ADAPTING METRICS FOR MUSIC SIMILARITY USING COMPARATIVE RATINGS

Daniel Wolff and Tillman Weyde
Department of Computing
City University London
{daniel.wolff.1, t.e.weyde}@soi.city.ac.uk

ABSTRACT

Understanding how we relate and compare pieces of music has been a topic of great interest in musicology as well as for business applications, such as music recommender systems. The way music is compared seems to vary among both individuals and cultures. Adapting a generic model to user ratings is useful for personalisation and can help to better understand such differences. This paper presents an approach to use machine learning techniques for analysing user data that specifies song similarity. We explore the potential for learning generalisable similarity measures with two state-of-the-art algorithms for learning metrics. We use the audio clips and user ratings in the MagnaTagATune dataset, enriched with genre annotations from the Magnatune label.

1. MOTIVATION

In the recent years, increased efforts have been made to adapt MIR techniques, especially for music recommendation, to specific contexts or user groups. This is encouraged by developments in machine learning that make more algorithms applicable to accumulated user data, like user preferences or click-trough data for ranked search results, and enable the involvement of crowd wisdom into general classification and distance learning tasks. Moreover, the combination of different information sources has been proven successful for improving music recommendation and for classification into cultural categories such as musical genres.

This paper shows the results of some experiments on learning a musical distance metric from user similarity comparisons. Similarity models of mixed acoustic and tag features are trained using comparative user judgent data on song similarities. We derive information of the form ”Song A is more similar to Song B than to Song C”, represented by binary rankings, which allows for the application of more generic algorithms designed for learning from such data.

Although the above type of rating data is not as readily accessible as customer preference or social network data, it provides a valuable change of focus from general classification and recommendation success towards modelling musical similarity and the users' perception of it when engaged in a comparison task. Thus, instead of targeting a general relevance criterion, the optimisation task tackled in the following experiments addresses reported perceived similarity, which only constitutes one of the many variable aspects of relevance. As distance measures we use Mahalanobis distance metrics, which allow for a direct analysis as well as the easy comparison of learning results [5], and therefore encourage evaluation from a musicological perspective.

2. RELATED WORK

The distance metrics learning in this paper can be seen as an extension of feature selection techniques developed earlier in the MIR field, regarding feature selection as a binary weighting of features. E.g., Dash and Liu [4] assembled a comprehensive survey of general techniques for feature selection in classification tasks. They pointed out attributes relevant for diverse application scenarios, e.g. compability considering dataset size, number of classes or robustness against noise. These attributes enable a systematic comparison of the various approaches when given the parameters of a specific application. Pickens [13] categorised selection techniques for music retrieval using symbolic data, calling for special attention to features' musicological properties.

A set-based method for learning a feature weightings was applied by Allan et al. [1]. Users could specify their perceived similarity using two example song sets: one containing similar and one dissimilar songs. A detailed discussion on how to generate a successful stimulus partitioning for a survey involving comparison within triplets of clips supported the design of their Balanced Complete Block Partitioning.

2.1 Optimising Recommendation via Metadata and User Information

Out of the many data sources available for music description, genre annotations provide particularly valuable data for indexing and presenting music in recommendation settings. Musical genre has been used for the general evaluation of similarity measures, using the correlation of songs' genres and data clusters derived from the learned similarity [11,12].

Barrington et al. showed a training of linear combinations of SVM kernels relating to similarity measurements on acoustic, tagging and web-mined annotation data, for building classifiers for automatic annotation [3]. They also provide relevance levels of the different feature types for different tag classifiers.

A user-data based similarity measure for multimedia objects was introduced by Slaney [15]. Here, similarity of objects was based on users votings for them. Songs which feature the same grade of likeability by the same group of users were considered similar. The resulting similarity measure was evaluated via analysing artist consistency in rankings. Inferring similarity from similar metadata sources as well as music blog titles, Slaney et al. evaluated the performance of several methods for learning a Mahalanobis distance metric for music in [16]. McFee et al. [10] used the MLR algorithm (see below) for parametrising a content-based music similarity metric. A Mahalanobis metric was trained on collected crowd data in form of rankings. This approach is very similar to ours, but their emphasis has been on the need for reliable content-based classifiers for music discovery in sparsely annotated data.

