Automatically Adapting the Structure of Audio Similarity Spaces

Tim Pohle[1], Peter Knees[1],
Markus Schedl[1] and Gerhard Widmer[1,2]

[1] Department of Computational Perception
Johannes Kepler University Linz, Austria
[2] Austrian Research Institute
for Artificial Intelligence (OFAI)
Vienna, Austria
music@jku.at

Abstract. Today, among the best-performing audio-based music similarity measures 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. One drawback of this approach is that the distance space obtained this way has some undesirable properties.

In this paper, a number of approaches to correct these undesirable properties are investigated. They use knowledge about the properties of music by using other music tracks as a reference. These reference tracks can either be the music collection itself, or they may be an external set of reference tracks.

Our results show that the proposed techniques clearly improve the quality of this audio similarity measure. Furthermore, preliminary experiments indicate that the techniques also help to improve other similarity measures. They may even be useful in completely different domains, most notably text information retrieval.

1 Introduction

Music similarity measures are central to many music information retrieval (MIR) applications. Today, among the best-performing audio-based music similarity measures are algorithms based on Mel Frequency Cepstrum Coefficients (MFCCs). In these algorithms (e.g. [1–4]), 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 [3–6], the distance space obtained this way has some undesirable properties, as explained in the next sections. In this paper, we evaluate algorithms to tackle these problems, and find that they help to improve the quality of the similarity measure.

Always Similar. When calculating the distances between pieces of a music collection with such a music similarity measure, it turns out that some particular pieces frequently show up as close neighbours of many other pieces in the collection without sounding similar. Such pieces are called *hubs* [5, 6]. In [5, 6], this problem is extensively studied, and it is pointed out that the same problem also appears in a number of other domains, including fingerprint, speech and speaker recognition. A measure that supports the investigation of this phenomenon is *n-occurrence* [5, 6]: For each piece in the collection, the n nearest neighbours are determined. We call this set of n nearest tracks of track A the *n-NN set* of track A. Then, for each piece in the collection, it is counted in how many n-NN sets it appears. This number is the *n-occurrence* of the respective piece. Thus, the first problem with the similarity space is that some pieces have a very high n-occurrence. For example, in one of our collections consisting of 729 tracks, the highest occurring 20-occurrence is 175, indicating the existence of hubs. Thus, by reducing the highest occurring n-occurrences, the quality of the audio similarity measure may be improved.

Never Similar. Analogously to the preceding section, there are tracks that have a very low n-occurrence. Although there are some few outliers in most collections that do not sound similar to any other track, in general this is an undesirable property.

Triangle Inequality. For existing fast indexing and search algorithms to work (e.g., [7]), it is necessary that the triangle inequality is fulfilled by the distances of the similarity space. Unfortunately, this is not the case for the class of algorithms discussed above. Thus, it would be desirable that this property is changed.

2 Proposed Techniques

In this section, we describe several techniques for improving the aspects of the similarity space that are discussed above. Music tracks are very unevenly distributed in the similarity space. Thus, the basic idea is to transform the similarity space via a normalization operation.

All techniques are based on a modification of the distances between tracks that are computed by a music similarity measure. This basic similarity measure is arbitrary (although we will use the similarity measure mentioned in the introduction). We call it D_{basic}. When calculating the similarity of two tracks A and B, the proposed algorithms calculate $D_{basic}(A, B)$, and also the distances of the two tracks to a *set of known music tracks*. We call this set *reference set*. The reference set may be the music collection itself, or it may be a completely different set of tracks. The latter is of importance if there is no a priori knowledge about which music the algorithm will compare. Thus, the algorithms apply knowledge about the reference set to improve an existing audio similarity measure. Based on this information, the final distance between the tracks A and B is computed. The details of this step differ for the various approaches, as described in the next sections.

