COMBINING FEATURES REDUCES HUBNESS IN AUDIO SIMILARITY

Arthur Flexer,[1] **Dominik Schnitzer,**[1,2] **Martin Gasser,**[1] **Tim Pohle**[2]
[1]Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria
[2]Department of Computational Perception
Johannes Kepler University Linz, Austria
arthur.flexer@ofai.at, dominik.schnitzer@ofai.at
martin.gasser@ofai.at, tim.pohle@jku.at

ABSTRACT

In audio based music similarity, a well known effect is the existence of hubs, i.e. songs which appear similar to many other songs without showing any meaningful perceptual similarity. We verify that this effect also exists in very large databases ($>$ 250000 songs) and that it even gets worse with growing size of databases. By combining different aspects of audio similarity we are able to reduce the hub problem while at the same time maintaining a high overall quality of audio similarity.

1. INTRODUCTION

One of the central goals in music information retrieval is the computation of audio similarity. Proper modeling of audio similarity enables a whole range of applications: genre classification, play list generation, music recommendation, etc. The de facto standard approach to computation of audio similarity is timbre similarity based on parameterization of audio using Mel Frequency Cepstrum Coefficients (MFCCs) plus Gaussian mixtures as statistical modeling (see Section 3.1). However, it is also an established fact that this approach suffers from the so-called hub problem [3]: songs which are, according to the audio similarity function, similar to very many other songs without showing any meaningful perceptual similarity to them. The hub problem of course interferes with all applications of audio similarity: hub songs keep appearing unwontedly often in recommendation lists and play lists, they degrade genre classification performance, etc.

Although the phenomenon of hubs is not yet fully understood, a number of results already exist. Aucouturier and Pachet [1] established that hubs are distributed along a scale-free distribution, i.e. non-hub songs are extremely common and large hubs are extremely rare. This is true for MFCCs modelled with different kinds of Gaussian mixtures as well as Hidden Markov Models, irrespective whether parametric Kullback-Leibler divergence or non-parametric histograms plus Euclidean distances are used for computation of similarity. But is also true that hubness is not the property of a song per se since non-parametric and parametric approaches produce very different hubs. It has also been noted that audio recorded from urban soundscapes, different from polyphonic music, does not produce hubs [2] since its spectral content seems to be more homogeneous and therefore probably easier to model. Direct interference with the Gaussian models during or after learning has also been tried (e.g. homogenization of model variances) although with mixed results. Whereas some authors report an increase in hubness [1], others observed the opposite [5]. Using a Hierarchical Dirichlet Process instead of Gaussians for modeling MFCCs seems to avoid the hub problem altogether [6].

Our contribution to the understanding of the hub problem is threefold: (i) since all results on the hub problem so far were achieved on rather small data sets (from $\sim$ 100 to $\sim$ 15000 songs), we first establish that the problem also exists in very large data sets ($>$ 250000 songs); (ii) we show that a non-timbre based parameterization is not prone to hubness; (iii) finally we show how combining timbre based audio similarity with other aspects of audio similarity is able to reduce the hub problem while maintaining a high overall quality of audio similarity.

2. DATA

2.1 Web shop data

For our experiments we used a data set $D(ALL)$ of $S_W = 254398$ song excerpts (30 seconds) from a popular web shop selling music. The freely available preview song excerpts were obtained with an automated web-crawl. All meta information (artist name, album title, song title, genres) is parsed automatically from the hmtl-code. The excerpts are from $U = 18386$ albums from $A = 1700$ artists. From the 280 existing different hierarchical genres, only the $G_W = 22$ general ones on top of the hierarchy are being kept for further analysis (e.g. "Pop/General" is kept but not "Pop/Vocal Pop"). The names of the genres plus percentages of songs belonging to each of the genres are given in Table 1. Please note that every song is allowed to belong to more than one genre, hence the percentages in Table 1 add up to more than 100%. The genre information is identical for all songs on an album. The numbers of

genre labels per albums range from 1 to 8. Our database was set up so that every artist contributes between 6 to 29 albums.

To study the influence of the size of the database on results, we created random non-overlapping splits of the entire data set: $D(1/2)$ - two data sets with mean number of song excerpts $= 127199$, $D(1/20)$ - twenty data sets with mean number of songs excerpts $= 12719.9$, $D(1/100)$ - one hundred data sets with mean number of songs excerpts $= 2543.98$. An artist with all their albums is always a member of a single data set.

