HIDDEN MARKOV MODELS FOR SPECTRAL SIMILARITY OF SONGS

Arthur Flexer[1,2], *Elias Pampalk*[2], *Gerhard Widmer*[2,3]

[1]Institute of Medical Cybernetics and Artificial Intelligence
Center for Brain Research, Medical University of Vienna
Freyung 6/2, A-1010 Vienna, Austria
[2]The Austrian Research Institute for Artificial Intelligence
Freyung 6/6, A-1010 Vienna, Austria
[3]Department of Computational Perception
Johannes Kepler University (JKU) Linz
Altenberger Str. 69, A-4040 Linz, Austria
arthur@ai.univie.ac.at, elias@ofai.at, gerhard.widmer@jku.at

ABSTRACT

Hidden Markov Models (HMM) are compared to Gaussian Mixture Models (GMM) for describing spectral similarity of songs. Contrary to previous work we make a direct comparison based on the log-likelihood of songs given an HMM or GMM. Whereas the direct comparison of log-likelihoods clearly favors HMMs, this advantage in terms of modeling power does not allow for any gain in genre classification accuracy.

1. INTRODUCTION

The general goal of a music information retrieval system can be broken down into two major objectives: the automatic structuring and organization of large collections of digital music, and intelligent music retrieval in such structured "music spaces". To achieve this, a concept of central importance is the notion of musical similarity. Similarity metrics define the inherent structure of a music collection, and the acceptance of a music retrieval system crucially depends on whether the user can recognize some similarity between the query and the retrieved sound files. There are a number of different aspects of music similarity which together influence the perceived similarity between two pieces of music: timbre, rhythm, harmony, melody, to name the most important.

The following approach to music similarity based on spectral similarity pioneered by [Logan & Salomon 2001] and [Aucouturier & Pachet 2002] is now seen as one of the standard approaches in the field of music information retrieval. For a given music collection of S songs, each belonging to one of G music genres, it consists of the following basic steps:

- for each song, divide raw data into overlapping frames of short duration (around $25ms$)
- compute Mel Frequency Cepstrum Coefficients (MFCC) for each frame (up to 20)
- train a Gaussian Mixture Model (GMM, number of mixtures up to 50) for each of the songs
- compute a similarity matrix between all songs using the likelihood of a song given a GMM
- based on the genre information, do k-nearest neighbor classification using the similarity matrix

The last step of genre classification can be seen as a form of evaluation. Since usually no ground truth with respect to music similarity exists, each song is labeled as belonging to a music genre using e.g. music expert advice. High genre classification results indicate good similarity measures. The winning entry to the ISMIR 2004 genre classification contest[1] by Elias Pampalk followed basically the above described approach.

This approach based on GMMs disregards the temporal order of the frames, i.e. to the algorithm it makes no difference whether the frames in a song are ordered in time or whether this order is completely reversed or scrambled. Research on perception of musical timbre of single musical instruments clearly shows that temporal aspects of the audio signals play a crucial role (see e.g. [Grey 1977]). Aspects like spectral fluctuation, attack or decay of an event cannot be modelled without respecting the temporal order of the audio signals.

A natural way to incorporate temporal context into the above described framework is the usage of Hidden Markov Models (HMM) instead of GMMs. HMMs trained on MFCCs have already been used for music summarization ([Logan & Chu 2000], [Aucouturier & Sandler 2001], [Peeters et al. 2002]) and genre classification [Aucouturier & Pachet 2004] but with rather limited success. This paper describes experiments using HMMs to compute similarity between songs based on spectral information. The results are compared to GMMs using goodness-of-fit criteria (log-likelihoods) between songs and models as well as genre classification for evaluation. Whereas the direct comparison of log-likelihoods clearly favors HMMs, this advantage in terms of modeling power does not allow for any gain in genre classification accuracy. Only by directly looking at the goodness-of-fit of the models the possible benefit of using HMMs for music analysis becomes apparant. After introducing the data base used in the study as well as the employed preprocessing (Sec. 2), we will describe the methods of GMMs and HMMs (Sec. 3), present our experiments and results (Sec. 4) which is followed by discussion (Sec. 5) and conclusion (Sec. 6).

