IMPROVING VISUALIZATION OF HIGH-DIMENSIONAL MUSIC SIMILARITY SPACES

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

ABSTRACT

Visualizations of music databases are a popular form of interface allowing intuitive exploration of music catalogs. They are often based on lower dimensional projections of high dimensional music similarity spaces. Such similarity spaces have already been shown to be negatively impacted by so-called hubs and anti-hubs. These are points that appear very close or very far to many other data points due to a problem of measuring distances in high-dimensional spaces. We present an empirical study on how this phenomenon impacts three popular approaches to compute two-dimensional visualizations of music databases. We also show how the negative impact of hubs and anti-hubs can be reduced by re-scaling the high dimensional spaces before low dimensional projection.

1. INTRODUCTION

Visualization via low dimensional projections is one way to produce interfaces that allow navigation and access to music data sets. A very popular and influential approach is the islands-of-music metaphor [14], where representations of similar music form islands on a two-dimensional display. Numerous variations of this approach have been published within the music information retrieval (MIR) community (see e.g. [5, 9, 16, 24]). A recent trend towards more holistic MIR approaches [18, 23] including human computer interaction aspects is likely to increase interest in visualization in the near future. State-of-the-art visualization algorithms are said to be able to visualize high-dimensional data [28]. Precisely for such high-dimensional data a new aspect of the curse of dimensionality, the so called hubness, has been discovered and described within the MIR community [1,8]. This paper investigates the impact of hubness on visualization of high-dimensional music similarity spaces. In an empirical evaluation of three methods for dimensionality reduction the negative impact of hubness is explored and it is shown how re-scaling of the similarity spaces as a preprocessing step can greatly improve the visualizations.

2. RELATED WORK

Hubness is a general problem of learning in high-dimensional space which has been discovered in MIR [1], but then gained attention in a general machine learning context where it has been discussed as a new aspect of the curse of dimensionality [15,20]. Hub objects appear very close to many other data objects and anti-hubs very far from most other data objects. The effect is related to the phenomenon of concentration of distances and has been shown to have a negative impact on many tasks including classification [15], nearest neighbor based recommendation [3] and retrieval [21], outlier detection [15] and clustering [19,26].

Visualization of music similarity spaces via low dimensional projections has a long tradition within MIR. Starting from the influential islands-of-music approach [14,16], numerous extensions and variations have been developed (see e.g. [5,9,24]). Although different methods for dimensionality reduction have been explored, the most popular approach seems to be self-organizing maps [10]. Despite the popularity of these interfaces based on lower dimensional projections, it has not yet been clarified how hubness influences these visualizations. To the best of our knowledge, there is only a single publication concerned with the impact of hubness on visualization [6]. In an analysis of dimensionality reduction of three audio databases to two dimensions using multidimensional scaling, the authors show that projected data tends to be concentrated in a single large cluster centered around the largest hub.

It is important to note that simple dimensionality reduction does not reduce hubness. On the contrary it has been shown that only projections to very few dimensions, well below the intrinsic dimensionality of a data set, are able to reduce hubness, but at the cost of a loss of distance information [15]. On the other hand, results on re-scaling methods to reduce hubness [20] show that it is possible to decrease hubness without changing the intrinsic dimensionality and therefore the information content of the data. Thus a good approach to visualization of high dimensional data might be to first re-scale to reduce hubness without changing the intrinsic dimensionality, and then to apply dimensionality reduction to the re-scaled data.

3. DATA

For our experiments we used two standard music databases: the "GTZAN" collection consisting of $N = 1000$ audio tracks (each 30 s length) evenly spread over ten music genres [27]; the "ISMIR2004" [1] collection containing $N = 1458$ tracks of six genres, with full-length audio being available and exhibiting a highly imbalanced genre distribution with classical music comprising almost half of the tracks.

We decided to compute timbre information from the audio, since this is an integral part of many MIR systems and at the same time has already been shown to be susceptible to hubness [3]. Every track is divided into overlapping frames for which 20 MFCCs are being computed which are modeled via a single Gaussian with full covariance matrix. To compute a distance value between two Gaussians the symmetrized Kullback-Leibler (SKL) divergence is used [11]. This results in $N \times N$ distance matrices D_I and D_G for the ISMIR and GTZAN data sets. Please note that SKL is symmetric and non-negative, but does not fulfill the triangle inequality and therefore is not a full metric.

