MUSIC GENRE CLASSIFICATION BASED ON DYNAMICAL MODELS

Alberto García-Durán[1], Jerónimo Arenas-García[1], Darío García-García[2]
and Emilio Parrado-Hernández[1]
[1]*Dept. of Signal Processing and Communications, Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain*
[2]*Research School for Computer Science, Australian National University, Canberra, Australia*

Keywords: Genre classification, HMMs, Dynamical features, Music retrieval.

Abstract: This paper studies several alternatives to extract dynamical features from hidden Markov Models (HMMs) that are meaningful for music genre supervised classification. Songs are modelled using a three scale approach: a first stage of short term (milliseconds) features, followed by two layers of dynamical models: a multivariate AR that provides mid term (seconds) features for each song followed by an HMM stage that captures long term (song) features shared among similar songs. We study from an empirical point of view which features are relevant for the genre classification task. Experiments on a database including pieces of heavy metal, punk, classical and reggae music illustrate the advantages of each set of features.

1 INTRODUCTION

Automatic music classification (Tzanetakis and Cook, 2002; Meng et al., 2007; Guaus, 2009; Mckinney and Breebaart, 2003) has become a hot topic in the machine learning community due to the recent widespread adoption of personal music repositories and players. Automatic music classification helps users to organize and efficiently browse their growing collections, as well as to discover new music that may result of interest to them. Trivial approaches to music classification rely on metadata associated to each item in the collection, such as composer, performers, style, year, genre and so on. The more elaborated content based approaches, relying on the analysis of musical features extracted from the song waveform, are more suited for the tasks of music discovering and automatic compilation of reproducing lists from examples. Among all the possible criteria to classify music for these purposes, genre based classification is the most used since user preferences are generally identified with some particular musical genres. In this sense, genre is often a subjective and imprecisely defined feature, specially in overlapping cases, such as techno vs. electronic, rock vs. alternative rock, etc. Therefore, content based automatic classification of music can alleviate the need for each user to tedious and carefully label their complete music collection.

Some reasonably successful approaches to genre classification with machine learning discard the time information. They model songs as sets of i.i.d. feature vectors and classify individually each one of these vectors. Finally, the genre with the majority of votes among all its vectors determines each song overall classification (Fu et al., 2011).

Our previous work (García-García et al., 2010) points out that features exploiting the time dynamics of songs through their sequential modeling can result in a significant improvement in the genre classification rate. These features come from the transition matrix induced by each song in a common hidden Markov Model that represents the complete song collection. This paper extends this study in the following directions:

- Analyse the impact of learning the hidden states from a global model trained with songs from all genres or from individual models trained only with songs from a determined genre. This is critical for the trade-off between scalability of the training and accuracy of the final model.
- Study which are the relevant features for the genre classification task: On the one hand, (García-García et al., 2010) shows that genre is captured inside the transitions profile of each song across the hidden states; on the other hand, common bag of features representations (Fu et al., 2011) look at the frequency of permanence in each hidden state.

García-Durán A., Arenas-García J., García-García D. and Parrado-Hernández E. (2012).
MUSIC GENRE CLASSIFICATION BASED ON DYNAMICAL MODELS.
In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pages 250-256
DOI: 10.5220/0003859002500256

The experimental section of the paper gives some insight on the advantages and differences of these two sets of features.

The remainder of the paper is organized as follows: Section 2 briefly reviews some background material including HMMs, used to model the song collection, and Support Vector Machines (SVM), that serve as final classifier. Section 3 describes in detail the genre classification scheme with all the analyzed alternatives. Section 4 illustrates the capabilities of each set of features with some experiments on a real dataset with four genres of different a priori separability: classical, punk, heavy metal and reggae. Finally, Section 5 draws the main conclusions of this work and suggests some lines for future research.

2 BACKGROUND

In this section we briefly review the basic technology for the understanding of the genre classification method. We focus on HMMs, the core of the sequential processing, a recent and probably not very well known metric for sequences based on the transition matrices of HMMs presented in (García-García et al., 2011) and SVMs, the final classifiers.

