IMPROVEMENTS OF AUDIO-BASED MUSIC SIMILARITY AND GENRE CLASSIFICATON

Elias Pampalk[1], **Arthur Flexer**[1,2], **Gerhard Widmer**[1,3]
[1] Austrian Research Institute for Artificial Intelligence (OFAI), Freyung 6/6, 1010 Vienna
[2] Institute of Medical Cybernetics and Artificial Intelligence, Center for Brain Research, Medical University of Vienna
[3] Department of Computational Perception, Johannes Kepler University, Linz, Austria

ABSTRACT

Audio-based music similarity measures can be applied to automatically generate playlists or recommendations. In this paper spectral similarity is combined with complementary information from fluctuation patterns including two new descriptors derived thereof. The performance is evaluated in a series of experiments on four music collections. The evaluations are based on genre classification, assuming that very similar tracks belong to the same genre. The main findings are that, (1) although the improvements are substantial on two of the four collections our extensive experiments confirm earlier findings that we are approaching the limit of how far we can get using simple audio statistics. (2) We have found that evaluating similarity through genre classification is biased by the music collection (and genre taxonomy) used. Furthermore, (3) in a cross validation no pieces from the same artist should be in both training and test set.

1 INTRODUCTION

Audio-based music similarity measures can be applied to playlist generation, recommendation of unknown pieces or artists, organization and visualization of music collections, or retrieval by example.

In general, music similarity is very complex, multi-dimensional, context-dependent, and ill-defined. To evaluate algorithms which model the perception of similarity would require extensive listening tests. A simpler alternative is to evaluate the performance in terms of genre classification. The assumption is that very similar pieces belong to the same genre. We believe this assumption holds in most cases despite the fact that music genre taxonomies have several limitations (see e.g. [1]). An obvious issue is that many artists have a very individual mix of several styles which is often difficult to pigeonhole. Nevertheless, genres are widely used to manage large music collections, and genre labels for artists are readily available.

In this paper we demonstrate how the performance can be improved by combining spectral similarity with complementary information. In particular, we combine spectral similarity (which describes aspects related to timbre) with fluctuation patterns (which describe periodic loudness fluctuations over time) and two new descriptors derived thereof.

The results are evaluated using four music collections with a total of almost 6000 pieces and up to 22 genres per collection. One of these collections was used as training set for the ISMIR'04 genre classification contest. Using last year's winning algorithm as baseline our findings show improvements of up to 41% (12 percentage points) on one of the collections, while the improvements on the contest training set are hardly significant. This confirms the findings of Aucouturier and Pachet [2] who suggest the existence of a glass ceiling which cannot be surpassed without taking higher level cognitive processing into account.

Another observation is that not using different music collections (with different structures and contents) can easily lead to overfitting. Finally, we recommend the use of an artist filter which ensures that none of the artists in the test set are not also present in the training set. Our results show that the classification accuracy is significantly lower if an artist filter is used. Not using an artist filter might transform the genre classification task into an artist identfication task.

2 RELATED WORK

There is a significant amount of research on audio-based genre classification with one of the first approaches presented in [3]. More recent approaches include, for example [4, 5]. Most of these approaches do not focus on similarity measures (and do not use nearest neighbor classifiers to evaluate the performance). However, content-based descriptors which work well for classifiers are also good candidates to be included in a similarity measure. An overview and evaluation of many of the descriptors used for classification can be found in [6]. In addition, recent work suggests that it is possible to automatically extract features [7].

For our work the most important ingredient is spectral similarity based on Mel Frequency Cepstrum Coefficients [2, 8, 9, 10]. Similar audio frames are grouped into clusters which are used to compare pieces (we describe the spectral similarity in detail later on). For these similarity measures the focus in terms of applications is mainly on playlist generation and recommendation (e.g. [11, 12]). Alternatives include the anchor space similarity [13] and the fluctuation patterns [14, 15].

Comparing approaches published by different authors is difficult. First, most implementations have not been made freely available. Second, sharing the same music collections is infeasible due to copyright restrictions. Third, results on different music collections (and genre

taxonomies) are not comparable.

One solution has been to reimplement approaches by other authors. For example, in [16] five different approaches were implemented. However, there is no guarantee that these implementations are correct (in fact one approach was not implemented correctly in [16]).