4. METHODS

In what follows we present three methods for dimensionality reduction (TSNE, SAMMON, SOM) and two methods to re-scale distance matrices in order to reduce hubness (MP, SNN). In Section 5 we will use MP and SNN as a preprocessing step before dimensionality reduction. This gives nine different combination of methods to compare: TSNE, MP TSNE, SNN TSNE, SOM, MP SOM, SNN SOM, SAMMON, MP SAMMON, SNN SAMMON. But first we present evaluation indices that will be used to measure the performance achieved in original and rescaled data spaces.

4.1 Evaluation measures

Hubness (S^n): To characterize the strength of the hubness phenomenon in a data set we use the so called hubness measure [15]. This is based on the n-occurrences of points x, which is the number of times x occurs in the n-nearest neighbor lists of all other objects in the collection. Hubness is then defined as the skewness of the distribution of n-occurrences O^n:

$$S^n = \frac{\mathrm{E}\left[(O^n - \mu_{O^n})^3\right]}{\sigma^3_{O^n}}. \quad (1)$$

A data set having high hubness produces few hub objects with very high n-occurrence and many anti-hubs with n-occurrence of zero. This makes the distribution of n-occurrences skewed with positive skewness indicating high hubness. Previous results [22] show that values above 1.4 are problematic.

Nearest neighbor overlap (L^k): To quantify the degree to which neighborhood relations are preserved we compute the overlap between nearest neighbor lists in the high dimensional input space ($NN(x)$) and the low-dimensional output space ($NN(\hat{x})$):

$$L^k = \frac{1}{N} \sum_{i=1 \ldots N} |NN(x) \cap NN(\hat{x})|/k. \quad (2)$$

Nearest neighbor classification accuracy (C^k): We report the k-nearest neighbor (kNN) classification accuracy C^k using leave-one-out cross-validation, where classification is performed via a majority vote among the k nearest neighbors, with the class of the nearest neighbor used for breaking ties.

4.2 Dimensionality reduction

Dimensionality reduction algorithms try to map high dimensional input data to lower dimensional output spaces while preserving information of the topology of the input space, i.e. preserving similarities or similarity orderings. All three methods used in this study are based on optimization algorithms that are initiated randomly and therefore can give different solutions for different initializations. All results reported in Section 5 are based on single runs since repeated runs have shown that all three methods give comparable solutions even for different initializations. Please note that the original and re-scaled distance matrices D_I and D_G are normalized to have a smallest value of 0 and a largest value of 1 and, if necessary, changed to similarities before dimensionality reduction.

t-Stochastic Neighbor Embedding (TSNE): A particularly successful algorithm for dimensionality reduction is t-SNE [28]. It first converts similarities of high dimensional points x_i and x_j into conditional probabilities $p_{j|i}$ that x_i and x_j are neighbors given a Gaussian probability density estimate centered at x_i. It computes a similar probability $q_{j|i}$ for the low dimensional counterparts y_i and y_j based on a Student-t density estimate. The mapping to the lower dimension is then achieved by minimizing the sum of the Kullback-Leibler divergences over all data points using gradient descent:

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \quad (3)$$

We used the implementation by Laurens van der Maaten [2] that accepts similarity matrices as input (function "tsne_p") using standard settings as provided by the software and 1000 iterations for all experiments.

Sammon mapping (SAMMON): Sammon mapping [17] does dimensionality reduction by minimizing the following via steepest descent:

$$\frac{1}{\sum_{i=0}^{N-1} \sum_{j<i} d(x_i, x_j)} \sum_{i=0}^{N-1} \sum_{j<i} \frac{(d(x_i, x_j) - \hat{d}(\hat{x}_i, \hat{x}_j))^2}{d(x_i, x_j)} \quad (4)$$

where $\hat{d}(\hat{x}_i, \hat{x}_j)$ is the distance in the output space that corresponds to the distance $d(x_i, x_j)$ in the input space and N is the number of points to be mapped. We used the implementation from the SOM Toolbox [3] for all experiments with standard settings and 100 iterations.

Self Organizing Map (SOM): The SOM [10] is an unsupervised neural network that visualizes high dimensional

[1] http://ismir2004.ismir.net/genre_contest/index.htm

[2] http://lvdmaaten.github.io/tsne/

[3] http://www.cis.hut.fi/projects/somtoolbox/

data by mapping it to a two dimensional map grid. Data points that are similar in the original high dimensional space are mapped onto locations close to each other on the grid. In essence the SOM consists of an ordered set of so called map units r_i, each of which is assigned a reference vector (or model vector) m_i in the high dimensional input space. In an iterative learning procedure the model vectors m_i are adapted to the input data, very much like cluster centers in a k-means clustering procedure. The main difference is that model vectors corresponding to neighboring map units r_i are adapted together, based on a neighborhood weighting function. This yields a topological organization of the model vectors m_i in the two dimensional output space.