2.1 Hidden Markov Models

Hidden Markov models (HMMs) (Rabiner, 1989) are a type of parametric, discrete state-space model widely used in applications concerning sequential data. Their main assumptions are the independence of the observations given the hidden states and that these states follow a Markov chain.

Consider a sequence $\mathbf{S}$ of T observation vectors $\mathbf{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$. The HMM assumes that $\mathbf{x}_t$, the t^{th} observation of the sequence, is generated according to the conditional emission density $p(\mathbf{x}_t|q_t)$, with q_t being the hidden state at time t. The state q_t can take values from a discrete set $\{s_1, \ldots, s_K\}$ of size K. The hidden states evolve following a time-homogeneous first-order Markov chain, so that $p(q_t|q_{t-1}, q_{t-2}, \ldots, q_0) = p(q_t|q_{t-1})$.

An HMM is completely defined in terms of the following distributions:

- The initial probabilities vector $\pi = \{\pi_i\}_{i=1}^K$, where $\pi_i = p(q_0 = s_i)$.
- The state transition probability, encoded in a matrix $\mathbf{A} = \{a_{ij}\}_{i,j=1}^K$ with $a_{ij} = p(q_{t+1} = s_j|q_t = s_i)$, $1 \leq i, j \leq K$.
- The emission pdf for each hidden state $p(\mathbf{x}_t|q_t = s_i), 1 \leq i \leq K$.

From these definitions, the likelihood of a sequence $\mathbf{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ can be written in the following factorized way:

$$p(\mathbf{S}|\theta) = \sum_{q_0,\ldots,q_T} \pi_{q_0} p(\mathbf{x}_0|q_0) \prod_{t=1}^{T} p(\mathbf{x}_t|q_t) a_{q_{t-1},q_t}. \quad (1)$$

The training of this kind of models in a maximum likelihood setting is usually accomplished using the Baum-Welch method (Rabiner, 1989), which is a particularization of the well-known EM algorithm. The E-step finds the expected state occupancy and transition probabilities, which can be done efficiently using the forward-backward algorithm (Rabiner, 1989). Then, the M-step updates the parameters in order to maximize the likelihood given the expected hidden states sequence. These two steps are then iterated until convergence. It is worth noting that the likelihood function can have many local maxima, and this algorithm does not guarantee convergence to the global optimum. Due to this, it is common practice to repeat the training several times using different initializations and then select as the correct run the one providing a larger likelihood.

The forward-backward algorithm implies the calculation of both the forward α and backward β variables that are defined as follows:

$$\alpha_k(t) = p(\mathbf{x}_1, \ldots, \mathbf{x}_t, q_t = s_k) \quad (2)$$

$$\beta_k(t) = p(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_T|q_t = s_k). \quad (3)$$

These variables can be obtained in $O(K^2T)$ time through a recursive procedure and can be used to rewrite the likelihood from Eq. (1) in the following manner:

$$p(\mathbf{S}|\theta) = \sum_{k=1}^{K} \alpha_k(t)\beta_k(t), \quad (4)$$

which holds for all values of $t \in \{1, \ldots, T\}$.

Given a previously estimated $\mathbf{A}$, the state transition probabilities can be updated using the forward/backward variables and that previous estimation, yielding:

$$\tilde{a}_{ij} \propto \sum_{t'=1}^{T} \alpha_i(t') a_{ij} p(\mathbf{x}_{t'+1}|q_{t'+1} = s_j)\beta_j(t'+1). \quad (5)$$

2.2 State-space Dynamics metric for sequences

Sometimes the relevant information to be extracted from sequences modeled with HMMs does not rely in how often each state is visited, but in its visiting pattern: which hidden states usually precede each state and what are the most probable next states. The State-Space Dynamics metric (García-García et al., 2011) is aimed at capturing such information.

