Arthur Flexer*[*] and Dominik Schnitzer*[*†]
[*]Austrian Research Institute for Artificial Intelligence
Freyung 6/6
A-1010 Vienna, Austria
arthur.flexer@ofai.at
[†]Department of Computational Perception
Johannes Kepler University Linz
Altenberger Str. 69
A-4040 Linz, Austria
dominik.schnitzer@jku.at

Effects of Album and Artist Filters in Audio Similarity Computed for Very Large Music Databases

In music information retrieval, one of the central goals is to automatically recommend music to users based on a query song or query artist. This can be done using expert knowledge (e.g., www.pandora.com), social meta-data (e.g., www.last.fm), collaborative filtering (e.g., www.amazon.com/mp3), or by extracting information directly from the audio (e.g., www.muffin.com). In audio-based music recommendation, a well-known effect is the dominance of songs from the same artist as the query song in recommendation lists.

This effect has been studied mainly in the context of genre-classification experiments. Because no ground truth with respect to music similarity usually exists, genre classification is widely used for evaluation of music similarity. Each song is labelled as belonging to a music genre using, e.g., advice of a music expert. High genre classification results indicate good similarity measures. If, in genre classification experiments, songs from the same artist are allowed in both training and test sets, this can lead to over-optimistic results since usually all songs from an artist have the same genre label. It can be argued that in such a scenario one is doing artist classification rather than genre classification. One could even speculate that the specific sound of an album (mastering and production effects) is being classified. In Pampalk, Flexer, and Widmer (2005) the use of a so-called "artist filter" that ensures that a given artist's songs are either all in the training set, or all in the test set, is proposed. Those authors found that the use of such an artist filter can lower the classification results quite considerably (as much as from 71 percent down to 27 percent, for one of their music collections). These over-optimistic accuracy results due to not using an artist filter have been confirmed in other studies (Flexer 2006; Pampalk 2006). Other results suggest that the use of an artist filter not only lowers genre classification accuracy but may also erode the differences in accuracies between different techniques (Flexer 2007).

Computer Music Journal, 34:3, pp. 20–28, Fall 2010
© 2010 Massachusetts Institute of Technology.

All these results were achieved on rather small databases (from 700 to 15,000 songs). Often whole albums from an artist were part of the database, perhaps even more than one. These specifics of the databases are often unclear and not properly documented. The present article extends these results by analyzing a very large data set (over 250,000 songs) containing multiple albums from individual artists. We try to answer the following questions:

1. Is there an album and artist effect even in very large databases?
2. Is the album effect larger than the artist effect?
3. What is the influence of database size on music recommendation and classification?

As will be seen, we find that the artist effect does exist in very large databases, and the album effect is bigger than the artist effect.

Data

For our experiments we used a data set $D(ALL)$ of $S = 254{,}398$ song excerpts (30 seconds each) from a popular Web store selling music. The freely

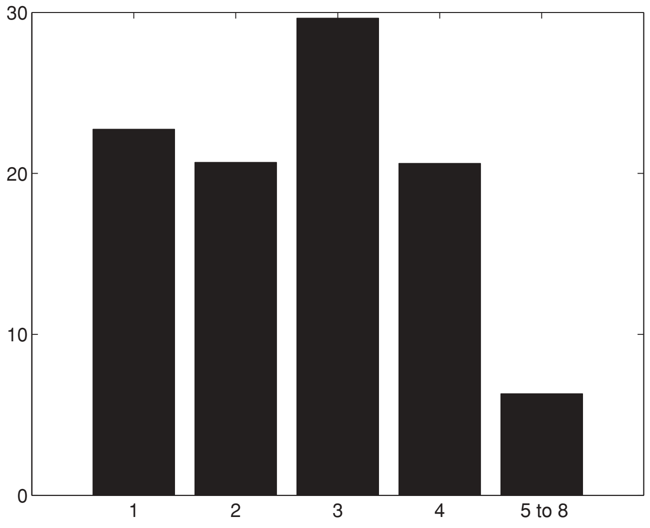


Figure 1. Percentages (y-axis) of albums having 1, 2, 3, 4, or 5 to 8 genre labels (x-axis).