[1]ISMIR 2004, 5th International Conference on Music Information Retrieval, Audiovisual Institute, Universitat Pompeu Fabra Barcelona, Spain, October 10-14, 2004; see http://ismir2004.ismir.net/ISMIRContest.html.

2. DATA

For our experiments we used the data set of the ISMIR 2004 genre classification contest[2]. The data base consist of $S = 729$ songs belonging to $G = 6$ genres. The different genres plus the numbers of songs belonging to each genre are given in Table 1.

Table 1: ISMIR 2004 contest data base (Genre, number of songs, percentage).

Genre	No.	%
Classical	320	43.9
Electronic	115	15.8
Jazz Blues	26	3.6
Metal Punk	45	6.2
Pop Rock	101	13.9
World	122	16.7
Sum	729	100.0

We divide the raw audio data into overlapping frames of short duration and use Mel Frequency Cepstrum Coefficients (MFCC) to represent the spectrum of each frame. MFCCs are a perceptually meaningful and spectrally smoothed representation of audio signals. MFCCs are now a standard technique for computation of spectral similarity in music analysis (see e.g. [Logan 2000]). The frame size for computation of MFCCs for our experiments was $23.2ms$ (512 samples), with a hop-size of $11.6ms$ (256 samples) for the overlap of frames. Although improved results have been reported with numbers of MFCCs of up to 20 [Aucouturier & Pachet 2004], we used only the first 8 MFCCs for all our experiments to limit the computational burden.

In order to allow modeling of a bigger temporal context we also used so-called texture windows [Tzanetakis & Cook 2002]: we computed means and variances of MFCCs across the following numbers of frames and used them as alternative input to the models: 22 frames, hop-size 11 ($510.4ms$, $255.2ms$), 10 frames, hop-size 5 ($232ms$, $116ms$), 10 frames, hop-size 2 ($232ms$, $46.4ms$). This means that if a texture window is being used, after preprocessing a single data point x^t is a 16-dimensional vector (8 mean MFCCs plus 8 variances across MFCCs) instead of a 8-dimensional vector if no texture window is used.

3. METHODS

A Gaussian Mixture Model (GMM) models the density of the input data by a mixture model of the form

$$p^{GMM}(x) = \sum_{m=1}^{M} P_m \mathcal{N}[x, \mu_m, U_m] \tag{1}$$

where P_m is the mixture coefficient for the m-th mixture, $\mathcal{N}$ is the normal density and μ_m and U_m are the mean vector and covariance matrix of the m-th mixture. The log-likelihood function is given by

$$L^{GMM} = \frac{1}{T} \sum_{t=1}^{T} log(p^{GMM}(x^t)) \tag{2}$$

for a data set containing T data points. This function is maximized both with respect to the mixing coefficients P_m and with respect to the parameters of the Gaussian basis functions using Expectation-Maximization (see e.g. [Bishop 1995]).

Hidden Markov Models (HMM) [Rabiner & Juang 1986] allow analysis of non-stationary multi-variate time series by modeling both the probability density functions of locally stationary multi-variate data and the transition probabilities between these stable states. If the probability density functions are modelled with mixtures of Gaussians, HMMs can be seen as GMMs plus transition probabilities. An HMM can be characterized as having a finite number N of states Q:

$$Q = \{q_1, q_2, \ldots, q_N\} \tag{3}$$

A new state q_j is entered based upon a transition probability distribution A which depends on the previous state (the Markovian property):

$$A = \{a_{ij}\}, a_{ij} = P(q_j(t) \mid q_i(t-1)) \tag{4}$$

where $t = 1, \ldots, T$ is a time index with T being the length of the observation sequence. After each transition an observation output symbol is produced according to a probability distribution B which depends on the current state. Although the classical HMM uses a set of discrete symbols as observation output, [Rabiner & Juang 1986] already discuss the extension to continuous observation symbols. We use a Gaussian Observation Hidden Markov Model (GOHMM) where the observation symbol probability distribution for state j is given by a mixture of Gaussians:

$$B = \{b_j(x)\}, b_j(x) = p_j^{GMM}(x) \tag{5}$$

where $p_j^{GMM}(x)$ is the density as defined for a mixture of Gaussians in Equ. 1.