2.1 Using the N^{th} Nearest Neighbour: *N_{dist} Normalization*

The first approach we evaluate is based on the distance to the n^{th} nearest neighbour of a piece. Analogous to [8, 9], we call this distance of a piece A to its n^{th} nearest neighbour $n_{dist}(A)$. The basic idea is that if there are many pieces in close proximity to a piece A, then A is likely to appear in many NN sets of these close tracks. So, in this case the distances between A and the other pieces in the collection should be increased. This is done via normalization by $n_{dist}(A)$, where $n_{dist}(A)$ is the distance of A to its n^{th} closest track in the reference set.

However, when comparing tracks A and B, just normalizing $D_{basic}(A, B)$ with $n_{dist}(A)$ would not result in a symmetric distance measure. Thus, the two normalization factors are combined:

$$D_{n_{dist}}(A, B) := \frac{D_{basic}(A, B)}{n_{dist}(A) \cdot n_{dist}(B)} \tag{1}$$

2.2 Using the N-occurrence: *N-occurrence Normalization*

The second proposed approach aims to find a factor analogous to Section 2.1, but defined in a way that each A occurs in n n-NN sets of pieces in the reference set. Thus, it is aimed to normalize the n-occurrence of all pieces to n.

In detail, the factor associated with piece A is determined the following way.

1. Determine all distances $D_{basic}(A, P_j)$ to the pieces P_j in the reference set. Express these distances as a fraction of the respective distance to the n^{th} nearest neighbour of P_j from the reference set: $g(A, P_j) = D_{basic}(A, P_j)/n_{dist}(P_j)$.
2. Sort $g(A, P_j)$ for all P_j in ascending order. Take the n^{th} value as the factor for A, denoted $f(A)$.

The distance $g(A, P_j)$ calculated in the first step is greater than one if piece A does *not* belong to the n-NN set of piece P_j, and is smaller than one if it *does* belong to it. Dividing all distances from piece A to other pieces by the n^{th} smallest of these g (which is determined in Step 2) aims to modify these distances in a way that A belongs to n n-NN sets of the tracks in the reference set, as g measures the relative distance of A to the n_{dist} of P_j.

However, as in Section 2.1, when applying this for calculating the distance between two pieces, there is a factor f for both the pieces A and B. We apply the same technique as before to combine these two factors for transforming the original distance between A and B:

$$D_{\texttt{n-NN norm}}(A, B) := \frac{D_{basic}(A, B)}{f(A) \cdot f(B)} \tag{2}$$

2.3 Using Symmetric Ranking: *Proximity Verification*

The third proposed approach is based on the ranking of the tracks. For calculating the distance between two tracks A and B, first all distances $D_{basic}(A, P_j)$

between A and the tracks P_j of the reference set are calculated. These distances are sorted in ascending order. Then it is determined which rank the distance $D_{basic}(A, B)$ would have in these sorted values. As in general, $D_{basic}(A, B)$ is not exactly equal to one of the $D_{basic}(A, P_j)$, this value is determined by interpolating the next smaller and next larger distances $D_{basic}(A, P_i)$ and $D_{basic}(A, P_j)$, respectively.[3] We denote this (interpolated) rank as $D_P(A, B)$.

As pointed out in [10], in general such neighbourhood rankings are not symmetric (i.e., $D_P(A, B) \neq D_P(B, A)$). Thus, to obtain a distance measure, we define

$$D_{PV}(A, B) := D_P(A, B) + D_P(B, A) \tag{3}$$

We call this measure *Proximity Verification*, because $D_{PV}(A, B)$ is only small if both A is a close neighbour of B, *and* B is a close neighbour of A.

3 Evaluation Techniques

In the remainder of this work, we present an empirical evaluation of the proposed techniques. In this section, the experimental setup is explained. The results obtained are presented in Section 4.

Basic Audio Similarity Measure. In our experiments, we use an algorithm similar to the one from [11]. Each track is resampled to 22 *kHz*, divided into frames of about 23 *ms*, and 25 MFCCs are calculated on each of these. The MFCCs of a song are described by only one Gaussian by taking their overall mean and calculating the full covariance matrix. For comparing the models, a closed-form symmetric KL distance is applied [11]. As pointed out in [4], this algorithm is magnitudes faster than the basic variant with multiple Gaussians [1–3], and still the obtained song models are very similar to those of the original algorithm. We follow the practice from [4] and use this faster version as a representative of this kind of audio similarity algorithms.

