Inf Retrieval (2014) 17:109–136
DOI 10.1007/s10791-013-9229-0

Learning music similarity from relative user ratings

Daniel Wolff · Tillman Weyde

Received: 29 October 2012 / Accepted: 23 March 2013 / Published online: 4 July 2013

Abstract Computational modelling of music similarity is an increasingly important part of personalisation and optimisation in music information retrieval and research in music perception and cognition. The use of relative similarity ratings is a new and promising approach to modelling similarity that avoids well known problems with absolute ratings. In this article, we use relative ratings from the MagnaTagATune dataset with new and existing variants of state-of-the-art algorithms and provide the first comprehensive and rigorous evaluation of this approach. We compare metric learning based on support vector machines (SVMs) and metric-learning-to-rank (MLR), including a diagonal and a novel weighted variant, and relative distance learning with neural networks (RDNN). We further evaluate the effectiveness of different high and low level audio features and genre data, as well as dimensionality reduction methods, weighting of similarity ratings, and different sampling methods. Our results show that music similarity measures learnt on relative ratings can be significantly better than a standard Euclidian metric, depending on the choice of learning algorithm, feature sets and application scenario. MLR and SVM outperform DMLR and RDNN, while MLR with weighted ratings leads to no further performance gain. Timbral and music-structural features are most effective, and all features jointly are significantly better than any other combination of feature sets. Sharing audio clips (but not the similarity ratings) between test and training sets improves performance, in particular for the SVM-based methods, which is useful for some applications scenarios. A testing framework has been implemented in Matlab and made publicly available http://mi.soi.city.ac.uk/datasets/ir2012framework so that these results are reproducible.

Keywords Music similarity · Relative similarity ratings · Metric learning · Support vector machines · Metric learning to rank · Neural networks

D. Wolff (✉) · T. Weyde
Department of Computing, School of Informatics, City University, London, UK
e-mail: Daniel.Wolff.1@city.ac.uk

T. Weyde
e-mail: t.e.weyde@city.ac.uk

1 Introduction

Similarity plays a central role in music information retrieval as well as in music recommendation and in musicology. Storing music digitally has become less expensive, so that increasing amounts of data are available for algorithmic music analysis and comparison today. Increasing numbers applications and multimedia devices require the development of more elaborate techniques to automatically analyse, classify, index, and retrieve music. One requirement is the modelling of relationships between music clips, especially similarity as addressed in this paper.

Most commercial systems successfully use collaborative filtering for finding these relationships in music search and recommendation, The main drawback of collaborative filtering is that it relies on user behavioural data for every item to retrieve. But often there are little or no user behavioural data available, e.g. for new or less popular music, as has been pointed out by Celma (2008).

On the other hand, content-based approaches for music similarity and recommendation avoid these issues by modelling similarity based on the audio data. They have been shown to work well in some scenarios, and are now being used on a wider scale in web services like The Echo Nest (Jehan 2005) or The Freesound Project (Akkermans et al. 2011). Content based music similarity models need to incorporate the extraction of acoustic, psychoacoustic and music theoretic information derived from audio. The applicability of such extraction and the models is highly dependent on the context of the music, the application, and the user. Learning models that generalise from limited amounts of user data can help adapt the system to the users' needs and the designers' intentions for music where user data is not available.

Context-based and user-adapted retrieval have become popular research topics in music information retrieval (MIR) and computational musicology (e.g. see Ricci 2012; Serra 2012), following and fostering developments in machine learning that provide suitable algorithms. This work is part of a project on culture-aware music information retrieval, where the long-term aim is to use adaptable models to accommodate different cultural contexts and provide personalised search and recommendation.

So far, mostly tags or class information, such as genre labels, have been used to optimise distance measures. In this work we use relative similarity ratings collected during the collaborative *game with a purpose* (GWAP) MagnaTagATune . These ratings carry similarity information of the form: clip C_i is more similar to C_j than to C_k. The relative nature of the ratings avoids known problems with classes or absolute ratings. However, the relative ratings complicate the learning of the similarity measure. We apply in this study two types of models for learning similarity measures: Mahalanobis metrics optimised with a support vector machine (SVM) and metric learning to rank, including a novel weighted variant (WMLR), and a non-metric distance measure based on neural networks (RDNN). We evaluate these methods with cross-validation, assessing the training and generalisation error, and significance-tests. We further study the influence of the feature sets and feature dimensionality as well as the preparation of sampling methods in correspondence to different application scenarios.

The remainder of this article is organised as follows: Sect. 2 reports on related work and Sect. 3 introduces our methods for this study. Section 4 provides an analysis of the MagnaTagATune similarity data and the methods used for deriving audio and genre features for the clips. We present our experiments in Sect. 5 and discuss the results in Sect. 6. Section 7 closes this article with conclusions and perspectives for future work.

1.1 Our contribution

Our original contribution in this paper can be summarised as follows:

We introduce novel variants of methods for learning similarity measures from relative data:

- building and pruning of similarity relation graphs from odd-one-out experiments (Sect. 3.2.1)
- the WMLR/WDMLR method for learning from weighted relative similarity data (Sect. 3.4.3)
- a new approach of using RDNN for similarity learning (Sect. 3.5)
- the *inductive sampling* method for unbiased sampling of relative similarity data for cross-validation (Sect. 4.1.3)

We present the first comprehensive experimental comparison of the learning methods and an analysis of the dataset, in particular:

- a comparison and statistical analysis of the learning methods
- a comparison of different types of audio and genre features (Sect. 5.2)
- a novel analysis of the MagnaTagATune dataset (Sect. 4)

Some of the methods presented here include revisions of approaches presented in Wolff and Weyde (2011a, b, c, 2012), which are indicated in the text. The experimental results presented here are new, and new analyses of the dataset, novel algorithms and an extended evaluation with statistical analysis are provided.