Bade et al. [2] train a set of song-adaptive music similarity msasures for folksongs, inferring training data from expert classifications: Several known similarity measures for the symboloc music data and metadata are combined linearly via a weighted sum specific to the measured songs, its corresponding clusters or database. For optimisation, the expert classification information is transferred into relative distance statements enforcing the class members to be nearer than songs from foreign classes.

2.2 Metric Learning from Comparative Ratings

Many common algorithms for metric learning use class annotations and nearest neighbour classifications for optimising and evaluating metrics [18]. As we intend to learn music similarity from relative comparisons, such approaches are difficult to apply considering the missing ground truth data for clusters of perceptually similar music pieces or equivalents.

Based on a framework for Support Vector Machines, Schultz and Joachims [14] presented an optimisation using relative constraints we apply on the task of music similarity learning. Davis et al. formulated a metric learning problem as an LogDet optimisation task [5]. In this case, a fully parametrised Mahalanobis metric was learned, allowing for a regularisation towards another predefined Mahalanobis metric.

McFee et al. have designed an algorithm for learning a Mahalanobis metric to rankings (MLR) [9]. In our experiments, MLR is applied to learning a distance metric on music, using the implementation provided by the authors. In their publication mentioned above [10], this algorithm has been adapted to enable learning from collaborative filtering data.

3. THE MAGNATAGATUNE DATABASE

The MagnaTagATune database combines the results of a web-based game called "TagATune" together with the music clips used therein and extracted audio features [7]. These roughly 30-second long clips are provided by the Magnatune online music label on a creative commons license. Magnatune has labelled the clips in this database with 44 genre-tags, which are not mutually exclusive. The majority of the data can be divided into four disjoint main groups using the genre tags "classical", "electronica", "world" and "rock", each containing more than 17% of the total number of clips. The MagnaTagATune game is a collaborative online game with two modes: a regular mode for collecting tags and a bonus mode for collecting similarity ratings.

3.1 Captured Similarity Ratings

We extract relative similarity information from data collected during the "bonus" mode of the "TagATune" game. In that mode, two players earn points if they vote the same clip as the outlier out of three clips provided [8]. All votes made (matching or not) are saved into a histogram $h_i = \{h_a, h_b, h_c\} \in H$ for that triplet of songs. 533 such histograms are included in the MagnaTagATune database, describing the vote distribution (between 1 and 153 votes per triplet, 14 on average). Not counting permutations of triplets, there are 346 unique triplets comprising 1019 unique clips. Many histograms do not show a clear agreement on one outlier. This may be caused by the diverse nature of the clips, causing triplets normally to range over various genres, as discussed in [11]. However, many other variables like users' cultural backgrounds can equally affect their decisions. Content is homogeneously distributed throughout the complete 25863-clip database, but the small number of triplets available and the varying number of permutations do not allow for choosing a suitable subset featuring a Balanced Block Partitioning. This has been pointet out as important in [1] to obtain a relatively unbiased survey data set.

The above data was transferred into a ranking representation like in [9]. Treating the histograms as votings on the similarity between the outlier and the other clips, for each clip C_a, a set r_a^s of similar and, respectively, dissimilar clips r_a^d was calculated.

$$r_a^s = \{b \mid \exists h_i \in H : h_a < h_c \wedge h_b < h_c\} \quad (1)$$

$$r_a^d = \{c \mid \exists h_i \in H : h_a < h_c \wedge h_b < h_c\} \quad (2)$$

The complete set of derived rankings is then given by

$$O = \Big\{(r_a^s, r_a^d) \mid \exists C_a \wedge r_a^s, r_a^d \neq \emptyset \wedge r_a^d \cap r_a^s = \emptyset\Big\}. \quad (3)$$

Inconsistent rankings with $r_a^d \cap r_a^s \neq \emptyset$ were excluded to enable the following training process. In order to use the data with other algorithms, we removed further triplets[1]. All but 12 of the resuting rankings contain a single clip on each side: $|r_a^d| = |r_a^s| = 1$. This resulted in 533 rankings.