Pop	Classical	Broadway
49.79	12.89	7.45
Soundtracks	Christian/Gospel	New Age
1.00	10.20	2.48
Miscellaneous	Opera/Vocal	Alternative Rock
6.11	3.24	27.13
Rock	Rap/Hip-Hop	R&B
51.78	0.98	4.26
Hard Rock/Metal	Classic Rock	Country
15.85	15.95	4.07
Jazz	Children's Music	International
6.98	7.78	9.69
Latin Music	Folk	Dance & DJ
0.54	11.18	5.24
Blues		
11.24		

Table 1. Percentages of songs belonging to the 22 genres with multiple membership allowed for the **web shop data**.

2.2 Music portal data

We also used a smaller data base comprised of the music of an Austrian music portal. The FM4 Soundpark is an internet platform [1] of the Austrian public radio station FM4. This internet platform allows artists to present their music free of any cost in the WWW. All interested parties can download this music free of any charge. This music collection contains about 10000 songs and is organized in a rather coarse genre taxonomy. The artists themselves choose which of the $G_M = 6$ genre labels "Hip Hop, Reggae, Funk, Electronic, Pop and Rock" best describe their music. The artists are allowed to choose one or two of the genre labels. We use a data base of $S_M = 7665$ songs for our experiments. Number of songs and percentages across genres are given in Table 2. Please note that every song is allowed to belong to more than one genre, hence the percentages in Table 2 add up to more than 100%.

[1] `http://fm4.orf.at/soundpark`

HiHo	Regg	Funk	Elec	Pop	Rock
15.34	4.64	21.87	46.25	34.39	44.03

Table 2. Percentages of songs belonging to genres with multiple membership allowed for the **music portal data**. Genres are Hip Hop, Reggae, Funk, Electronic, Pop and Rock.

3. METHODS

We compare two approaches based on different parameterizations of the data. Whereas Mel Frequency Cepstrum Coefficients (MFCCs) are a quite direct representation of the spectral information of a signal and therefore of the specific "sound" or "timbre" of a song, Fluctuation Patterns (FPs) are a more abstract kind of feature describing the amplitude modulation of the loudness per frequency band.

3.1 Mel Frequency Cepstrum Coefficients and Single Gaussians (G1)

We use the following approach to compute music similarity based on spectral similarity. For a given music collection of songs, it consists of the following steps:

1. for each song, compute MFCCs for short overlapping frames
2. train a single Gaussian (G1) to model each of the songs
3. compute a distance matrix M_{G1} between all songs using the symmetrized Kullback-Leibler divergence between respective G1 models

For the web shop data the 30 seconds song excerpts in mp3-format are recomputed to 22050Hz mono audio signals. For the music portal data, the two minutes from the center of each song are recomputed to 22050Hz mono audio signals. 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 $46.4ms$ (1024 samples), the hop size $23.2ms$ (512 samples). We used the first $d = 25$ MFCCs for all experiments with the web shop data and the first $d = 20$ MFCCs for all experiments with the music portal data.

A single Gaussian (G1) with full covariance represents the MFCCs of each song [8]. For two single Gaussians, $p(x) = \mathcal{N}(x; \mu_p, \Sigma_p)$ and $q(x) = \mathcal{N}(x; \mu_q, \Sigma_q)$, the closed form of the Kullback-Leibler divergence is defined as [14]:

$$KL_N(p\|q) = \frac{1}{2}\left(\log\left(\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) + Tr\left(\Sigma_p^{-1}\Sigma_q\right) + (\mu_p - \mu_q)'\,\Sigma_p^{-1}\,(\mu_q - \mu_p) - d\right) \quad (1)$$

where $Tr(M)$ denotes the trace of the matrix M, $Tr(M) = \Sigma_{i=1..n} m_{i,i}$. The divergence is symmetrized by computing:

$$KL_{sym} = \frac{KL_N(p\|q) + KL_N(q\|p)}{2} \quad (2)$$

3.2 Fluctuation Patterns and Euclidean Distance (FP)

Fluctuation Patterns (FP) [9] [12] describe the amplitude modulation of the loudness per frequency band and are based on ideas developed in [4]. For a given music collection of songs, computation of music similarity based on FPs consists of the following steps:

1. for each song, compute a Fluctuation Pattern (FP)
2. compute a distance matrix M_{FP} between all songs using the Euclidean distance of the FP patterns

Closely following the implementation outlined in [10], an FP is computed by: (i) cutting an MFCC spectrogram into three second 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 emphasize certain patterns and smooth the result. The resulting FP is a 12 (frequency bands according to 12 critical bands of the Bark scale [15]) times 30 (modulation frequencies, ranging from 0 to $10Hz$) matrix for each song. The distance between two FPs i and j is computed as the squared Euclidean distance:

$$D(FP^i, FP^j) = \sum_{k=1}^{12}\sum_{l=1}^{30}(FP^i_{k,l} - FP^j_{k,l})^2 \quad (3)$$

For the web shop data an FP pattern is computed from the full 30 second song excerpts. For the music portal data an FP pattern is computed from the central minute of each song.