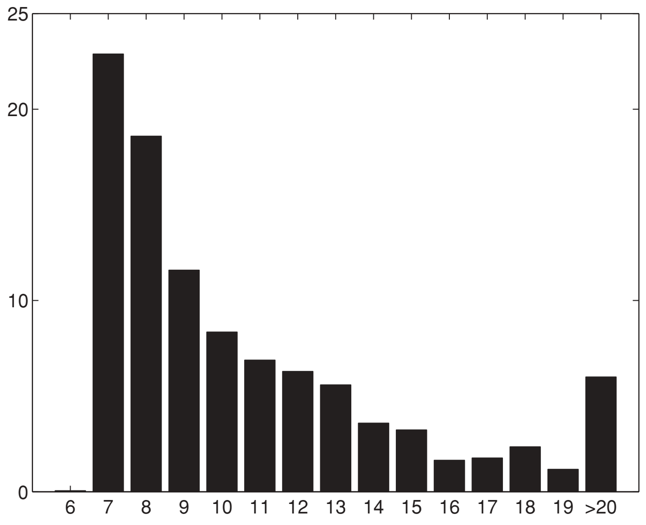


Figure 2. Percentages (y-axis) of artists having 6, 7, ..., 19, or 20 to 29 albums (x-axis).

Table 1. Percentages of Songs Belonging to the 22 Genres, with Multiple Membership Allowed

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

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 HTML code. The excerpts are from $U = 18{,}386$ albums from $A = 1{,}700$ artists. From the 280 existing different hierarchical genres, only the $G = 22$ 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. (Each song is allowed to belong to more than one genre, hence the percentages in Table 1 add up to more than 100 percent.) The genre information is identical for all songs on an album. The numbers of genre labels per album are given in Figure 1. Our database was set up so that every artist contributes between 6 and 29 albums (see Figure 2).

To study the influence of the size of the database on results, we created random non-overlapping splits of the entire data set: $D(1/2)$ is two data sets with the mean number of song excerpts = 127,199; $D(1/20)$ is twenty data sets with the mean number of songs excerpts = 12,719.9; and $D(1/100)$ is one hundred data sets with the mean number of songs excerpts = 2,543.98. An artist with all their albums is always a member of a single data set.

Methods

We compare two approaches based on different parameterizations of the data. Whereas mel-frequency cepstral coefficients (MFCCs) are a relatively 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. It is our hypothesis that MFCCs are more prone to pick up production and mastering effects of a single album as well as the specific "sound" of an artist (voice, instrumentation, etc.).

Mel-Frequency Cepstral Coefficients and Single Gaussians (G1)

We use the following approach to 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 similarity matrix between all songs using the symmetrized Kullback-Leibler divergence between respective G1 models.

The 30-second song excerpts in MP3 format are recomputed to 22,050-Hz mono audio signals. We divide the raw audio data into non-overlapping frames of short duration and use MFCCs 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., Logan 2000). The frame size for computation of MFCCs for our experiments was 46.4 msec (1,024 samples). We used the first 25 MFCCs for all our experiments. A G1 with full covariance represents the MFCCs of each song (Mandel and Ellis 2005). 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 (Penny 2001):

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

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:

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

Fluctuation Patterns and Euclidean Distance (FP)

FP (Pampalk 2001; Pampalk, Rauber, and Merkl 2002) describe the amplitude modulation of the loudness per frequency band and are based on ideas developed in Fruehwirt and Rauber (2001). For a given music collection of songs, computation of music similarity based on FPs consists of the following steps:

1. For each song, compute an FP.
2. Compute a similarity matrix between all songs using the Euclidean distance of the FP patterns.

Closely following the implementation outlined in Pampalk (2006), 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 – 10 Hz) for each segment and frequency band; (iii) weighting the modulation frequencies based on a model of perceived fluctuation strength; and (iv) applying filters to emphasize certain patterns and smooth the result. The resulting FP is a 12 (frequency bands according to twelve critical bands of the Bark scale [Zwicker and Fastl 1999]) × 30 (modulation frequencies, ranging from 0 to 10 Hz) matrix for each song. The distance between two FPs i and j is computed as the Euclidean distance:

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

Table 2. Nearest Neighbor Characteristics for Methods G1 and FP

Method	*1st AL*	*1st AR*	*AL prec*	*AR prec*
G1	27.87	35.76	13.86	8.14
FP	2.24	26.85	0.90	1.63

1st AL = same album; 1st AR = same artist; ALprec = album precision; AR = artist precision.

where k indexes the frequency band and l indexes the modulation frequency.