Music Collections. For evaluation, we use two music collections:

1. The first music collection is the one from the ISMIR'04 Genre Classification Contest, consisting of 729 audio tracks from six genres. One advantage of using this collection is that it is downloadable[4], thus other researchers are able to compare their results to those reported here. We call this collection *IGC* (short for *Ismir Genre Contest* collection).
2. The second music collection is an in-house collection consisting of 2446 tracks by 103 artists, grouped into 22 genres. The biggest genres are Punk, consisting of 255 tracks, and Folk-Rock, consisting of 233 tracks.

[3] Note that interpolating the value makes this approach more stable than determining an integral rank, because the order is maintained in crowded regions.

[4] http://ismir2005.ismir.net/genre_contest/index.htm

Quality Measures. We apply several quality measures for examining the effects of the proposed techniques. *Always similar* and *never similar* are measured by the n-occurrence. The percentage of tracks A, B and C whose distances do not satisfy the triangle inequality can easily be estimated on the distance matrix (via sampling). However, measuring the quality of the algorithm's similarity judgements is not as straightforward, as discussed in the next paragraph.

Audio Similarity Performance. One important goal is to improve the algorithm with respect to the quality of its suggestions (i.e., those tracks that are close according to the algorithm should sound "similar"). The best way to measure this is human judgement. As user studies are expensive and time intensive, we use a common approach to estimate the algorithm's performance. Assuming that tracks that belong to the same genre sound more similar than tracks belonging to different genres, we use the leave-one-out k Nearest Neighbour (k-NN) genre classification accuracy as a quality measure.

4 Evaluation Results

In this section, we first describe how the parameter n was chosen for the n_{dist} normalization and for the n-occurrence normalization. Proximity Verification has no parameter. Afterwards, the results obtained in the experiments are presented.

4.1 Choosing Parameters

For the n_{dist} normalization and for the n-occurrence normalization, the respective values for n have to be chosen. In the following, we investigate the effect of specific values for n with respect to genre classification accuracy, measured by a k nearest neighbour (k-NN) classifier, with k set to 1. Our goal is to get an impression which parameter choice is well suited for the two parametric algorithms, so that their potential can be compared with the parameterless approach (i.e., Proximity Verification).

Choosing n for n_{dist} normalization. In Figure 1, the effect of n_{dist} normalization on the two music collections is shown for n in the range of 1 to 50. It can be seen that there is a common tendency. Values below $n = 10$ seem not to be a good choice, and there seems to be no additional benefit when choosing values above $n = 15$. Thus, we choose $n = 15$ for all subsequent experiments.

Choosing n for n-occurrence normalization. When doing the analogous analysis for n-occurrence (not depicted here), for the 103 artists collection roughly similar tendencies as before show up. However, for the *IGC* collection, the values seem to be quite inconclusive. Thus, we do not take into account the values of the *IGC* collection for choosing n, and opt for a value of $n = 15$ for all subsequent experiments, which has the positive side effect that the experimental results may be better comparable to n_{dist} normalization.

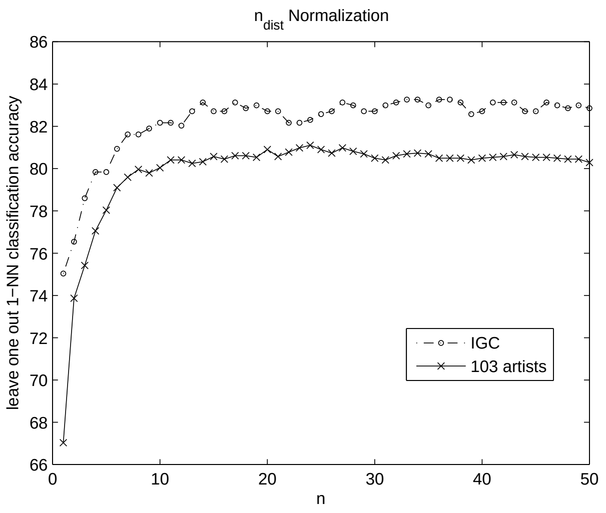

Fig. 1. Leave one out 1-NN genre classification accuracies on the two test collections after n_{dist} normalization for n in the range from 1 to 50. The classification accuracies without normalization are 81.1% for the IGC collection and 77.4% for the 103 artists collection. We think that the curve for the 103 artists collection is smoother than the one of the *IGC* collection because the size of the collections differ by a factor of about three, resulting in more stable results for the larger collection.