2 Related work

The context of this study is music information retrieval, where a standard architecture for adaptive systems as sketched in Fig. 1 has become prevalent for information retrieval involving audio data (Bosma et al. 2006; Casey et al. 2008; Page et al. 2012). In this architecture, an audio clip is analysed with regards to a number of features using a diverse range of signal processing methods. The features are presented as a single vector per audio clip, representing a range from low-level features like zero-crossings to higher level properties, for instance dancability. The audio features can be complemented with professionally produced metadata and user annotations. When a query is processed, a matching process takes place, that typically involves classification or similarity. In adaptive systems the matching process is optimised, typically using supervised machine learning techniques. In most cases, similarity models optimise the dual problem of a distance measure. Ground truth consists of information on actual class membership or similarity values, against which the the adapted system is evaluated, typically with cross-validation. From this perspective we discuss in this section general and music specific work on similarity models, methods for collecting similarity data, and computational methods to learn from the data.

2.1 Learning similarity models from data

There is a considerable variety of computational approaches for learning similarity measures. Most similarity models are based on features, as proposed by Tversky (1977). Distance measures normally treat the feature dimensions uniformly, which ignores the different natures of features and their relations, e.g. the aspect of systematicity as pointed

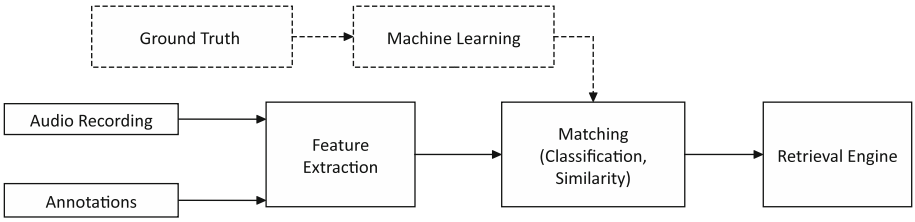

Fig. 1 Schematic architecture of an adaptive music information retrieval system

out by Gentner and Markman (1997). This can be addressed to some degree by using a Mahalanobis distance (Mahalanobis 1936) (see Sect. 3.3), which models weights and correlations between features.

2.1.1 Learning from absolute similarity ground truth

The following sections discuss the use of survey data for similarity learning. Many surveys collect *absolute similarity data* by asking for similarity ratings of two clips on a fixed scale, e.g. in the MIREX similarity evaluation[1] or in Ferrer and Eerola (2010). This approach expects the subject to make consistent similarity statements over time, which seems rather optimistic given the subjective nature of music similarity.

2.1.2 Learning similarity from classes

Consistency is less problematic in class information, which is a standard part of many datasets, e.g. genre labels. Therefore, it is interesting to use class information to adapt similarity ratings. A considerable range of distance learning methods has been used for learning from class information, including linear discriminant analysis, nearest-neighbour-based optimisation, and kernelised approaches such as SVM (Davis et al. 2007; McFee et al. 2010; Weinberger and Saul 2009; Yang 2006). The assumption is here that distances within classes should generally be smaller than distances between classes.

E.g. Novello et al. (2006) apply this in a "perceptual evaluation of music similarity". They collected relative similarity judgements from 36 participants on triplets of songs, and found a positive correlation of users' similarity ratings with musical genres. However, this is not by design, as class data are normally not designed to model similarity, but to represent other, often cultural, criteria.

An alternative approach is to gather *class-based similarity data* by asking subjects to classify clips by assigning them to one of a fixed number of unlabelled classes (e.g. Musil et al. 2012). This type of experiment typically requires choosing an appropriate number of classes beforehand, and does not solve the problem of inter and intra class similarities. Also, depending on the number of classes, class-based data often contains relatively little information.

2.1.3 Learning similarity to relative constraints

The problems of consistency in absolute ratings and the limitations of using classes could be avoided by learning from relative similarity ratings. This has been occasionally been adressed in MIR in the last decade.

[1] http://www.music-ir.org/mirex/wiki/2011:Evalutron6000_Walkthrough

Ellis and Whitman (2002) use relative similarity data from a comparative survey on artist similarity to evaluate similarity metrics based on similar artist lists from the All Music Guide[2] to define their ERDÖS distance. Their artist similarity data covers 412 popular musicians, for whom they gathered 16,385 relative comparisons. Moreover, they compare crowd-sourced similarity measures based on listening patterns and text analysis of web pages. The distance measures are regularised using multidimensional scaling (MDS) to fit metric requirements of symmetry and transitivity. They find that the unregularised ERDÖS distance outperforms the cultural crowd-sourced similarity measures.

Allan et al. (2007) discuss the challenges of gathering consistent relative similarity data via surveys. Besides introducing an interface for the interactive collection of song similarity data, they tackle the problem of subjects' coverage of survey examples. As already pointed out by Novello et al. (2006), it is usually not feasible to present all triplet permutations for even a medium-sized dataset to a single subject. Their approach of a *balanced complete block design* guarantees a balanced number of occurrences for individual clips and also accomplishes a balancing of the positioning of the clips within the triplets presented to a particular subject.

In Wolff and Weyde (2011a, b), we used the MagnaTagATune dataset (Ellis and Whitman 2002) to adapt similarity measures based on the relative similarity data in this set. This included SVM-Light to train a weighted Euclidean distance by Schultz and Joachims (2003) (see Sect. 3.4.4) and *Metric learning to rank* (MLR), adapting a full Mahalanobis distance (see Sect. 3.4.1) to relative similarity data. For a reduced version of the similarity data, our experiments showed some learning success. Stober and Nürnberger (2010) have also worked on the MagnaTagATune dataset, comparing algorithms for linear and quadratic optimisation of a similarity measure based on feature weighting. They apply early fusion of the feature data followed by adapting a linear model. They analyse the training methods on two different subsets of the similarity constraints (see Sect. 4.1). The smaller of which is designed to be solvable by all of the optimisation approaches, showing the learnability of a large subset of the data. For the larger set, where not all constraints can be learned, their SVM-based method achieves the best results. The early fusion approach can support better user understanding and interaction, and the results are similar to a late fusion approach(Wolff and Weyde 2012).