The Expectation-Maximization (EM) algorithm is used to train the GOHMM thereby estimating the parameter sets A and B. The log-likelihood function is given by

$$L^{HMM} = \frac{1}{T} \sum_{t=1}^{T} log(b_{q_t}(x^t)) + log(a_{q_{t,t-1}}) \tag{6}$$

for an observation sequence of length $t = 1, \ldots, T$ with $q_1, \ldots, q_T$ being the most likely state sequence and q_0 a start state. The forward algorithm is used to identify most likely state sequences corresponding to a particular time series and enables the computation of the log-likelihoods. Full details of the algorithms can be found in [Rabiner & Juang 1986].

It is informative to have a closer look at how the transition probabilities influence the state sequence characteristics. The inherent duration probability density $p_i(d)$ associated with state q_i, with self transition coefficient a_{ii} is of the form

$$p_i(d) = (a_{ii})^{d-1}(1 - a_{ii}) \tag{7}$$

This is the probability of d consecutive observations in state q_i, i.e. the duration probability of staying d times in one of the locally stationary states modeled with a mixture of Gaussians. As [Rabiner 1989] noted, this exponential state duration density is not optimal for a lot of physical signals. The duration of a single data point in our case is dependent on the window length win of the frame used for computing the MFCCs or the size of the texture window as well as the hop size hop. The length l of staying in the same state expressed in ms is then:

[2] To be more precise, we used the training set of the contest.

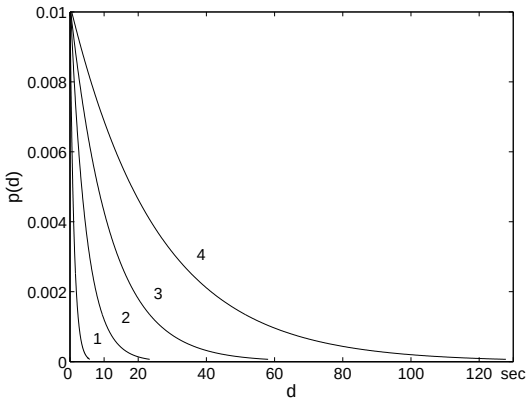

Figure 1: Duration probability densities $p(d)$ (y-axis) for durations d (x-axis) in seconds for different combinations of window and hop sizes: line (1) win $23.2ms$, hop $11.6ms$, line (2) win $232ms$, hop $46.4ms$, line (3) win $232ms$, hop $116ms$, line (4) win $510.4ms$, hop $255.2ms$.

$$l = (d-1)hop + win \tag{8}$$

with hop and win given in ms. Fig. 1 gives duration probability densities for all different combinations of hop and win used for preprocessing in Sec. 2 with a_{ii} set to .99 (which is a reasonable choice for audio data). One can see that whereas for $hop = 11.6$ and $win = 23.2$ the duration probability at five seconds is already almost zero, there still is an albeit small probability for durations up to 120 seconds for $hop = 255.2$ and $win = 510.4$. Our choice of different frame sizes and texture windows seems to guarantee a range of different duration probabilities. The shorter the state durations in HMMs are, the more often the state sequence will switch from state to state and the less clear the boundaries between the mixture of Gaussians of the individual states will be. Therefore, with shorter state durations the HMMs will be more akin to GMMs in their modeling behavior.

An important open issue is the model topology of the HMM. Looking again at the work by [Rabiner & Juang 1986] on speech analysis, we can see that the standard model for isolated word recognition is a left-to-right HMM. No transitions are allowed to states whose indices are lower than the current state, i.e. as time increases the state index increases. This has been found to account well for modeling of words which rarely have repeating vowels or sounds. For songs, a fully connected so-called ergodic HMM seems to be more suitable for modeling than the constrained left-to-right model. After all, repeating patterns seem to be an integral part of music. Therefore it makes sense to allow states to be entered more than once and hence use ergodic HMMs.