3.2 Feature Generation

The MagnaTagATune dataset comes with precalculated features for all clips extracted by the "The Echo Nest" API 1.0, via the "analyse" interface. These features are also included in other online databases such as the Million Song Dataset[2]. This also allows for a wider application of the feature extraction procedure detailed below and facilitates comparability with other studies. Of the wide feature range provided[3], we only use the chroma and timbre information. The chroma and timbre features are sampled on a non-uniform time scale. In order to aggregate to the clip level, we use a k-means based algorithm to extract $n = 4$ cluster centres for both of these features. In order to keep the features invariant to key, whilst preserving the harmonic and structural information, the chroma features are then transposed to fit the main key as estimated in the provided features, in the first chroma bin. This is achieved using a circular shift on the n chroma mean vectors. The resulting shifted chroma mean vectors are now separately normalised to a maximum value of 1.

The timbre features provided within the dataset very much resemble the output of a 2-dimensional convolution with 12 different filters, corresponding to characteristic spectral shapes. After clustering the timbre data to $n = 4$ mean vectors, these are scaled and clipped to retain 85% of the data within the interval of [0, 1] for the set of the 1019 clips. Additionally, the cluster weights for each of the included chroma and timbre cluster centroids are included in the features.

3.2.1 Genre Features

These acoustic features are enriched using the genre tags assigned by the Magnatune label. This way, up to four genre tags are assigned to each of the clips. For each clip, a binary 44-dimensional vector indicates the annotation according to the tags found for all of the clips in the dataset. The combination results in one feature vector $x_i \in (\mathbb{R} \cap [0,1])^{148}$ per clip $C_i, i \in \{1, \cdots, 1019\}$.

4. LEARNING SIMILARITY FROM COMPARISONS

The distance measure $d(x_i, x_j)$ we intend to optimise using the following algorithms is defined on the clip level. Generally, our approach and the corresponding features are intended to model a perceived distance, assumed to resemble the inverse similarity of two songs.

The ranking data in the following experiments has been approximated as a consensus from decisive triplet histograms, and is therefore simpler, e.g. contains fewer contradictory elements than the original data. Concerning the gathering of the histograms themselves, the authors of [1] emphasise that both the representation and especially the selection of combinations of the rated stimuli, in this case the clips, presented to the users, affect the balance of the resulting ratings. They only accept a set containing all possible triplet combinations of a set of stimuli for an unbiased test. Unfortunately, the triplets contained in the MagnaTagATune comparison data and the resulting ratings r_i are unbalanced. This may well include a bias caused by the specific constellation of graphical and acoustical presentations.

Using a metric for modelling song similarity implies several assumptions. These assumptions have already been questioned by Tversky [17], arguing that perceived similarity is not necessarily a linear, positive definite and symmetric function, which satisfies the triangle inequality. Instead, perceived similarity, in many circumstances, is assumed be directional, considering specific functions of the objects in comparison, e.g. prototype and referent.

However, the properties of a metric support efficient and robust learning algorithms for dealing with the highly sparse and often contradictory data involved in learning the song similarity. Also, metrics have a straightforward geometric interpretation. Thus, besides the comparison of songs, frameworks are available for comparing the metrics themselves. We now give a quick overview of the family of metrics used in our experiments before we focus on the way they are used in Section 5.

4.1 Mahalanobis distances

The two algorithms summarised below are designed to learn parametrised distance functions. These functions are special cases of Mahalanobis distances, which are defined as

$$d_W(x_i, x_j) = \sqrt{(x_i - x_j)^T W (x_i - x_j)}, \quad (4)$$

[1] Two histograms $\{h_a, h_b, h_c\}$ and $\{h_a, h_b, h_d\}$ were removed if they did not agree on the outlier, except if the outliers were c and d.

[2] http://labrosa.ee.columbia.edu/millionsong/

[3] http://developer.echonest.com/docs/v4/_static/AnalyzeDocumentation_2.2.pdf

where $x_i, x_j \in \mathbb{R}^N$ and $W \in \mathbb{R}^{N \times N}$.

To qualify as a metric, W has to be positive definite [19]. The algorithms we use only guarantee W to be positive semidefinite. The corresponding distance functions still satisfy the conditions of symmetry, non-negativity and the triangle inequality, but allow for $d_W(x_i, x_j) = 0$ whilst $x_i \neq x_j$ and therefore are called pseudometrics. This function is the Euclidean metric if W is the unit matrix. As detailed below, a Mahalanobis distance can be described as a weighted Euclidean distance applied to previously linearly transformed features.

4.2 SC03

In [14], Matthew Schultz and Thorsten Joachims present an SVM approach to learning a distance metric. The function learned here is parametrised by two matrices, a linear transformation A and the positive semidefinite W. For our experiments, $A = I$ contains the identity transformation and W is constrained to be a diagonal matrix. Thus d_W describes a weighted Euclidean distance metric.