4. RESULTS

4.1 Hubs in very large data bases

As a measure of the hubness of a given song we use the so-called n-occurrence [1], i.e. the number of times the songs occurs in the first n nearest neighbors of all the other songs in the data base. Please note that the mean n-occurrence across all songs in a data base is equal to n. Any n-occurrence significantly bigger than n therefore indicates existence of a hub. For every song in the data bases $D(ALL)$, $D(1/2)$, $D(1/20)$ and $D(1/100)$ (see Section 2.1) we computed the first n nearest neighbors for both methods G1 and FP. For method G1, the first n nearest neighbors are the n songs with minimum Kullback Leibler divergence (Equation 2) to the query song. For method FP, the first n nearest neighbors are the songs with minimum Euclidean distance of the FP pattern (Equation 3) to the query song. To compare results for data bases of different sizes S_W, we keep the relation n/S_W constant at 0.001965: e.g. for $D(ALL)$ $S_W = 254398$ and $n = 500$, for $D(1/100)$ $S_W = 2543.98$ and therefore $n = 5$.

data set	n	maxhub	maxhub%	$hub3\%$
D(ALL)	500	29588	11.63	7.75
D(1/2)	250	12094	9.52	7.56
D(1/20)	25	590	4.68	6.13
D(1/100)	5	62	2.49	4.62

Table 3. Hub analysis results for **web shop data** using method **G1**. See Section 4.1 for details.

data set	n	maxhub	maxhub%	$hub3\%$
D(ALL)	500	3386	1.33	1.18
D(1/2)	250	1639	1.29	1.18
D(1/20)	25	137	1.08	1.12
D(1/100)	5	25	1.02	1.22

Table 4. Hub analysis results for **web shop data** using method **FP**. See Section 4.1 for details.

The results given in Tables 3 and 4 show mean values over 100 ($D(1/100)$), 20 ($D(1/20)$), 2 ($D(1/2)$) data sets or the respective single result for the full data set $D(ALL)$. We give the number of nearest neighbors n, the absolute number of the maximum n-occurrence $maxhub$ (i.e. the biggest hub), the percentage of songs in whose nearest neighbor lists this biggest hub can be found $maxhub\% = maxhub/S_W$ and the percentage of hubs $hub3\%$ (i.e. the percentage of songs of which the n-occurrence is more than three times n).

When looking at the results for method G1 (Table 3) it is clear that hubs do exist even for very large data bases. As a matter of fact, the hub problem increases significantly with the size of the data base. Whereas for the small data sets $D(1/100)$ on average the biggest hub is in the neighbor lists of 2.49% of all songs, the biggest hub for $D(ALL)$ is a neighbor to 11.63% of all songs. The number of hubs increases from an average 4.62% of all songs in $D(1/100)$ to 7.75% in $D(ALL)$. To sum up, there are more and bigger hubs in larger data bases when using method G1 for computation of audio similarity.

The results for method FP in Table 4 show a quite different picture. The size of the biggest hub is much smaller and the number of hubs is also much reduced. There is also very little influence of the size of the data bases on the results. We like to conclude that method FP is not as prone to hubness as method G1.

w_{G1}	w_{FP}	maxhub	maxhub%	$hub3\%$	$hub10\%$	$hub15\%$	$hub20\%$	acc
1.0	**0.0**	**879**	**11.47**	**8.05**	**0.94**	**0.40**	**0.22**	**48.47**
0.9	0.1	598	7.80	8.15	0.86	0.35	0.09	49.84
0.8	0.2	445	5.81	8.23	0.80	0.23	0.08	49.47
0.7	0.3	342	4.46	8.11	0.72	0.16	0.05	48.44
0.6	**0.4**	**352**	**4.59**	**8.06**	**0.57**	**0.09**	**0.01**	**47.80**
0.5	0.5	344	4.49	8.04	0.51	0.07	0.01	46.58
0.4	0.6	334	4.36	7.91	0.31	0.04	0.01	45.73
0.3	0.7	315	4.11	7.80	0.21	0.01	0.01	44.93
0.2	0.8	247	3.22	7.21	0.17	0.01	0.0	43.94
0.1	0.9	215	2.81	6.72	0.04	0.0	0.0	42.82
0.0	**1.0**	**145**	**1.89**	**5.38**	**0.0**	**0.0**	**0.0**	**38.45**

Table 5. Hub analysis result for **music portal data** using combinations of **G1** and **FP**. Results for using G1 or FP alone as well as for a moderate combination are in bold face. See Section 4.2 for details.