There is a small number of papers describing applications of HMMs to the modeling of some form of spectral similarity. [Logan & Chu 2000] compare HMMs and static clustering for music summarization. Fully ergodic HMMs with five to twelve states of single Gaussians are trained on the first 13 MFCCs (computed from $25.6ms$ overlapping windows). Key phrases are chosen based on state frequencies and evaluated in a user study. Clustering performs best and HMMs do not even surpass the performance of a random algorithm. [Aucouturier & Sandler 2001] use fully ergodic three state HMMs with single Gaussians per state trained on the first ten MFCCs (computed from $30ms$ overlapping windows) for segmentation of songs into chorus, verse, etc. The authors found little improvement over using static k-means clustering for the problem. The same approach is used as part of a bigger system for audio thumb-nailing in [Aucouturier & Sandler 2002]. [Peeters et al. 2002] also compare HMMs and k-means clustering for music audio summary generation. The authors report about achieving smoother state jumps using HMMs.

[Aucouturier & Pachet 2004] report about genre classification experiments using HMMs with numbers of states ranging from 3 to 30 where the states are mixtures of four Gaussians. For their genre classification task the best HMM is the one with 12 states. Its performance is slightly worse than that of a GMM with a mixture of 50. The authors do not give any detail about the topology of the HMM, i.e. whether it is a fully ergodic one or one with left-to-right topology. It is also unclear whether they use full covariance matrices for the mixtures of Gaussians. From the graph in their paper (Figure 6) it is evident that HMMs with numbers of states ranging from 4 to 25 perform at a very comparable level in terms of genre classification accuracy.

HMMs have also been used successfully for audio fingerprinting (see e.g. [Batlle et al. 2003]). There HMMs with tailor made topologies trained on MFCCs are used to fully represent each detail of a song in a huge database. The emphasis is on exact identification of a specific song and not on generalization to songs with similar characteristics.

4. RESULTS

For our experiments with GMMs and HMMs we used the following parameters (abbreviations correspond to those used in Table 2):

- **preprocessing:** we used combinations of window (*win*) and hop sizes (*hop*) and texture windows (*tex* set to yes (*'y'*) or no (*'n'*)) as described in Sec. 2
- **topology:** 3, 6 and 10 state ergodic (fully connected) HMMs with mixtures of 1, 3 or 5 Gaussians per state, GMMs with mixtures of 9, 10 or 30 Gaussians (see *states* and *mix* in Table 2 for combinations used); Gaussians use diagonal covariance matrices for HMMs and GMMs
- **computation of similarity:** similarity is computed using Equ. 6 for HMMs and Equ. 2 for GMMs

The combinations of parameters *states*, *mix*, *win*, *hop* and *tex* used for this study yielded twelve different model classes: six types of HMMs and six types of GMMs. We made sure to employ comparable types of GMMs and HMMs by having comparable degrees of freedom for pairs of model classes: HMM (*states* 10, *mix* 1) vs. GMM (*mix* 10), HMM (*states* 3, *mix* 3) vs. GMM (*mix* 9), HMM (*states* 6, *mix* 5) vs. GMM (*mix* 30). The degrees of freedom (number of free parameters) for HMMs and GMMs are

$$df^{GMM} = mix \times 2 \times dim(x) \tag{9}$$

$$df^{HMM} = states \times mix \times 2 \times dim(x) + states^2 \tag{10}$$

with $dim(x)$ being the dimensionality of the input vectors (see Sec. 2). Column df in Table 2 gives the degrees of freedom for all types of models. With the first column nr indexing the different models, odd numbered models are always HMMs and the

Table 2: Overview of all types of models used and results achieved: index of model *nr*, model type *model*, number of states *states*, size of mixture *mix*, window size *win*, hop size *hop*, texture window *tex*, degrees of freedom *df*, mean log-likelihood *likeli*, number of HMM based log-likelihoods bigger than GMM based log-likelihoods $H > G$, z-statistic *z*, mean accuracy *acc*, standard deviation *stddev*, t-statistic *t*.