For all our experiments we use SOMs with 40×40 output maps, thereby ensuring that we always have more model vectors than input vectors which is advantageous for using SOM for visualization (see [2] for more on the usage of SOMs for clustering and visualization). We use the NETLAB [12] SOM implementation with standard settings for the learning parameters (initial neighborhood size of 8 shrunk to 1 during an ordering phase lasting 50 iterations, followed by 400 iterations of a convergence phase). Since SOMs need data vectors and not distance matrices as input data, we use the full rows of the distance matrices as inputs (see e.g. [9, 13] for more detail or [22] for a criticism of this rather crude but standard approach).

4.3 Reducing hubness

We introduce the two methods we will apply to reduce hubness by using each method on the whole distance matrix and computing re-scaled distances. Both methods aim at repairing asymmetric nearest neighbor relations which are a consequence of the presence of hubs. A hub y is the nearest neighbor of x, but the nearest neighbor of the hub y is another point a ($a \neq x$). This is because hubs are by definition nearest neighbors to very many data points but only one data point can be the nearest neighbor to a hub.

Mutual Proximity (MP): MP reinterprets the original distance space so that two objects sharing similar nearest neighbors are more closely tied to each other, while two objects with dissimilar neighborhoods are repelled from each other. This is done by transforming the distance of two objects into a mutual proximity in terms of their distribution of distances. It was shown that by using this mutual reinterpretation of distances hubness is decisively reduced, while the intrinsic dimensionality of the data stays the same [20]. To compute MP, we assume that the distances $D_{x,i=1..N}$ from an object x to all other objects in our data set follow a certain probability distribution, thus any distance $D_{x,y}$ can be reinterpreted as the probability of y being the nearest neighbor of x, given their distance $D_{x,y}$ and the probability distribution $P(X)$. In this work we assume that the distances $D_{x,i=1..N}$ follow a Gaussian distribution. MP is defined as the probability that y is the nearest neighbor of x given $P(X)$ and x is the nearest neighbor of y given $P(Y)$:

$$MP(D_{x,y}) = P(X > D_{x,y} \cap Y > D_{y,x}). \qquad (5)$$

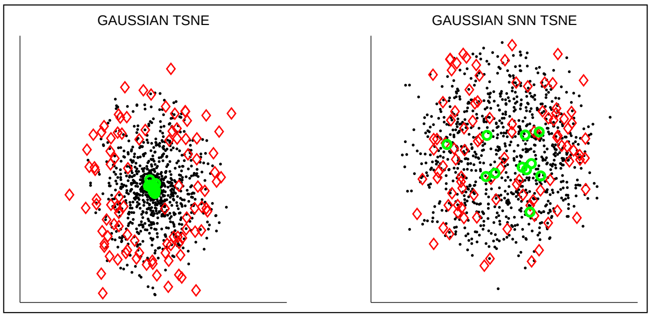

Figure 1. Maps obtained for Gaussian artificial data via TSNE (left) and SNN TSNE (right). Hubs are shown as green circles and anti-hubs as red diamonds.

Shared Nearest Neighbors (SNN): SNN [7] uses the neighborhood information to help enforce pairwise stability. SNN is computed as a set intersection of the k-nearest neighbor lists NN of two objects x, y:

$$SNN(x, y) = |NN(x) \cap NN(y)|/k. \qquad (6)$$

This way SNN strictly strengthens symmetric nearest neighbor relations which in turn should also manifest itself in a reduction of hubness. We use SNN with $k = 50$ because this already yields hubness values S^5 (see Section 4.1) below 1 for both ISMIR and GTZAN.

5. RESULTS

Before we present our results using the ISMIR and GTZAN data sets we give a first illustration based on artificial data. We sampled 1000 data points from a 50-dimensional Gaussian distribution and used Euclidean distance to compute a distance matrix. The hubness S^5 of this data set is 2.95. Similar to other work [20], we defined anti-hubs as points with a $O^5 = 0$, i.e. points never appearing in any nearest neighbor list of size 5. Hubs are points with $O^5 > 25$, i.e. points that appear more than five times as expected. This definition of hubs and anti-hubs is used for all results in this paper and hubs and anti-hubs are always computed in the high-dimensional spaces. Figure 1 plots two dimensional results obtained using TSNE alone (left plot) and SNN plus TSNE (right plot). As can be seen, TSNE maps all hubs (green circles) to the center of the points and maps all anti-hubs (red diamonds) to the edges. The right plot shows that SNN TSNE is able to map hubs and anti-hubs much more evenly across the whole set of mapped points. The hubness S^5 of the re-scaled distance space after application of SNN is 0.81.