2.1.4 Infering music similarity from other user data

Instead of directly learning from similarity ratings, other data can be used to learn music similarity measures. Crowd-sourcing, as such a data source, makes use of the large numbers of people that can be reached through the Internet. Based on users' playlists, 'like' data, music purchase history and tag annotations, substantial datasets can be collected (Bogdanov et al. 2009; McFee and Lanckriet 2012). Models learnt from such data have been introduced in the recent years, but their applicability depends on the relationship of the data source to the application scenario. The approaches discussed below use data from crowd-sourcing to derive music similarity or relevance models structurally similar to those presented in this paper.

McFee et al. (2010) parametrise a music similarity metric using collaborative filtering data. They use Mahalanobis metrics to describe a parametrised linear combination of content-based features, using MLR for training. Post-training analysis of feature weights revealed that tags relating to genre or radio stations were assigned greater weights than

[2] http://www.allmusic.com/

those related to music theoretical terms. In our experiments in Sect. 5, we use MLR to adapt a music similarity metric to user ratings.

Slaney and White (2007) also presented a general method for learning a Mahalanobis distance metric. They adapt similarity on user "like" data. Their experiments evaluate the similarity metrics based on artist name identity of k nearest neighbours (kNN). They find that the collaborative-filtering based measure outperforms a content-based metric. The unknown variety of style given an artist is an instance of a general problem associated to using vaguely defined labels as classes. Secondly the imbalanced distribution of collaborative-filtering information in their data is discussed, as the pre-selection of the users' playlists influences the items they can "like". The variety of similarity models is later extended by Slaney et al. (2008), comparing six approaches of adapting content-based similarity on the same ground truth (unmodified, whitening, LDA, NCA, LMNN and RCA), showing significant improvement through training for all models.

The results for learning distance metrics from collaborative filtering and the availability of data from GWAPs motivate a systematic evaluation of such methods for similarity learning. The psychological view of similarity perception including asymmetry made clear that care is necessary when interpreting the results of learning similarity from data, as they depend on the information in the data, the context it has been collected in and the limitations of the preprocessing and the learning method. In the following, we introduce and develop the analysis and learning methods for ground truth similarity data as given in the MagnaTagATune dataset.

3 Modelling music similarity from relative user ratings

In this section we consider data mentioned from an odd-one-out game, like the MagnaTagATune dataset which we use in this study. In the game, three clips are presented to the players, who are asked to choose the one which least fits with the others. This selection indicates a relatively higher similarity between the two remaining clips than to the selected one. In the following we describe data structures and algorithms for using this data to optimise similarity measures.

3.1 Relative ratings from odd-one-out games

In an odd-one-out game, we gather relative similarity data in the form of relations between two pairs of clips. For example, given the clips C_i, C_j, C_k and C_l, we can express a similarity relation using the following:

$$(C_i, C_j) \overset{\text{sim}}{>} (C_k, C_l), \tag{1}$$

where the relation $\overset{\text{sim}}{>}$ denotes "more similar than". This can easily be applied to an odd-one-out survey: Given three clips C_i, C_j and C_k, a vote for C_k as the odd-one-out can be interpreted using the following two relations:

$$\begin{aligned} & (C_i, C_j) \overset{\text{sim}}{>} (C_i, C_k) \\ \wedge \quad & (C_i, C_j) \overset{\text{sim}}{>} (C_j, C_k). \end{aligned} \tag{2}$$

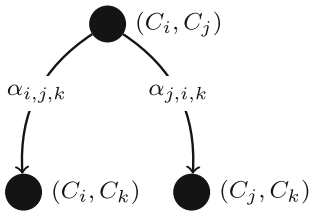

Fig. 2 Graph induced by a single "odd-one-out" statement, C_k is the odd-one-out as in Eq. 2. *Nodes* represent pairs of clips and edges represent the relation *more-similar-than*

3.2 Similarity graphs

Relative similarity relations can be represented as edges in a directed weighted graph of pairs of clips (McFee and Lanckriet 2009; Stober and Nürnberger 2011): Given the clip index I for all clips C_i, $i \in I$ and similarity information $\hat{Q}$ containing constraints in form (1), our Graph $G = (V, E)$ consists of vertices representing clip pairs

$$V = \{(C_i, C_j) \mid i,j \in I\}$$

and edges

$$E = \left\{\left((C_i, C_j), (C_i, C_k), \alpha_{i,j,k}\right) \mid (i,j,k) \in \hat{Q}, \alpha_{i,j,k} \in \mathbb{N} \setminus 0\right\}$$

representing the pairs' similarity relations. The weights $\alpha_{i,j,k}$ assigned to the edges represent the number of occurrences of a particular constraint (i, j, k). Such a graph as corresponding to Eq. 2 is shown in Fig. 2.

The induced graph can include inconsistent similarity information, for instance from users directly disagreeing on the outlying clip in a triplet, or multiple votes leading to an inconsistency when considering the transitivity of the induced similarity metric. Inconsistencies appear as cycles in the graph as shown in Figs. 3 and 4. Such cycles can be found and analysed using standard methods for extracting strongly connected components in directed graphs.

3.2.1 Removing cycles

The SVM and MLR training algorithms we use here require the similarity data to be consistent.