4.2 *k*-NN Genre Classification Evaluation

The results of the k-NN genre classification evaluation when using the same collection for evaluation and as the reference set are summarized in Table 1. It can be seen that in general all approaches improve classification accuracies. Improvements are better for the 103 artists collection than for the *IGC* collection. For the 15_{dist} normalization and the 15-occurrence normalization, accuracies are below the baseline (no normalization) in 3 of the 8 tested cases. For Proximity Verification, the classification accuracy improved in all cases.

4.3 Triangle Inequality

We estimate the percentage where the triangle inequality does not hold by randomly sampling over 100.000 triplets of tracks P_i, P_j and P_k from a collection and directly checking for the inequality. Without normalization, the inequality is violated in 50.9% and 33.7% of the cases for the *IGC* and for the 103 artists collection, respectively. All investigated approaches improve these values. However, the obtained results are still too high to be of use for fast indexing algorithms. The best performing algorithm is Proximity Verification, yielding 24.3% and 25.2%, respectively.

IGC collection	1-NN	5-NN	10-NN	20-NN
No normalization	81.1%	65.6%	56.6%	53.1%
15_{dist} normalization	82.7%	64.6%	57.3%	55.6%
15-occurrence norm.	80.9%	66.3%	58.3%	57.8%
Proximity Verification	84.0%	66.8%	57.7%	54.2%

103 artists collection	1-NN	5-NN	10-NN	20-NN
No normalization	77.4%	63.2%	47.5%	30.4%
15_{dist} normalization	80.6%	63.9%	46.3%	26.8%
15-occurrence norm.	82.6%	65.8%	46.3%	27.0%
Proximity Verification	82.3%	68.2%	51.4%	32.3%

Table 1. Average k-NN leave one out classification accuracy for the various methods for $k = \{1, 5, 10, 20\}$, when using the collection itself as reference set. Above: *IGC* collection, below: 103 artists collection.

4.4 20-occurrences

In Figure 2, the 20-occurrences are shown that are obtained when the various approaches are applied to the *IGC* collection. It can be seen that all approaches yield a much more uniform distribution of 20-occurrences for the tracks of the collection. The maximum value drops drastically (from 582 to 61 and below), and there are much fewer tracks with low 20-occurrences. The best performing algorithm is 15-occurrence normalization. The corresponding results on the 103 artists collection (not depicted here) are quite comparable.

4.5 Stability of the Approaches

In this section, we briefly examine the behaviour of the algorithms when the tracks used as a reference are changed. In the experiments presented so far, the reference set was equal to the music collection. Here, we examine the effect of using a completely different set of songs for reference. We use the tracks of the respective other collection as the reference set, as the two collections differ both in size (with a factor of about three) and in the composition of the kind of music they contain (e.g., the 103 artists collection does not contain the genre *classical*, which is the biggest genre in the *IGC* collection). Also, the 103 artists collection contains only commercial music (i.e., music that was bought at record stores), while the *IGC* collection contains a significant amount of amateur-made music.

From the results in Table 2, it can be seen that with the exchanged reference set, in most cases (21 out of 24) the classification accuracy was improved. This indicates that the performance of the proposed algorithms is not highly dependant on having the music collection they are applied on as a reference set. In the case of the 103 artists collection, 15-occurrence normalization even performed better with this different reference set than using the 103 artists collection as reference set. We think this indicates that this algorithm has a high potential, but the choice of the reference set is crucial, and the algorithm seems to be less stable than the other approaches.

