Post Processing Music Similarity Computations

Tim Pohle
Department of Computational Perception
Johannes Kepler University Linz, Austria
tim.pohle@jku.at

Abstract

Today, among the best-performing algorithms for music similarity computations are algorithms based on Mel Frequency Cepstrum Coefficients (MFCCs). In these algorithms, each music track is modelled as a Gaussian Mixture Model (GMM) of MFCCs. The similarity between two tracks is computed by comparing their GMMs. As pointed out in [1, 2, 3], the distance space obtained this way has some undesirable properties. In this MIREX'06 submission, a technique has been implemented that aims to correct such anomalies to a certain extent[1]. The described algorithm ranked second (out of six) in the MIREX evaluation based on human listeners (note that the differences between the top-five ranked algorithms are not statistically significant). There is indication that it works better for artist identification than the other submitted algorithms.

1. Feature Extraction and Basic Distance Computation

The basic feature extraction process is quite similar to the one in [5]. It was chosen because its good tradeoff between runtime and quality, and because algorithms based on related techniques yielded good results in MIREX'05.

- The input wave files (22.050 Hz sampling rate, mono) are divided into frames of 512 samples length, with 256 samples overlap, disregarding the first and last 30 seconds.

- The number of frames corresponding to 2 minutes (i.e. 20.672 frames) are used for feature extraction. In the submitted algorithm, these frames are not chosen to be consecutive. Instead, the length of the wave data is divided into 20.672 fragments of equal length, and from each of those fragments, randomly 512 consecutive samples are chosen for feature extraction. By randomly choosing the frames possible aliasing effects with respect to the track's meter are reduced. It seems that this approach yields better results than choosing the frames in a fully random manner, or taking all frames from the two minutes in the middle of the track.

- From the chosen frames, 25 MFCCs are computed.

- A song is represented as the overall mean of the MFCCs, and the full covariance matrix.

The feature extraction process was implemented using the MA-Toolbox ([6]). Two songs are compared by the Kullback-Leiber (KL) distance. If the inverse of a song's covariance matrix can not be found, it is assumed that it is dissimilar to all other songs.

One drawback of this technique is that it does not take into consideration the temporal order of frames, thus aspects related to time are not modelled. An approach to add time-dependent features is propsed in [2]. However, the version used here it is a good starting point for the post-processing step described in the next section.

2. Post Processing

As pointed out in [1, 2, 3], the distance space obtained with such an algorithm has some undesired properties. Some tracks may be very similar to many other tracks (so-called "hubs" [3], e.g. in a collection containing about 2.500 tracks, one track may appear in the ten nearest neighbours of 250 other tracks). Also, there may be tracks that are *not* similar to other tracks. Reducing the effects of these properties may improve the quality of the algorithm's output. One could think of various ways to approach this, including those described below.

2.1. Calculations on the Distance Matrix

After computing the distance matrix of all tracks in a collection, all values in a column and all values in a row are divided by the distance of the (e.g.) 25th nearest neighbour of the track that corresponds to the index of this row or column, respectively. On in-house collections, this typically improved the leave-one-out 5 nearest neighbour (NN) genre classification by one or more percent points, depending on the collection.

In the MIREX'06 audio similarity contest, it is not allowed to use such knowledge about the whole collection for which the distances should be computed (the *test collection*). Thus, it would be necessary to provide the models of another collection (the *reference collection*), that serves for reference during the (pairwise) distance computations of the contest.

As it turned out in preliminary experiments, using a reference collection for this approach did not work satisfactorily

[1] For more detailed evaluations, please refer to [4]

in all cases. Thus, no further investigations into this directions were made[2].

2.2. Counting the Number of Appearances in k-NN Sets

Another possible approach is to determine for each piece A of the test collection a measure which is called k*-occurrences*. For calculating this, it is assumed that A is part of the reference collection. Count how often it appears in the set of k nearest neighbours (e.g. $k = 10$) of tracks in the reference collection. The k-occurrence may then be used to either filter out those pieces with a high k-occurrence, or to accordingly modify the distances of track A to other tracks in the test collection. However, this approach has not been tested yet because the approach described in the next section showed to be effective in preliminary experiments. Thus, an evaluation of this approach is left as future work [2] .