An alternative is the collaboration of authors (e.g. [17]) or the use of creative commons music which is easily available (e.g. the Magnatune collection used for the ISMIR'04 contest). For commercially interesting (and therefor copyright protected) music a solution is a centralized evaluation system [18] as used for the ISMIR'05 evaluation exchange (MIREX).

Related work on similarity measures for music includes approaches using cultural information retrieved from the Internet such as playlists, reviews, lyrics, and web pages (e.g. [19, 20, 21, 22, 23]). These web-based approaches can complement audio-based approaches.

3 AUDIO-BASED MUSIC SIMILARITY

In this section we review spectral similarity and the fluctuation patterns from which we extract two new descriptors, namely "Focus" and "Gravity". Furthermore we describe the linear combination of these.

3.1 Spectral Similarity

Spectral similarity describes aspects related to timbre. However, important timbre characteristics such as the attack or decay of a sound are not modeled. Instead the audio signal is chopped into thousands of very short (e.g. 20ms) frames and their order in time is ignored. Each frame is described by Mel Frequency Cepstrum Coefficients (MFCCs). The large set of frames is summarized by a model obtained by clustering the frames. The distance between two pieces is computed by comparing their cluster models.

The first approach was presented by Foote [9] based on a global set of clusters for all pieces in the collection. This global set was obtained from a classifier.

The first localized approach was presented by Logan and Salomon [10]. For each piece an individual set of clusters is used. The distances between these are computed using the Earth Movers Distance [24]. Aucouturier and Pachet suggested using the computationally more expensive Monte Carlo sampling instead [8].

For the experiments described in this paper we use the spectral similarity implemented in the MA Toolbox [25]. We apply the findings of Aucouturier and Pachet in [2], thus we refer to it as "AP".

From the 22050Hz mono audio signals two minutes from the center are used for further analysis. The signal is chopped into frames with a length of 512 samples (about 23ms) with 50% overlap. The average energy of each frame's spectrum is subtracted. The 40 Mel frequency bands (in the range of 20Hz to 16kHz) are represented by the first 20 MFCC coefficients. For clustering we use a Gaussian Mixture Model with 30 clusters trained using expectation maximization (after k-means initialization). The cluster model similarity is computed with Monte Carlo sampling.

3.2 Fluctuation Patterns

Fluctuation Patterns (FPs) describe loudness fluctuations in frequency bands [14, 15]. They describe characteristics of the audio signal which are not described by the spectral similarity measure.

A FP is a matrix with 20 rows (frequency bands) and 60 columns (modulation frequencies, in the range of 0-10Hz). The elements of this matrix describe the fluctuation strength. The distance between pieces is computed by interpreting the FP matrix as 1200-dimensional vector and computing the Euclidean distance.

From the FPs we extract two new descriptors. The first one, describes how distinctive the fluctuations at specific frequencies are, we call it *Focus*. The second one which we call *Gravity*, is related to the overall perceived tempo.

3.2.1 Focus

The Focus (FP.F) describes the distribution of energy in the FP. In particular, FP.F is low if the energy is focused in small regions of the FP, and high if the energy is spread out over the whole FP. The FP.F is computed as mean value of all values in the FP matrix, after normalizing the FP such that the maximum value equals 1. The distance between two pieces of music is computed as the absolute difference between their FP.F values.

3.2.2 Gravity

The Gravity (FP.G) describes the center of gravity of the FP on the modulation frequency axis. Given 60 modulation frequency-bins (linearly spaced in the range from 0-10Hz) the center usually lies between the 20^{th} and the 30^{th} bin.We compute FP.G by subtracting the theoretical mean of the fluctuation model (which is around the 31^{st} band) from the center.

Low values indicate that the piece might be perceived slow. However, FP.G is not intended to model the perception of tempo. Effects such as vibrato or tremolo are also reflected in the FP. The distance between two pieces of music is computed as the absolute difference between their FP.G values.