Let us assume we have an HMM of K states $\Theta = \{\pi, \mathbf{A}, p(\mathbf{x}_t|q_t = s_i)\}$ that models the complete set of training sequences (songs in our case). From the transition matrix $\mathbf{A}$, we obtain, for each particular sequence $\mathbf{S}_n$, an induced transition matrix $\tilde{\mathbf{A}}^n$ by running a single M-step of the Forward-Backward algorithm (equation (5) with $\alpha_i(t')^n$ and $\beta_i(t')^n$ particularized for sequence $\mathbf{S}_n$). The SSD metric expresses the similarity between two sequences $\mathbf{S}_n$ and $\mathbf{S}_m$ as a distance between their induced $\tilde{\mathbf{A}}^n$ and $\tilde{\mathbf{A}}^m$. For this purpose, each row $\mathbf{a}^n_k$ in $\tilde{\mathbf{A}}^n$ is regarded as a discrete probability function of the transitions from the k-th hidden state to the other states in $\mathbf{S}_n$. Therefore, one can compute the similarity between rows corresponding to the same hidden state through any divergence between discrete probabilities. In this paper we adopt the Bhattacharyya affinity (Bhattacharyya, 1943):

$$D_B(\mathbf{a}^n_k, \mathbf{a}^m_k) = \sum_{i=1}^{K} \sqrt{a^n_{ki} a^m_{ki}} \tag{6}$$

The distance between $\tilde{\mathbf{A}}^n$ and $\tilde{\mathbf{A}}^m$ is computed from the mean affinity between their rows as follows:

$$d_{nm} = -\log \frac{1}{K} \sum_{k=1}^{K} D_B(\mathbf{a}^n_k, \mathbf{a}^m_k) \tag{7}$$

This distance can be further transformed into a scale sensitive kernel for songs by exponentiation:

$$\kappa(\mathbf{S}_n, \mathbf{S}_m) = \exp(-\gamma d_{nm}) \tag{8}$$

where γ is a scale parameter that has to be either fixed with domain knowledge or crossvalidated.

2.3 Support Vector Machines

The supervised genre classifier is based on Support Vector Machines (SVM) (Boser et al., 1992) endowed with the kernel matrices that incorporate similarities between sequences. The multiclass classifier is implemented by a pool of one-versus-all binary SVMs (Rifkin and Klautau, 2004), each one trained to discriminate between one of the genres and the rest. Given a kernel function on songs $\kappa(\mathbf{S}_1, \mathbf{S}_2)$, for each genre we wish to construct a scoring function $f_c(\mathbf{S})$ that takes highly positive values (greater than one) when $\mathbf{S}$ is a positive example for genre c and highly negative values otherwise. This scoring function is

$$f_c(\mathbf{S}) = \sum_{i=1}^{l} y^c_i \alpha^c_i \kappa(\mathbf{S}_i, \mathbf{S}) \tag{9}$$

where $\{\mathbf{S}_i, y^c_i\}^l_{i=1}$ are the pairs song/label in the training set. Label $y^c_i \in \{1, -1\}$ marks $\mathbf{S}_i$ as a positive or negative example for genre c. The classifier is then defined by weights α^c_i, that result from the following optimization:

$$\max_{\alpha^c_1,\ldots,\alpha^c_l} \sum_{i=1}^{l} \alpha^c_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha^c_i \alpha^c_j \kappa(\mathbf{S}_i, \mathbf{S}_j)$$

subject to

$$0 \leq \alpha^c_i \leq C \qquad i = 1, \ldots, l \tag{10}$$

where C is a regularization parameter that has to be fixed using prior knowledge or crossvalidated.

After all the scoring functions are determined, the overall classification consists in assigning each song to the genre that achieves the higher value of its scoring function:

3 GENRE CLASSIFICATION SYSTEM BASED ON DYNAMICAL FEATURES

The musical genre classification studied in this work is a generalization of that presented in (García-García et al., 2010). It consists in a multiclass pool of one-vs-all SVMs (see Section 2.3) endowed with a kernel that incorporates dynamic features from the songs relevant for the identification of the genre. We use a three level feature selection to capture such information. The first two levels are based on the song representation of (Meng et al., 2007), that provides with features describing intervals of 1-2 seconds. Then, these subsong features are completed with dynamical features extracted from the HMMs (Section 2.1). This third level actually captures relevant information for the identification of the genre.