2.3. Proximity Verification

The basic idea behind this approach is to replace the absolute distance (obtained by computing the KL distance) by a relative distance based on the ranking of tracks. Thus, the divergence of track A and B, denoted $D(A, B)$, is k, where B is the k nearest neighbour of A (i.e. there are $D(A, B)-1$ other tracks in the collection that are more similar to A than B, measured by the KL distance).

As in general $D(A, B) \neq D(B, A)$, a symmetric distance measure is obtained by defining

$$D_{PV} = D(A, B) + D(B, A)$$

This approach is called *proximity verification* here, as D_{PV} has a low value only if both A is a close neighbour of B *and* B is a close neighbour of A. Obviously, the average value of the divergence between track A and all tracks T_i in the collection $\frac{1}{N} \sum_{i=1}^{N} D(A, T_i)$ is the same, regardless if a track A is determined by initial similarity algorithm as being similar to many other tracks, or if it even is regarded as being dissimilar to all other tracks. Thus, these effects are reduced.

Again, in the MIREX contest, as it is not allowed to use knowledge about the test collection, $D_{PV}(A, B)$ has to be determined on the models of a reference collection, with the models of tracks A and B being the only information available from the test collection. In the submitted implementation, this results in values for $D(A, B)$ that are usually not integers, as the rank of B is interpolated.

2.3.1. *Preliminary Evaluation*

Preliminary experiments on in-house test collections and on the ISMIR'04 Genre Classification Contest Training Collection indicate that this approach also is beneficial when using a reference collection for post processing instead of using the test collection itself. Another tendency seems to be that merits are larger when using a larger reference collection. Examples of evaluation results with the largest possible reference collection consisting of all available tracks (more than 8.000, including the tracks in the test collection, and some tracks multiple times) are given below [3] . The test collection consists of 2447 tracks from 22 genres.

Tables 1 and 2 show that the percentage of the closest tracks that are in the same genre is improved by applying proximity verification. These results are quite promising; however, on the ISMIR'04 Genre Classification Contest Training Collection, the corresponding 5-NN value was only improved by approximately 1.4% before artist filtering.

No Artist Filter	5	10	20	50
Basic	67.9%	60.8%	51.8%	39.1%
ProxiVeri	73.2%	66.5%	56.1%	42.8%

Table 1. Percentage of closest n tracks of each track that are in the same genre for $n = \{5, 10, 20, 50\}$, *without* artist filter. $Basic$ is the basic algorithm described in Section 1, $ProxiVeri$ is the same algorithm with additional proximity verification.

With Artist Filter	5	10	20	50
Basic	28.6%	26.9%	25.0%	21.3%
ProxiVeri	31.2%	29.8%	27.8%	24.1%

Table 2. Percentage of closest n tracks of each track that are in the same genre for $n = \{5, 10, 20, 50\}$, *with* artist filter. Same algorithms as in Table 1.

Figure 1 indicates that proximity verification has a positive effect both on the number of tracks that are considered as being similar to many other tracks, and on the number of tracks that are similar to only few other tracks in the collection. Also, there is a positive effect on the number of track triples where the triangle inequality is fulfilled (e.g. the number of track triples where it is not fulfilled dropped from about 41% to about 32%).

3. Conclusions

The main motivations for submitting this algorithm to MIREX are to compare the properties of the resulting similarity space to the other submissions, and also to get feedback about how human evaluators assess its performance. Future work includes an in-depth evaluation of the proposed post processing approaches, and – most importantly – their application to other feature extraction routines and distance measures, most notably such that model time aspects of the music signal.

[2] In the meantime, further work is available [4].

[3] More detailed evaluations are future work. In particular, evaluations with reference collections that are smaller than the test collection are important.