3.3 Illustrations

Figure 1 illustrates the extracted features for five songs. All five *clusters models* have low energy in high frequencies and high energy (with a high variance) in the low frequencies. As the cluster models are a very low-level representation it is difficult to guess the actual instrumentation by looking at the figures. In the *FPs* vertical lines indicate reoccurring periodic beats. The song Spider, by Flex, which is a typical example of the genre eurodance, has the strongest vertical lines. The highest *FP.F* value (0.42) is computed for Black Jesus by Everlast (belonging to the genre alternative). The song has a strong focus on guitar chords and vocals, while the drums are hardly noticeable. Spider by Flex has the lowest FP.F value (0.16). Most of the songs energy is in the strong periodic beats. The highest *FP.G* value (-5.0) is computed for Spider by Flex. The

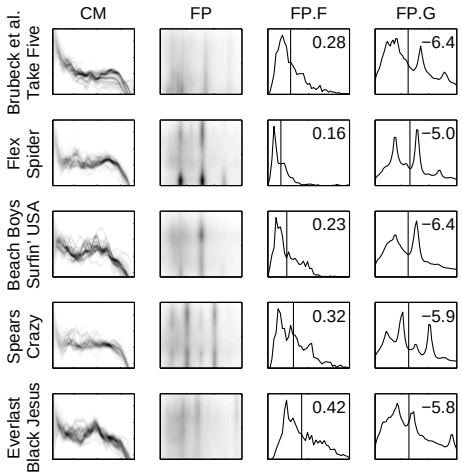


Figure 1: Visualization of the features. On the y-axis of the cluster model (CM) is the loudness (dB-SPL), on the x-axis are the Mel frequency bands. The plots show the 30 centers and their variances on top of each other. On the y-axis of the FP are the Bark frequency bands, the x-axis is the modulation frequency (in the range from 0-10Hz). The y-axis on the FP.F histogram plots are the counts, on the x-axis are the values of the FP (from 0 to 1). The mean is marked with a vertical line. The y-axis of the FP.G is the sum of values per FP column, the x-axis is the modulation frequency (from 0-10Hz). The center of gravity is marked with a vertical line.

				Artists/Genre		Tracks/Genre	
	Genres	Artists	Tracks	Min	Max	Min	Max
DB-S	16	63	100	2	7	4	8
DB-L	22	103	2522	3	6	45	259
DB-MS	6	128	729	5	40	26	320
DB-ML	10	147	3248	2	40	22	1277

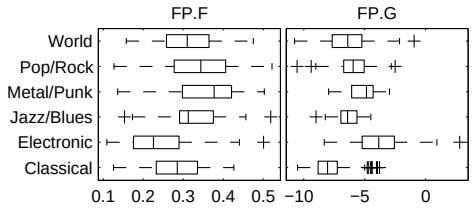
Table 1: Statistics of the four collections.

lowest value (-6.4) is computed for Take Five by the Dave Brubeck Quartet and Surfin' USA by the Beach Boys.

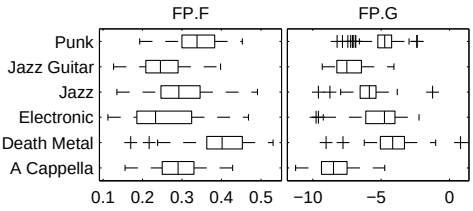
Figure 2 shows the distribution of FP.F and FP.G over different genres. The *FP.F* values have a large deviation and the overlap between quite different genres is significant. Electronic has the lowest values while punk/metal has the highest. The amount of overlap is an important factor for the quality of the descriptor. As we will see later, in the optimal combination of all similarity measures, FP.F has the smallest contribution. The *FP.G* values have a smaller deviation compared to FP.F and there is less overlap between different genres. Classical and a capella have the lowest values, while electronic, metal, and punk have the highest values.

3.4 Combination

We combine the distance matrices linearly, similar to the approach used for the aligned Self-Organizing Maps (SOMs) [26]. Before combining the distances we normalize the four distances such that the standard deviation of all pairwise distances within a music collection equals 1. In contrast to the aligned-SOMs we do not rely on the user to set the optimum weights for the linear combination, instead we automatically optimize the weights for genre classification.



(a) DB-MS



(b) DB-L

Figure 2: Boxplots showing the distribution of the descriptors per genre on two music collections. A description of the collections can be found in Section 4.1. The boxes have lines at the lower quartile, median, and upper quartile values. The whiskers show the extent of the rest of the data (the maximum length is 1.5 of the inter-quartile range). Data beyond the ends of the whiskers are marked with plus-signs.

