Computation and Visualization of Subjective Artist Similarity for Music Libraries on Android Devices

Manuel Maly

Institute of Software Technology and Interactive Systems
Vienna University of Technology

Abstract. Abstract goes here.

Keywords: subjective artist similarity, multi-dimensional scaling, audio analysis

Table of Contents

1 Introduction

2 Related Work

In this chapter, the reader will be introduced to the proceedings in science which are relevant or related to this thesis. They are grouped into the following topics of interest:

- **Motivation for this thesis** - Provides an explanation on why the chosen topic of this thesis is generally of interest, and why similarity measures in music are feasible (given optimal circumstances).
- **Features of digital music** - Gives an overview of existing methods of feature extraction which are purely based on computational methods.
- **Subjective music similarity computation** - Lists literature which is related to this thesis' problem of computing the subjective similarity of artists, which is based on subjectiveness as experienced by humans.
- **Visualization of artist similarity** - Provides an overview of existing methods and models of visualization of data similarity in general, and music similarity in particular.

2.1 Motivation for the Topic of This Thesis

Music is an integral part of the daily life in nearly all societies, and the list of published titles is growing every day. As huge amounts of data tend to be hard to digest, ontologies have to be created, by which music can be categorized in a hierarchical fashion. Aside from the author's motivation of choosing the topic of this thesis, a great interest in music classification can be observed in scientific literature. This is related to the problem that the categorization of arbitrary music titles is neither implicit nor trivial. Serving the demands of Electronic Music Distribution (EMD), the authors of [6] elaborate on the feasability of music similarity measures. It is found in [6] that the introduced similarity measure (timbre similarity) combined with other measures can yield interesting results. It is also mentioned that the interpretation of experimental results in the field of music similarity is challenging due to the subjective demands.

It is clear that even the best-educated music experts could hardly agree on any distinct similarity measure between two music titles, due to the implicit fuzziness of subjective measures. It can be assumed that it is rare that two humans would agree on the same similarity between music files if they vote independent of each other.

2.2 Features of Digital Music

As opposed to subjective artist similarity, there are music features or measures which can be retrieved by purely computational approaches. In the field of audio feature extraction, a wide range of classifiers (feature extractors) has been created. These classifiers in many cases run a bitstream analysis of a digitally stored music file and extract one or more reproducible measures characterizing the file. Interestingly, it is confirmed in [17] that the use of psycho-acoustic enhancements before feature extraction improves the classification accuracy significantly. It can be concluded that the outcomes of audio feature extraction are influenced by many factors which are not always intuitive. As has been mentioned previously, most audio classifiers analyze the bitstream of music files - however, the bitstream is only one dimension of a piece of music, if we regard it as a multidimensional object. For example, it is also possible to analyse the lyrics, as has been done in [19].

2.3 Subjective Music Similarity Computation

Subjective similarity, as the author understands it, expresses human opinions on a certain object. As previously mentioned, it is obvious that humans will hardly agree on attributes of music, and the same person might even make different statements in the course of time, depending e.g. on her mood. The following applies to both artist similarities and music file similarities, since the former may be constructed from aggregations of the latter (it has to be noted at this point that many artists tend to produce music from multiple genres, thus making an artist-to-artist-comparison difficult or even infeasible). In article [10] it is found that it is doubtable that a common ground truth for subjective artist similarity even exists, because of the inhomegeneity of measures made by the involved users. It can be deduced that a meaningful model of subjective music similarity will in most cases only resemble a compromise between different stakeholders. As inferred from [7] and [20] there are different approaches to retrieving a model of subjective similarity for a given set of music files, which include:

– Conduction of surveys with end users
– Opinions of experts
– Co-occurrence of files in end users' libraries or playlists
– Data mining of text in web sources, as performed in [30]
– Leveraging data gathered by social music services

As it is intended by this thesis to provide a concept for a fast and fully automatic approach to similarity measuring, we will concentrate on the last approach, the usage of data provided by social music services. Hybrid computation methods, such as the method described by [20] (combining acoustic features with text excerpts and tags retrieved from online services) turn out to be hardly feasible on a mobile device because of performance requirements. It is assumed by the author that for a rough estimation of music file or artist similarity, the data provided by social music services (as opposed to hybrid approaches) is sufficiently meaningful, as their daily user base is in the millions and still growing.

Apart from the source of similarity metrics, also the scope of computation has to be given thought to. Depending on a user's scenario, the user might want to explore her own music collection, or she might want to discover music similar to her own. In the following subsections, both cases are considered.

