VARIABLE-SIZE GAUSSIAN MIXTURE MODELS FOR MUSIC SIMILARITY MEASURES

Wietse Balkema
Robert Bosch GmbH - Corporate Research
Po. Box 77 77 77
31137 Hildesheim, Germany

ABSTRACT

An algorithm to efficiently determine an appropriate number of components for a Gaussian mixture model is presented. For determining the optimal model complexity we do not use a classical iterative procedure, but use the strong correlation between a simple clustering method (BSAS [13]) and an MDL-based method [6]. This approach is computationally efficient and prevents the model from representing statistically irrelevant data.

The performance of these variable size mixture models is evaluated with respect to hub occurrences, genre classification and computational complexity. Our variable size modelling approach marginally reduces the number of hubs, yields 3-4% better genre classification precision and is approximately 40% less computationally expensive.

1 INTRODUCTION

In the current boom of web 2.0, the market for music recommender systems seems to have taken off. Last.fm, iLike, myStrands and others analyze a user's listening behaviour and compare it with other user's profiles. Music can be tagged with personal tags which allows new ways to explore music collections.

One of the major problems of these community based recommender systems is their robustness in new databases and dealing with underrepresented data. Once a song is chosen as favourite, a loop mechanism can keep this song as favourite for a long time. Community based recommenders can be sensitive to attacks that try to influence a specific song's rating. Two kinds of attack strategies are *shilling* (promoting an item) and *nuking* (demoting an item) [8].

Another approach for recommender systems that do not suffer from loops, shilling or nuking is content-based recommendation. The acoustical content of a song is analyzed, and songs found to be 'similar' to songs a user likes are recommended. Audio similarity is multifaceted, so a common approach to evaluate audio similarity measures is to perform a genre classification task. Pampalk et al. [11] found that a combination of 70% timbral and 30% temporal features provide a good audio similarity measure.

Hubs, songs that are found to be very similar to a very large number of other songs, are a major problem for audio-based music recommender systems. Aucouturier and Pachet [3] showed that in a purely timbre-based nearest neighbor retrieval system, the number of hubs significantly increases when discarding the 5% least significant clusters from a Gaussian mixture model.

The computational complexity for calculating the distance between Gaussian mixture models scales linearly with the number of clusters in a mixture model for most distance measures. Reducing the number of clusters in a model thus has great impact in computational complexity, but influences performance.

We present a method to reduce the number of mixture components without sacrificing retrieval performance. The required number of Gaussians in a Gaussian mixture model is estimated for each song individually. The number of clusters is then used by the Estimation Maximization algorithm to model the song data. Using individual song complexity estimation prevents overfitted models on 'simple' songs with too complex models, while still offering 'complex' songs an adequate model complexity.

The remainder of this paper is organized as follows: In section 2 we present a short overview of related work. Section 3 describes our feature modelling approach in detail. This section is followed by a performance analysis with respect to the effect of our algorithm on hubs and on a genre classification task. In our last section we summarize our results and give some recommendations for further research.

2 RELATED WORK

Berenzweig et al. [4] compare anchor space based and GMM based similarity measures with a similarity matrix retrieved from a user survey. The anchor space method performs very similarly to the GMM method.

Aucouturier and Pachet [2] systematically explore feature parameter space for timbre similarity experiments and evaluate performance with a nearest neighbour retrieval task. Optimal R-precision was found with 20 dimensional MFCCs and a Gaussian mixture model with 50 components. The number of model components however is of less influence than the number of feature dimensions. Their conclusion is that 'Everything performs the same' and that there seems to be a glass ceiling in R-precision.

Flexer et al. [7] compare Hidden Markov Models (HMMs) with Gaussian Mixture Models (GMM) describing spectral similarity of songs. It is shown that HMMs are capable of representing the underlying data better than GMMs, even if the GMM has more degrees of freedom. In a genre classification task, both methods show very similar results.

3 FEATURE MODELLING

We calculate song similarity on 15 dimensional MFCC vectors (without the 0^{th} coefficient), modelled with a Gaussian Mixture Model:

$$p\left(\boldsymbol{x}, \boldsymbol{\Theta}\right) = \sum_{i=1}^{k} \alpha_i \mathcal{G}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}_{\mathcal{X}}) \tag{1}$$

with $\boldsymbol{x}$ a single feature vector and $\boldsymbol{\Theta}$ the model parameters: cluster mean $\boldsymbol{\mu}$ and cluster covariance $\boldsymbol{\Sigma}$. The mixture weights α_i are nonnegative and add up to one.

