Limitations of interactive music recommendation based on audio content

Arthur Flexer
Austrian Research Institute for
Artificial Intelligence
Vienna, Austria
arthur.flexer@ofai.at

Martin Gasser
Austrian Research Institute for
Artificial Intelligence
Vienna, Austria
martin.gasser@ofai.at

Dominik Schnitzer
Department of Computational
Perception
Johannes Kepler University
Linz, Austria
dominik.schnitzer@ofai.at

ABSTRACT

We present a study on the limitations of an interactive music recommendation service based on automatic computation of audio similarity. Songs which are, according to the audio similarity function, similar to very many other songs and hence appear unwantedly often in recommendation lists keep a significant proportion of the audio collection from being recommended at all. This problem is studied in-depth with a series of computer experiments including analysis of alternative audio similarity functions and comparison with actual download data.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.1 [**Information Systems**]: Information Storage and Retrieval—*Content Analysis and Indexing*; H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

General Terms

Experimentation

Keywords

Music recommendation, audio analysis, hubs

1. INTRODUCTION

This paper reports on limitations of an interactive music recommendation service based on automatic computation of audio similarity (FM4 Soundpark)[1]. Whenever users listen to a song from the data base, they are presented with lists of most similar songs. These lists can be explored interactively by going from song to song and expanding the similarity lists. In theory, any song from the data base should have the same chance to be a member of the similarity lists and hence have the same chance of being exposed to the audience. Compared to the strictly chronological and alphabetical access that was possible before implementation of our music recommender, the average number of distinct songs being downloaded per day did indeed double [6]. However, it seems that certain peculiarities of the audio similarity function prevent a significant proportion of the data base from being recommended. In studying this problem our contribution consists of: (i) a series of computer experiments exploring accessibility of our audio catalogue; (ii) evaluation of an alternative audio similarity function; (iii) analysis of actual download data from the music recommendation service.

2. RELATED WORK

For our music recommendation service we use the de facto standard approach to computation of audio similarity: timbre similarity based on parameterization of audio using Mel Frequency Cepstrum Coefficients (MFCCs) plus Gaussian mixtures as statistical modeling (see Section 5.1). 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. Because of the hub problem hub songs keep appearing unwantedly often in recommendation lists and prevent other songs from being recommended at all.

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. The same has been observed for monophonic sounds from individual instruments [7]. 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 au-

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

thors report an increase in hubness [1], others observed the opposite [8]. Using a Hierarchical Dirichlet Process instead of Gaussians for modeling MFCCs seems to avoid the hub problem altogether [9]. The existence of the hub problem has also been reported for music recommendation based on collaborative filtering instead of on audio content analysis [4].

While many research prototypes of recommendation systems/visualizations of music collections that use content-based audio similarity have been described in the literature (e.g., [10, 16, 12, 15, 11], to name just a few), very little has been reported about successful adoption of such approaches to real-life scenarios. *Mufin*[2] is advertised as a music discovery engine that uses purely content-based methods. MusicIP[3] offers the *Mixer* application that uses a combination of content-based methods and metadata to generate playlists. Bang&Olufsen recently released the Beosound 5[4] home entertainment center, which integrates content-based audio similarity with a simple "More Of The Same Music"-user interface, that allows users to create playlists by choosing an arbitrary seed song.