In order to apply these methods, we use an approach presented by Stober and Nürnberger (2011) for filtering inconsistent data. The information to be discarded is selected based on a minimal number of associated user votes: For removing direct inconsistencies we remove cycles of length 2 by removing the edge (i, j, k) with the smaller weight $\alpha_{i,j,k}$ and subtracting its weight from the weight $\alpha_{i,k,j}$ of the edge in the opposite direction. If two inconsistent edges have equal weight, both are deleted, possibly leaving a vertex disconnected from the graph.

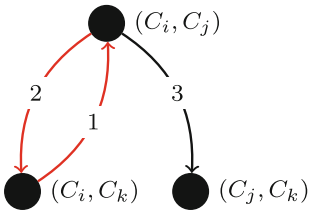

Fig. 3 Graph containing a length-2 cycle

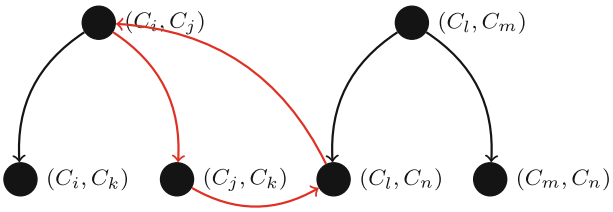

Fig. 4 Graph containing a length-3 cycle. Edge weights have been hidden

Removing cycles of greater length and finding the maximal acyclic subgraph of G is an NP-hard problem (Karp 1972). McFee and Lanckriet (2009) use a randomised algorithm by Aho et al. (1972) to extract an acyclic subgraph for this application. The graph is created by iteratively adding edges to a new graph and testing for cycles. Edges that complete a cycle are omitted. Depending on the similarity data, different means of finding an acyclic subgraph may give better or even optimal results. See Sect. 4.1 for the structure of the MagnaTagATune similarity data.

The resulting acyclic weighted graph provides the similarity constraints $(i, j, k) \in Q$ that we use to train the similarity measures. The analysis of the adjacent components in this graph gives information on transitive similarity relations expressed by the constraints (see Sect. 4.1).

3.3 Mahalanobis distance

The MLR algorithm, which we introduce in the next section, adapts a metric that was introduced by Mahalanobis (1936). The Mahalanobis metric d_W, which can be seen as a generalisation of the Euclidian metric, is defined as

$$d_W(x_i, x_j) = \sqrt{(x_i - x_j)^T W (x_i - x_j)}, \tag{3}$$

where $x_i, x_j \in \mathbb{R}^N$ represent our feature vectors and $W \in \mathbb{R}^{N \times N}$ is a *Mahalanobis matrix*, parametrising the similarity space. If W is the identity matrix, d_W is the Euclidean metric. If W is diagonal the feature dimensions are separately weighted within the distance function, as it is used with the SVM-Light and the DMLR algorithms introduced in the next section. If the full matrix W is positive definite, d_W satisfies all conditions of a metric (symmetry, non-negativity and the triangle inequality). We require W only to be positive semidefinite, so that $d_W(x_i,x_j) = 0$ for $x_i \neq x_j$ is possible, which makes the distance function a pseudometric (Weinberger and Saul 2009).

As described by Davis et al. (2007), each Mahalanobis matrix W induces a multivariate Gaussian distribution

$$P(x_i; W) = \frac{1}{\beta} \exp\left(-\frac{1}{2} d_W(x_i, \mu)\right). \tag{4}$$

Here, as in the standard definition (Mahalanobis 1936) of the Mahalanobis distance, W^{-1} represents the covariance of the distribution, β represents a normalising factor and μ the mean of the feature data.

With W derived from data covariances, the Mahalanobis distance can be used to calculate the distance from the data average or any another point in relation to the distribution of the data.

3.4 Metric learning

In this study we evaluate two state-of-the-art methods and a new variant for learning a Mahalanobis distance from relative similarity data: (D)MLR, the new W(D)MLR variant and SVM-Light are applicable to a multitude of data sources, with relatively little preprocessing and conversion required. They are based on Support Vector Machines, and thus work effectively with high-dimensional feature vectors that are commonly used for describing the music clips (see Sect. 4.2). Implementations of (D)MLR and SVM-Light algorithms are available as open source. Thus, modifications can be applied to the code as described in the following sections and comparisons of experiment results can be made easily by other researchers.

These algorithms parametrise a Mahalanobis distance from similarity constraints. Instead of using the covariance of the feature data data, the Mahalanobis matrix W is adapted to satisfy similarity constraints as derived in Sect. 4.1. Thus, not the feature data of the clip but the human similarity votes determine the similarity space. The resulting Mahalanobis matrix transforms the feature space when calculating similarity, allowing for dilations, rotations and translations to match the given similarity constraints. The rest of this section introduces the different algorithms used for optimising W.

3.4.1 Metric learning to rank (MLR)

McFee and Lanckriet (2010) describe the MLR algorithm for learning a fully parametrised Mahalanobis distance based on the SVM$^{\text{struct}}$ framework of Tsochantaridis et al. (2004). Specifically well-suited for use in retrieval environments, this method utilises rankings for the specification of training data as well as for the in-training evaluation of candidates for distance metrics. Such rankings assign a ranking position to each of the clips in our dataset given one of these as query item. For all constraints $(i, j, k) \in Q$, referring to $(C_i, C_j) \overset{\text{sim}}{>} (C_i, C_k)$, the final metric should rank C_j before C_k, when the query is C_i.

During the optimisation, ranking losses resulting from suboptimal metrics are determined using standard information retrieval performance measures. We use the area under the ROC curve as the measure for ranking loss. Violations of constraints are allowed for, but penalised using a single slack variable. Apart from the minimisation of the shared slack penalty, a regularisation term based on the trace $tr(W)$ of the Mahalanobis matrix is used in the optimisation.