3.1 Parameter Estimation

When the number of components in a mixture is known in advance, the Expectation Maximization (EM) algorithm [5] provides an efficient method to estimate the parameters of the distribution of $\boldsymbol{n}$ data samples $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. The EM algorithm is an iterative procedure and is guaranteed to converge to a local maximum of the maximum (log-)likelihood estimate of the mixture parameters:

$$\hat{\boldsymbol{\Theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\Theta}} \left(\log p\left(\mathcal{X} | \boldsymbol{\Theta}\right)\right) \tag{2}$$

Each iteration consists of two steps:

- **E-step:** Assign each sample to the mixture component that is most likely to have generated the sample, based on the current estimate of the model parameters.
- **M-step:** Recompute the model parameters based on the current sample membership estimation.

These steps are repeated until the likelihood estimate converges.

3.2 Complexity Estimation

During the training phase of the EM algorithm, the number of mixture components remains constant, even if the model over- or underfits the data. When listening to various kinds of music, it is clear that there are broad variations in musical structure and sound. Mörchen et al. [9] recognized this issue on a genre level and used different feature sets for each genre for determining genre likelihood.

Pampalk [10] allows variable-size models for each individual song. A k-means model is fitted to the song features and a minimal distance between clusters is defined. When two cluster centers are within this minimal distance, they are merged.

We introduce a similar approach to Pampalk, and use it to generate gaussian mixture models.

3.2.1 Optimal Models

Model selection algorithms try to find the number of components k, that minimize the cost function $\mathcal{C}(\hat{\boldsymbol{\Theta}}(k), k)$:

$$\hat{k} = \arg\min_{k} \{\mathcal{C}(\hat{\boldsymbol{\Theta}}(k), k)\}, \quad k = k_{\min}, \ldots, k_{\max} \tag{3}$$

The cost function $\mathcal{C}(\hat{\boldsymbol{\Theta}}(k), k)$ consists of two parts:

- A part expressing the goodness of fit of a model with k components. This function is a monotonically increasing function of k.
- A part penalizing models with higher k.

Figueiredo and Jain [6] presented an algorithm that optimizes a cost function based on the Minimum Description Length (MDL) criterion. This criterion is based on the assumption that if one can describe some observed data with a short code, one has a good model of the source generating the data. Other algorithms optimizing a cost function like in Equation 3 exist (eg. [12], based on the Bayesian Information Criterion), but have not been investigated.

Figueiredo uses a modified version of the EM algorithm for fitting a GMM in the dataset. The algorithm starts with a high number of components and eliminates components of the mixture in the M-step.

$$\text{for} \quad i = 1, \ldots, k:$$

$$\hat{\alpha}_i(t+1) = \frac{\max\left\{0, \left(\sum_{j=1}^{n} w_i^{(j)}\right) - \frac{N}{2}\right\}}{\sum_{i=1}^{k} \max\left\{0, \left(\sum_{j=1}^{n} w_i^{(j)}\right) - \frac{N}{2}\right\}} \tag{4}$$

where $w_i^{(j)}$ is the conditional expectation that sample j belongs to mixture component i. When the EM algorithm is converged and the number of components is still larger than $k_{\min}$, the component with the smallest support is forced to zero. This procedure is repeated until $k = k_{\min}$.

3.2.2 Optimal Model Approximation

As a consequence of using EM to iterate through the various model sizes, Figueiredo's algorithm is very slow. We found that the number of clusters found by the much simpler 'Basic Sequential Algorithmic Scheme' (BSAS, [13]) shows high correlation with the number of clusters as found by Figueiredo. This algorithm only takes two parameters: the threshold θ for determining whether a new cluster has to be formed, and N_{maxClust}, the maximum number of clusters to be formed.