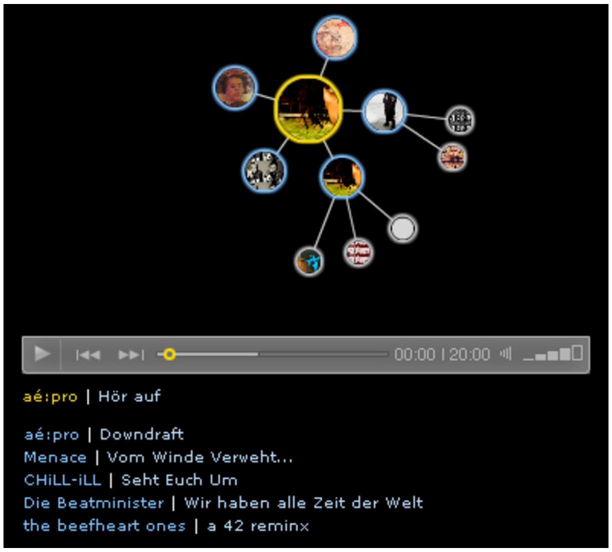
3. THE MUSIC RECOMMENDER

The FM4 Soundpark[5] is a web platform run by the Austrian public radio station FM4, a subsidiary of the Austrian Broadcasting Corporation (ORF). The FM4 Soundpark was launched in 2001 and gained significant public perception since then. Registered artists can upload and present their music free of any charge. After a short editorial review period, new tracks are published on the frontpage of the website. Older tracks remain accessible in the order of their publication date and in a large alphabetical list. Visitors of the website can listen to and download all the music at no cost. The FM4 Soundpark attracts a large and lively community interested in up and coming music, and the radio station FM4 also picks out selected artists and plays them on terrestrial radio. At the time of writing this paper, there are about 11000 tracks by about 5000 artists enlisted in the online catalogue.

Whereas chronological publishing is suitable to promote new releases, older releases tend to disappear from the users' attention. In the case of the FM4 Soundpark this had the effect of users mostly listening to music that is advertised on the frontpage, and therefore missing the full musical bandwidth. To allow access to the full data base regardless of publication date of a song, we implemented a recommendation system utilizing a content-based music similarity measure.

3.1 Web player

The user interface to the music recommender has been implemented as a Adobe Flash-based MP3 player with integrated visualization of similar songs to the currently played one. This web player can be launched from within an artist's web page on the Soundpark website by clicking on one of the artist's songs. Additionally to offering the usual player interface (start, stop, skipping forward/backward) it shows similar songs to the currently playing one in a text list and in a graph-based visualization (see Figure 1). The similar songs are being computed with the audio similarity function described in Section 5.1.



aé:pro | Hör auf

aé:pro | Downdraft
Menace | Vom Winde Verweht...
CHiLL-iLL | Seht Euch Um
Die Beatminister | Wir haben alle Zeit der Welt
the beefheart ones | a 42 reminx

Figure 1: Web player including graph-based visualization

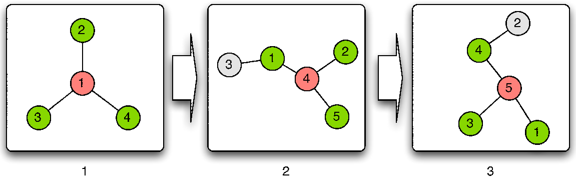


Figure 2: Interaction sequence with graph-based visualization

The graph visualization displays an incrementally constructed nearest neighbor graph (number of nearest neighbors = 5), where nodes having an edge distance greater than two from the central node are blinded out. Figure 2 demonstrates the dynamic behavior of visualization (to simplify things, we have chosen a nearest neighbor number of 3 for this sketch): (1) User clicks on a song, visualization displays song (red) and the 3 nearest neighbors (green), (2) user selects song 4 by clicking on it, the visualization shows the song and its 3 nearest neighbors; note that song 2 – which is amongst the nearest neighbors to song 1 – is also in the nearest neighbor set of song 4; song number 3 (grey) is still displayed since its edge distance to song 4 is equal 2; (3) user selects song 5 as the new center, song 1 – which was nearest neighbor to song 4 in the preceding step – is also nearest neighbor to song 5. In the long run, the re-occurrence of songs in the nearest neighbor sets indicates the existence of several connected components in the nearest neighbor graph.

[2] http://www.mufin.com/
[3] http://www.musicip.com/
[4] http://www.beosound5.com/
[5] http://fm4.orf.at/soundpark

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

Table 1: Percentages of songs belonging to genres with multiple membership allowed. Genres are Hip Hop, Reggae, Funk, Electronic, Pop and Rock.

4. DATA

We now describe the data we used for our computer experiments and the download data we used for analyzing our music recommendation service.

4.1 Data set for computer experiments

For our computer experiments we use a data base of $S_M = 7665$ songs. This data base represents the full FM4 Soundpark collection as of early 2008 when we started working on the music recommender. The data base 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. Percentages across 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%.

4.2 Download data

We collected webserver log files for the time from 6.5.2009 to 16.9.2009 (134 days). The log files register every song that has been actually listened to using the Web player. During the observation period, users have listened to songs 432808 times. The number of distinct songs in the log files is 10099. Please note that we have no knowledge of the number of songs that have not been listened to during the observation period. This is due to the fact that the data base is changing on a daily basis. About ten new songs are added every day and an unregistered number of songs are being removed by the artists themselves.