In this study, we use a Matlab® implementation of the MLR algorithm, which McFee has published online[3].

3.4.2 DMLR

A variant of the MLR algorithm (DMLR) restrains W to a diagonal matrix with $W_{ij} = 0$ for $i \neq j$. Whilst still allowing for the weighting of different feature dimensions, rotations and

[3] http://cseweb.ucsd.edu/∼bmcfee/code/mlr/

translations in features space are ruled out by this restriction. For feature vectors $x_i \in \mathbb{R}^n$, this reduces the number of training parameters from n^2 to n.

3.4.3 Weighted learning with W(D)MLR

To our knowledge, no methods for weighted training with MLR have been published. MLR uses a 1-slack approach, prohibiting the weighting of individual constraints via their slack penalty. Instead we implemented the weighting by repeating individual constraints according to their weight. The repeated constraints gain their respective weight during slack aggregation, as the error is averaged along the training constraints. We call this method W(D)MLR. This approach is obviously not efficient, but for the MagnaTagATune similarity dataset it is feasible and the efficiency is improved by quantising the constraint weights. Experiments showed similar performance with using only fractions (10 %) of data overhead, which improves the scalability to larger datasets. The performance of weighted learning with WMLR and WDMLR is presented in Sect. 5.4.

3.4.4 Metric learning with SVM-Light

In Schultz and Joachims (2003), Schultz and Joachims present a metric learning strategy based on their SVM-Light framework[4]. Here, the matrix W, as introduced in Eq. 3 is factorised into a linear kernel transformation A and a diagonal matrix W. We use the identity transform as kernel $A = I$. Thus, d_W describes the Euclidean metric based on weighted features.

The proposed algorithm optimises the distance measure by representing it as the hyperplane dividing triplets (i, j, k), referring to $(C_i, C_j) \overset{\text{sim}}{>} (C_i, C_k)$, from triplets representing the contrary information (i, k, j). Clip pairs (C_i, C_j) are represented by the clips' feature difference: for each constraint triplet (i, j, k), we consider the component-wise squared difference of the involved clip pairs' features: $\Delta^{x_i,x_j} = \left((x_{i_1} - x_{j_1})^2, \ldots, (x_{i_N} - x_{j_N})^2\right)$. The differences of the pairs

$$\Delta^{\Delta}_{(i,j,k)} = (\Delta^{x_i,x_k} - \Delta^{x_i,x_j}) \tag{5}$$

are then used as constraints for the following optimisation problem:

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}||W||_F^2 + c_{SC03} \cdot \sum_{(i,j,k)\in Q_{train}} \xi_{(i,j,k)} \\ \text{s.t.} \quad & \forall (i,j,k) \in Q_{\text{train}} : <\text{diag}(W), \Delta^{\Delta}_{(i,j,k)} > \geq 1 - \xi_{abc} \\ & w_{ij} \geq 0, \ \xi_{abc} \geq 0 \end{aligned} \tag{6}$$

This minimises the loss defined by the sum of the per-constraint slack variables $\xi_{(i,j,k)}$ and regularises W using the squared Frobenius norm $\|W\|_F^2 = [3]tr(W^T \cdot W)$. Here, $c_{SC03} > 0$ determines the tradeoff between regularisation and slack loss. The implementation calculates the diagonal in W in its dual form on the basis of the support vectors. Given the support vectors $\Delta^{\Delta}_{(i,j,k)}$ and their weights $a_i\, y_i$, W can be easily retrieved using

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

$$\mathrm{diag}(W) = \sum_{(i,j,k)} a_{(i,j,k)} y_{(i,j,k)} \Delta^{\Delta}_{(i,j,k)}. \tag{7}$$

The resulting d_W normally turns out positive semidefinite, but this is not guaranteed. Cases occur where some of the $W_{ii} < 0$ are slightly below zero. This behaviour has also been reported for the LIBLINEAR framework by Stober and Nürnberger (2011). In these cases, the measure does not qualify as a metric or pseudometric but may still perform well in terms of training error and generalisation.

The SVM-Light toolbox allows for weights associated to constraints to be directly applied during training, by effectively weighting the individual slack variables $\xi_{(i,j,k)}$ in the penalty term of Eq. 6.

3.5 Distance learning using RDNN neural networks

Unlike the previous models, neural networks, specifically multi layer perceptrons (MLP), are capable of approximating arbitrary functions (cf. Hornik et al. (1989)). This means that more complex interactions of the features can be modelled than with a metric. This includes the distances measures where the triangle inequality doesn't hold or asymmetrical distance functions as discussed in Sect. 2.1. We don't do the latter in this study, as order information is not available in our dataset.

For our experiments, we have adapted a strategy presented by Hörnel (2004), based on earlier work by Braun et al. (1991), for making a neural network learn an absolute rating from relative information. This strategy is based on a combined network sketched in Fig. 5 with two MLP networks, *net*1 and *net*2, that have the same structure and share their weights. The input of each net is the vector of absolute differences a pair of feature vectors. From a similarity constraint, *net*1 gets the vector of the most similar pair, and should thus output a higher distance value than *net*2, getting the less similar pair. The outputs of *net*1 and *net*2 are connected to a comparator neuron c with negative fixed weight $-/+\,v$ for *net*1/*net*2 respectively. Thus c outputs a higher value if the correct input has not been achieved. The activation function of c is chosen to produce non-negative values, and the whole network can now be trained with target values of 0 for every training example.

Hörnel used a comparator neuron with sigmoid activation function, and a weight fixed with a negative sign for the 'left' network and a negative sign for the 'right' network. An alternative suggested by Braun (1997) is the use of a semi-linear activation function f_c for the comparator neuron, which we use as indicated in Fig. 5. We also introduce a margin between the higher and the lower ratings with a variable γ.