Results

In what follows, we present our results concerning the effect of album and artist filters on album and artist precision as well as on genre classification performance. This is done for the full database and all its subsets to study the influence of the database size.

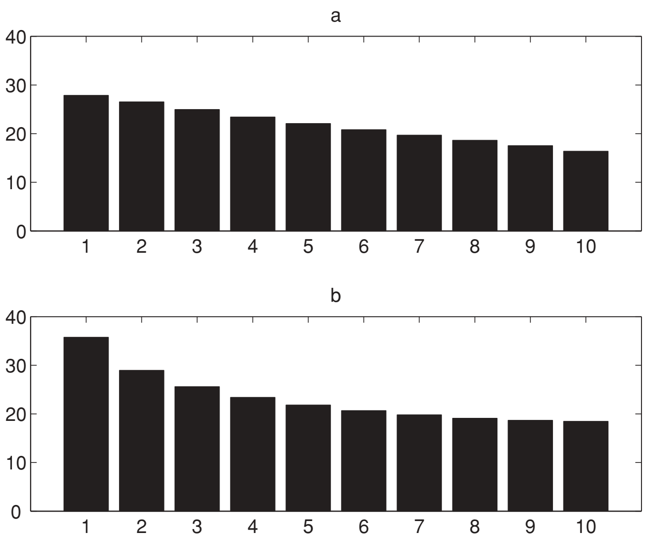
Album/Artist Precision

For the Full Database, $D(ALL)$

For every song in the database $D(ALL)$, we computed the first nearest neighbor for both methods G1 and FP. For method G1, the first nearest neighbor is the song with minimum Kullback-Leibler divergence (Equation 2) from the query song. For method FP, the first nearest neighbor is the song with minimum Euclidean distance of the FP pattern (Equation 3) from the query song. We then computed the percentage of instances in which the first nearest neighbor is from the same album (1st AL) or from other albums by the same artist (1st AR) as the query song (see Table 2).

For method G1, 27.87 percent are from the same album and 35.76 percent from other albums by the same artist. On average, there are 13.46 songs on an album and 131.2 songs from one artist. Considered that there are always more than 250,000 songs from other artists, it is quite astonishing that only in 36.37 percent a song from a different artist turns up as a first nearest neighbor. For method FP, percentages are quite a bit lower, with only 2.24 percent from the same album and 26.85 percent from other albums by the same artist.

Figure 3. Percentage (y-axis) of songs whose n*th nearest neighbor (*n = *1, . . . , 10) is from the same album (a), or from another album by the same artist (b), for method G1.*



We also computed lists of the nth nearest neighbors ($n = 1, \ldots, 10$) for every song in the database $D(ALL)$, for both methods, G1 and FP. We then computed the percentage of instances in which members of these lists of size $n = 1, \ldots, 10$ are from the same album or from other albums by the same artist as the query song (see Figure 3 for method G1 and Figure 4 for method FP). As can be seen, the degradation of percentages with growing list size for method G1 is quite graceful (see Figure 3): For example, the percentage of instances where the five nearest neighbors are from the same album is at 22.07 percent compared to 27.87 percent for the first nearest neighbor. The percentage of instances where the five nearest neighbors are from other albums by the same artist is at 21.83 percent compared to 35.76 percent for the first nearest neighbor. There is a similar behavior for method FP at a generally lower level (see Figure 4).