3.1 Subsong Level Features

Following (Meng et al., 2007) audio features at two different time levels are extracted from each song:

- **Short-time Feature Extraction.** First, MFCCs are extracted in overlapped windows of short duration. These parameters were originally developed for automatic speech recognition tasks, but they have also been extensively applied in Music Information Retrieval (MIR) tasks (Sigurdsson et al., 2006) with generally good results.

  In this work we follow (Sigurdsson et al., 2006), using a bank with 30 filters, and keeping just the initial 6 coefficients (however, the first coefficient, which is associated to perceptual dimension of loudness, is discarded (Meng et al., 2007)). The window size and hopsize have been fixed to 30 and 15 ms, respectively. Therefore, a fragment of 60 seconds of music is represented by a 4000×6 matrix after MFCC feature extraction.

- **Temporal Feature Integration.** It is well-known that the direct use of MFCCs does not provide an adequate representation for music genre identification. Thus, a time integration process based on a Multivariate Autoregressive (MAR) model (Meng et al., 2007) recovers this more relevant information. For a set of consecutive MFCCs vectors, we fit an MAR model of lag three:

$$\mathbf{z}_j = \sum_{p=1}^{3} \mathbf{B}_p \mathbf{z}_{j-p} + \mathbf{e}_j,$$

where $\mathbf{z}_n$ are the MFCCs extracted at the jth window, $\mathbf{e}_j$ is the prediction error, and $\mathbf{B}_p$ are the model parameters. The values of matrices $\mathbf{B}_p$, $p = 1, \ldots, 3$, together with the mean and covariance of the residuals $\mathbf{e}_j$ are concatenated into a 135×1 single feature vector (MAR vector). For this temporal integration phase, we have considered a window size and hopsize of 2 and 1 seconds, respectively. Thus, an audio fragment of 60 seconds is represented by a matrix of size 60×135 after time integration.

3.2 Song Level Dynamical Features

This section collects the main contributions of this paper. The key component of the classification system is the song level features that determine the information that is fed into the classification stage in the form of a kernel for songs.

Our previous work (García-García et al., 2010) shows that the time evolution of the MAR coefficients is quite relevant to determine the genre. This paper proposes to learn a common HMM with all the training songs and use the SSD metric (Section 2.2) to characterize each song by its transition profile across all the hidden states. Such strategy yields significantly better classification rates than not using the time dynamics information or using the steady state probability of each song visiting each state (computed considering $(\tilde{\mathbf{A}}^n)^\infty$ instead of $\tilde{\mathbf{A}}^n$ as induced transition matrix for song S_n).

From this previous experience we identify two critical elements that determine the quality of the genre classification:

- What is the best way to define the hidden states?
- Which information encoded in the HMMs is actually relevant for the genre discrimination task?

With respect to the first question, (García-García et al., 2010) points out that a single HMM with enough number of hidden states yields a pretty useful set of hidden states. The alternative would be to learn different hidden states for each genre and merge them in the common model. The former guarantees a larger number of examples to train each hidden state, whilst the latter indirectly helps genre discrimination since hidden states learned from different genres will be more separated.

With respect to the second question, this paper proposes a third alternative between the transition profile explored in (García-García et al., 2010) and the stationary frequency of each hidden state: a dynamically computed bag of acoustic words song representation. In this approach, each hidden state is considered as an acoustic feature analogous to the role of words in the bag of words approach to parameterize document collections (Fu et al., 2011). These frequencies are computed dynamically by evolving the song across the common HMM. The bags of words are fed into the SVM through an standard RBF gaussian kernel

$$\kappa(\mathbf{S}_n, \mathbf{S}_m) = \exp(-\gamma \|\mathbf{z}_n - \mathbf{z}_m\|^2)$$

where $\mathbf{z}_n$ and $\mathbf{z}_m$ are the bag of words corresponding to songs $\mathbf{S}_n$ and $\mathbf{S}_m$.

We aim at answering these questions by studying the impact on the genre discrimination accuracy of the following five sets of song level features :

1HMM+SSD. Learn a single HMM with all the songs and use the SSD metric to form the kernel for the SVM. This is the approach of (García-García et al., 2010).