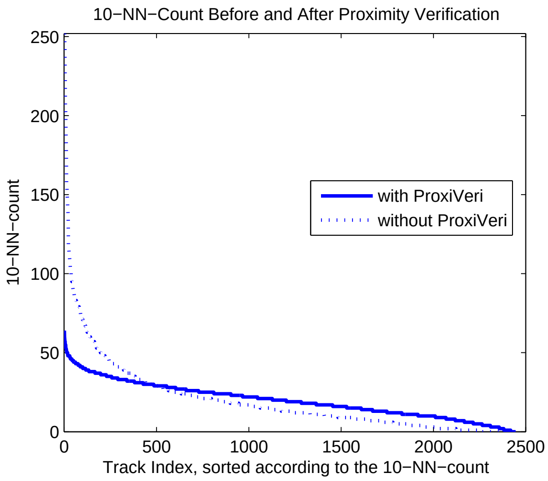

Figure 1. 10-occurrences before and after proximity verification. The y-axis is cut off at 252, corresponding to the second highest value before proximity verification. The highest value is 359. The highest value after proximity verification is 64.

4. Comments on the MIREX results

This final section contains a brief discussion of the MIREX results. The most important performance measure is the listening test, as the other measures do not directly take into account how the songs that were rated "most similar" actually sound.

4.1. Listening Test Results

The algorithm described in this abstract ("TP") ranked second in the listening tests. However, from the six submitted algorithms, the differences of the top five ranked algorithms were not statistically significant. This was the first MIREX AudioSim with a listening test, so there were no previous results that could be used to design the experimental setup. Knowing that the state-of-the-art algorithm have close scores in this listening test, for future evaluations a modified setup (e.g. more queries, a more diverse music collection) could be considered to improve the significance of the results.

4.2. Distance Matrix Statistics

Several metrics were calculated on the distance matrices that were produced by each of the submitted algorithms. Many of them use the genre labels of the songs. Unfortunately, the number of tracks per genre were very skewed, and the genre labels probably are not assigned very carefully. For example, Britney Spears was classified as Rock, and the music from the genre Rap & Hip-Hop and Rock constituted more than 70% of the collection. Thus, the results of these metrics should be regarded very cautiously.

In micro-averaged 5-NN genre classification, TP ranked first when no artist filter was used. If the genre labels are considered near noise, then this result may indicate that TP is better suited for artist identification than the other submissions, which is consistent with the results of last year's MIREX, and the observation that TP produced the lowest Artist / Genre ratio (i.e., the ratio of the distance between tracks of the same artist and the distance of tracks of the same genre).

4.2.1. Always Similar

One of the motivations for using proximity verification was to remove hubs. To my knowledge, only one of the other submitted algorithms (G1C) makes use of a technique that is known to potentially produce hubs. Depending on if the cover song tracks were also considered for calculating the k-occurrences, the statistics were different. When they are considered, the highest 50-occurrence of G1C is 2058, indicating the presence of a hub. It turns out that when not regarding the cover song tracks, there seems to be no hub for G1C. In the latter case, the highest 50-occurrence of G1C is 434, which is within the range of the other algorithms. The corresponding values of TP were 557 and 554, respectively. These values seem acceptable for non-hub tracks, although for a final decision, it would be necessary to listen to the tracks with the highest k-occurrences.

5. Acknowledgements

This research is supported by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) under project number L112-N04, and by the EU 6th FP project S2S2 ("Sound to Sense, Sense to Sound", IST-2004-03773).

References

[1] Jean-Julien Aucouturier and Francois Pachet, "Improving timbre similarity: How high is the sky?," *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[2] Elias Pampalk, *Computational Models of Music Similarity and their Application in Music Information Retrieval*, Ph.D. thesis, Technische Universität Wien, 2006.

[3] J.-J. Aucouturier and F. Pachet, "A scale-free distribution of false positives for a large class of audio similarity measures," 2006, Submitted.

[4] Tim Pohle, Peter Knees, Markus Schedl, and Gerhard Widmer, "Automatically Adapting the Structure of Audio Similarity Spaces," in *Proc.* 1^{st} *Workshop on Learning the Semantics of Audio Signals (LSAS)*, Athens, Greece, December 2006.

[5] Michael Mandel and Dan Ellis, "Song-Level Features and Support Vector Machines for Music Classification," in *Proc. International Symposium on Music Information Retrieval (ISMIR'05)*, London, UK, 2005.

[6] Elias Pampalk, "A Matlab Toolbox to Compute Music Similarity From Audio," in *Proceedings of the Fifth International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 10-14 2004.