Next we computed the album and artist precision at n. Album precision at n (AL prec) is the percentage of songs from the album in a list of the n nearest neighbors, with n being equal to the number of other

Figure 4. Percentage (y-axis) of songs whose n*th nearest neighbor* (n = *1, . . . , 10) is from the same album (a), or from another album by the same artist (b), for method FP.*

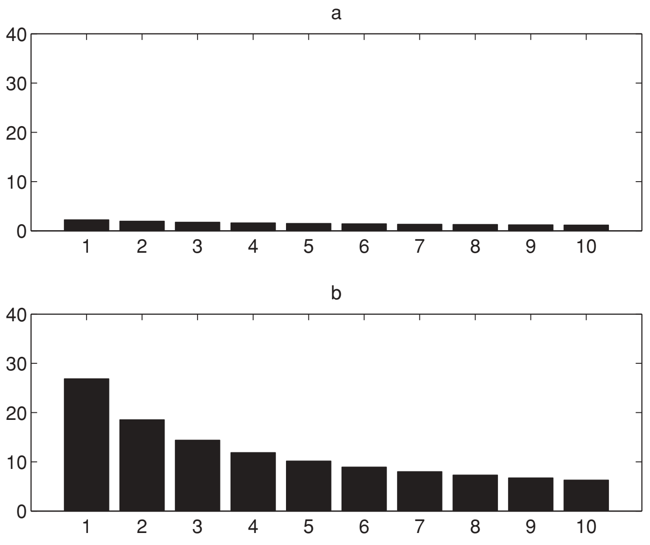
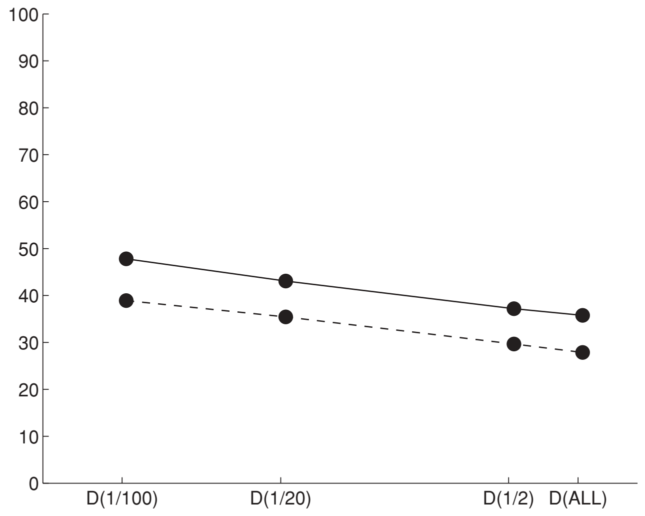


*Figure 5. Percentage (y-axis) of first nearest neighbors from the same album (dashed line), and from other albums by the same artist (solid), for different sizes of the data set (*x*-axis, log scale), using method G1.*



songs in the same album as the query song. Artist precision at n (AR prec) is the percentage of songs from the artist in a list of the n nearest neighbors, with n being equal to the number of other songs from the same artist as the query song. For $D(ALL)$ and method G1, album precision is at 13.86 percent and artist precision at 8.14 percent (see Table 2). Precision values for method FP are very small. To sum up, there is both an album and an artist effect in nearest-neighbor-based music recommendation for method G1. For this timbre-based method, the album effect is even bigger, relatively speaking, than the artist effect, given that, on average, the number of songs on the same album is only a tenth of the number of songs on other albums by the same artist. For method FP, there is only a smaller artist effect, and no album effect.

Influence of the Database Size

We repeated the experiments for all the subsets of the database as described in the Data section. The results depicted in Figures 5, 6, 7 and 8 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 $D(ALL)$. Note that in all these figures, the x-axis is given in log scale to better depict the large range of values of the different sizes of data sets (from 2,543.98 for $D(1/100)$ to 254,398.00 for $D(ALL)$). The percentage of songs whose first nearest neighbor is from the same album decreases from 38.91 percent for $D(1/100)$ to 27.87 percent for $D(ALL)$, for method G1 (Figure 5). There is a parallel decrease for the first nearest neighbor from other albums from the same artist for method G1 (see Figure 5). A similar decrease at lower levels can be seen for method FP (see Figure 6). As the data sets get larger, there clearly seems to be an increased probability that songs from other artists are more similar to the query song than are songs from the same album or artist.