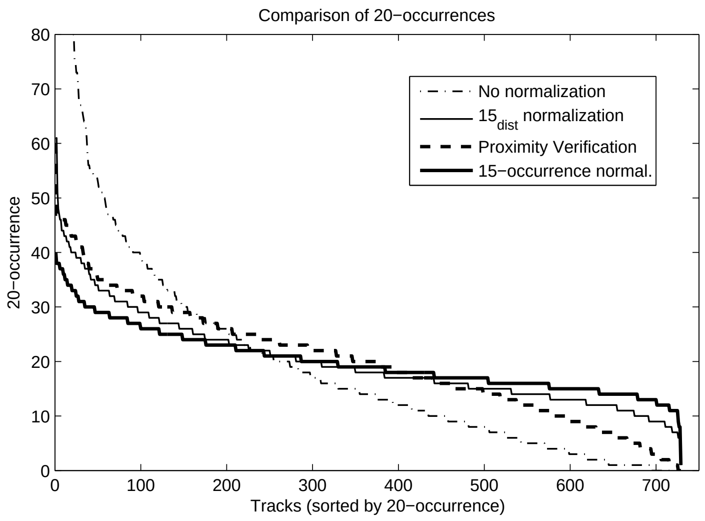

Fig. 2. 20-occurrences after applying the various approaches for the tracks of the *IGC* collection. For better comparability, the track indices are sorted according to their 20-occurrences (x-axis). Values are cut off at 80. The highest values are: baseline (i.e., no normalization): 582, 15_{dist} normalization: 61, Proximity Verification: 60, and 15-occurrence normalization: 40. Note that the purpose of this graphic is to depict the *shape* of the curves. Their average height is the same, as as the average n-occurrence value is the same (cf. [5, 6]).

5 Application to Web Based Artist Similarity

Encouraged by the results presented in the previous section, we evaluate if the proposed techniques may also successfully be applied to other similarity measures. We choose to evaluate the proposed post processing steps on web-based artist similarity data [12–14]. Such data is obtained using algorithms inspired by text information retrieval techniques. We use the data obtained from the web pages of 103 artists, grouped into 22 genres, which are analysed with $TF \times IDF$. Pairwise artist similarities are computed using the cosine measure. On this data, the three highest 5-occurrences are 31, 23 and 18, belonging to the artists *Genius*, *The Speed Freak* and *Ominus*, respectively. Although the music by these artists is off-mainstream and rather genre-specific, they appear in the 5-NN sets of artists from 15, 10 and 8 different genres, respectively. Thus, also in this domain there may be a problem with hubs. Note that according to [15], also the construction of user-rating or user-behaviour derived *artist networks* is not guaranteed to be uniquely guided by similarity criteria and that such networks may contain *hub artists* (i.e. certain artists that keep the network together), which is not expected for a network based on expert opinions.

The results obtained in our experiments are given in Table 3. It can be seen that the techniques also work for this completely different domain. We will investigate this in more detail in future work. Note that for a metric space (as

IGC collection	1-NN	5-NN	10-NN	20-NN
No normalization	81.1%	65.6%	56.6%	53.1%
15_{dist} normalization	80.7%	62.4%	56.9%	54.0%
15-occurrence norm.	81.5%	65.7%	58.0%	53.9%
Proximity Verification	82.9%	65.2%	57.8%	53.8%

103 artists collection	1-NN	5-NN	10-NN	20-NN
No normalization	77.4%	63.2%	47.5%	30.4%
15_{dist} normalization	81.2%	67.5%	50.8%	33.2%
15-occurrence norm.	82.7%	69.4%	52.3%	34.1%
Proximity Verification	81.2%	67.9%	51.5%	33.2%

Table 2. Average k-NN leave one out classification accuracy for the various methods for $k = \{1, 5, 10, 20\}$, when using the respective *other* collection as reference set. Top: results for the *IGC* collection with the 103 artists collection used as reference set, bottom: vice versa.

it is given in this case), techniques like those presented in [10, 8] may be useful for a more efficient implementation of Proximity Verification.