5. METHODS

We compare two approaches for music similarity based on different parameterizations of the data and also explore a combination of both of them: (i) Mel Frequency Cepstrum Coefficients and Single Gaussians (G1, see Section 5.1) are used in the actual music recommender at the moment; (ii) Fluctuation Patterns (FP, see Section 5.2) are investigated as a possible extension of the current implementation; (iii) linear combinations of G1 and FP are also explored (see Section 5.3). Whereas G1 is 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.

5.1 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. [13]). 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 $d = 20$ MFCCs for all experiments.

A single Gaussian (G1) with full covariance represents the MFCCs of each song [14]. 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 [21]:

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

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:

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

5.2 Fluctuation Patterns and Euclidean Distance (FP)

Fluctuation Patterns (FP) [16] [19] describe the amplitude modulation of the loudness per frequency band and are based on ideas developed in [5]. 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 [17], 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 [23]) 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:

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

An FP pattern is computed from the central minute of each song.

5.3 Combination

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 successful [18, 20]. Such a combination approach was able to rank first at the 2009 MIREX "Audio Music Similarity and Retrieval"-contest[6]. Following previous approaches towards combination of features [18, 17] 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}:

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

6. EVALUATION

Our analysis of the incrementally constructed nearest neighbor graphs concentrates on how likely it is that individual songs are reached when users browse through the graph. To compute the evaluation measures described below, we first compute all nearest neighbor lists with $n = 5$ for all songs using all different methods described in Section 5. 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. For all combinations of G1 and FP, the first n nearest neighbors are the songs with minimum distance according to combination matrix $\bar{M}_C$ (see Equation 5).

Reachability (reach): This is the percentage of songs from the whole data base that are part of at least one of the recommendation lists. If a song is not part of any of the recommendation lists of size $n = 5$ it cannot be reached using our recommendation function.

Strongly connected Component (scc, #scc, s̄c̄c): For our incrementally constructed nearest neighbor graph, a strongly connected component (SCC) is a subgraph where every song is connected to all other songs traveling along the nearest neighbor connections. We use Tarjan's algorithm [22] to find all SCC-graphs in our nearest neighbor graph with $n = 5$. We report the size of the largest strongly connected component as a percentage of the whole data base (scc), the number of additional strongly connected components (#scc) and the average size of all SCCs except the largest one as a percentage of the whole data base ($\bar{s}\bar{c}c$).

n-occurrence (maxhub, hub10, hub20): 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. Since our music recommender always shows the five most similar songs we used $n = 5$. We compute the absolute number of the maximum n-occurrence $maxhub$ (i.e. the biggest hub) and the number of songs of which the n-occurrence is more than ten or twenty times n (i.e. the number of hubs $hub10|20$).

Accuracy (acc): To evaluate the quality of audio similarity achieved using methods G1, FP and its combinations we computed the genre classification performance. Since usually no ground truth with respect to audio similarity exists, genre classification is widely used for evaluation of audio similarity. Each song has been labelled as belonging to one or two music genres by the artists themselves when uploading their music to the FM4 Soundpark (see Section 4.1). High genre classification results indicate good similarity measures. We used nearest neighbor classification as a classifier. 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:

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

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 account 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.

7. RESULTS

7.1 Computer experiments

To analyze the behavior of our music recommender and possible improvements we ran a series of experiments using a number of different weight combinations w_{G1} and w_{FP}. The

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

w_{G1}	w_{FP}	reach	scc	#scc	$\bar{scc}$	maxhub	hub10	hub20	acc
1.0	**0.0**	**65.28**	**29.11**	**409**	**0.04**	**419**	**75**	**24**	**48.47**
0.9	0.1	65.23	30.33	373	0.04	291	74	18	49.84
0.8	0.2	66.24	31.51	373	0.04	195	73	11	49.47
0.7	0.3	67.03	33.87	361	0.04	178	71	10	48.44
0.6	**0.4**	**68.26**	**34.12**	**337**	**0.04**	**180**	**58**	**8**	**47.80**
0.5	0.5	68.78	34.61	358	0.04	167	53	2	46.58
0.4	0.6	69.65	39.13	306	0.04	162	44	2	45.73
0.3	0.7	70.92	39.97	317	0.04	145	33	1	44.93
0.2	0.8	73.24	43.90	308	0.03	118	23	1	43.94
0.1	0.9	77.16	46.52	312	0.03	91	14	0	42.82
0.0	**1.0**	**81.51**	**53.41**	**267**	**0.03**	**63**	**2**	**0**	**38.45**

Table 2: Analysis results 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 7.1 for details.

results given in Table 2 show: the weights w_{G1} and w_{FP}, the reachability $reach$, the size of the largest strongly connected component scc, the number of additional strongly connected components #scc, the average size of all SCCs except the largest one $\bar{scc}$, the absolute number of the maximum n-occurrence $maxhub$ (i.e. the biggest hub), the number of hubs $hub10|20$ and the genre classification accuracy acc.

