A CLOSER LOOK ON ARTIST FILTERS FOR MUSICAL GENRE CLASSIFICATION

Arthur Flexer
Institute of Medical Cybernetics and Artificial Intelligence
Center for Brain Research, Medical University of Vienna, Austria
Freyung 6/2, A-1010 Vienna, Austria
arthur.flexer@meduniwien.ac.at

ABSTRACT

Musical genre classification is the automatic classification of audio signals into user defined labels describing pieces of music. A problem inherent to genre classification experiments in music information retrieval research is the use of songs from the same artist in both training and test sets. We show that this does not only lead to over-optimistic accuracy results but also selectively favours particular classification approaches. The advantage of using models of songs rather than models of genres vanishes when applying an artist filter. The same holds true for the use of spectral features versus fluctuation patterns for preprocessing of the audio files.

1 INTRODUCTION

Music information retrieval (MIR) is the science of extracting information from music. Probably the most popular form of such information is musical genre (see [2] and [15] for comprehensive overviews). Genre information can be used to describe music in interpersonal communication, in publications about music as well as to structure music databases, libraries and music stores. Although musical genre is a somewhat poorly defined concept, automation of the genre classification process remains an important topic in MIR [9]. Besides being a goal in its own right, genre classification results are often used as a means to quantify success in modelling musical similarity.

A problem inherent to genre classification experiments in MIR research is the use of songs from the same artist in both training and test sets. It can be argued that in such a scenario one is doing artist classification rather than genre classification. Specific mastering and production effects could also play a role in such a scenario. In [14] the use of a so-called "artist filter" ensuring that all songs from an artist are in either the training or the test set is proposed. The authors found that the use of such an artist filter can lower the classification results quite considerably (with one of their music collection even from 71% down to 27%). These over-optimistic accuracy results due to not using an artist filter have been confirmed in other studies [12] [5].

In extending these results, we show that the failure to use an artist filter also selectively favours particular genre classification approaches. In two genre classification experiments we are able to show that: (i) the advantage of using models of songs rather than models of genres vanishes when applying an artist filter; (ii) the same holds true for the use of spectral features versus fluctuation patterns for preprocessing of the audio files.

2 DATA

For our experiments we used a data set of the ISMIR 2004 genre classification contest. The data base consist of $S = 729$ songs from $A = 128$ artists belonging to $G = 6$ genres. The different genres plus the numbers of artists and songs belonging to each genre are given in Table 2.

Genre	No. artists	No. songs	% of songs
Classical	40	320	43.9
Electronic	30	115	15.8
Jazz Blues	5	26	3.6
Metal Punk	8	45	6.2
Pop Rock	26	101	13.9
World	19	122	16.7
Sum	128	729	100.0

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

3 METHODS

We performed two genre classification experiments to show the influence of the use of artist filters. The first shows the effect of artist filters on the usage of models of songs versus models of genres. The second shows the effect of artist filters on the choice of features used for genre classification.

3.1 Experiment 1: one model per song versus one model per genre

The following approach based on spectral similarity is now seen as one of the standard approaches in genre classification (see [8] and [1] for early references). For a given music collection of S songs, divided into S_{train} training and S_{test} test songs, each belonging to one of G music genres, it consists of the following basic steps:

One model per song (GMMsong)

1. for each song, compute Mel Frequency Cepstrum Coefficients (MFCCs) for short overlapping frames
2. train a Gaussian Mixture Model (GMM) for each of the S_{train} training *songs*
3. compute an $S_{train} \times S_{test}$ distance matrix using the likelihood of each test song given all GMMs estimated on the training songs (Equ. 2, see below)
4. based on the genre information, do nearest neighbour classification for all test songs using the distance matrix

To be more precise, step number four means that for each test song, we find its closest neighbour amongst the S_{train} GMMs (where likelihood of test song is maximal) and assign the label of this nearest neighbour to the test song. This is actually a version of an earlier approach [16] which used one GMM per genre and not per song [1]:

One model per genre (GMMgenre)

1. for each song, compute Mel Frequency Cepstrum Coefficients (MFCCs) for short overlapping frames
2. train a Gaussian Mixture Model (GMM) for each of the *genres*
3. compute a $G \times S_{test}$ distance matrix using the likelihood of each test song given all GMMs estimated on genres (Equ. 2, see below)
4. based on the genre information, do nearest neighbour classification for all test songs using the distance matrix

Step number four means that for each test song, we find its closest neighbour amongst the G GMMs (where likelihood of test song is maximal) and assign the respective label of this GMM to the test song.

For step number one, 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. [7]). 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. We used the first 8 MFCCs for all our experiments. The MA Toolbox [11] was used for computation of the MFCCs.

For step number two, we use Gaussian Mixture Models (GMM) to model the density of the input data by a mixture model of the form

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

where P_m is the mixture coefficient for the m-th component, $\mathcal{N}$ is the Normal density and μ_m and U_m are the mean vector and covariance matrix of the m-th mixture.