In order to use the users' similarity data r_i^d and r_i^s, the rankings are converted into singular similarity statements of the form (a,b,c), where the clip C_a is more similar to C_b than to C_c. This leads to the following set of triplet constraints:

$$Q = \left\{ (a, b, c) \mid \exists \, (r_a^s, r_a^d) \in O : b \in r_a^s \ \wedge \ c \in r_a^d \right\} \quad (5)$$

For each training triplet (a, b, c), Schultz et al. consider the squared pointwise difference Δ^{x_i,x_j} of the transformed clips' features, which in this application case reduces to $\Delta^{x_i,x_j} = (x_i - x_j) \cdot (x_i - x_j)$ (note the point-wise product). The weighted differences of

$$\Delta^{\Delta}_{(a,b,c)} = (\Delta^{x_a,x_c} - \Delta^{x_a,x_b}) \quad (6)$$

are then used as constraints for the following optimisation problem (with $w = diag(W)$):

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2} w^T w + c_{SC03} \cdot \sum_{abc} \xi_{abc} \qquad (7) \\ \text{s.t.} \quad & \forall (a, b, c) \in Q_{train} : w^T \Delta^{\Delta}_{(a,b,c)} \geq 1 - \xi_{abc} \\ & w_{i,j} \geq 0, \ \xi_{abc} \geq 0. \end{aligned}$$

This minimises the loss defined by the sum of the slack variables ξ_{abc}, whilst regularising W using the Frobenius norm with $\frac{1}{2} \|W\|_F^2$. We used the SVM^{light} C++ implementation[4] to minimise the above term. The software returns w in form of its support vector expansion, containing the support (difference) vectors Δ_i^{Δ} of the corresponding hyperplane and their weights $\alpha_i y_i$. w can be easily retrieved using $w = \sum_{i=1}^n \alpha_i y_i \Delta_i^{\Delta}$.

4.3 Metric Learning to Rank

In [9], McFee et al. describe an algorithm for learning a fullly parametrised Mahalanobis distance (see Equation (4)) using ranking information. Presenting an algorithm based on Structural SVM, they compute W whilst assuring the margin between the given training rankings and possible different rankings of the training data [10]. This method uses binary rankings and evaluates results by the relative positioning of clips marked as relevant or irrelevant. A fully correct ranking positions the relevant clips r_a^s before the ones in r_a^d. The calculation of the associated loss involves standard IR measures for estimating the ranking loss, e.g. the area under ROC curve. For selecting the most effective constraints, a cutting-planes method [6] is used. Note that clips not named in the rankings stay neutral and have no effect on the loss.

The MATLAB® implementation of the MLR framework, available online[5], provides several options for choosing the cutting-planes method and loss function. In the experiments below, we selected the AUC-related methods for simplicity. In the literature, W is regularised by its trace $tr(W)$, but the implementation provided by McFee also allows to use a squared Frobenius norm, similar to the quadratic regularisation in (7).

5. EXPERIMENTS

All experiments were performed using five-fold cross-validation on the rankings. The ranking set O was divided into five disjoint batches of 106 or 107 rankings, respectively. Each batch was used once as a test set against the remaining four batches combined as training set. For smaller sized training sets, subsets were picked randomly from each of the training batches. The size of the test sets was kept constant for all training set sizes.

We tested three different variations of learning metrics: SC03 for learning a weighted Euclidean distance, MLR for calculating a full Mahalanobis matrix, and MLR with W constrained to be diagonal. The slack-loss / regularisation trade-off factors c were set to $c_{mlr} = 10000$ for both the diagonal and the full-W MLR, and $c_{SC03} = 100$ for the SC03 algorithm (Section 4.2). The squared Frobenius norm was used for regularising W in all experiments. These parameters were determined in earlier experiments using the present dataset with non-reduced training sets.

For evaluation, we compare the rankings in the ground truth with rankings induced by the learned distance functions. We also tested an unweighted Euclidean distance metric as a baseline. As we deal with binary rankings as described in Section 3.1, any ranking featuring the clips in r_a^s before the ones in r_a^d for a query clip a qualifies as correct,

[4] http://svmlight.joachims.org/

[5] http://cseweb.ucsd.edu/~bmcfee/code/mlr/

the absolute ranking positions were not taken into account.