We developed an implementation of this scheme using a single network. This is based on the observation that the derivatives of the sum-of-squares error ($SSE(P)$) on a set of inputs P with regards to the output $n_1^{(p)}$ and $n_2^{(p)}$ of *net* 1 and *net* 2 for input p are

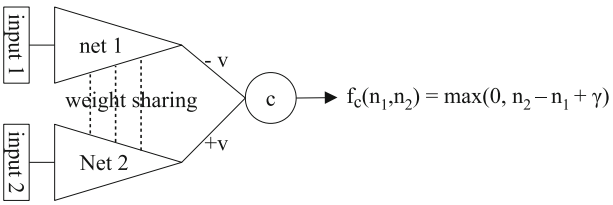

Fig. 5 Scheme for RDNN neural network learning from relative ratings

$$\frac{\partial sse(P)}{\partial n_1^{(p)}} = v \cdot (n_2^{(p)} - n_1^{(p)} + \gamma) \text{ and } \frac{\partial sse(P)}{\partial n_2^{(p)}} = v \cdot (n_1^{(p)} - n_2^{(p)} + \gamma). \quad (8)$$

This is equivalent to defining the target values of each net in terms of output of the other net:

$$t_1 = (n_2 - n_1 + \gamma) \text{ and } t_2 = (n_1 - n_2 + \gamma). \quad (9)$$

We used this to implement training on a single network with $\gamma = 0.5$ with resilient backpropagation (cf. Riedmiller and Braun (1993)) with regularisation. The procedure is described in listing 1.

Algorithm 1 RDNN Training

Require: Constraints Q_{train}, features $x_i \, \forall i \in I$, # of cycles k
Ensure:
  Define $D := \{(\delta^{x_i,x_j}, \delta^{x_i,x_k}) \mid \exists (i,j,k) \in Q^*\}$ ▷ training data
  Define $T := \{(t_{i,j}, t_{i,k}) \mid \exists (i,j,k) \in Q^*\}$ ▷ training targets
  $\text{MLP} = \text{initRandomMLP}()$ ▷ initialise MLP with random weights
  $Q^* = \{(i,j,k) \in Q_{train} \mid d_{\text{MLP}}(x_i, x_j) + 2\gamma > d_{\text{MLP}}(x_i, x_k)\}$ ▷ violated constraints
  $\text{cycles} = 0$
  while $\text{cycles} \leq k \wedge Q^* \neq \emptyset$ **do**
    for all $(i,j,k) \in Q^*$ **do**
      $\bar{d}_{i,j,k} = \frac{1}{2} * (\text{MLP}(\delta^{x_i,x_j}) + \text{MLP}(\delta^{x_i,x_k}))$ ▷ update training targets
      $t_{i,j} = \bar{d}_{i,j,k} - \gamma$ ▷ decrease distance for more similar pair by margin γ
      $t_{i,k} = \bar{d}_{i,j,k} + \gamma$ ▷ add margin for less similar pair
      $\text{MLP} = \text{trainRp}(\text{MLP}, Q^*, D, T, r)$ ▷ Train MLP with new targets
      $Q^* = \{(i,j,k) \in Q_{train} \mid d_{\text{MLP}}(x_i, x_j) + 2\gamma > d_{\text{MLP}}(x_i, x_k)\}$ ▷ update train set
      cycles++
    end for
  end while

The resulting MLP calculates a distance measure between two clips C_i, C_j, given the vector $\delta^{x_i,x_j} := |x_i - x_j|$ of absolute differences of the two clips' features:

$$d_{\text{MLP}}(x_i, x_j) = \text{MLP}(\delta^{x_i,x_j}). \quad (10)$$

4 MagnaTagATune dataset, analysis and preprocessing

The MagnaTagATune dataset is to our knowledge the only similarity dataset that is freely available[5] with the corresponding music data. Our experiments are based on this set to make our results reproducible and comparable.

4.1 Similarity data

In the bonus mode of the TagATune game, a team of two players is asked to agree on the odd-one-out of three audio clips. This is a typical instance of an output-agreement game with a purpose. Regardless of the success of the team, the votes of both users are saved in the history for this triplet. The MagnaTagATune dataset contains 7,650 such votes for a total 346 of triplets, referring to 1,019 clips. Some of the triplets have been presented as permutations, and the order of display is in the dataset, as well, but not the order of

[5] http://mi.soi.city.ac.uk/datasets/magnatagatune

listening. On average, each instance of a triplet permutation counts 14 votes. In our experiments, the information of each player's vote, e.g. C_k being the outlier in (C_i, C_j, C_k) is used to derive two relative similarity constraints as stated in Eq. 2.

The induced weighted graph, derived from $2 \cdot 7650 = \sum_{(i,j,k)\in\hat{Q}} \alpha_{i,j,k}$ votes, includes cycles of length 2, but no cycles of greater length. Thus, removing the cycles of length 2, removing 8,402 weight points, resolves all cycles existing in the initial graph. The resulting directed acyclic weighted graph consists of 337 connected subgraphs G^i_{sub}, each containing 3 vertices or clip pairs. The 6,898 weight points for 860 unique connections contain the remaining similarity information Q. Equal vote counts for inconsistent statements lead to the isolation of 27 vertices. Thereby, 26 songs are left without reference to any remaining similarity constraints, reducing the number of referenced clips to 993.

When excluding the isolated vertices with no associated similarity information, the combination of clips in the remaining subgraphs corresponds the triplets in the initial dataset, now associated with modified weights. This is due to the similarity triplets presented to the users, as explained above, and thus no information about interrelations of the different clip triplets can be directly extracted from the similarity data.