Similarity Computation for Collections of Music

In order to provide a meaningful, semantic overview of a collection of objects, data must be available or generated for all objects in the collection, or for most of them.

As has been mentioned before, **extraction of features** of music files may be used to gather metrics about the analysed files. The data retrieved may then be combined into a feature vector of N dimensions, where N is the number of features. When all files in the analysed music collection have been classified in this way, a **self-organizing map** can be trained with the resulting feature vectors, mapping vectors with small Euclidean distances spatially near to each other [27]. This results in a map where elevation represents the frequency of vectors in certain areas - thus, a high elevation at a point on the map means that the feature vector attached to the point is similar to a big number of music files. It is obvious that self-organizing maps are well suited for visually clustering music files by their features. However, it must be noted that this approach is only feasible if a huge amount of music metadata can be retrieved, either by feature extraction or other methods like text mining. Therefore, self-organizing maps can be considered not being quite suitable for the goals and scope of this thesis.

An alternative to feature extraction in this context is the construction of a **similarity matrix** containing all objects of the analysed music collection. A similarity matrix contains the similarities (or rather the dissimilarities) of all items to each other, in the form of distances. Intuitively, the measured item-item distance is higher if they are more dissimilar to each other (a distance of zero expressing equalness between items). Data sources for similarity matrices describing music include (as described earlier in this chapter):

- Surveys, playlist co-occurrences, user collection co-occurrences, web text mining,...
- Similarity rankings or measures from public Web APIs

As mentioned before, we will concentrate on the latter for complexity reasons. The task of building up a similarity matrix for a collection of music titles would then be reduced to a number of Web API calls, mapping the resulting distances or rankings to the objects in the collection. Similarity matrices can then be processed to find a suitable two-dimensional representation in Euclidean space, where the spatial distance between objects represents their similarity, or rather the dissimilarity. This process is called multi-dimensional scaling (MDS), and it has found broad adoption in scientific and industrial applications.

One of the most common multi-dimensional scaling techniques is the adoption of a spring model, which emulates the physical behaviour of steel rings connected by metal springs [22]. To continue with the analogy, MDS starts off with the steel rings at random positions, and in multiple iterations tries to satisfy the springs' forces. A spring's force is usually proportional to the discrepancy of two objects' high-dimensional distance and their low-dimensional distance. The fitness of the model (all steel rings being at their optimal position, considering all connected springs) is described by a stress function. The lower the stress function's output, the better suits the new low-dimensional model the original high-dimensional model. In most cases a perfect match between high-dimensional and low-dimensional Euclidean distances (stress function output = 0) is not possible, and thus for the MDS calculation to terminate, sensible termination criteria have to be defined - e.g., if the velocity of objects moving at each iteration falls below a predefined threshold, the algorithm terminates and the resulting low-dimensional representation is accepted as being "good enough".

Unfortunately, common spring model (or: metric distance) MDS is inflexible in the sense that the whole computation has to be performed all over again if slight changes to the data set occur [22]. Therefore, a computationally advantageous and more flexible approach to multi-dimensional scaling has been presented in [22], combining MDS with sampling and interpolation. As opposed to common multi-dimensional scaling, this hybrid methodology starts off with a random sample of size $\sqrt{N}$, where N is the number of objects contained in the dataset. After the spring model computation described above has been performed on the subset, the rest of the data set ($N - \sqrt{N}$ objects) is integrated into the low-dimensional model by an interpolation process which is described in detail in [22]. It has been found in the article that this combination of algorithms improves greatly on the accuracy of the model (i.e., a lower stress function output) and offers a sub-quadratic run time of $\mathcal{O}(N * \sqrt{N})$.

Discovering Objects Similar to a Given Object

After the elaboration of means of exploring the semi-static collection of objects in a user's library, heed must be given to the recommendation of unknown objects. Equipped with similarity data retrieved from various web APIs it is not only possible to compute relations of objects within a library, but also to find related objects which are currently not present in the library. It is clear that the same algorithms which have been previously described would compute usable outcomes by simply adding unknown objects to a library's representation (e.g. spring model MDS); yet, it can be assumed that in most cases only one object in a library is the starting point of a search for similar objects. This renders a big part of the library's representation irrelevant for this use case - consider a user searching for interprets similar to "The Beatles" - most likely, she will not be interested in how these unknown interprets relate to other bands in her library. Also, integrating such previously unknown objects into the representation of a big library will be computationally and query-wise infeasible, as dissimilarity distances to all objects in the library have to be determined. It must be noted that the representation for unknown object recommendation can only be a crude approximation (because of the previously described volatility of subjective similarity), and for this reason not only numerical measures, but also similarity rankings are considered sufficient for this use case.