5.1 Results

Figure 1 shows the results for running the above configuration on the features described in Section 3.2. The upper plot (a) shows the percentage of correctly induced rankings for the three metric learning approaches as well as the results for an unmodified Euclidean metric, serving as baseline. With 81.81% correctly reproduced test rankings and a standard deviation of 4.78% over the five test sets, the fully parametrised MLR-trained distance produces the best results, followed by the diagonal-MLR (71.85%, 2.69%) and SC03 (69.61%, 4.27%), barely superceeding the baseline of 67.74%. Both of the diagonal-W methods score rather low compared to the MLR-trained metric. Although the number of variables to determine is rather high, given the feature dimensionality, MLR proves successful in finding the best solution, except for the training with less than 50 rankings.

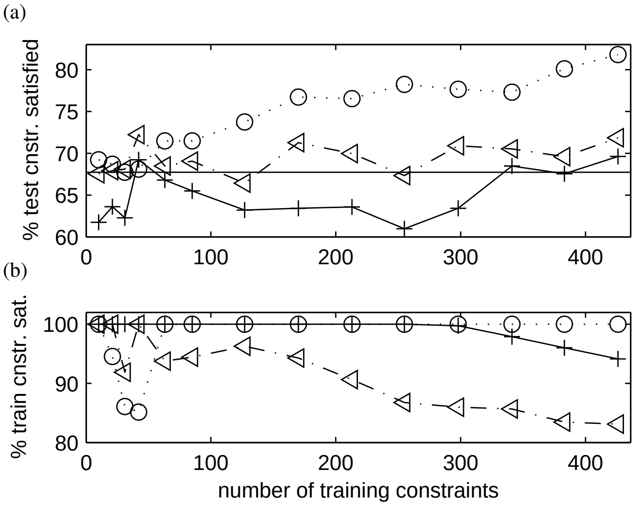

Figure 1. Results for increasing training set size. Plotted are the mean percentages of fulfilled rankings. MLR algorithm (○), MLR with diagonal W (◁), and SC03 (+). The performance of the Euclidean metric is represented by a straight line.

SC03 performs worst in this comparison, even dropping below the baseline during the medium-sized test-sets. As can be seen in Figure 1(b), SC03 performs much better than the diagonal MLR on the training set. This suggests an overfitting of SC03 and possibly insufficient influence of the regularisation loss. Overfitting depends strongly on the choice of c_{SC03}. The fact that the more flexible fully parametrised MLR-trained distance metric shows more flexibility towards the satisfaction of training constraints appears intuitive (Figure 1(b)). Lesser so, the better generalisation, which might be explained by the ability to spread the necessary adjustments in the metric across many parameters compared to the diagonally parametrised metrics.

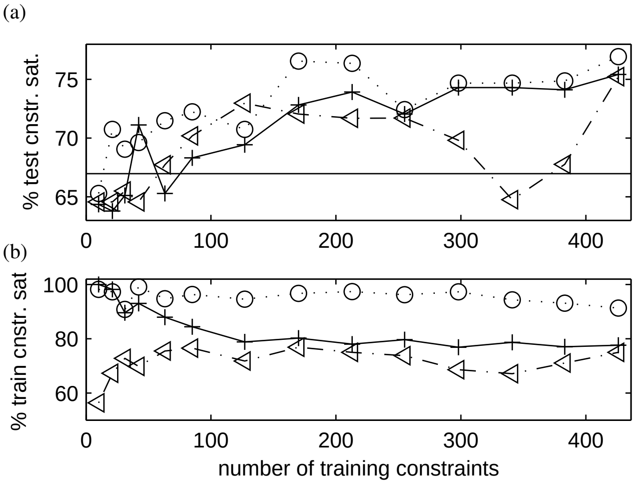

Figure 2. Results for increasing training set size using PCA features. Labels are as above.