Album and artist precision also decrease with increasing size of data set. For method G1, artist precision drops from 35.99 percent for $D(1/100)$ to 8.14 percent for $D(ALL)$ even falling below album precision (see Figure 7). For method FP, artist precision drops from 19.19 percent for $D(1/100)$ to 1.63 percent for $D(ALL)$ which is at the same low level as album precision (see Figure 8). To sum up, both first-nearest-neighbor rates and precision values are overestimated when smaller data sets are used.

Genre Classification

For the Full Database, $D(ALL)$

We also did experiments on the influence of album and artist filters on genre classification performance.

Figure 6. Percentage (y-axis) of first nearest neighbors from the same album (dashed line), and from other albums by the same artist (solid), for different sizes of the data set (x-axis, log scale), using method FP.

Figure 7. Precision (y-axis) of album (dashed line) and artist (solid) for different sizes of the data set (x-axis, log scale), using method G1.

Figure 8. Precision (y-axis) of album (dashed line) and artist (solid) for different sizes of the data set (x-axis, log scale), using method FP.

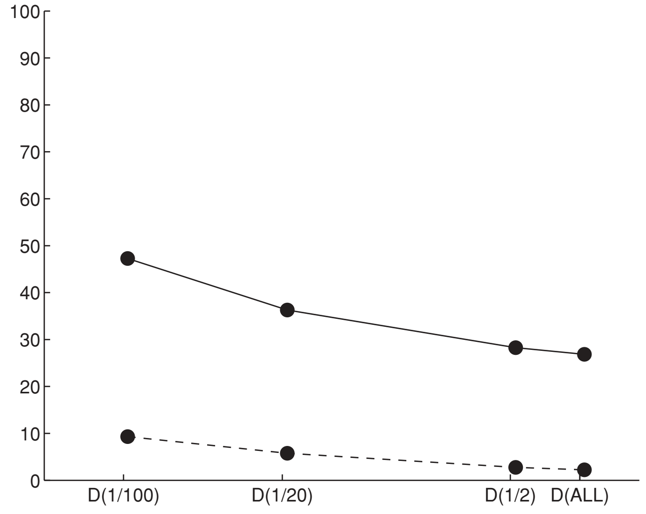
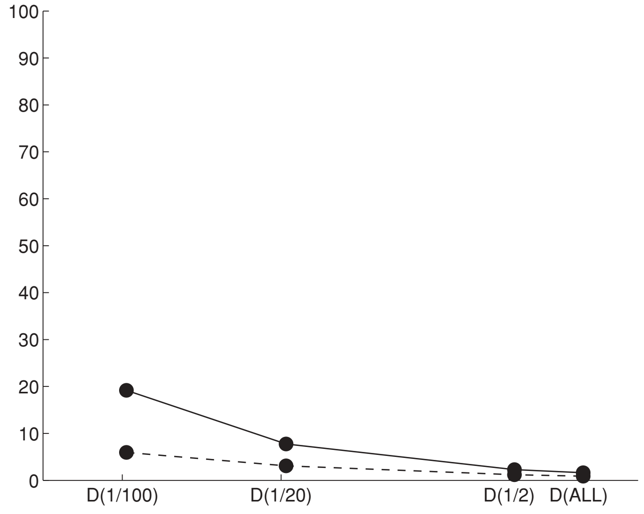


Figure 6.



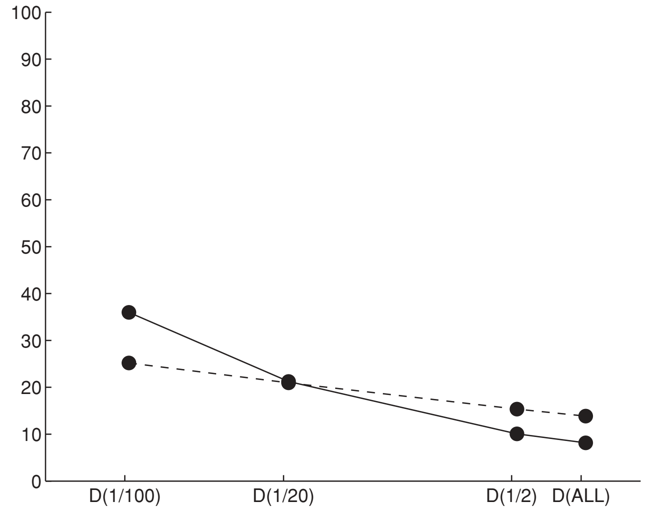


Figure 7.