A derivation from a previously proposed method can be considered here: Suiting the requirements well is a spring model MDS approach applied to a small dataset consisting of the starting point object (in the example being "The Beatles") and previously unkown objects which are most similar to it (retrieved via web APIs). The size of such a dataset would presumably peak at 30-40 objects, making common spring model multi-dimensional scaling computationally feasible. Naturally, the sampling/interpolation approach described in the previous subsection would also apply here, further decreasing the computation time.

However, a computationally less expensive algorithm for such means has been proposed in [18], consisting of a fusion of similarity rankings from various social music services. In this article it is demonstrated that various methods of embedding (fusing) similarity rankings from online services can provide different meaningful similarity models, some of which give more weight to unknown artists. Intuitively, this methodology is able to compute a ranked list of similar objects, based on multiple sources for greater reliability, in a customizable way. The fusion methods reach from rank-average to concordet-fusion (unweighted directed graph). Three major benefits speak in this approach's favour over global numerical similarity measurements:

- **Potentially insignificant computation times** while preserving a stable similarity ranking very well suited for mobile end users.
- **Simple but effective customization** achieved by easily exchangable fusion algorithm components.
- **Reducing the number of web API queries to a minimum** greatly reduces the overall number of network requests, making the method even better suited for mobile usage.

It must be noted at this point that the rank fusion algorithm is neither able to define distances between arbitrary objects in the dataset - only a dissimilarity ranking between the starting point object (e.g. "The Beatles") and the remaining objects is obtained - nor is it able to force the inclusion of objects (from a local library) in the ranking. This algorithm depends fully on the objects provided by external sources, meaning that if the starting point object is not contained in external sources, no meaningful result can be obtained. However, the author considers the amount of objects which can be obtained from third party web APIs sufficient for the algorithm to perform well for most of all objects in a typical user's library.

2.4 Visualization of Artist Similarity

The mode or fashion of data visualization can be considered a crucial aspect of interfaces for humans (i.e., graphical user interfaces). Today, humans' ability to apprehend information presented to them is limited in several ways, some of which are physical, and some of which are of psychological nature. Some of these limiting factors include:

- Restriction of short-term memory,
- Limited power of concentration,

– Narrow attention span (especially while using mobile device),
– Color blindness,...

Therefore, it is desirable to give heed to the choice of visualization method to achieve optimal apprehension results, without hindering information understanding through visualization errors. Since the field of music and music collection visualization is broad and not all algorithms can be presented within this thesis, the author decided to select only the field of two-dimensional collection visualizations for further investigation. It must be noted that several fields of visualizations are left out of the scope here, including:

– **Abstract visualization as an artform** - Certain artforms try to make music more tangible by creating matching images, as in the movie "2001 - A Space Odyssey" by Stanley Kubrick, or in works by the demo scene in Germany [29].
– **Realtime computed images as abstract visualization** have become common components of many desktop audio players, presenting the user with animated images (fractals, 3D-animations,...) which somehow resemble certain features of the currently played music.
– **3D environments resembling music content** give users the ability to roam through a virtual space similar to the way they interact with the physical world, as described in [9].

The scope of this thesis confines itself to the visualization of music tracks as objects, relating these objects to each other, and disregarding their real-time aspects (i.e., not generating any visualizations during playback). Intuitively, the computation and visualization of those relations (also, the quality of relations, e.g. ranking or dissimilarity distances) are closely related to each other. In some cases, a certain mode of computation of object relations more or less forces or forbids the usage of certain visualization approaches. Therefore, the presented modes of visualization are at least closely related to their computational counterparts from the previous subsection.

Visualization of Collections of Music

Previous work has shown that **self-organizing maps (SOM)**, which are in this context also called "islands of music" are well suited as visualizations of related music objects [8]. This methodology depends on raw audio stream analysis (performed by aforementioned feature extraction algorithms), and subsequently displaying them on an elevation-map, similar to a geographical map. Proof-of-concepts have been successfully implemented, as has been demonstrated in [24], featuring the PlaySOM. The information such self-organizing map visualizations want to give the user is: There are clusters of similar pieces of music in the provided collection, and within one cluster the contained music files most similar to each other. Additionally, each cluster has its own weight vector which can be used to add semantic height annotation to the map - for example, clusters whose objects contain a high tempo can be marked "high" (as opposed to clusters with slower music being marked as "low"), generating a corresponding height profile.