DB-S	alternative, blues, classic orchestra, classic piano, dance, eurodance, happy sound, hard pop, hip hop, mystera, pop, punk rock, rock, rock & roll, romantic dinner, talk
DB-L	a cappella, acid jazz, blues, bossa nova, celtic, death metal, DnB, downtempo, electronic, euro-dance, folk-rock, German hip hop, hard core rap, heavy metal/thrash, Italian, jazz, jazz guitar, melodic metal, punk, reggae, trance, trance2
DB-MS	classical, electronic, jazz/blues, metal/punk, pop/rock, world
DB-ML	ambient, classical, electronic, jazz, metal, new age, pop, punk, rock, world

Table 2: List of genres for each collection.

4 GENRE CLASSIFICATION

We use a nearest neighbor classifier and leave-one-out cross validation for the evaluation. The accuracies are computed as ratio of the correctly classified compared to the total number of tracks (without normalizing the accuracies with respect to the different class probabilities).

In contrast to the ISMIR'04 genre contest we apply an artist filter which ensures that all pieces of an artist are either in the training set or test set. Otherwise the genre classification might be transformed into an artist identification task since all pieces of an artist are in the same genre (in all of the collections we use). The resulting performance is significantly worse. For example, on the ISMIR 2004 genre classification training set (using the same algorithm we submitted last year) we obtain 79% accuracy without and only 64% with artist filter. On the large in-house collection (using the same algorithm) we obtain 71% without and only 27% with filter. Therefor, in the results described below we always use an artist filter if not stated otherwise.

In the remainder of this section first the four music

collections we use are described. Second, results using only one similarity measure are presented. Third, pairwise combinations with spectral similarity (AP) are evaluated. Fourth, all four measures are combined. Finally, the performances on all collections is evaluated to avoid overfitting.

4.1 Data

We use four music collections with a total of almost 6000 pieces. Details are given in Tables 1 and 2. An important characteristic is that the collections are structured differently and have different types of contents. This helps to avoid overfitting.

4.1.1 In-House Small (DB-S)

The smallest collection consists of 100 pieces. It is the same used in [25]. However, we removed all classes consisting of one artist only. The categories are not strictly genres (one of them is romantic dinner music). Furthermore, the collection also includes one non-music category, namely speech (German cabaret). This collection has a very good (i.e low) ratio of tracks per artist. However, due to its size the results need to be treated with caution.

4.1.2 In-House Large (DB-L)

The second largest collection has mainly been organized according to genre/artist/album. Thus, all pieces from an artist (and album) are assigned to the same genre, which is questionable but common practice. Only two pieces overlap between DB-L and DB-S, namely Take Five and Blue Rondo by the Dave Brubeck Quartet. The genres are user defined and inconsistent. In particular, there are two different definitions of trance. Furthermore, there are overlaps, for example, jazz and jazz guitar, heavy metal and death metal etc.

4.1.3 Magnatune Small (DB-MS)

This collection was used as training set for the ISMIR'04 genre classification contest. The music originates from Magnatune[1] and is licensed as creative commons. MTG[2] compiled the collection. Although it is a subset of DB-ML we use it to compare our results to those of the ISMIR'04 results. However, while we report 79% accuracy for our last year's submission on the training set, the accuracy on the test set was 84%. We believe this is related to the artist filter issue, as half of the pieces of each album were split between training and test set and all pieces from an artist belong to the same genre.

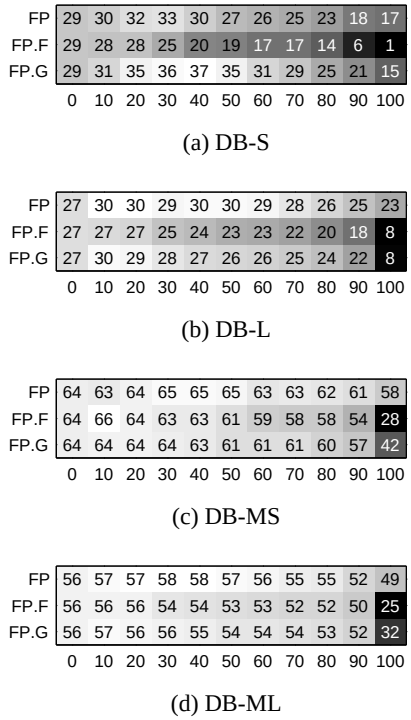
The genre labels are given on the Magnatune website. The collection is very unbalanced. Most pieces belong to the genre classical and a large number of pieces in world sound like classical music. Some of the original Magnatune classes were merged by MTG due to ambiguities and the small number of tracks in some of the genres.