For the classifier, we used nearest-neighbor classification. For every song in the database $D(ALL)$, we computed the first nearest neighbor for both methods G1 and FP. For method G1, the first nearest neighbor is the song with minimum Kullback-Leibler divergence (Equation 2) to the query song. For method FP, the first nearest neighbor is the song with minimum Euclidean distance of the FP pattern (Equation 3) to the query song. When using an album filter (ALF), all other songs from the same album as the query song were excluded from becoming the first nearest neighbor. When using an artist filter (ARF), all other songs from the same artist as the query song were excluded from becoming the first nearest neighbor. When using no filter (NOF), any song was allowed to become the first nearest neighbor. 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:

$$acc(s_{query}, s_{nn}) = \frac{|g_{query} \cap g_{nn}|}{|g_{query} \cup g_{nn}|} \qquad (4)$$

with g_{query} being a set of all genre labels for the query song, g_{nn} being the analogous set for the 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}. The latter is done to penalize nearest-neighbor songs that have high numbers of genre labels. The range of values for accuracy is between 0 and 1. The baseline accuracy achieved by always guessing the three most probable genres ("Rock," "Pop," and "Alternative Rock," see Table 1) is 32.42 percent. We decided to use three genres, rather than some other number of genres, for this baseline accuracy, because the greatest

Figure 9. Accuracy (y-axis) for no filter (dotted line), album filter (dashed line), and artist filter (solid line), for different sizes of the data set (x-axis, log scale), using method G1.

Figure 10. Accuracy (y-axis) for no filter (dotted line), album filter (dashed line), and artist filter (solid line), for different sizes of the data set (x-axis, log scale), using method FP.

Table 3. Average Accuracies for G1 and FP without (NOF) and with Album Filter (ALF) and Artist Filter (ARF)

Method	*NOF*	*ALF*	*ARF*
G1	68.98	56.07	35.98
FP	44.35	43.03	28.96

number of songs are labeled with three genres (see Figure 1). Average accuracy results for methods G1 and FP are given in Table 3. Without using any filter (NOF), G1 clearly outperforms FP (68.98 percent vs. 44.35 percent). Using an album filter (ALF) strongly degrades the performance of G1 down to 56.07 percent, but hardly impairs method FP. Using an artist filter (ARF) further degrades the performance of G1 but also of FP. The difference between G1 and FP is now much closer (35.98 percent vs. 28.96 percent). However, method G1 barely outperforms the baseline accuracy of 32.42 percent and method FP clearly falls below it.

That is, not using any filter yields very over-optimistic accuracy results. As a matter fact, results after artist filtering are very close to, or even below, baseline accuracy. There is both an album and an artist filter effect for G1. There is only an album filter effect for FP. Using filters diminishes the differences in accuracies between methods G1 and FP, since filters have a bigger impact on G1 than FP.

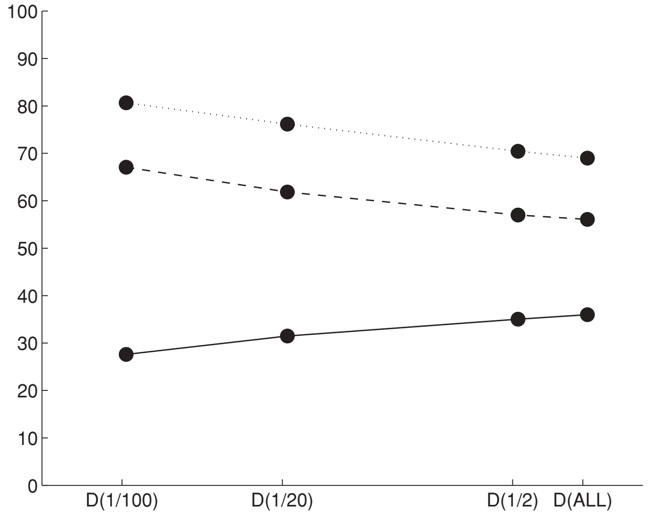


Figure 9.