For a data set X^i containing T data points given a GMM trained on a song (GMMsong) or genre (GMMgenre) j, the negative log-likelihood function is given by

$$L(X^i|GMM_j) = -\frac{1}{T}\sum_{t=1}^{T} log(p_j(x_t^i)) \qquad (2)$$

For learning a GMM for a song or genre i, $L(X^i|GMM_i)$ is minimised both with respect to the mixing coefficients P_m and with respect to the parameters of the Gaussian basis functions using Expectation-Maximisation (see e.g. [4]). For all our experiments we used $M = 10$ components and diagonal covariances. For GMMsong, we used all MFCCs from the whole duration of a song for training of a GMM, as well as for evaluation of $L(X^i|GMM_j)$. For GMMgenre, we used only a total of 5000 frames from all training songs belonging to a genre for training of a GMM, as well as for evaluation of $L(X^i|GMM_j)$. This corresponds to only about one minute of music to represent a genre. The share of frames taken from each song belonging to a genre was taken randomly from the middle minute of the songs. The amount of frames and corresponding MFCCs taken from each song was equal irrespective of the song's total length.

3.2 Experiment 2: Mel Frequency Cepstrum Coefficients versus Fluctuation Patterns

This experiment compares the results obtained with the GMMsong approach described above to those obtained by substituting the MFCC features with Fluctuation Patterns:

One Fluctuation Pattern per song (FPsong)

1. for each song, compute a Fluctuation Pattern (FP)
2. compute an $S_{train} \times S_{test}$ distance matrix using the Euclidean distance of each test song to all training songs
3. based on the genre information, do nearest neighbour classification for all test songs using the distance matrix

Fluctuation Patterns (FP) [10] [13] describe the amplitude modulation of the loudness per frequency band and are based on ideas developed in [6]. Closely following the implementation outlined in [12], an FP is computed by: (i) cutting an MFCC spectrogram into three second

[1] The authors used more than just MFCCs as features.

segments, (ii) using an FFT to compute amplitude modulation frequencies of loudness (range $0 - 10Hz$) for each segment and frequency band, (iii) weighting the modulation frequencies based on a model of perceived fluctuation strength, (iv) applying filters to emphasise certain patterns and smooth the result. The resulting FP is a 20 (frequency bands according to 20 critical bands of the Bark scale [17]) times 60 (modulation frequencies, ranging from 0 to $10Hz$) matrix for each song. The distance between two FPs i and j is computed as the Euclidean distance:

$$D(FP^i, FP^j) = \sum_{k=1}^{20} \sum_{l=1}^{60} (FP^i_{k,l} - FP^j_{k,l})^2 \quad (3)$$

4 RESULTS

For experiment 1, we computed two 10-fold cross-validations to compare approaches GMMsong and GMMgenre: one with and one without the use of an artist filter. During cross-validation without artist filter, assignment of songs to training and test sets was totally random irrespective of its association with an artist. During cross-validation with artist filter it was ensured that all songs from an artist were either in the training or test set. All other assignments were again done randomly. Average accuracy rates (i.e. percentage correctly classified test songs) and standard deviations are given in Table 4.

Method	no AF	with AF
GMMsong	75.72 ± 3.35	58.50 ± 10.29
GMMgenre	69.00 ± 3.36	61.22 ± 10.42

Table 2. Experiment 1: average accuracies $\pm$ standard deviations for GMMsong and GMMgenre without and with artist filter (AF).

The difference in genre classification accuracy between GMMsong and GMMgenre without artist filter is significant: $|t| = |-4.1650| > t_{(95,df=9)} = 2.26$. GMMsong outperforms GMMgenre by about seven percentage points (76% versus 69%). The difference in genre classification accuracy between GMMsong and GMMgenre with artist filter is however not significant: $|t| = |1.1523| < t_{(95,df=9)} = 2.26$. GMMsong and GMMgenre perform at the same level of around 60%. Whereas both approaches decrease in performance due to the use of the artist filter, this decrease is more severe for GMMsong.

For experiment 2, we compared the results from experiment 1 obtained for GMMsong with those obtained with FPsong. We used the identical cross-validation folds from experiment 1 to compare FPsong to GMMsong both with and without artist filter. Average accuracy rates (i.e. percentage correctly classified test songs) and standard deviations are given in Table 4.

The difference in genre classification accuracy between GMMsong and FPsong without artist filter is significant:

Method	no AF	with AF
GMMsong	75.72 ± 3.35	58.50 ± 10.29
FPsong	63.10 ± 2.38	55.63 ± 8.61

Table 3. Experiment 2: average accuracies $\pm$ standard deviations for GMMsong and FPsong without and with artist filter (AF).

$|t| = |8.5779| > t_{(95,df=9)} = 2.26$. GMMsong outperforms FPsong by about thirteen percentage points (76% versus 63%). The difference in genre classification accuracy between GMMsong and FPsong with artist filter is however not significant: $|t| = |0.8514| < t_{(95,df=9)} = 2.26$. GMMsong and FPsong perform at the same level of around 55 to 58%. Whereas both approaches decrease in performance due to the use of the artist filter, this decrease is more severe for GMMsong.