Next we present the visualization results obtained for the ISMIR data set using different combinations of TSNE, SOM, SAMMON and MP and SNN in Figure 2. The hubness S^5 of the ISMIR data set is 3.94. Re-scaling reduces this value to 1.25 for MP and 0.89 for SNN. Hubs are again shown as green circles and anti-hubs as red diamonds. When using TSNE (top row), we again see that the hub points are mapped to the center of the visualization and anti-hubs appearing all over the plot but also at the edges

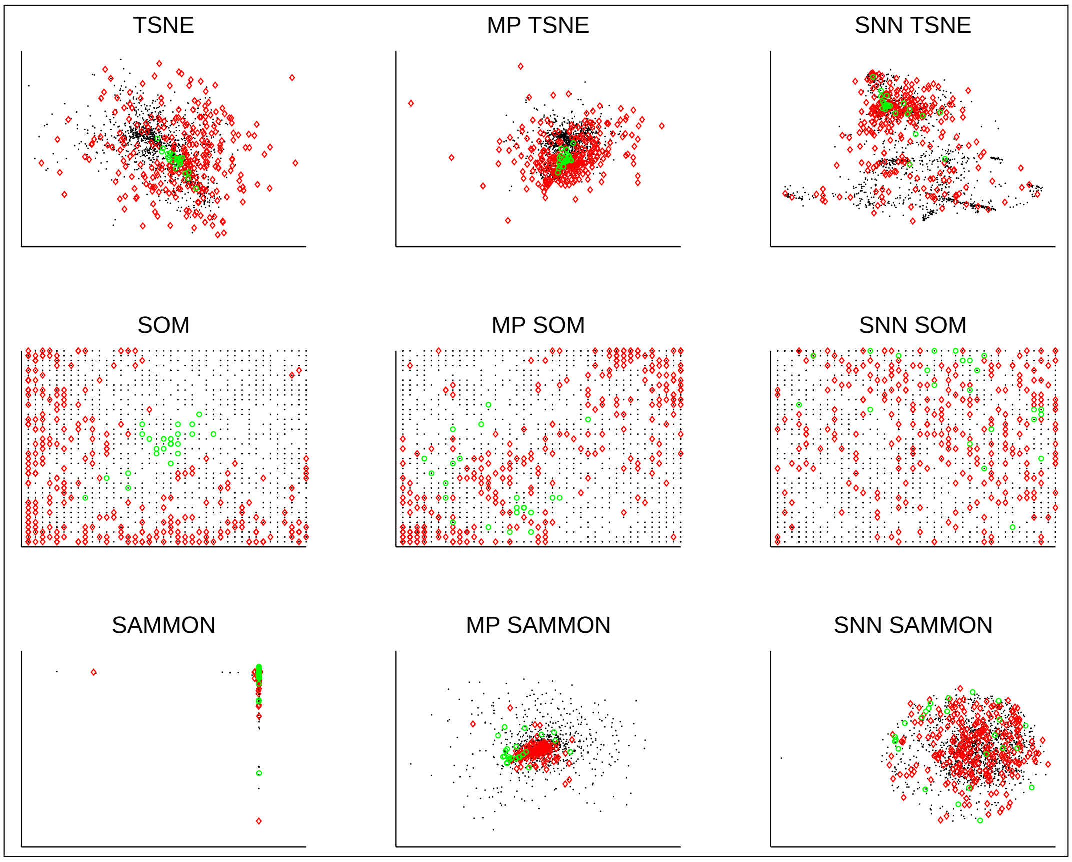

Figure 2. Visualization of ISMIR data set using different combinations of TSNE, SOM, SAMMON and MP and SNN. Hubs are shown as green circles and anti-hubs as red diamonds.

where no other data points are mapped to. When using the combination MP TSNE, this situation shows only little improvement with some anti-hubs still being mapped to areas where no other points can be found. Hub points are still mapped to the center of the plot. When using the combination SNN TSNE the result seems to be much improved, the plot showing much more structure and the hubs and anti-hubs no longer confined to the center or edges. Looking at the results obtained for SOM (middle row), we can again see that the hub points are mapped to the center of the plot whereas the anti-hubs are confined to the left and bottom edge areas. When using MP SOM or even better SNN SOM, hubs and anti-hubs are much more scattered across the whole plots. When using SAMMON (bottom row), we can see that the visualization is heavily distorted with a few data points being mapped far away from the rest of the data. When using MP SAMMON, this distortion is no longer visible but both hubs and anti-hubs are mapped to the more central parts of the plots. Only SNN SAMMON seems to be able to map anti-hubs more or less evenly across the plot, with hubs being mapped closer to the edges. Overall the combination SNN TSNE seems to yield the best visualization results. Results are similar for GTZAN, but are not depicted for lack of space.