When discussing our results our attention is on using method G1 alone (i.e. $w_{G1} = 1.0$ and $w_{FP} = 0.0$) since this is what our music recommender does, on using the alternative method FP alone (i.e. $w_{G1} = 0.0$ and $w_{FP} = 1.0$) and on a moderate combination using weights $w_{G1} = 0.6$ and $w_{FP} = 0.4$ since this has been reported to yield good overall quality of audio similarity. This is corroborated when looking at our accuracy results. The moderate combination yields 47.80% accuracy which is at the level of using method G1 alone yielding 48.47%. Using method FP alone gives an accuracy of only 38.45%. The baseline accuracy achieved by always guessing the most probable genre “Electronic” (see Table 1) is 29.11%. Always guessing the two most probable genres “Electronic” and “Rock” yields 36.46%.

When using method G1 alone $reach = 65.28\%$ of the songs can be reached in principle. Whereas using the moderate combination hardly improves this result (68.28%), using method FP alone shows an improved reachability of 81.51%. With the weight w_{FP} for method FP growing, the reachability improves. This seems to be in direct correspondence with our results regarding hubness. It is evident that with the weight w_{FP} for method FP growing, the hubs become smaller and less in number. Whereas using method G1 alone yields a maximum hub of size $maxhub = 419$, the moderate combination already diminishes the biggest hub to a size of 180. Also the number of large hubs decreases: e.g. the number of songs of which the n-occurrence is more than 20 times n ($hub20$) drops from 24 to 8; the number of more moderate sized hubs ($hub10$) is still diminished from 75 to 58. Using method FP alone yields even more improved results concerning hubness.

Results concerning the strongly connected components (SCC) indicate that the situation concerning reachability might be even worse. For all methods, there exists one single largest SCC which increases in size with increasing weight w_{FP} for method FP. This SCC contains 29.11% of all songs for method G1, 34.12% for the moderate combination and 53.41% for method FP. All other existing SCC for all methods are very small ($\bar{scc} = 0.03$ to 0.04% of all songs) and almost negligible. The number of additional SCCs #scc is smallest for method FP. All these results indicate that there exists one large tightly connected subgraph that all other songs lead to when travelling along the nearest neighbor connections. For method G1 implemented in our music recommender this seems to indicate that whereas about two thirds of all songs can be reached in principle, the majority of recommended songs stems from a subset of only about a third of all songs. This subset is slightly larger when using the moderate combination.

To sum up, the accessibility of the full audio data base through our music recommender indeed does seem to be limited. Using fluctuation patterns as an alternative method does improve the accessibility but at the cost of impaired quality of audio similarity. A moderate combination of methods retains the quality in terms of audio similarity but only vaguely improves access to the full data base.

7.2 Analysis of download data

As explained in Section 4.2 the webserver log files only indicate songs that have actually been listened to using the Web player. Therefore we have no knowledge about songs not being listened to. As a matter of fact, we do not even know how many distinct songs existed in the data base during the observation period since the data base is changing every single day. This makes any statement concerning the reachability of songs very hard. However, given that the full size of the data base is about 11000 as of writing this paper (May 2010) and the number of distinct songs in the log files is 10099, it seems that almost all songs have been reached at least once. According to our computer experiments, only about two thirds of the data base are reachable through the music recommender. This discrepancy can be explained since users can start the Web player, and the interactive music recommendation process, from any song in the data base using e.g. the alphabetical list of all songs. Another explanation could be automatic web crawlers copying every single song in the data base. This did actually happen and we tried to clean the log files from all traces of automatic crawlers.