Looking at Tables 4 and 4, it is noticeable that the standard deviations of the accuracy results increase for all experiments when using artist filters. Since some of the genres have fewer artists (e.g. five artists in "Jazz Blues" or eight in "Metal Punk") than the number of cross-validation folds (ten), some of the test folds inevitably do not contain songs from these genres. Since some of the genres are harder to classify than others, the fact that not all genres are present in all of the test folds introduces additional variance in the results. Please note that this increased variance does not change any of our results. Even with standard deviations at the level of the "no artist filter"-results, performance differences when using artist filters would still not have been significant, i.e. our conclusions would not be different.

5 DISCUSSION

In experiment 1 we examined the effect of an artist filter on a popular and successful genre classification approach: using statistical models of MFCC representations of individual songs plus nearest neighbour classification. In particular we compared it to building models of whole genres (GMMgenre) instead of individual songs (GMMsong). Whereas GMMsong is significantly better than GMMgenre without the use of an artist filter, both approaches show reduced but similar performance with an artist filter employed. Our explanation is that comparing models of individual songs is prone to finding songs from the same artist during nearest neighbour search. With GMMsong and no artist filter, this is the case in 48.84% of all test songs. Instead of actually learning the spectral characteristics of a certain genre due to its preferred instrumentation and respective sound, the peculiar style of individual artists (production effects, vocal characteristics, etc.) might be modelled.

In experiment 2 we examined the effect of an artist filter on the choice of features used for preprocessing the audio files. In particular, Mel Frequency Cepstrum Coefficients (MFCCs) were compared to Fluctuation Patterns

(FPs). Whereas MFCCs are a quite direct representation of the spectral information of a signal and therefore of the specific "sound" or "timbre" of a song, FPs are a more abstract kind of feature describing the amplitude modulation of the loudness per frequency band. During nearest neighbour search of FPsong with no artist filter, songs from the same artist are found in only 24.69% of all test songs (compared to 48.84% with GMMsong). This also explains why classification based on MFCCs (GMMsong) degrades more than classification based on FPs (FPsong) when employing an artist filter. The advantage of using MFCCs vanishes when modelling sound characteristics of individual songs is no longer the main focus.

As with any empirical research, our results are limited to the data sets and algorithms used in the experiments. Therefore it remains an open question whether our results can be replicated when other classification algorithms are being employed, other parametrizations of the data or different, probably larger data sets are being used. Nevertheless it is our belief that there is enough evidence to discourage any further research on genre classification without the use of artist filters since the results obtained with and without such filters might be quite different.

6 CONCLUSION

Our work is concerned with a major problem of musical genre classification experiments: the use of songs from the same artist in both training and test sets. Our results suggest that use of an artist filter not only lowers genre classification accuracy but may also erode the differences in accuracies between different techniques. As a consequence it seems advisable to reconsider all results on music classification obtained without the use of artist filters.

7 ACKNOWLEDGMENT

Parts of the MA Toolbox [11] and the Netlab Toolbox (`http://www.ncrg.aston.ac.uk/netlab`) have been used for this work.

8 REFERENCES

[1] Aucouturier J.-J., Pachet F.: Music Similarity Measures: What's the Use?, in Proc. of the 3rd Int. Conf. on Music Information Retrieval, pp. 157-163, 2002.

[2] Aucouturier J.-J., Pachet F.: Representing Musical Genre: A State of the Art, Journal of New Music Research, Vol. 32, No. 1, pp.83-93, 2003.

[3] 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.

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

[5] Flexer A.: Statistical Evaluation of Music Information Retrieval Experiments, Journal of New Music Research, Vol. 35, No. 2, pp.113-120, 2006.

[6] Fruehwirt M., Rauber A.: Self-Organizing Maps for Content-Based Music Clustering, Proceedings of the Twelth Italian Workshop on Neural Nets, IIAS, 2001.

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

[8] Logan B., Salomon A.: A music similarity function based on signal analysis, IEEE International Conf. on Multimedia and Expo, Tokyo, Japan, 2001.

[9] McKay C., Fujinaga I.: Musical genre classification: Is it worth pursuing and how can it be improved?, Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006), Victoria, Canada, October 8-12, 2006.

[10] Pampalk E.: Islands of Music: Analysis, Organization, and Visualization of Music Archives, MSc Thesis, Technical University of Vienna, 2001.

[11] 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.

[12] Pampalk E.: Computational Models of Music Similarity and their Application to Music Information Retrieval, Vienna University of Technology, Austria, Doctoral Thesis, 2006.

[13] Pampalk E., Rauber A., Merkl D.: Content-based organization and visualization of music archives, Proceedings of the 10th ACM International Conference on Multimedia, Juan les Pins, France, pp. 570-579, 2002.

[14] Pampalk E., Flexer A., and Widmer G.: Improvements of Audio-Based Music Similarity and Genre Classification , Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05), London, UK, September 11-15, pp.628-633, 2005.

[15] Scaringella N., Zoia G., Mlynek D.: Automatic genre classification of music content: a survey, IEEE Signal Processing Magazine, Vol. 23, Issue 2, pp. 133-141, 2006.

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

[17] Zwicker E., Fastl H.: Psychoaccoustics, Facts and Models, Springer Series of Information Sciences, Volume 22, 2nd edition, 1999.