The basic algorithm in pseudocode consists of the following steps:

```
$N_{\text{clust}} = 1$
$C_1 = \{\boldsymbol{x}_1\}$
for $j = 2$ to $n$ do
    find $C_i$: $d(\boldsymbol{x}_j, C_i) = \min_{\forall k} d(\boldsymbol{x}_j, C_k)$
    if ($d(\boldsymbol{x}_j, C_k) > \theta$) and ($N_{\text{clust}} < N_{\text{maxClust}}$) then
        $N_{\text{clust}} = N_{\text{clust}} + 1$
        $C_{N_{\text{clust}}} = \{\boldsymbol{x}_j\}$
    else
        $C_k = C_k \cup \{\boldsymbol{x}_j\}$
    end if
end for
```

This algorithm was modified to accept new clusters even if the maximum number of clusters has already been reached, but only if there is a cluster that was assigned less than 1% of the data. This smallest cluster is then discarded and replaced by the new cluster. After the algorithm has finished, all clusters containing less than 1% of the data are discarded. The cluster centers found by BSAS are used as input for an EM algorithm to fit a GMM in the data. The EM algorithm uses all samples, including those in the clusters that were discarded in the BSAS algorithm.

Initializing the EM process with the clustering result of BSAS significantly decreases the number of iterations the EM algorithm needs to converge when we compare it with methods that discard insignificant clusters in the training phase of the algorithm.

We compared the number of components found by Figueiredo with that of BSAS, on a subset of 234 songs from the Magnatune dataset. This subset covers all music genres available in the Magnatune data. The Pearson's correlation coefficient between the number of clusters found by both algorithms is 0.78.

4 EVALUATION

We selected a subset of 331 songs from the Magnatune dataset, covering six genres. This dataset is modelled both with fixed size GMMs with 20 clusters and with variable size GMMs with a maximum 30 clusters. The number of clusters in the variable size model case is determined by the BSAS algorithm as presented in section 3.2.2. The mean number of clusters over our dataset was 15. The EM modelling complexity scales approximately linearly with the number of clusters, we therefore obtained a 25% computing time gain for the variable size models. We use the Earth Mover's Distance to determine the distance between the GMMs [4]. Computation of the full song similarity matrix was approximately 40% faster for the variable size models.

4.1 Hub-analysis

Robustness of music similarity measures can be evaluated by means of a hub-analysis. Aucouturier [3] uses two methods to assess the hubness of various algorithms: *N-occurrences* and *Neighbour Angle*. In this subsection we evaluate the hubness of our dataset, modelled with the variable and the fixed size models, using the N-occurrence measure.

4.1.1 N-occurrences

The N-occurrence measure counts the number of times a song occurs within the N nearest neighbours of all songs in a data set. The measure is a constant-sum measure: the mean N-occurrence is equal to N. When studying hubs, we are interested in the amount of songs that occur much more frequently in the N nearest neighbours than the expected average value. In Figure 1 we show the N-occurrence histograms for $N = 50$. The number of songs that occur more than 150 times differs only marginally between the two model types: 11 for the variable size models and 12 for the fixed size models. The use of variable size models thus seems to have small positive impact on hub occurrences.

Aucouturier [3] showed that discarding statistically irrelevant clusters (*homogenization*) caused a dramatic increase in the number of hubs. With this experiment we showed that reducing the number of mixture components can be done without having negative influence on the number of hubs. Apparently, not all songs require the same number of mixture components.

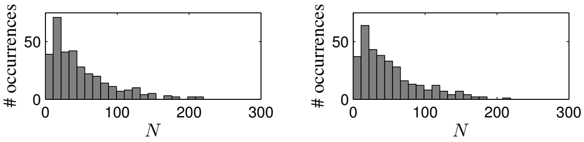
4.2 Genre classification

The most common evaluation procedure for music similarity measures is genre classification. Since we are only interested in the comparison between fixed- and variable size models, we do not apply an artist-filter as has been suggested by Pampalk et al. in [11].

Aucouturier and Pachet [1] dispute the use of genre classification for evaluating timbre similarity. Different artists within one single genre may have a very broad 'timbral' spectrum. Our data set only contains very few artists per genre. As a consequence of this, and under the assumption that each single artist only uses a narrow timbral spectrum, we can generalize genre distance to timbral distance.