Another broad group of (dis-)similarity visualizations is made up of **force-directed graph layouts**. They all consist of nodes (music objects) and edges (relations between objects). Additionally, the distances (edge lengths) between nodes approximate a function over the previously determined dissimilarities between the music objects. As has been described in [14], the application of pseudo-physical forces on an undirected graph provides for a improvement of the graph layout. This is achieved by adding attractive or repulsive forces to all nodes in the graph, such that nodes push away from or attract each other. As long as there is energy left in a graph (i.e., there are objects which are not in their optimal position), the nodes are moved in a way that satisfies the applied forces. The authors of [23] have described and experimented with several graph-based layouts, and among them was a force-directed layout algorithm called LinLog [25], which has been found to deliver the most aestethic results.

The forces in a force-directed graph layout can behave like springs connecting nodes, and for this reason a subset of force-directed graph layouts is called **spring model**, which has been described in length in the previous subchapter in the context of multi-dimensional scaling (MDS). The calculation of a spring model's layout is very tightly coupled with the overall MDS computation, even in hybrid approaches [22] - in the case of MDS in combination with spring models, the visualization approach cannot be cleanly separated from the computation approach.

Other graph layouts which pose options for music collection browsing include [23]:

– Principal Component Analysis (PCA) layouts
– Tree map layouts
– Space filling curve layouts

Visualizations for the Recommendation of Unknown Objects

As has been discussed in the previous subchapter, the computational approaches for the recommendation of new objects can be either very similar to the computation of ordinary collection visualization, or they can use more simplified rank-based models. The former are clearly covered properly by the broad range of previously described visualization methods for collections of objects. On the contrary, visualization possibilities for rank-based computation models are not as manifold, due to the fuzzy dissimilarities between objects - a ranking can not be used to acquire deterministic object distances. However, since the goal of the visualization of such a ranking is to provide a very rough overview to the user, a deterministic visualization is not necessary. It can even be considered to omit the ranks, and just display these objects as relations of the same relevance, as has been done by the authors of [28]. It seems that also for this use case, a force-directed graph layout provides for the most aestethic results [13]. Additionally, such layouts can execute their self-optimization in realtime while being presented to their user without affecting the user experience in a negative way, as is shown by [1]. Additionally, the nodes in such layouts are user-manipulable in realtime.

2.5 Summary of this Section

3 Scenario and Scope of this Thesis

In this chapter, the scope of this thesis will be defined and a user scenario outlined.

3.1 Scope Definition

3.2 Selected Artist Similarity Computation

Rationale...

3.3 Selected Visualization Computation

Rationale...

3.4 Summary of this Section

4 Computation of Artist Similarity based on Webservices

4.1 Matching of Data-items from Different Sources

4.2 Basic Artist Similarity

4.3 Optimizations for Better Subjective Similarity

4.4 Summary of this Section

5 Visualization of Artist Similarity

There are visualization-only aspects to graph based layouts which can be formulated as the following questions:

– How are edges between objects displayed, or are they displayed at all?
– How are intersecting edges dealt with?
– How are objects presented?
– How are dissimilarities scaled to geometric distances (linearly, exponentially, logarithmically,...)?
– How is perfect (or near-perfect) similarity displayed (overlapping objects, minimum edge length,...)?

They can be viewed as implementation subtleties.