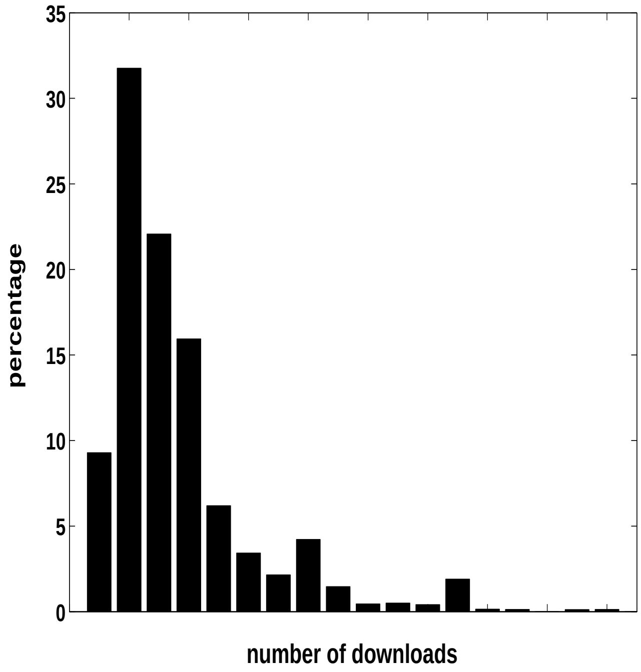


Figure 3: Histogram plot of number of downloads. Bars on x-axis indicate number of downloads (from left to right): 1, 2-5, 6-10, 11-20, 21-30, 31-40, 41-50, 51-100, 101-200, 201-300, 301-400, 401-500, 501-1000, 1001-2000, 2001-3000, 3001-4000, 4001-5000, 5001-50000. Y-axis gives percentage of downloaded songs falling in respective bin.

To find out whether hubs do exist in the download data we made a histogram plot showing how many songs have been downloaded one time, 2 to 5 times, 6 to 10 times, 11 to 20 times, etc (exact bin sizes for the bar plot are given in the caption of Figure 3). The plot indeed shows the typical scale-free distribution: non-hub songs are extremely common and hub songs are extremely rare. There are three especially large hub songs that have been listened to 47547, 27195 and 10065 times. There are two more songs that have been listened to between 4001 and 5000 times, three between 2001 and 3000 times, fifteen between 1001 and 2000 times. The vast majority has been listened to only a single time (9.29% of all songs), 2 to 5 times (31.76%) or 6 to 10 times (22.07%).

To sum up, analysis of actual download data shows a scale-free distribution similar to results from our computer experiments. Comparison in terms of reachability is hard to make due to deficiencies of the log files.

8. CONCLUSION

We have presented a study on the limitations of an interactive music recommendation service based on automatic computation of audio similarity. A series of computer experiments as well as analysis of actual download data shows that a significant proportion of the audio catalogue is being recommended very rarely or not all. About two thirds of the songs can be reached using the automatic music recommendation, but the majority of recommended songs stems from a subset of only about a third of all songs. This is due to songs which are, according to the audio similarity function, similar to very many other songs and hence appear unwantedly often in recommendation lists. Usage of alternative audio similarity functions is able to somewhat improve this situation.

Our music recommendation service is based on timbre similarity using Gaussian mixtures as statistical models. This is the de facto standard approach to computation of audio similarity known to yield high quality results. Any music recommendation service based on this approach is likely to run into the same problems described in this paper.

9. ACKNOWLEDGMENTS

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”).

10. 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] Celma, O.: Music Recommendation and Discovery in the Long Tail, PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

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

[6] Gasser M., Flexer A.: FM4 Soundpark: Audio-based Music Recommendation in Everyday Use, Proceedings of the 6th Sound and Music Computing Conference (SMC'09), 2009.

[7] Gasser M., Flexer A., Schnitzer D.: Hubs and Orphans - an Explorative Approach, Proceedings of the 7th Sound and Music Computing Conference (SMC'10), 2010.

[8] 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), 2008.

[9] 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), 2008.

[10] Knees P., Schedl M., Pohle T., Widmer G.: An

innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web, Proceedings of the ACM Multimedia, 2006.

[11] Lamere P., Eck D.: Using 3d visualizations to explore and discover music, Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07), 2007.

[12] Leitich S., Topf M.: Globe of music - music library visualization using geosom, Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07), 2007.

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

[14] 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), 2005.

[15] Neumayer R., Dittenbach D., Rauber A.: Playsom and pocketsomplayer: Alternative interfaces to large music collections, Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR'05), 2005.

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

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

[18] 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), 2005.

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

[20] 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), 2009.

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

[22] Tarjan R.: Depth-first search and linear graph algorithms, SIAM Journal on Computing, Vol. 1, No. 2, pp. 146-160, 1972.

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