nr	model	states	mix	win	hop	tex	df	likeli	$H > G$	z	acc	stddev	t
1	HMM	10	1	23.2	11.6	n	260	-31.10	22	-24.43	74.20	5.43	-2.33
2	GMM	-	10	23.2	11.6	n	160	-29.89			76.54	3.64	
3	HMM	3	3	23.2	11.6	n	153	-29.26	698	24.76	77.08	4.73	3.20
4	GMM	-	9	23.2	11.6	n	144	-29.91			73.38	5.00	
5	HMM	6	5	23.2	11.6	n	516	-28.95	706	25.46	78.18	4.59	-0.01
6	GMM	-	30	23.2	11.6	n	480	-29.93			78.19	3.32	
7	HMM	3	3	510.4	255.2	y	297	-29.31	692	24.26	74.20	4.85	-0.43
8	GMM	-	9	510.4	255.2	y	288	-29.92			74.62	3.67	
9	HMM	3	3	232.0	116.0	y	297	-29.30	690	24.11	76.67	2.22	0.71
10	GMM	-	9	232.0	116.0	y	288	-29.90			76.26	3.13	
11	HMM	3	3	232.0	46.4	y	297	-29.34	677	23.13	73.79	4.81	-0.36
12	GMM	-	9	232.0	46.4	y	288	-29.89			74.20	3.27	

next even numbered model is always the associated GMM. The difference in degrees of freedom between two associated types of GMMs and HMMs is always the number of transition probabilities ($states^2$).

4.1. Comparing log-likelihoods directly

The first line of experiments compares goodness-of-fit criteria (log-likelihoods) between songs and models in order to explore which type of model best describes the data. Out-of-sample log-likelihoods were computed in the following way:

- train HMMs and GMMs for each of the twelve model types for each of the songs in the training set, using only the first half of each song
- use the second half of each song to compute log-likelihoods L^{HMM} and L^{GMM}

This yielded $S = 729$ log-likelihoods for each of the twelve model types. Average log-likelihoods per model type are given in column *likeli* in Table 2. Since the absolute values of log-likelihoods very much depend on the type of songs used, it is much more informative to compare log-likelihoods on a song-by-song basis. In Fig. 2 histogram plots of the differences of log-likelihoods $L_i - L_{i+1}$ between associated model types are shown:

$$L_i - L_{i+1} = L^{HMM(i)} - L^{GMM(i+1)} \tag{11}$$

with $HMM(i)$ being an HMM of model type index $nr = i$ and $GMM(i+1)$ being the associated GMM of model type index $nr = i + 1$ and $i = 1, 3, 5, 7, 9, 11$. The differences $L_i - L_{i+1}$ are computed for all the $S = 729$ songs before doing the histogram plots. As can be seen in Fig. 2, except for one histogram plot the majority of HMM models show a better goodness-of-fit of the data than their associated GMMs (i.e. their log-likelihoods are higher for most of the songs). The only exception is the comparison of model types 1 and 2 (HMM (*states* 10, *mix* 1) vs. GMM (*mix* 10)) which is interesting because in this case the HMMs have the biggest advantage in terms of degrees of freedom (180 vs. 80) over the GMMs of all the comparisons. This is due to the fact that this type of HMM models has the highest number of states with $states = 10$. But it also has only a single Gaussian per state to model probability density functions. Experiments on isolated word recognition in speech analysis [Rabiner & Juang 1986] have shown that small sizes of the mixtures of Gaussians used in HMMs do not catch the full detail of the emission probabilities which often are not Gaussian at all. Mixtures of five Gaussians with diagonal covariances per state have been found to be a good choice.