4.1.1 Genre distribution over triplets

In Sect. 2.1 we discussed the role of genre regarding the perceived similarity of music. Unfortunately, with this dataset, genre-specific similarity measures cannot be studied, as the datasets per genre are too small for similarity learning (Table 1). To give an impression of the dataset's structure, we divided the genre groups using the most frequently annotated genres:

4.1.2 Similarity weights

For the MagnaTagATune dataset, the numbers of votes (see Sect. 4.1) per constraint varies.

Since the weights of the edges are determined as the differences of conflicting votes, there is a compensation between total vote number and vote proportion: constraints with a small relatvie majority of votes but many votes in total can get the same weight as songs with a large relative majority but fewer total votes. We view this compensation as useful, because either factor can contribute to the confidence in the constraint. The separate use of proportion and vote count is interesting, e.g. in a probabilistic model, but is left for future work.

4.1.3 Sampling methods

In our experiments, the performance of the learnt metrics regarding the similarity data is evaluated using cross-validation. In k-fold cross-validation, the complete constraint set is divided into k disjoint subsets of approximately equal size. One of the subsets is held out during training and used for testing the performance. Our training data consist of three

Table 1 Number of triplets with n clips sharing the same genre tag

Genres	$n = 3$ of 3	2 of 3	1 of 3
Electronica, New Age, Ambient	43	159	447
Classical, Baroque	8	65	257
Rock, Alt Rock, Hard Rock, Metal	6	59	251

layers: the clips, the clip pairs, and the similarity constraints on the pairs. Disjoint sets of constraints can be based on the same pairs or individual clips, and disjoint sets of pairs can be based on the same clips.

Sampling for transduction In the odd-one-out dataset, the constraints are defined on triplets of clip pairs, and each pair of constraints on a triplet has one referenced pair of clips in common and references all clips in the triplet. Thus, when constraints from one triplet are divided between the test and training set, the two sets both reference one pair of clips and all individual clips in common. In our experiments presented in Sect. 5.3, the similarity constraints Q are randomly sampled subsets of constraints for 10-fold cross-validation, so that clips and clip pairs appear in several sets. One of these subsets is used as the test set Q^k_{test} of 86 constraints, while the remaining 9 subsets are combined to the training set Q^k_{train} of 774 constraints. Because of the random sampling of constraints, a triplet with 2 constraints, where one of the constraints is in the test set, has a chance of 90 % of the other constraint being in the training set. If the triplet has 3 constraints and one of them is in the test set, the chance of one of the other 2 being in the training set is 99 %. In our tests, the training sets referenced on average 989 clips out of the 993 total referenced clips.

We call this method *transductive sampling* (TD-sampling) because it enables transductive learning (cf. Gammerman et al. 1998). As our results in Sect. 5.3 show, the SVM-based approaches achieve better results with TD-sampling. TD-sampling can be an appropriate method for evaluation, e.g. for recommendation within a static database, but it does not support accurate performance predictions for unseen clip data.

Sampling for induction For assessing the capacity of a model to generalise over unknown pairs or individual items, *transductive sampling* is not suitable. In Wolff et al. (2012) we introduced and tested *inductive sampling* (ID-sampling), which separates similarity data the clip pair level. Rather than defining the subsets on the basis of constraints $(i, j, k) \in Q$, we use the disjoint subgraphs G^i_{sub} of the full similarity graph G (see Sect. 3.1). Choosing disjoint sets on the basis of these 337 disjoint subgraphs guarantees the sets to be disjoint with regards to the clip pairs (the vertices of G). In the MagnaTagATune dataset, after removing inconsistent edges, the subgraphs are also disjoint in terms of clips.

The G^i_{sub} differ in their number of edges because of unanimous votes or edge cancellation. Therefore the cross-validation sets vary slightly in their size. For the experiments in Sect. 5, 337 subgraphs have been divided into 10 subsets, each corresponding to 33 or 34 subgraphs. This results in subsets containing 85 constraints on average. The maximal training set size varies from 771 to 779 constraints referencing on average 896 clips, about 10 % less than in the TD-sampling, as expected. We use ID-sampling throughout this study, except where we explicitly test TD-sampling.

4.2 Content-based feature data

In this paper we use three types of features for representing clips in our models: low-level and higher level audio features, which we introduce in this section, and genre features that will be explained in the next section.

4.2.1 Low-level audio features

For the experiments in Wolff and Weyde (2011a, b), we only used the precomputed chroma and timbre vectors provided with the dataset. These were extracted with The Echo Nest API, version 1.0. This information as the basis for our features allows more reliable

reconstruction of audio features compared to the web-based and regularly updated API of The Echo Nest.

The chroma and timbre vectors are provided on a per-segment basis, with the clips divided into segments of relatively stable frequency distribution (details can be found in Jehan 2005). For each of these segments, the MagnaTagATune dataset contains a single chroma and timbre vector, each $\in \mathbb{R}^{12}$. We used two modes of aggregation, averaging and clustering, which we compare in Sect. 5.2.

In most of our experiments, we aggregate this information to the 30 s time scale of a clip. Like in Stober and Nürnberger (2010), a straightforward approach is to take the mean and variance of the features over time and use these values for representing the clip. We conducted experiments with the variance of chroma and timbre, but found them not to be helpful features. Thus, in Sect. 5.2 we only evaluate features based on the means of chroma and timbre values, i.e. for each clip $C_i,\ i \in \{1, \ldots 1019\}$, a single timbre average t_i^1 and chroma average $c_i^1, t_i^1 \in \mathbb{R}^{12}$ and $c_i^1 \in \mathbb{R}^{12}_{\geq 0}$, are extracted.