4.2 Reducing hubs by combining G1 and FP

Recent advances in computing audio similarity rely on combining timbre-based approaches (MFCCs plus Gaussian models) with a range of other features derived from audio. In particular, combinations of timbre and, among other features, fluctuation patterns or variants thereof have proven sucessfull [11, 13]. Such a combination approach was able to rank first at the 2009 MIREX "Audio Music Similarity and Retrieval"-contest [2]. Since our method based on fluctuation patterns is less prone to hubness than the timbre based approach, we tried to combine distances obtained with methods G1 and FP. It is our hypothesis that such a combination could reduce hubness and at the same time preserve the good quality of timbre based methods in terms of audio similarity.

Following previous approaches towards combination of features [10, 11] we first normalize the distance matrices M_{G1} and M_{FP} by subtracting the respective overall means and dividing by the standard deviations:

$$\bar{M}_{G1} = \frac{M_{G1} - \mu_{G1}}{s_{G1}} \quad \bar{M}_{FP} = \frac{M_{FP} - \mu_{FP}}{s_{FP}} \qquad (4)$$

We combine the normalized distance matrices linearly using weights w_{G1} and w_{FP}:

$$\bar{M}_C = w_{G1}\bar{M}_{G1} + w_{FP}\bar{M}_{FP} \qquad (5)$$

To evaluate the quality of audio similarity achieved by combining methods G1 and FP we computed the genre classification performance. We used nearest neighbor classification as a classifier. For every song in the data base we computed the first nearest neighbor using the distance matrix $\bar{M}_C$. The first nearest neighbor to a query song is the song with minimum distance according to $\bar{M}_C$. To estimate genre classification accuracy, the genre label of a query song s_{query} and its first nearest neighbor s_{nn} were compared. The accuracy is defined as:

$$acc(s_{query}, s_{nn}) = \frac{|g_{query} \cap g_{nn}|}{|g_{query} \cup g_{nn}|} \times 100 \qquad (6)$$

[2] http://www.music-ir.org/mirex/2009/

with g_{query} (g_{nn}) being a set of all genre labels for the query song (nearest neighbor song) and $|.|$ counting the number of members in a set. Therefore accuracy is defined as the number of shared genre labels divided by the set size of the union of sets g_{query} and g_{nn} times 100. The latter is done to acount for nearest neighbor songs with two genre labels as compared to only one genre label. The range of values for accuracy is between 0 and 100. All genre classification results are averaged over ten fold cross validations.

We ran a series of experiments using the music portal data base (see Section 2.2) and a number of different weight combinations w_{G1} and w_{FP}. To measure the hubness of a given song we use n-occurrence with n equal 15. The results given in Table 5 show: the weights w_{G1} and w_{FP}, the absolute number of the maximum n-occurrence $maxhub$ (i.e. the biggest hub), the percentage of songs in whose nearest neighbor lists this biggest hub can be found $maxhub\%$, the percentage of hubs $hub3|10|15|20\%$ (i.e. the percentage of songs of which the n-occurrence is more than $3|10|15|20$ times n) and the genre classification accuracy acc.

It is evident that with the weight w_{FP} for method FP growing, the hubs become smaller and less in number but the genre classification accuracy also degrades. Whereas using method G1 alone (i.e. $w_{G1} = 1.0$ and $w_{FP} = 0.0$) yields a maximum hub of size 879 that is in the nearest neighbor lists of 11.47% of all songs, a moderate combination using weights $w_{G1} = 0.6$ and $w_{FP} = 0.4$ diminishes the biggest hub to a size of 352. This reduced hub is now a member of only 4.59% of the nearest neighbor lists. Also the number of especially large hubs decreases: e.g. the percentage of songs of which the n-occurrence is more than 20 times n ($hub20\%$) drops from 0.22% to 0.01% (in absolute numbers from 17 to 1); the number of more moderate sized hubs ($hub10\%$) is still about halfed (from 0.94% to 0.57%, or from 72 to 44 in absolute numbers). Such a moderate combination does not impair the overall quality of audio similarity as measured with genre classification accuracy: it is at 47.80% which is at the level of using method G1 alone yielding 48.47%. The baseline accuracy achieved by always guessing the most probable