4HMM+SSD. Learn a separate HMM with the training songs of each genre. Merge their states in a single HMM and use all the songs to learn the transition matrices and initial states probabilities. Then form the SVM kernel with the SSD metric in the common HMM (but where the hidden states were learned independently).

1HMM+BoW. Learn a single HMM with all the songs as in (1HMM+SSD) but instead of the SSD metric, use the dynamically computed bag of acoustic words as features for the SVM.

4HMM+BoW. Learn the hidden states separately as in (4HMM+SSD) and replace the SSD metric with the dynamically computed bag of acoustic words.

4HMM+4BoW. Learn one independent complete HMM per genre (i.e. the hidden states will not be shared and the transition probabilities will also be learnt independently per each genre). The bag of acoustic words that is passed to the classification stage results from the concatenation of all the bags of words from all the models. Note that when one learns an independent HMM per genre the kernel based on transition matrices is pointless.

4 EXPERIMENTS

The ability of the song level features presented in Section 3.2 to discriminate musical genre is evaluated in the following classification task. We use a subset of the *garageband* dataset described in (Arenas-García et al., 2007). The data set consists of snippets of 60 seconds of songs downloaded from the online music site `http://www.garageband.com`[1]. The songs are in MP3 format, and belong to different genres. For the experiments we consider a simplified problem where the goal is to discriminate between four different genres: "Punk", "Heavy Metal", "Classical", and "Reggae". The dataset includes genres that are a priori hard to distinguish, like Punk and Heavy Metal plus others which are easily separated. Each genre is represented by a subset of 300 songs. MFCC and MAR extraction settings proceed as described in Section 3.1. For completeness, we have included in the comparison the results of a classifier that assigns each song to the genre whose HMM yields a maximum likelihood (no SVM as final classifier). This baseline classifier is named **4HMM** in the tables.

For each experiment we adopted repeated random sub-sampling validation as our evaluation scheme. The training and testing subsets are composed of 175 and 25 songs, respectively. The hyperparameters γ and C of the SVMs are determined after 5-fold cross validation over the training set. The presented results correspond to the average over 10 different random training/test partitions.

In order to ensure a fair comparison we have chosen the number of hidden states for the HMMs in a way that the resulting hidden states space has the same size. This way, single HMMs trained with all the songs are endowed with 24 hidden states whilst the independent HMMs trained only with songs of a same genre have 6 hidden states. The density functions are spherical covariance gaussians.

Table 1 shows the average accuracy achieved by each feature set together with the standard deviation. The best average performance is obtained by the `4HMM+BoW` features, although it is remarkable the higher stability in terms of small standard deviation obtained by the SSD based features. In fact, the performance of the `1HMM+SSD` features is almost as good in spite of the emission pdfs being learned with all the songs. It seems that the SSD focus on the transition probabilities compensates for the not so discriminative hidden states. The worse performance of `4HMM+4BoW` brings out the advantage of a joint learning of the transition probabilities.

The individual confusion matrices corresponding to each feature set, showed in Tables 2–7 open a more detailed genre-wise discussion. Moreover, Figure 1 shows a Hinton plot of the average occupancy frequency of each hidden state per each genre in the `4HMM` cases. The bigger the rectangle, the more frequently is that state visited by the songs belonging to that genre. States are sorted according to their HMM (states 1–6 come from the Classic, states 7–12 from the Punk, states 13–18 from Reggae and states 19–24 from Heavy HMM). Finally, Figure 2 shows the Hinton plot of average transition matrices for the four genres. The bigger the rectangle in position (i, j), the most probable the transition from state s_i to state s_j is. The states follow the same order as in Figure 1.

Table 1: Comparison among all the strategies on the *garageband* data set with the same experimental setup.

Strategy	Accuracy
4HMM + 4BoW	65.0 ± 3.10 %
1HMM + SSD	75.30 ± 0.04 %
4HMM + BoW	78.0 ± 3.70 %
4HMM + SSD	72.20 ± 0.03 %
1HMM + BoW	71.40 ± 3.40 %
4HMM	69.5 ±3.7%

Table 2: Confusion matrix for 4HMM + 4BoW.