4.2.1 Classification results

We use a simple k-nearest neighbour classifier and classify with a leave one out cross validation procedure. In Figure 2(a) we depict the classification accuracy for a range of k, for variable-size models with 15 Gaussians on average and for fixed-size models with 20 Gaussians. We see that the variable-size models consequently outperform

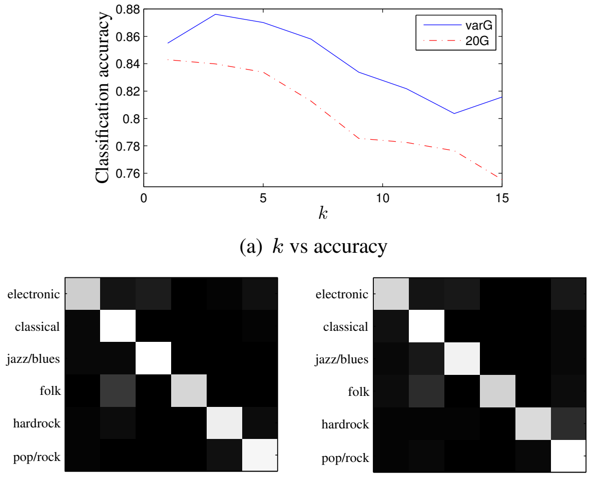


(a) Fixed size models, 20 Gaussians

(b) Variable size models

Figure 1. N-occurrence analysis

the fixed-size models, even with an average lower model complexity.



(a) k vs accuracy

(b) Variable size

(c) 20 Gaussians

Figure 2. Genre classification performance

4.2.2 *Inter- and intra genre distance*

Aucouturier and Pachet [1] use the mean distance between songs within a single genre and between different genres to express the limited use of genre classification for timbral similarity evaluations. Artists within a certain genre without a 'coherent' sound make it difficult to find a direct relationship between timbre similarity and genre similarity.

Our database consists of few artists per genre and all have a 'coherent' sound. We can thus use the measure to compare timbre discrimination performance of the fixed size Gaussian models with the variable size models to each other. Both for the 20 Gaussians and the variable size models we find a ratio of $1 : 1.32$. Although the variable size models have a lower mean number of components, the timbre information seems to be captured just as well as by the more complex fixed size models.

5 CONCLUSIONS

In this paper we presented an algorithm to estimate an optimal number of cluster components for each individual song. We compared the number of hub occurrences between a 20-Gaussian model and our variable size modelling approach with 15 clusters on average. Our variable size modelling approach marginally reduces the number of hubs.

We analyzed the timbral discrimination performance of our measure with a genre classification task on a small database with homogenous genres. Variable size models outperformed fixed size models with respect to genre classification by 3-4% and shows the same mean inter- to intra genre distance ratio at average lower model complexity.

Computation of a full song distance matrix using the Earth Mover's Distance is approximately 40% faster for the variable size models.

References

[1] J. J. Aucouturier and F. Pachet. Music similarity measures: What 's the use? In *Proc. ISMIR 02*, 2002.

[2] J. J. Aucouturier and F. Pachet. Improving timbre similarity: How high's the sky? *Journal of Negative Results in Speech and Audio Sciences*, 2004.

[3] J. J. Aucouturier and F. Pachet. A scale-free distribution of false positives for a large class of audio similarity measures. *Pattern Recognition*, 2007.

[4] A. Berenzweig, B. T. Logan, D. P. W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. In *Proc. ISMIR 03*, 2003.

[5] A. P. Dempster, D. B. Rubin, and N. Laird. Maximum likelihood from incomplete data via the em algorithm. *J.Royal Statistical Soc., Series B (Methodological)*, 1977.

[6] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[7] A. Flexer, E. Pampalk, and G. Widmer. Hidden markov models for spectral similarity of songs. In *Proc. DAFx 05*, 2005.

[8] B. Mobasher, R. Burke, C. Williams, and R. Bhaumik. Analysis and detection of segment-focused attacks against collaborative recommendation. In *WEBKDD*, volume 4198 of *Lecture Notes in Computer Science*, 2005.

[9] F. Mörchen, I. Mierswa, and A. Ultsch. Understandable models of music collections based on exhaustive feature generation with temporal statistics. In *Proc. 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2006.

[10] E. Pampalk. Speeding up music similarity. Technical report, 2005.

[11] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *Proc. ISMIR 05*, 2005.

[12] D. Pelleg and A. W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proc. 7th Int. Conf. on Machine Learning*, 2000.

[13] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Elsevier Academic Press, second edition, 2003.