To quantify the success in visualization, we compute the nearest neighbor overlap L^k between high- and low-dimensional spaces for TSNE, SOM and SAMMON which is shown in Figure 3 for both ISMIR (top row) and GTZAN (bottom row) data sets. In all six plots solid lines show results when using dimensionality reduction only (TSNE, SOM or SAMMON), dash-dotted lines give results when MP is used for preprocessing, dashed lines when SNN is used for preprocessing. The overlap L^k is computed for a range of $k = 5 \dots 500$ plotted on the x-axis to quantify preservation of local and more global neighborhoods. We can see that for all three dimensionality reduction methods and over the full range of k, preprocessing via MP and SNN is able to increase the overlap L^k. The only exception is SAMMON when applied to ISMIR, where SNN gives worse results than using no preprocessing for $k > 200$. Preprocessing with SNN is superior to using MP in combination with TSNE and SOM. In combination with SAMMON, MP works a little better than SNN. Overall TSNE performs better than SOM, which is again better than SAMMON. Again the combination SNN TSNE gives the best results of all.

Next we present a more detailed analysis of the nearest neighbor overlap results by concentrating on L^k with $k = 50$ since this is where the difference in performance is largest. In Figure 4 we give separate results for "all" data points, "hub", "anti-hub" and "normal" (i.e. not hubs or anti-hubs) data points as bar plots for TSNE, SOM and SAMMON. Every bar plot shows a black bar for dimensionality reduction only, a gray bar for results when MP is used for preprocessing, a white bar when SNN is used. For all three dimensionality reduction algorithms, L^{50} is high-

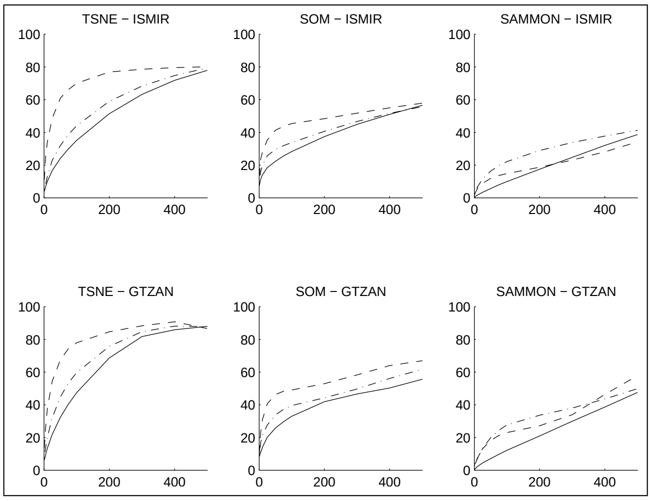

Figure 3. Overlap of nearest neighbors in high and low dimensions in percent (y-axis) vs. number of neighbors (x-axis) for ISMIR (top) and GTZAN (bottom). Plots given for TSNE, SOM and SAMMON with solid lines for using dimensionality reduction only, dash-dotted lines for using MP for preprocessing, dashed lines when SNN is used.

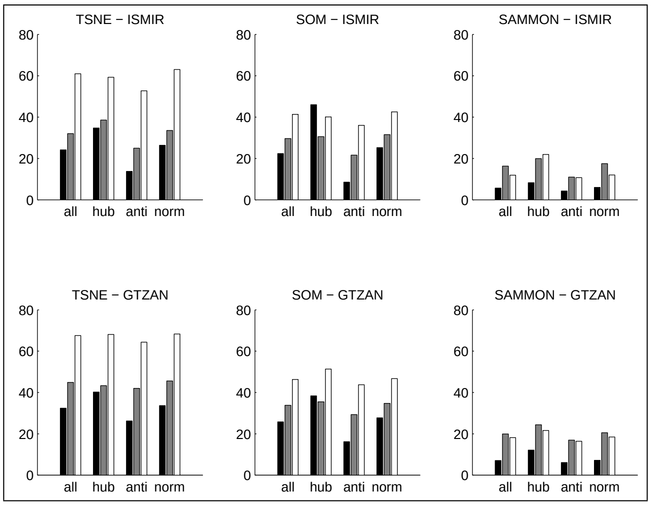

Figure 4. Analysis of overlap of 50 nearest neighbors in high and low dimensions in percent (y-axis) vs. type of data points (x-axis: all, hub, anti-hub, normal) for ISMIR (top) and GTZAN (bottom). Bar plots are given for TSNE, SOM and SAMMON with black bars showing results for dimensionality reduction only, gray bars for using MP for preprocessing, white bars when SNN is used.