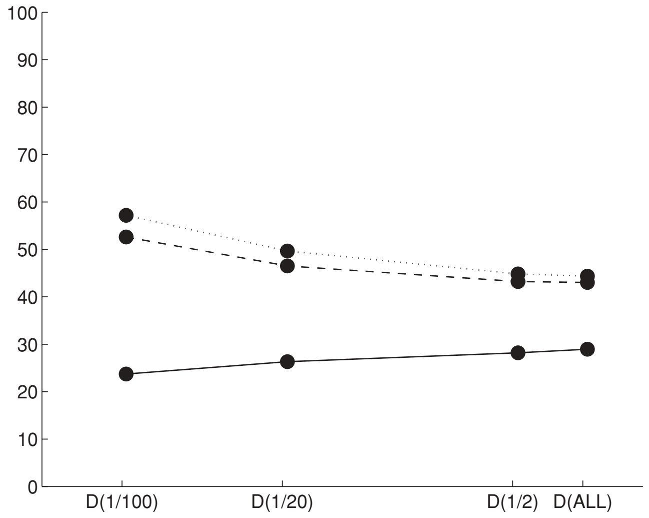


Figure 10.

Influence of the Size of the Database

We repeated the experiments for all the subsets of the database as described in the section "Data." The results depicted in Figures 9 and 10 show mean accuracy values over 100 ($D(1/100)$), 20 ($D(1/20)$), and 2 ($D(1/2)$) data sets or the respective single result for the full data set $D(ALL)$. For both methods G1 and FP, the accuracy without using a filter (dotted lines in Figures 9 and 10) decreases with increasing size of data set. For G1, the decrease is from 80.65 percent for $D(1/100)$ to 68.98 percent for $D(ALL)$. For FP, it is from 57.19 percent for $D(1/100)$ to 44.35 percent for $D(ALL)$. There is an almost parallel decrease in accuracy when using album filters (dashed lines in Figures 9 and 10). For both methods G1 and FP, the accuracy when using an artist filter (solid lines in Figures 9 and 10) increases with increasing size of data set. For G1, the increase is from 27.60 percent for $D(1/100)$ to 35.98 percent for $D(ALL)$. For FP, it is from 23.72 percent for $D(1/100)$ to 28.96 percent for $D(ALL)$.

How can this contrary behavior of decreasing accuracy for no filter and album filter versus increasing accuracy for artist filters be explained? Larger data sets allow for a larger choice of songs to become the first nearest neighbor. This larger choice of songs can come with correct or incorrect genre labels. If we use artist filters, this larger choice seems to make it more probable that a song with the correct genre label is first nearest neighbor. Otherwise we would not see the increase in accuracy. If we use no filter or only an album filter, the larger choice seems to interfere with the songs from the same artist still in the database. Songs from the larger choice sometimes end up being first nearest neighbor instead of a song from the same artist as the query song. Because most songs from an artist share the same genre labels, the larger choice in this case diminishes the accuracy.

In other words, there clearly is an influence of the database size on accuracy performance. Small data sets are too pessimistic when artist filters are used. But they are overly optimistic if no filters are used, or only album filters.

Conclusion

There are clearly both an album effect and an artist effect in music recommendation, even in very large databases. For the timbre-based method G1, about one third of the first recommendations are from the same album and about another third from other albums by the same artist as the query song. Considering that every artist has multiple albums in the database and that an album contains only about 13 songs on average, the album effect is bigger, relatively speaking, than the artist effect. This suggests that the direct representation of the spectral information is more sensitive to the specific "sound" of an album. This "sound" can be due to instrumentation, production and mastering effects, and other aspects that remain constant for all songs on an individual album.

Note that we have no way to know whether an artist is working with the same recording studio or sound engineer for more than one album, nor whether instrumentation, style, etc., change from album to album. For method FP, there is only a smaller artist effect, and no album effect. This suggests that the more abstract signal representation of the fluctuation patterns is not sensitive to the "sound" of individual albums. But it is still able to model the common musical "language" of an artist across different albums. Our experiments also show that album and artist effects in music recommendation are overestimated when smaller data sets are being used. Because most research on artist filters so far concentrated on genre classification, we also did large scale experiments on classification accuracy. We corroborated earlier results that not using any filter yields very over-optimistic accuracy results. Using artist filters even reduces results close to or below baseline accuracy. As reported before, using artist filters also diminishes the differences in accuracies between methods that are affected distinctly by filtering. Additionally, there clearly is an influence of database size on accuracy performance.