Finding a correct statistical test for comparing likelihoods of so-called non-nested models is far from trivial (see e.g. [McAleer 1995] or [Golden 2000]). HMMs and GMMs are non-nested models because one is not just a subset of the other as would e.g. be the case with a mixture of five Gaussians compared to a mixture of six Gaussians. What makes the models non-nested is the fact that it is not clear how to weigh the parameter of a transition probability a_{ij} against, say, a mean μ_m of a Gaussian. Nevertheless, it is correct to compare the log-likelihoods since we use out-of-sample estimates, which automatically punishes over-fitting due to excessive free parameters. It is just the distribution characteristics of the log-likelihoods which are hard to describe. Therefore we resorted to the distribution free sign test which relies only on the rank of results (see e.g. [Siegel 1956]). Let C_I be the score under condition I and C_{II} the score under condition II then the null hypothesis tested by the sign test is

$$H_0 : p(C_I > C_{II}) = p(C_I < C_{II}) = \frac{1}{2} \tag{12}$$

In our case the two scores C_I and C_{II} are the matched pairs of log-likelihoods for a song given associated models HMM_I and GMM_{II}. If c is the number of times that $C_I > C_{II}$ and the number of matched pairs N is greater than 25 then the sampling distribution is the normal distribution with

$$z = \frac{c - \frac{1}{2}N}{\frac{1}{2}\sqrt{N}} \tag{13}$$

Column $H > G$ in Table 2 gives the count c of HMM based log-likelihoods being bigger than GMM based log-likelihoods for all pairs of associated model types. Column z gives the corresponding z-values obtained using Equ. 13. All z-values are highly

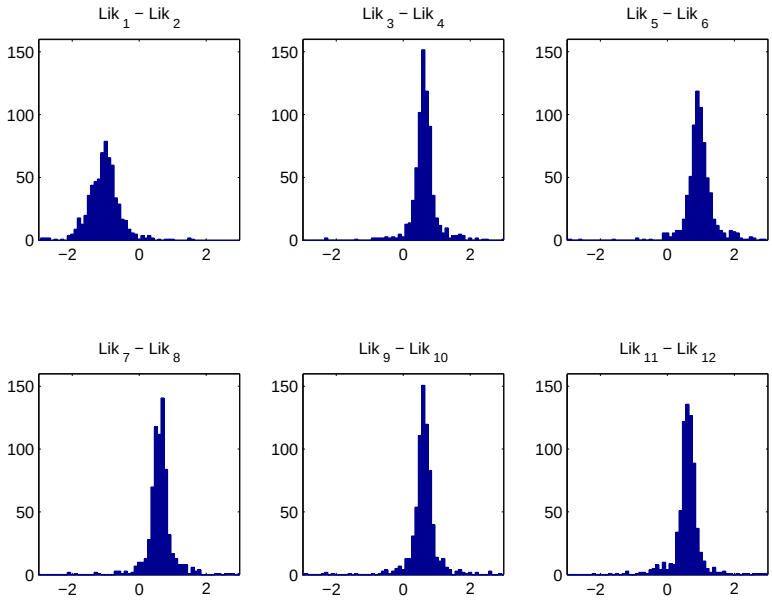

Figure 2: Histogram plots of differences in log-likelihood between associated models.

significant at the 99% error level since all $|z| > z_{99} = 2.58$. Therefore HMMs always better describe the data compared to their associated GMMs with the exception of the comparison of model types 1 and 2 (HMM (*states* 10, *mix* 1) vs. GMM (*mix* 10)).

To counter the argument that the superior performance of the HMMs is due to their extra number of degrees of freedom (i.e. number of transition probabilities, see column *df* in Table 2) we also compared the smallest type of HMMs (model nr 3: HMM (*states* 3, *mix* 3), *df = 153*) with the biggest type of GMMs (model nr 6: GMM (*mix* 30), *df = 480*). This comparison yielded a count c ($H > G$) of 635, and a z-value of $z = 20.14 > z_{99} = 2.58$ again being highly significant. We conclude that it is not the sheer number of degrees of freedom in the models but the quality of the free parameters which decides which type of model better fits the data. After all, the degrees of freedom of the HMMs in our last comparison are outnumbered three times by those of the GMMs.