	ISMIR			GTZAN		
	-	mp	snn	-	mp	snn
orig	71.4	78.1	76.5	61.7	67.8	61.6
tsne	62.6	63.9	70.4	39.1	41.0	52.8
som	65.2	65.0	69.2	40.4	37.7	49.7
sammon	47.0	51.3	46.3	16.1	27.3	25.3

Table 1. Genre classification accuracy in percent using 50-nearest neighbor classification for ISMIR and GTZAN data sets different combinations of TSNE, SOM, SAMMON and MP and SNN as well as for the original (orig) high dimensional data space.

est for hub points and worst for anti-hub points, with normal points somewhere in between. Applying either MP or SNN as preprocessing generally increases L^{50} for all different kinds of points, but also makes L^{50} for hubs, anti-hubs and normal points perform much more comparable. Anti-hub points now perform almost as well as all other points. The only exception is again SAMMON, which generally performs very poorly and where SNN is not able to improve the overall situation.

As a further analysis of our visualization results, we give kNN genre classification [4] accuracy results C^{50} when using different combinations of TSNE, SOM, SAMMON and MP and SNN as well as for the original high dimensional data space in Table 1. As for the original input space (row "orig"), MP and SNN increase C^{50} for ISMIR, but only MP for GTZAN. This is in line with previous comparison of MP and SNN [4]. Classification results for low-dimensional spaces are of course lower than those achieved on the original input spaces since any dimensionality reduction incurs some loss of information. But for both TSNE and SOM on both data sets, SNN is able to increase C^{50}, which is additional indication that SNN is the preprocessing method to prefer. Results for SAMMON are generally very low and rather mixed.

Finally we show visualization results of the ISMIR data set when using TSNE as well as SNN TSNE, which is the best performing combination, in Figure 5 with different genres given in different colors. The color coding is as follows: classical - black, jazz_blues - blue, rock_pop - red, world - green, metal_punk - yellow, electronic - cyan. Although it is hard to verbalize the information contained in a visualization, it seems apparent that the result for SNN TSNE (right plot) shows much more structure than the result for TSNE only. This enables a more detailed picture of the overlap between "classical" (black) and "world" (green) music. Also the position of genre "jazz_blues" (blue) is now clearer between "classical/world" and the remaining three genres. Also "electronic" (cyan) music seems to be a little more apart from "rock_pop" (red) and "metal_punk" (yellow). Results for GTZAN, which consists of music from ten genres, are similar in tendency but not shown for space considerations.

6. DISCUSSION

Summing up the results presented in the previous section, we like to state that all three visualization methods are affected by the hubness problem. Looking at the visualizations, checking the amount of overlap between nearest neighbors in high and low dimensions for hub, anti-hub and normal points makes it clear that there is a problem for dimensionality reduction of data with high values of hubness. It is also evident that preprocessing with either MP or SNN can help in this situation. Especially the combination

[4] Please note that we are of course aware of the controversial role of genre classification in MIR, especially in the context of GTZAN [25], but that accuracy only serves as a further illustration of results in this context.