4.1.4 Magnatune Large (DB-ML)

This is the largest set in our experiments. DB-MS is a subset of this collection. The number of artists is not much higher than in DB-MS and the genres are equally unbalanced. The genres which were merged for the ISMIR'04 contest are separated.

[1] http://www.magnatune.com

[2] http://www.iua.upf.es/mtg



(a) DB-S

(b) DB-L

(c) DB-MS

(d) DB-ML

Figure 3: Results for combining AP with one of the other measures. All values are given in percent. The values on the x-axis are the mixing coefficients. For example, the fourth column in the second row is the accuracy for combining 70% AP with 30% of FP.F.

4.2 Individual Performance

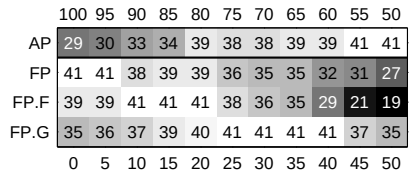
The performances using the similarity measures individually are given in Figure 3 in the first (only spectral similarity, AP) and last columns (FP, FP.F, FP.G). AP clearly performs best, followed by FP. The performance of FP.F is extremely poor on DB-S while it is equal to FP.G on DB-L. For DB-MS without artist filter we obtain: AP 79%, FP 66%, FP.F 30%, and FP.G 43% (using each individually). Always guessing that a piece is classical gives 44% accuracy.

4.3 Combining Two

The results for combining AP with one of the other measures are given in Figure 3. The main findings are that combining AP with FP or FP.G performs better than combining AP with FP.F (except for 10% FP.F and 90% AP in DB-MS). For all collections a combination can be found which improves the performance. However, the improvements on the Magnatune collection are marginal. The smooth changes of the accuracy with respect to the mixing coefficient are an indicator that the the approach is relatively robust (within each collection).

4.4 Combining All

Figure 4 shows the accuracies obtained when all similarity measures are combined. There are a total of 270 possi-



	100	95	90	85	80	75	70	65	60	55	50
AP	29	30	33	34	39	38	38	39	39	41	41
FP	41	41	38	39	39	36	35	35	32	31	27
FP.F	39	39	41	41	41	38	36	35	29	21	19
FP.G	35	36	37	39	40	41	41	41	41	37	35
	0	5	10	15	20	25	30	35	40	45	50

(a) DB-S

	100	95	90	85	80	75	70	65	60	55	50
AP	27	30	31	32	32	32	32	32	31	32	31
FP	30	32	32	32	32	31	31	32	31	31	30
FP.F	31	32	32	32	31	31	30	29	28	26	23
FP.G	32	32	32	32	31	30	29	29	29	28	26
	0	5	10	15	20	25	30	35	40	45	50

(b) DB-L

	100	95	90	85	80	75	70	65	60	55	50
AP	64	67	68	67	67	67	67	67	67	67	67
FP	68	67	67	67	67	67	66	67	67	65	65
FP.F	66	68	67	67	66	66	65	65	64	64	61
FP.G	67	68	67	67	67	66	65	65	65	64	61
	0	5	10	15	20	25	30	35	40	45	50

(c) DB-MS

	100	95	90	85	80	75	70	65	60	55	50
AP	56	57	57	58	58	58	58	58	58	58	57
FP	57	58	58	58	58	58	58	58	58	57	57
FP.F	58	58	58	58	57	57	56	56	55	55	53
FP.G	58	58	58	58	58	57	57	57	56	56	54
	0	5	10	15	20	25	30	35	40	45	50

(d) DB-ML

Figure 4: Results for combining all similarity measures. A total of 270 combinations are summarized in each table. All values are given in percent. The mixing coefficients for AP (the first row) are given above the table, for all other rows below. For each entry in the table of all possible combinations the highest accuracy is given. For example, the second row, third column depicts the highest accuracy obtained from all possible combinations with 10% FP. The not specified 90% can have any combination of mixing coefficients, e.g. 90% AP, or 80% AP and 10% FP.G etc.

ble combinations using a step size of 5 percent-points and limiting AP to a mixing coefficient between 100-50% and the other measures to 0-50%.