4.2. Genre Classification

The second line of experiments compares genre classification results. In a 10-fold cross validation we did the following:

- train HMMs and GMMs for each of the twelve model types for each of the songs in the training set (the nine training folds), this time using the complete songs
- for each of the model types, compute a similarity matrix between all songs using the log-likelihood of a song given a HMM or a GMM (L^{HMM} and L^{GMM})
- based on the genre information, do one-nearest neighbor classification for all songs in the test fold using the similarity matrices

Average accuracies and standard deviations across the ten folds of the cross validation are given in columns *acc* and *stddev* in Table 2. Looking at the results one can see that the achieved accuracies range from around 73% to around 78% with standard deviations of up to 5%. We compared accuracy results of associated model types in a series of paired t-tests (model *nr* 1 vs. *nr* 2, . . ., *nr* 11 vs. *nr* 12). The resulting t-values are given in column *t* in Table 2. All t-values are not significant at the 99% error level since all $|t| < t_{(99,df=9)} = 3.25$. Peak performances are reached with model types *nr* 5, HMM (*states* 6, *mix* 5), and *nr* 6, GMM (*mix* 30), with almost identical accuracies of 78.18% and 78.19%. We therefore conclude that there is no systematic difference in genre classification performance between HMMs and GMMs.

5. DISCUSSION

There are two main results of our work:

(i) HMMs better describe spectral similarity of songs than the standard technique of GMMs. Comparison of log-likelihoods clearly shows that HMMs allow for a better fit of the data. This holds not only if looking at competing models with comparable numbers of degrees of freedom but also for GMMs with numbers of parameters that are much larger than of those of the HMMs. The only outlier in this respect is model type 1 (HMM (*states* 10, *mix* 1)). But as discussed in Sec. 4 this is probably due to the poor choice of single Gaussians for modeling the emission probabilities.

(ii) HMMs perform at the same level as GMMs when used for spectral similarity based genre classification. There is no significant gain in terms of classification accuracy. Genre classification is of course a rather indirect way of measuring differences between alternative similarity models. The human error in classifying some of the songs gives rise to a certain percentage of misclassification already. Inter-rater reliability between a number of music experts is far from perfect for genre classification.

Although we believe this work is the most comprehensive study on using HMMs for spectral similarity of songs so far, there is of course a lot still to be done. Two possible routes for further improvements come to mind: the topology of the HMMs and the handling of the state duration. Choosing a topology for an HMM

still is more of an art than a science (see e.g. [Durbin et al. 1998] for a discussion). Our limited set of examined combinations of numbers of states and sizes of mixtures could be extended. One should however notice that too large numbers for these parameters quickly lead to numerical problems due to insufficient training data. We also have not yet tried out left-to-right models.

With our choice of different frame sizes and texture windows we tried to explore a range of different state duration densities. There are of course a number of alternative and possibly more principled ways of doing this. The usage of so-called explicit state duration modeling could be explored. A duration parameter d per HMM state is added. Upon entering a state q_i a duration d_i is chosen according to a state duration density $p(d_i)$. Formulas are given in [Rabiner & Juang 1986]. Another idea is to use an array of n states with identical self transition probabilities where it is enforced to pass each state at least once. This gives rise to more flexible so-called Erlang duration density distributions (see [Durbin et al. 1998]).

An altogether different approach of representing the dynamical nature of audio signals is the computation of dynamic features by substituting the MFCCs with features that already code some temporal information (e.g. autocorrelation or reflection coefficients). Examples can be found in [Rabiner & Juang 1986].

Some of these ideas might be able to further improve the modeling of songs by HMMs but it is not clear whether this will also help the genre classification performance.

6. CONCLUSION

We were able to show by comparison of log-likelihoods that HMMs better describe the spectral similarity of songs than the standard technique of GMMs. This advantage in terms of modeling power does not buy any gain in accuracy when HMMs instead of GMMs are used for genre classification. These two results together seem to explain why so far in the literature little success in using HMMs for music analysis based on spectral similarity has been reported. Evaluation criteria reported before were rather indirect means of measurement.

7. ACKNOWLEDGEMENTS

Parts of the MA Toolbox [Pampalk 2004], the Netlab Toolbox[3] and the Hidden Markov Model Toolbox by Kevin Murphy[4] have been used for this work. This research was supported by the EU project FP6-507142 SIMAC[5]. The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Education, Science and Culture and the Austrian Federal Ministry for Transport, Innovation and Technology.