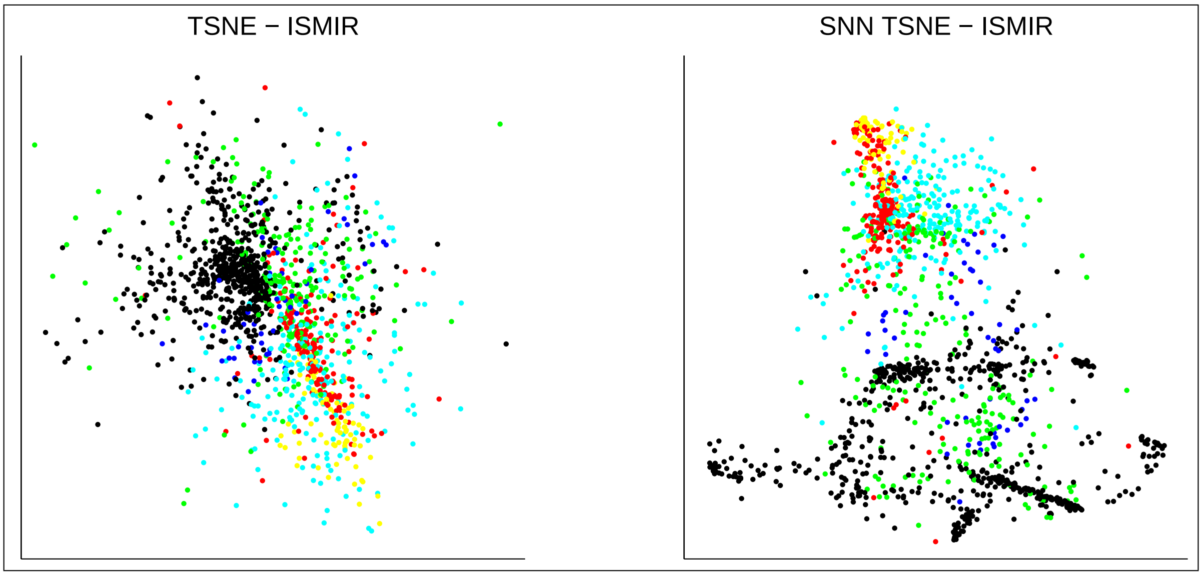

Figure 5. Visualization of ISMIR data set using TSNE (left) and SNN TSNE (right) with color coded genres (see Section 5).

of SNN and TSNE yields very improved results. Although this paper used only one particular approach to compute music similarity, previous work [3] has made it clear that many different approaches are affected by hubness. For the dimensionality reduction algorithms, we basically used standard settings since attempts to adjust parameters did not really improve results. But of course a more rigorous parameter tuning should be part of future research.

One particularity of the music similarity spaces used in this work is the fact they are based on Gaussian models of timbre information and therefore only distances between models are available but not vector representations. Therefore all dimensionality reduction methods need to be able to deal with distance/similarity information as input. Whereas this is natural for SAMMON, it already constitutes a problem for SOM. We resorted to the standard but somewhat crude approach to use the full rows of the distance matrices as input vectors (with length equal 1000 or 1485 for GTZAN and ISMIR). But there already exists a superior approach [22] of directly using Gaussian models as inputs to SOMs and it would be very interesting to research the impact of hubness on this version of SOM. For TSNE, we were able to use a variant ("tsne_p") that is able to deal directly with similarity matrices. But as stated by the authors [28], this should only be done if "these similarities can be interpreted as conditional probabilities" as explained in Section 4.2. A theoretic examination as to what extent MP and SNN fulfill this requirement will be part of future work. When TSNE is being used with input vectors instead of a similarity matrix, the width of the Gaussian probability densities are adapted locally according to a so-called perplexity term. This is an important part of the algorithm which is missing in case it is used with a similarity matrix directly. It is an interesting research question whether this local adaption in itself is able to counter some of the problems due to hubness. But this can only be studied if vectors are available as input to TSNE.

As has already been noted in Section 3, the music similarity spaces are based on symmetric Kullback-Leibler divergences which do not fulfill the triangle inequality and therefore do not exhibit all aspects of a true metric. There exists an extension of t-SNE [29] which uses multiple maps to visualize non-metric similarities. Even more interesting, this extension is motivated with the notion of data points which show high centrality, i.e. points which are similar to very many other data points. In contrast to the discussion of hub points, such central points are in this case not seen as problematic but as a special challenge for a visualization algorithm. It would therefore be very interesting to study and compare these central and hub points and to apply the t-SNE algorithm for non-metric similarities to data sets with high hubness.

7. CONCLUSION

We presented the first substantial empirical evaluation of the impact of hubness on visualization of high-dimensional music similarity spaces. Analyzing three popular methods for dimensionality reduction applied to two standard music data sets, we were able to show that hubs and anti-hubs distort the lower dimensional representations. Generally hubs are mapped to the central parts of plots and anti-hubs usually to the edges. We were able to show that preprocessing with methods that have been designed to reduce hubness can greatly improve this situation. This results in visualization where hubs and anti-hubs are no longer mapped to peculiar locations, which also gives improved preservation of neighborhood information when mapping to low dimensions. Particularly a combination of preprocessing via "shared nearest neighbors" followed by dimensionality reduction via "t-SNE" proved to be most successful. This approach could therefore be used as the core technology for future visualization interfaces to music catalogs.

Acknowledgements: This work was supported by the Austrian Science Fund (FWF, grant P27082).

8. REFERENCES

[1] Aucouturier, J.-J., Pachet F.: Improving Timbre Similarity: How high is the sky?, *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.

[2] Flexer A.: On the Use of Self-organizing Maps for Clustering and Visualization, *Intelligent Data Analysis*, Vol. 5, Number 5, pp. 373-384, 2001.

[3] Flexer A., Schnitzer D., Schlüter J.: A MIREX meta-analysis of hubness in audio music similarity, *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012.

[4] Flexer A., Schnitzer D.: Can Shared Nearest Neighbors Reduce Hubness in High-Dimensional Spaces?, *Proceedings of 1st International Workshop on High Dimensional Data Mining (HDM)*, IEEE International Conference on Data Mining, 2013.