web based artist-sim.	1-NN	3-NN	5-NN	10-NN
No normalization	62.1%	48.9%	29.8%	14.8%
15_{dist} normalization	69.9%	56.8%	35.9%	24.7%
15-occurrence norm.	69.9%	58.9%	37.0%	26.0%
Proximity Verification	68.0%	59.7%	34.0%	26.3%

Table 3. Web based similarities between 103 musical artists from 22 genres: Average k-NN leave one out classification accuracy for the various methods for $k = \{1, 3, 5, 10\}$. The same 103 artists serve as reference models (i.e. no external reference models are used). The maximum number of artists in a genre is six.

6 Conclusion and Future Work

We presented three different approaches for modifying the distances calculated by a similarity measure. These algorithms are based on using a set of music tracks as a reference, which we call *reference set*. In the evaluation of the approaches, it turned out that all three algorithms are able to improve the audio similarity measure under investigation. Preliminary experiments indicate that the proposed techniques also may be successfully applied to other similarity measures, which was shown by means of a text-based similarity measure. The presented results show that the approaches seem to be stable with respect to replacing the reference set with a different one.

The latter two points will be investigated in more depth in future work. In particular, we think it will be interesting to see how much the cardinality of the reference set can be reduced without degrading the algorithm's performance. Also, we will investigate in more detail the merits of the proposed algorithms in the domain of text information retrieval. Furthermore, we will conduct more theoretical investigations about the algorithm's impacts on the similarity space, and the relation of the resulting distributions to power-law and exponential decay.

7 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). The Austrian Research Institute for Artificial Intelligence also acknowledges financial support by the Austrian Federal Ministries BMBWK and BMVIT. The authors would like to thank the reviewers for their valuable comments.

References

1. B. Logan, "Mel frequency cepstral coefficients for music modeling," Read at the first Int. Symp. on Music Information Retrieval (ISMIR'00), 2000.
2. B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME'01)*, 2001.
3. J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high is the sky?," *J. Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.
4. E. Pampalk, *Computational Models of Music Similarity and their Application in Music Information Retrieval*, Ph.D. thesis, Technische Universität Wien, 2006.
5. J.-J. Aucouturier, *Ten Experiments on the Modelling of Polyphonic Timbre*, Ph.D. thesis, University of Paris 6, 2006.
6. J.-J. Aucouturier and F. Pachet, "A scale-free distribution of false positives for a large class of audio similarity measures," 2006, Submitted.
7. P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *Proc. 23rd Int. Conf. on Very Large Data Bases (VLDB'97)*, 1997.
8. W. Jin, A. K. H. Tung, J. Han, and W. Wang, "Ranking Outliers Using Symmetric Neighborhood Relationship.," in *Proc. Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining (PAKDD'06)*, 2006.
9. M. M. Breunig, H. P. Kriegel, R. T. Nag, and J. Sander, "Lof: Identifying density-based local outliers," in *Proc. ACM SIGMOD 2000 Conf.*, 2000.
10. F. Korn and S. Muthuskrishnan, "Influence Sets Based on Reverse Nearest Neighbor Queries," in *Proc. ACM SIGMOD 2000 Conf.*, 2000.
11. M. Mandel and D. Ellis, "Song-Level Features and Support Vector Machines for Music Classification," in *Proc. Int. Symp. on Music Information Retrieval (ISMIR'05)*, 2005.
12. B. Whitman and S. Lawrence, "Inferring descriptions and similarity for music from community metadata," in *Proc. Int. Computer Music Conf.*, 2002.
13. S. Baumann and O. Hummel, "Using Cultural Metadata for Artist Recommendations," in *Proc. Conf. on Web Delivering of Music (Wedelmusic2003)*, 2003.
14. P. Knees, E. Pampalk, and G. Widmer, "Artist classification with web-based data," in *Proc. Int. Symp. on Music Information Retrieval (ISMIR'04)*, 2004.
15. P. Cano, O. Celma, M. Koppenberger, and J. Martin-Buldú, "Topology of music recommendation networks," in *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2006.