2004, pp. 462–466.

[13] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, “On combining classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.