[5] Gasser M., Flexer A.: FM4 Soundpark: Audio-based Music Recommendation in Everyday Use, *Proc. of the 6th Sound and Music Computing Conference*, 2009.

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

[7] Jarvis R., Patrick E.A.: Clustering using a similarity measure based on shared near neighbors, *IEEE Transactions on Computers*, vol. 22, pp. 10251034, 1973.

[8] Karydis I., Radovanović M., Nanopoulos A., Ivanović M.: Looking through the "glass ceiling": A conceptual framework for the problems of spectral similarity, *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pp. 267-272, 2010.

[9] Knees P., Schedl M., Pohle T., Widmer G.: Exploring music collections in virtual landscapes, *IEEE multimedia*, 14(3):4654, 2007.

[10] Kohonen T.: *Self-Organizing Maps*, Springer, 2001.

[11] Mandel M., Ellis D.: Song-level features and support vector machines for music classification, *Proc. of the 6th Int. Conf. on Music Information Retrieval*, 2005.

[12] Nabney I.: *NETLAB: Algorithms for Pattern Recognition*, Springer Science & Business Media, 2002.

[13] Pampalk E.: *Computational models of music similarity and their application in music information retrieval*, Doctoral dissertation, Vienna University of Technology, Austria, 2006.

[14] Pampalk E., Dixon S., Widmer G.: Exploring music collections by browsing different views, *Computer Music Journal*, Vol. 28,No. 2, pp. 49-62, 2004.

[15] Radovanović M., Nanopoulos A., Ivanović M.: Hubs in space: Popular nearest neighbors in high-dimensional data, *Journal of Machine Learning Research*, 11:2487-2531, 2010.

[16] Rauber A., Frühwirth M.: Automatically Analyzing and Organizing Music Archives, *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries*, pp. 4-8, 2001.

[17] Sammon J.W.: A nonlinear mapping for data structure analysis, *IEEE Transactions on Computers*, vol. C-18, no. 5, pp. 401-409, 1969.

[18] Schedl M., Flexer A., Urbano J.: The Neglected User in Music Information Retrieval Research, *Journal of Intelligent Information Systems*, 41(3), 523-539, 2013.

[19] Schnitzer D., Flexer A.: The Unbalancing Effect of Hubs on K-medoids Clustering in High-Dimensional Spaces, *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2015.

[20] Schnitzer D., Flexer A., Schedl M., Widmer G.: Local and Global Scaling Reduce Hubs in Space, *Journal of Machine Learning Research*, 13:2871-2902, 2012.

[21] Schnitzer D., Flexer A., Tomašev N.: A Case for Hubness Removal in High-Dimensional Multimedia Retrieval, *Proceedings of the 36th European Conference on Information Retrieval (ECIR)*, 2014.

[22] Schnitzer D., Flexer A., Widmer G., Gasser M.: Islands of Gaussians: The Self Organizing Map and Gaussian Music Similarity Features, *Proceedings of the Eleventh International Society for Music Information Retrieval Conference (ISMIR'10)*, 2010.

[23] Serra X., Magas M., Benetos E., Chudy M., Dixon S., Flexer A., Gomez E., Gouyon F., Herrera P., Jorda S., Paytuvi O., Peeters G., Schlüter J., Vinet H., Widmer G., *Roadmap for Music Information ReSearch*, Peeters G. (editor), 2013.

[24] Stober S., Nürnberger A.: MusicGalaxy - an adaptive user-interface for exploratory music retrieval, *Proceedings of the 11th International Conference on Music Information Retrieval*, 2010.

[25] Sturm B. L.: Classification accuracy is not enough, *Journal of Intelligent Information Systems*, 41(3), 371-406, 2103.

[26] Tomašev N., Radovanović M., Mladenić D., Ivanović M.: The Role of Hubness in Clustering High-dimensional Data, *IEEE Transactions on Knowledge and Data Engineering*, Volume 26, Issue 3, 2013.

[27] Tzanetakis G., Cook P.: Musical genre classification of audio signals, *IEEE Transactions on Speech and Audio Processing*, Vol. 10, Issue 5, 293-302, 2002.

[28] van der Maaten L.J.P., Hinton G.E.: Visualizing High-Dimensional Data Using t-SNE, *Journal of Machine Learning Research*, 9(Nov):2579-2605, 2008.

[29] van der Maaten L.J.P., Hinton G.E.: Visualizing Non-Metric Similarities in Multiple Maps, *Machine Learning*, 87(1):33-55, 2012.