5.1 Summary of this Section

6 Implementation of Artist Similarity Visualization on Android

6.1 Structure of the Application

6.2 Web-Service Workflow

6.3 Android Environment

6.4 Artist Similarity Visualization Variants

6.5 Summary of this Section

7 User Study

7.1 Hypotheses

7.2 Experiment Setup

Population

Tasks

Metrics

7.3 Evaluation and Analysis of Study Results

7.4 Summary of this Section

8 Conclusion

References

1. http://audiomap.tuneglue.net/
2. http://developer.echonest.com/docs/v4/
3. http://devzone.wiki.meemix.com/index.php?title=MeeMix_API
4. http://www.lastfm.de/api
5. Agarwal, S., Wills, J., Cayton, L., Lanckriet, G., Kriegman, D., Belongie, S.: Generalized non-metric multidimensional scaling. In: AISTATS. San Juan, Puerto Rico (2007)
6. Aucouturier, J.J., Pachet, F.: Music similarity measures: What's the use? In: Ircam (ed.) Proceedings of the 3rd International Symposium on Music Information Retrieval. pp. 157–163. Paris, France (October 2002)
7. Berenzweig, A., Logan, B., Ellis, D.P.W., Whitman, B.: A large-scale evaluation of acoustic and subjective music similarity measures. In: Computer Music Journal (2003)
8. Cooper, M., Foote, J., Pampalk, E., Tzanetakis, G.: Visualization in audio-based music information retrieval. Comput. Music J. 30, 42–62 (June 2006), http://portal.acm.org/citation.cfm?id=1176357.1176365
9. Dittenbach, M., Berger, H., Genswaider, R., Pesenhofer, A., Rauber, A., Lidy, T., Merkl, D.: Shaping 3d multimedia environments: the mediasquare. In: Proceedings of the 6th ACM international conference on Image and video retrieval. pp. 85–88. CIVR '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1282280.1282292
10. Ellis, D.P.W., Whitman, B.: The quest for ground truth in musical artist similarity. In: in Proc. International Symposium on Music Information Retrieval ISMIR-2002. pp. 170–177 (2002)
11. Erten, C., Kobourov, S.G., Le, V., Navabi, A.: Simultaneous graph drawing: Layout algorithms and visualization schemes. In: In 11th Symposium on Graph Drawing (GD. pp. 437–449 (2003)
12. Frank, J., Lidy, T., Peiszer, E., Genswaider, R., Rauber, A.: Creating ambient music spaces in real and virtual worlds. Multimedia Tools Appl. 44(3), 449–468 (2009)
13. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Softw., Pract. Exper. 21(11), 1129–1164 (1991)
14. Gansner, E., North, S.: Improved force-directed layouts. In: Whitesides, S. (ed.) Graph Drawing, Lecture Notes in Computer Science, vol. 1547, pp. 364–373. Springer Berlin / Heidelberg (1998), http://dx.doi.org/10.1007/3-540-37623-2_28, 10.1007/3-540-37623-2_28
15. Geleijnse, G., Korst, J.: Tagging artists using cooccurrences on the web. In: Proceedings Third Philips Symposium on Intelligent Algorithms (SOIA 2006), pages 171 – 182 (2006)
16. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. Comput. Graph. Forum 28(3), 983–990 (2009)
17. Lidy, T., Rauber, A.: Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In: Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005). pp. 34–41. London, UK (September 11-15 2005)
18. Marshall, B.: Aggregating music recommendation web apis by artist. In: Information Reuse and Integration (IRI), 2010 IEEE International Conference on. pp. 75 –79 (2010)

19. Mayer, R., Neumayer, R., Rauber, A.: Rhyme and style features for musical genre classification by song lyrics. In: ISMIR. pp. 337–342 (2008)
20. McFee, B., Lanckriet, G.: Heterogeneous embedding for subjective artist similarity. In: Tenth International Symposium for Music Information Retrieval (ISMIR2009)) (October 2009)
21. McFee, B., Lanckriet, G.: Partial order embedding with multiple kernels. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 721–728. ICML '09, ACM, New York, NY, USA (2009), `http://doi.acm.org/10.1145/1553374.1553467`
22. Morrison, A., Ross, G., Chalmers, M.: Fast multidimensional scaling through sampling, springs and interpolation. Information Visualization 2, 68–77 (March 2003), `http://dx.doi.org/10.1057/palgrave.ivs.9500040`
23. Muelder, C., Provan, T., Ma, K.L.: Content based graph visualization of audio data for music library navigation. In: In Proceedings of The IEEE International Symposium on Multimedia (ISM2010) (December 2010)
24. Neumayer, R., Dittenbach, M., Rauber, A.: Playsom and pocketsomplayer: Alternative interfaces to large music collections. In: Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005). pp. 618–623. London, UK (September 11-15 2005)
25. Noack, A.: An energy model for visual graph clustering (2003), `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.3319`
26. Pampalk, E., Rauber, A., Merkl, D.: Content-based organization and visualization of music archives. pp. 570–579. ACM (2002)
27. Rauber, A., Pampalk, E., Merkl, D.: Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In: Proceedings of the 3rd International Symposium on Music Information Retrieval. pp. 71–80. Paris, France (October 13-17 2002), `http://www.ifs.tuwien.ac.at/ifs/research/publications.html`
28. Sarmento, L., Gouyon, F., Costa, B.G., Oliveira, E.C.: Visualizing networks of music artists with rama. In: WEBIST. pp. 232–237 (2009)
29. Scheib, V., Engell-Nielsen, T., Lehtinen, S., Haines, E., Taylor, P.: The demo scene. In: ACM SIGGRAPH 2002 conference abstracts and applications. pp. 96–97. SIGGRAPH '02, ACM, New York, NY, USA (2002), `http://doi.acm.org/10.1145/1242073.1242125`
30. Whitman, B., Lawrence, S.: Inferring descriptions and similarity for music from community metadata. In: In Proceedings of the 2002 International Computer Music Conference. pp. 591–598 (2002)

9 Appendix A