Analogously to the previous results FP.F has the weakest performance and the improvements for the Magnatune collection are hardly significant. As in Figure 3 the smooth changes of the accuracy with respect to the mixing coefficient are an indicator for the robustness of the approach (within each collection). Without the artist filter the combinations on the DB-MS reach a maximum of 81% (compared to 79% using only AP).

It is clearly noticeable that the results on the collections are quite different. For example, for DB-S using as little AP as possible (highest values around 45-50%) and a lot of FP.G (highest values around 25-40%) gives best results. On the other hand, for the DB-MS collection the best results are obtained using 90% AP and only 5% FP.G. These deviations indicate overfitting, thus we analyze the performances across collections in the next section.

	Weights				Classification Accuracy				
Rank	AP	FP	FP.F	FP.G	DB-S	DB-L	DB-MS	DB-ML	Score
1	65	15	5	15	38	32	67	58	1.14
2	65	10	10	15	38	31	67	57	1.14
3	70	10	5	15	38	31	67	58	1.14
4	55	20	5	20	39	31	65	57	1.14
5	60	15	10	15	38	31	66	57	1.14
6	60	15	5	20	39	31	66	57	1.13
7	75	10	5	10	37	31	67	58	1.13
8	75	5	5	15	38	31	66	58	1.13
9	65	10	5	20	38	30	66	58	1.13
10	55	5	10	30	41	29	65	56	1.13
248	100	0	0	0	29	27	64	56	1.00
270	50	0	50	0	19	23	61	53	0.85

Table 3: Overall performance on all collections. Columns 2-4 are the mixing coefficients in percent and columns 5-8 are the rounded accuracies in percent.

4.5 Overall Performance

To study overfitting we compute the relative performance gain compared to the AP baseline (i.e. using only AP). We compute the score (which we want to maximize) as the average of these gains over the four collections. The results are given in Table 3.

The worst combination (using 50% AP and 50% FP.F) yields a score of 0.85. (That is, in average, the accuracy using this combination is 15% lower compared to the AP baseline.) There are a total of 247 combinations which perform better than the AP baseline. Almost all of the 22 combinations that fall below AP have a large contribution of FP.F. The best score is 14% above the baseline. The ranges of the top 10 ranked combinations are 55-75% AP, 5-20% FP, 5-10% FP.F, 10-30% FP.G.

Without artist filter, for DB-MS the top three ranked combinations from Table 3 have the accuracies 1: 79%, 2: 78%, 3: 79% (the AP baseline is 79%, the best possible combination yields 81%). For the DB-S collection without artist filter the AP baseline is 52% and the top three ranked combinations have the accuracies 1: 63%, 2: 61%, 3: 62% (the best possible score achieved through combination is 64%).

Figure 5 shows the score of each combination ranked by their average (on all four collections) score. In several cases a combination performs well on one collection and poor on another. This indicates that there is a large potential for overfitting. On the other hand, the performance stays above the baseline for most of the combinations and there is a common trend. Truly reliable results would require further testing on additional collections.

5 CONCLUSIONS

We presented an improvement to audio-based music similarity and genre classification. We combined spectral similarity (in particular the approach presented by Aucouturier and Pachet) with three additional similarity measures based on fluctuation patterns. We presented two new descriptors and a series of experiments evaluating the combinations.

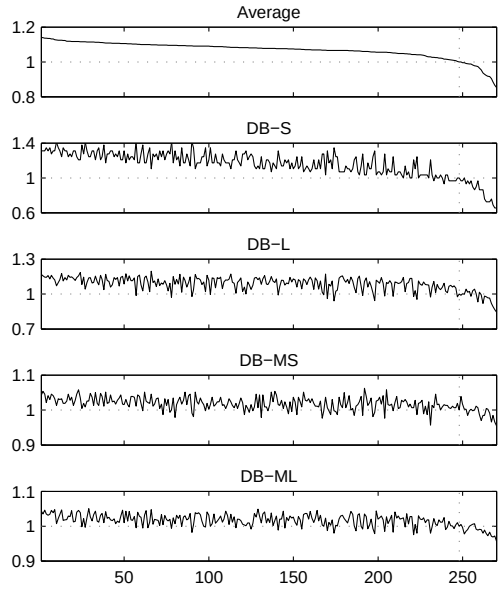


Figure 5: Score (y-axis) ranked by average performance (x-axis).