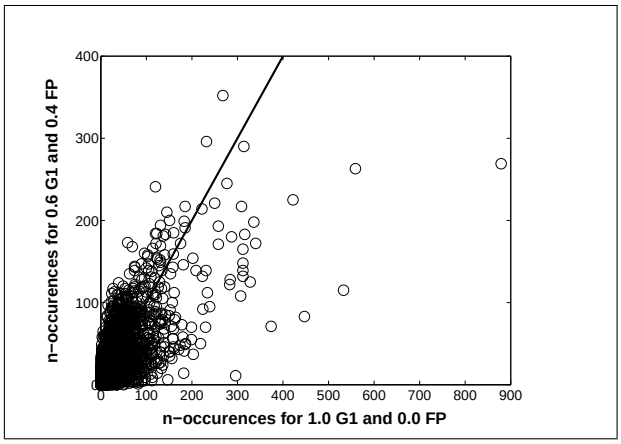

Figure 1. n-occurrences of using method G1 alone (x-axis) vs. n-occurrences using a moderate combination of G1 and FP (y-axis, $w_{G1} = 0.6$ and $w_{FP} = 0.4$) for **music portal data**. The diagonal line indicates songs for which the n-occurence does not change.

genre "Electronic" (see Table 2) is 29.11%. Always guessing the two most probable genres "Electronic" and "Rock" yields 36.46%.

In Figure 1 we have plotted the n-occurrences of using method G1 alone (i.e. $w_{G1} = 1.0$ and $w_{FP} = 0.0$) versus the n-occurrences of the moderate combination using weights $w_{G1} = 0.6$ and $w_{FP} = 0.4$. This is done for all songs in the music portal data base. The n-occurrence of every song beneath the diagonal line is reduced by using the combination. All large hubs with an n-occurrence bigger than 300 are clearly reduced. The same is true for the majority of hubs with n-occurrences between 200 and 300.

5. CONCLUSION

We were able to show that the so-called hub problem in audio based music similarity indeed does exist in very large data bases and therefore is not an artefact of using limited amounts of data. As a matter of fact, the relative amount and size of hubs is even growing with the size of the data base. On the same very large web shop data base we were able to show that a non-timbre based parameterization of audio similarity (fluctuation patterns) is by far not as prone to hubness as the standard approach of using Mel Frequency Cepstrum Coefficients (MFCCs) plus Gaussian modeling. Extending recent successful work on combining different features to compute overall audio similarity, we were able to show that this not only maintains a high quality of audio similarity but also decisively reduces the hub problem.

The combination result has so far only been shown on the smaller music portal data base, but there is no reason why this should not hold for the larger web shop data. Only limitations in computer run time led us to first evaluate the combination approach on the smaller data set. We are not claiming that our specific combination of features is the best general route towards audio similarity. But we are convinced that going beyond pure timbre-based similarity is able to achieve two goals simultaneously: high quality audio similarity and avoiding the hub problem.

6. ACKNOWLEDGEMENTS

This research is supported by the Austrian Science Fund (FWF, grants L511-N15 and P21247) and the Vienna Science and Technology Fund (WWTF, project "Audiominer").

7. REFERENCES

[1] Aucouturier J.-J., Pachet F.: A scale-free distribution of false positives for a large class of audio similarity measures, *Pattern Recognition*, Vol. 41(1), pp. 272-284, 2007.

[2] Aucouturier J.-J., Defreville B., Pachet F.: The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music, *Journal of the Acoustical Society of America*, 122 (2), 881-891, 2007.

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

[5] Godfrey M.T., Chordia P.: Hubs and Homogeneity: Improving Content-Based Music Modeling, *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'08)*, Philiadelphia, USA, 2008.

[6] Hoffman M., Blei D., Cook P.: Content-Based Musical Similarity Computation Using the Hierarchical Dirichlet Process, *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'08)*, Philiadelphia, USA, 2008.

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

[8] Mandel M.I., Ellis D.P.W.: Song-Level Features and Support Vector Machines for Music Classification, *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, 2005.

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

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

[11] Pampalk E., Flexer A., 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., 2005.

[12] 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.

[13] Pohle T., Schnitzer D., Schedl M., Knees P., Widmer G.: On rhythm and general music similarity, *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR'09)*, Kobe, Japan, 2009.

[14] Penny W.D.: Kullback-Leibler Divergences of Normal, Gamma, Dirichlet and Wishart Densities, Wellcome Department of Cognitive Neurology, 2001.

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