	Classical	Punk	Reggae	Heavy
Classical	0.70	0.12	0.12	0.06
Punk	0.06	0.57	0.01	0.36
Reggae	0.05	0.02	0.79	0.14
Heavy	0.02	0.36	0.08	0.54

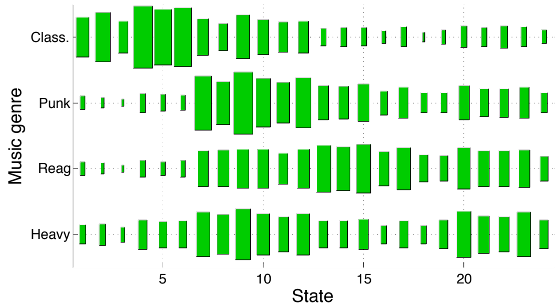

Figure 1: Hinton diagram of the visiting frequency to each state for the songs of each genre.

Classical is the easiest to discriminate genre, regardless of the feature set. Figures 1 and 2 show that this genre takes separate states from the rest. Reggae

[1] Downloaded in November, 2005.

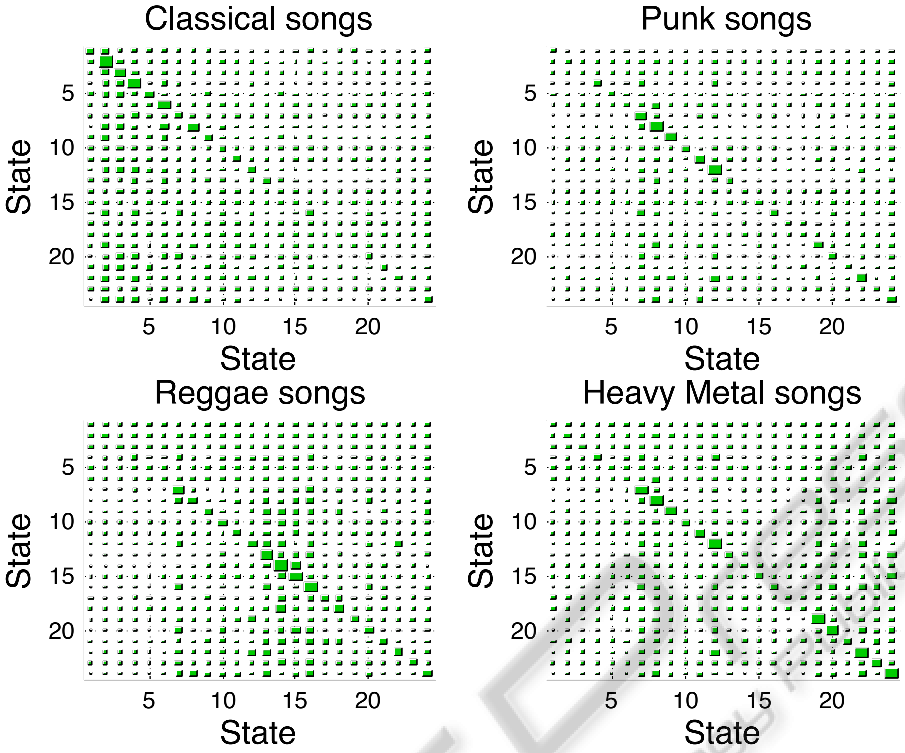

Figure 2: Hinton diagram of the transition matrices for the songs of each genre.

is also easy to discriminate, although there is a higher overlapping with Punk and Heavy states.

With respect to Heavy Metal and Punk, there is a higher overlapping in their bags of words, therefore the independent learning of the hidden states followed by `4HMM` is of certain advantage. In the case of Punk, Figure 1 shows enough spatial separability from the Heavy Metal states, so `4HMM+BoW` yields better performance than `4HMM+SSD`. However, the Heavy Metal transition matrices are more different than the Punk ones, as shown in Figure 2, making the SSD kernel more suited than the BoW for their separation.

Table 3: Confusion matrix for 1HMM + SSD.