As with all large-scale performance studies, there remains the question as to how representative and universally valid our results are. We are convinced that our database is representative of music that is generally listened to and available in the Western hemisphere, since it is a large and random subset of about 5 million songs from a popular Web store. As to the methods employed, we chose one method that closely models the audio signal and one that extracts information on a somewhat higher level. It is our guess that other methods' performance will be close to either of our methods, depending on their level of closeness to the analyzed audio signal. Systems using a Gaussian representation of spectral features together with the Kullback-Leibler divergence (like our method G1) regularly rank in the top places of the yearly Music Information Retrieval Evaluation Exchange (MIREX, www.music-ir.org/mirexwiki/) (Downie 2008), which is a community-based framework for the formal evaluation of music information retrieval systems and algorithms. This shows that at least one of our chosen music recommendation methods belongs to the most successful approaches in music similarity. The choice of our methods was also influenced by considerations of computability. After all, 250,000 song excerpts are a lot of data to analyze, and both our methods can be implemented very efficiently. Using nearest-neighbor methods for music recommendation seemed to be the obvious choice.

With audio-based music recommendation maturing to the scale of the Web, our work provides important insight into the behavior of music similarity for very large databases. Even with hundreds of thousands of songs, album and artist effects remain a problem.

Acknowledgments

Parts of this work were funded by the Austrian National Science Foundation (FWF) (Projects P21247 and L511-N15) and by the Austrian Research Promotion Agency (FFG) (Bridge-project 815474).

References

Downie, J. 2008. "The Music Information Retrieval Evaluation Exchange (2005–2007): A Window into Music Information Retrieval Research." *Acoustical Science and Technology* 29(4):247–255.

Flexer, A. 2006. "Statistical Evaluation of Music Information Retrieval Experiments." *Journal of New Music Research* 35(2):113–120.

Flexer, A. 2007. "A Closer Look on Artist Filters for Musical Genre Classification." *Proceedings of the Eighth International Conference on Music Information Retrieval (ISMIR '07)*. Vienna: Austrian Computer Society, pp. 341–344.

Fruehwirt, M., and A. Rauber. 2001. "Self-Organizing Maps for Content-Based Music Clustering." *Proceedings of the Twelth Italian Workshop on Neural Nets*. London: Springer, pp. 228–233.

Logan, B. 2000. "Mel Frequency Cepstral Coefficients for Music Modeling." *Proceedings of the International Symposium on Music Information Retrieval (ISMIR '00)*. Available at http://ismir2000.ismir.net/. Last accessed 26 April 2010.

Mandel, M., and D. Ellis. 2005. "Song-Level Features and Support Vector Machines for Music Classification." *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR '05)*. London: University of London, pp. 594–599.

Pampalk, E. 2001. "Islands of Music: Analysis, Organization, and Visualization of Music Archives." MSc thesis, Technical University of Vienna.

Pampalk, E. 2006. "Computational Models of Music Similarity and Their Application to Music Information Retrieval." Doctoral thesis, Vienna University of Technology.

Pampalk, E., A. Flexer, and G. Widmer. 2005. "Improvements of Audio-Based Music Similarity and Genre Classification." *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR '05)*. London: University of London, pp. 628–633.

Pampalk, E., A. Rauber, and D. Merkl. 2002. "Content-Based Organization and Visualization of Music Archives." *Proceedings of the Tenth ACM International Conference on Multimedia*. New York: ACM, pp. 570–579.

Penny, W. D. 2001. "Kullback-Leibler Divergences of Normal, Gamma, Dirichlet and Wishart Densities." Technical report, Wellcome Department of Cognitive Neurology, University College London.

Zwicker, E., and H. Fastl. 1999. *Psychoacoustics, Facts and Models*. Springer Series of Information Sciences, volume 22. Berlin: Springer, 2nd edition.