5.1.1 *PCA features*

Figure 2 shows the results of applying the metric learning to a feature set that was reduced to 20 dimensions using Principal Component Analysis (PCA). As in the earlier experiment, MLR scores best, with (76.94%, 3.1%). The degradation may be attributed to the smaller number of parameters ($W \in \mathbb{R}^{20\times20}$) available for adapting the metric. However, when analysing the weights for the single feature dimensions, the ordering (by absolute value of the eigenvalues) used for determining the relevant pca dimensions does not correspond to their influence on the rated similarity. Thus, information relevant for similarity is lost in these PCA reduced features, which has been validated by the training of metrics using more PCA coefficients. In this experiment with 20 coefficients we compare the ranking of PCA coefficients, as determined by PCA data variance, with the ranking of PCA coefficients derived from the SC03 weighting. They differ on average by more than 52% of the index range.

With the PCA features, the SC03 algorithm greatly improves in performance, 75.42% indicating a higher suitability of the low-dimensional vector space. This time, a less effective enforcement of training constraints apparently enables a better generalisation. In contrast, the diagonal MLR is less able to cope with the data. Especially for the training sets involving around 300 rankings, the decrease in performance on the test set can be explained by less consistent training sets leading to badly generalising metrics. The baseline Euclidean metric achieves 66.97% of correct ratings.

6. DISCUSSION

In the present paper, we apply general algorithms for metric learning to a music similarity modelling task Using simple and widely available features and comparative similarity ratings, we demonstrated that a considerable proportion of the ratings can be effectively learned and reproduced using Mahalanobis distances. This corroborates the initial hypothesis that the ratings sharing some concordant information. Whilst with both the original features and the low-dimensional PCA features the MLR algorithm shows superior results, the diagonal matrix algorithms show comparable generalisation abilities for the PCA features. However, PCA seems not suitable for reducing feature dimensionality in a musical similarity context. Instead, the metric leaning techniques may hint on the necessary transformations and on which features may be ommitted.

6.1 Future Work

Despite the sparse and sometimes contradictory nature of the rankings derived from MagnaTagATune, we find the our results encouraging to develop more elaborate data sets for further experiments. Special attention will be given to the variation of learned metrics when observing different culturally defined user groups. More research has to be done in the development of specialised regularisation terms for metric learning algorithms, e.g. allowing for a customised W as a regularisation target [5].

6.2 Acknowledgements

We would like to thank Brian McFee and Thorsten Joachims for their help with using their implementations.

7. REFERENCES

[1] H. Allan, D. Müllensiefen, and G. Wiggins. Methodological considerations in studies of musical similarity. In *8th International Conference on Music Information Retrieval*, 2007.

[2] Korinna Bade, Jörg Garbers, Sebastian Stober, Frans Wiering, and Andreas Nürnberger. Supporting folk-song research by automatic metric learning and ranking. In *ISMIR*, pages 741–746, Kobe, Japan, October 2009.

[3] L. Barrington, M. Yazdani, D. Turnbull, and G. Lanckriet. Combining feature kernels for semantic music retrieval. In *ISMIR*, pages 614–619, 2008.

[4] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4):131–156, 1997.

[5] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, ICML '07, pages 209–216, New York, NY, USA, 2007. ACM.

[6] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77:27–59, October 2009.

[7] E. Law, K. West, M. Mandel, M. Bay, and J. S. Downie. Evaluation of algorithms using games: the case of music annotation. In *ISMIR*, pages 387–392, October 2009.

[8] M.I. Mandel and D. P. W. Ellis. A web-based game for collecting music metadata. *Journal of New Music Research*, 37(2):151–165, 2008.

[9] B. Mcfee and G. Lanckriet. Metric learning to rank. In *ICML*, 2010.

[10] L. McFee, B.and Barrington and G. Lanckriet. Learning similarity from collaborative filters. In *ISMIR*, pages 345–350, 2010.

[11] A. Novello, M. F. Mckinney, and A. Kohlrausch. Perceptual evaluation of music similarity. In *ISMIR*, 2006.

[12] E. Pampalk. *Computational Models of Music Similarity and their Application in Music Information Retrieval*. PhD thesis, Vienna University of Technology, Vienna, Austria, March 2006.

[13] Jeremy Pickens. A survey of feature selection techniques for music information retrieval. In *ISMIR*, 2001.

[14] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2003.

[15] M. Slaney. Similarity based on rating data. In *ISMIR*, 2007.

[16] Malcolm Slaney, Kilian Q. Weinberger, and William White. Learning a metric for music similarity. In Juan Pablo Bello, Elaine Chew, and Douglas Turnbull, editors, *ISMIR*, pages 313–318, 2008.

[17] A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

[18] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, 2009.

[19] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, 2002.