	Classical	Punk	Reggae	Heavy
Classical	0.89	0.05	0.05	0.01
Punk	0.04	0.70	0.02	0.24
Reggae	0.04	0.06	0.84	0.06
Heavy	0.03	0.32	0.06	0.59

5 CONCLUSIONS

This paper has studied the suitability to discriminate musical genre of several feature sets extracted from an HMM based dynamical model of a song collection. The best classification rates are obtained when the hidden states of the model are learned independently for each genre but then merged in a single overall HMM where the probabilities of transition be-

Table 4: Confusion matrix for 4HMM + BoW.

	Classical	Punk	Reggae	Heavy
Classical	0.88	0.05	0.06	0.01
Punk	0.04	0.78	0.03	0.15
Reggae	0.04	0.07	0.84	0.05
Heavy	0.01	0.30	0.07	0.62

Table 5: Confusion matrix for 4HMM + SSD.

	Classical	Punk	Reggae	Heavy
Classical	0.82	0.09	0.05	0.04
Punk	0.05	0.67	0.04	0.24
Reggae	0.06	0.10	0.76	0.08
Heavy	0.02	0.26	0.08	0.64

Table 6: Confusion matrix for 1HMM + BoW.

	Classical	Punk	Reggae	Heavy
Classical	0.87	0.04	0.04	0.05
Punk	0.03	0.69	0.04	0.24
Reggae	0.04	0.03	0.80	0.13
Heavy	0.01	0.42	0.08	0.49

Table 7: Confusion matrix for 4HMM.

	Classical	Punk	Reggae	Heavy
Classical	0.71	0.21	0.02	0.06
Punk	0.03	0.82	0.01	0.14
Reggae	0.01	0.22	0.63	0.14
Heavy	0.00	0.38	0.00	0.62

tween any pair of states are more precisely acquired. These probabilities of transition carry relevant information for the genre discrimination task, as pointed out by the good results achieved by the SSD kernel when the states are learned in a common model. In this sense, this information is able to somehow compensate for the lack of a discriminative learning of the hidden states.

Future work will be focused on the extension to more musical genres, to other families of dynamical models different from HMMs. Another intersting line of research consists in the combination of the features related to the frequency of hidden state occupancy and to the dynamics of the transitions between hidden states in a multiple view learning framework.

ACKNOWLEDGEMENTS

This work has been partially supported by the Regional Government of Madrid through grant CCG10-UC3M/TIC-5511 and by the IST Programme of the European Community under the PASCAL2 Network of Excellence IST-2007-216886.

REFERENCES

Arenas-García, J., Parrado-Hernández, E., Meng, A., Hansen, L.-K., and Larsen, J. (2007). Discovering music structure via similarity fusion. In *Music, Brain and Cognition Workshop, NIPS'07*.

Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. Bull. Calcutta Math Soc.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152.

Fu, Z., Lu, G., Ting, K. M., and Zhang, D. (2011). Music classification via the bag-of-features approach. *Pattern Recognition Letters*, 32(14):1768(10).

García-García, D., Arenas-García, J., Parrado-Hernández, E., and de Maria F, D. (2010). Music genre classification using the temporal structure of songs. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, pages 266 –271.

García-García, D., Parrado-Hernández, E., and Diaz-de Maria, F. (2011). State-space dynamics distance for clustering sequential data. *Pattern Recogn.*, 44:1014–1022.

Guaus, E. (2009). *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers*. PhD thesis, Universitat Pompeu Fabra, Spain.

Mckinney, M. and Breebaart, J. (2003). Features for audio and music classification. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 151–158.

Meng, A., Ahrendt, P., Larsen, J., and Hansen, L. (2007). Temporal feature integration for music genre classification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(5):1654 –1664.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, pages 257–286.

Rifkin, R. and Klautau, A. (2004). In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5:101–141.

Sigurdsson, S., Petersen, K. B., and Lehn-Schiler, T. (2006). Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music. In *Proceedings of the Seventh International Conference on Music Information Retrieval (ISMIR)*, pages 286–289.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *Speech and Audio Processing, IEEE Transactions on*, 10(5):293 – 302.