Although we obtained an average genre classification performance increase of 14%, our findings confirm the glass ceiling observed in [2]. In particular, for the training set used for the ISMIR'04 genre classification contest our improvements are hardly significant. Furthermore, preliminary results with a larger number of descriptors indicate that the performance per collection can only be further improved by up to 1-2 percentage points.

Our results show a significant difference in the overall performance if pieces from the same artist are in the test and training set. We believe this shows the necessity to use an artist filter to evaluate genre classification performance (if all pieces from an artist are assigned to the same genre) and not the performance of artist identification. However, some the observed effects are partly also caused by the low number of artists per genre. For example, for DB-L in some cases up to one third of the pieces from the target genre are removed by the artist filter.

Another observation is that improvements on one collection might harm the performance on another. This danger of overfitting is imminent and a simple solution is the use of different collections (with different contents and different genre taxonomies).

In general, genre classification is not the ideal solution to evaluate similarity. Although the assumption that the most similar piece to a given piece belongs to the same genre holds in many cases, a true evaluation would require listening tests. However, a listening test where human listeners are required to sort a complete collection (i.e. $O(N^2)$ comparisons) is infeasible for large collections. Several alternatives exist and should be considered for future work.

Acknowledgements

This research was supported by the EU project SIMAC (FP6-507142). OFAI is supported by the Austrian Federal Ministries BMBWK and BMFIT. The authors wish to thank the anonymous reviewers for their helpful comments.

References

[1] F. Pachet and D. Cazaly. A taxonomy of musical genres. In *Proc RIAO Content-Based Multimedia Information Access*, 2000.

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

[3] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. In *Proc ISMIR*, 2001.

[4] M. F. McKinney and J. Breebaart. Features for audio and music classification. In *Proc ISMIR*, 2003.

[5] K. West and S. Cox. Features and classifiers for the automatic classification of musical audio signals. In *Proc ISMIR*, 2004.

[6] T. Pohle. Extraction of audio descriptors and their evaluation in music classification tasks. MSc thesis, TU Kaiserslautern, ÖFAI, DFKI, 2005.

[7] A. Zils and F. Pachet. Automatic extraction of music descriptors from acoustic signals. In *Proc ISMIR*, 2004.

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

[9] J. Foote. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II*, 1997.

[10] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *Proc IEEE Intl Conf on Multimedia and Expo*, 2001.

[11] B. Logan. Music recommendation from song sets. In *Proc ISMIR*, 2004.

[12] B. Logan. Content-based playlist generation: Exploratory experiments. In *Proc ISMIR*, 2002.

[13] A. Berenzweig, D. P.W. Ellis, and S. Lawrence. Anchor space for classification and similarity measurement of music. In *Proc IEEE Intl Conf on Multimedia and Expo*, 2003.

[14] E. Pampalk. Islands of music: Analysis, organization, and visualization of music archives. MSc thesis, Vienna University of Technology, 2001.

[15] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proc ACM Multimedia*, 2002.

[16] E. Pampalk, S. Dixon, and G. Widmer. On the evaluation of perceptual similarity measures for music. In *Proc Intl Conf on Digital Audio Effects*, 2003.

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

[18] J.S. Downie, J. Futrelle, and D. Tcheng. The international music information retrieval systems evaluation laboratory: governance, access and security In *Proc ISMIR*, 2004.

[19] S. Baumann and O. Hummel. Using cultural metadata for artist recommendation. In *Proc WedelMusic Conf*, 2003.

[20] P. Knees, E. Pampalk, and G. Widmer. Artist classification with web-based data. In *Proc ISMIR*, 2004.

[21] B. Logan, A. Kositsky, and P. Moreno. Semantic analysis of song lyrics. In *Proc IEEE Intl Conf on Multimedia and Expo 2004*, 2004.

[22] F. Pachet, G. Westerman, and D. Laigre. Musical data mining for electronic music distribution. In *Proc WedelMusic Conf*, 2001.

[23] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proc Intl Computer Music Conf*, 2002.

[24] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth movers distance as a metric for image retrieval. *Intl Journal of Computer Vision*, 40(2), 2000.

[25] E. Pampalk. A Matlab toolbox to compute music similarity from audio. In *Proc ISMIR*, 2004.

[26] E. Pampalk, W. Goebl, and G. Widmer. Visualizing changes in the structure of data for exploratory feature selection. In *Proc ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining*, 2003.