8. REFERENCES

[Aucouturier & Pachet 2002] Aucouturier J.-J., Pachet F.: Music Similarity Measures: What's the Use?, in Proceedings of the Third International Conference on Music Information Retrieval (ISMIR'02), pp. 157-163,IRCAM, 2002.

[Aucouturier & Pachet 2004] Aucouturier, J.-J., Pachet F.: Improving Timbre Similarity: How high is the sky?, Journal of Negative Results in Speech and Audio Sciences, 1(1), 2004.

[Aucouturier & Sandler 2001] Aucouturier, J.-J. and Sandler, M. Segmentation of Musical Signals Using Hidden Markov Models. Proceedings of the Audio Engineering Society 110th Convention, Amsterdam, May 12-15, 2001.

[Aucouturier & Sandler 2002] Aucouturier, J.-J. and Sandler, M. Finding repeating patterns in acoustic musical signals, in Proceedigs of the AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio, 2002.

[Batlle et al. 2003] Batlle E., Masip J., Cano M.: System Analysis and Performance Tuning for Broadcast Audio Fingerprinting, In Proc. of the 6th Int. Conference on Digital Audio Effects (DAFX-03), London, Uk, September 8-11, 2003.

[Bishop 1995] Bishop C.M.: Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

[Durbin et al. 1998] Durbin R., Eddy S., Krogh A., Mitchison G.: Biological sequence analysis, Cambridge Univ. Press, 1998.

[Golden 2000] Golden R.M.: Statistical Tests for Comparing Possibly Misspecified and Nonnested Models, Journal of Mathematical Psychology, 44, 153-170, 2000.

[Grey 1977] Grey J.M.: Multidimensional perceptual scaling of musical timbres, Journal of the Acoustical Society of America, Vol.61, No.5, pp.1270-1277, 1977.

[Logan 2000] Logan B.: Mel Frequency Cepstral Coefficients for Music ModelingMel Frequency Cepstral Coefficients for Music Modeling, Proceedings of the International Symposium on Music Information Retrieval (ISMIR'00), 2000.

[Logan & Chu 2000] Logan B., Chu S.: Music Summarization Using Key Phrases, in Proc. of the Intern. Conf. on Acoustics, Speech and Signal Processing, pp. II-749-752, 2000.

[Logan & Salomon 2001] Logan B., Salomon A.: A music similarity function based on signal analysis, IEEE International Conference on Multimedia and Expo, Tokio, Japan, 2001.

[McAleer 1995] McAleer M.: The significance of testing empirical non-nested models, Journal of Econometrics, 67, 149-171, 1995.

[Pampalk 2004] Pampalk E.: A Matlab Toolbox to compute music similarity from audio, in Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04), Universitat Pompeu Fabra, Barcelona, Spain, pp.254-257,2004.

[Peeters et al. 2002] Peeters G., La Burthe A., Rodet X.: Toward Automatic Music Audio Summary Generation from Signal Analysis, in Proceedings of the Third International Conference on Music Information Retrieval (ISMIR'02), pp. 157-163,IRCAM, 2002.

[Rabiner 1989] Rabiner L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, Vol.77, No. 2, p.257-285, 1989.

[Rabiner & Juang 1986] Rabiner L.R., Juang B.H.: An Introduction To Hidden Markov Models, IEEE ASSP Magazine, 3(1):4-16, 1986.

[Siegel 1956] Siegel S.: Nonparametric Statistics for the Behavioral Sciences, McGraw-Hill, 1956.

[Tzanetakis & Cook 2002] Tzanetakis G., Cook P.: Musical genre classification of audio signals, IEEE Trans. on Speech and Audio Processing, Vol. 10, Issue 5, 293-302, 2002.

[3] `http://www.ncrg.aston.ac.uk/netlab`

[4] `http://www.ai.mit.edu/~murphyk/Software/hmm.html`

[5] `http://www.semanticaudio.org`