Aggregation by Clustering Previous experiments (Wolff and Weyde 2011a, b, 2012) did not use a single average but 4 cluster centroids $t_i^j \in \mathbb{R}^{12}, c_i^j \in \mathbb{R}^{12}_{\geq 0}, j \in \{1, \cdots 4\}$ for each feature and clip $C_i,\ i \in \{1, \ldots 1019\}$. The idea of this approach is to preserve some of the variety of harmony and timbre in the clips. The centroids are extracted with a weighted k-means variant, which accounts for the differing durations of the individual segments: centroids are influenced more strongly by feature data from longer segments. The final relative temporal weights of the cluster centroids are saved in scalars $\lambda(c_i^j), \lambda(t_i^j) \in [0, 1]$.

Normalisation and clipping Following aggregation, the centroids or averages of the chroma features are normalised to fit the interval [0, 1] using

$$\tilde{c}_i^j = \frac{c_i^j}{\max_k(c_i^j(k))}. \tag{11}$$

The timbre data is provided in an open numerical range $[-\infty, \infty]$ by The Echo Nest . This also applies to the extracted centroids and averages. In order to adapt the timbre feature data's range to those of the chroma and other features, the values are clipped to a maximum threshold. The clipping threshold was chosen such that 85 % of the timbre data values for the similarity dataset are preserved. Afterwards, the timbre data is shifted and scaled to fit $t_i^j \in [0, 1]$.

4.2.2 Higher-level audio features

In Wolff and Weyde (2011a, b) we restricted the set of features to the easily extractable low-level features mentioned above. Slaney et al. (2008) introduced a complementary feature set to facilitate the adaptation of music similarity measures to ground truth based on annotations. In their experiments, the segment-based chroma and timbre features were not used. Instead, they use those features from the The Echo Nest API which are already given on the clip level, as well as statistics for segment and beat locations and their frequencies. These features are the result of different classification, structure analysis and optimisation algorithms for music, which have been described in detail in Tristan Jehan's (2005) PhD thesis.

In the experiments presented in this paper, we complement the low-level features with higher-level features by reproducing the features by Slaney et al. (2008), as far as the required information is available in the MagnaTagATune dataset. Features where this was

Table 2 Features from (Slaney et al. 2008) used in our experiments

segmentDurationMean	tempo
segmentDurationVariance	tempoConfidence
timeLoudnessMaxMean	beatVariance
loudness	tatum
loudnessMaxMean	tatumConfidence
loudnessMaxVariance	numTatumsPerBeat
loudnessBeginMean	timeSignature
loudnessBeginVariance	timeSignatureStability

not the case have been omitted to ensure reproducibility of the experiments. Table 2 shows a list of the features used in this study.

Most of the features in Table 2 are directly based on the dataset. The "-Mean" and "-Variance" features represent the respective statistical operation on the provided feature data, with no further processing apart from a final normalisation, as explained in the following paragraph. The *beatVariance* feature represents the variance of the time between detected beats. If no beats are detected, the variance is set to zero. The *tatum* feature contains the median length of the inter-tatum intervals. Analogously, the *numTatumsPerBeat* feature results from the division of the median inter-beat interval by the tatum length as described above. If no tatum positions are detected, the *tatum* and *tatumConfidence* features are set to zero, while the *numTatumsPerBeat* feature is set to a default of 2.

Finally, each of these features is separately normalised over the values for the clips in the whole similarity dataset: The values are scaled and their minimal value subtracted to result in a one-dimensional $s_i^j \in [0, 1]$, for clips C_i. The features are not whitened as described by Slaney et al (2008), as we are interested in keeping the features' original associations to properties in music theory. For a comparison of PCA-transformed features' performance see Sect. 5.2.1. Note that some of the features allocate only a small number of actual values. For example, the *timeSignature* feature uses only the values $\{\frac{0}{7}, \frac{1}{7}, \ldots, \frac{7}{7}\}$.

4.3 Genre features

In addition to the audio features explained above, we use contextual information on the clips via tag-based features. We employ genre tags from the Magnatune label's catalogue, which is available online[6]. It contains descriptions of the songs containing the MagnaTagATune dataset's clips: Each song is annotated with 2–4 genre descriptions, which are also ordered from the most general to the most specific associated genre. We assign these genres as one binary vector $c_i \in \{0, 1\}^{44}$ per clip, setting positions j to 1 for each genre c_i^j and 0 otherwise.

5 Experiments

In the following, we present results from experiments we conducted to study the feasibility of similarity learning from relative ratings and to compare the effect of different algorithms, training parameters, features, and evaluation approaches on the training and generalisation results. All performances are evaluated with cross-validation based on the

[6] http://magnatune.com/info/api.html

percentage of unique distance constraints being satisfied by the learnt distance function. The distance constraints used below are extracted as described in Sect. 4.1.3. Following the strategy from Wolff et al. (2012), we start from a set of 13 constraints on average and increase the training set size $|Q^k_{\text{test}}(p)|$ for each cross-validation by extending the subsets.

Because the sampling and the choice of starting set have an influence on the result we extend the strategy here by repeating the procedure 4 times and averaging the results. We also use the 4·10 cross validation test sets for significance testing, applying a non-parametric approach. We use a Wilcoxon two-tailed signed rank test to compare the model trained on the full training set with the standard Euclidian metric—or another model as indicated—on each test set.

The following section compares the algorithms described above using the full feature set. The different feature types will be compared individually and in combined form in Sect. 5.2. Section 5.3 compares the ID-sampling, which was used in all other experiments, to TD-sampling. Finally, Sect. 5.4 explores the use of weight information in the similarity graph.

5.1 Comparison of learning methods

We compare MLR, DMLR, SVM-Light and RDNN . DMLR and SVM-Light learn a weighted Euclidean distance, while MLR is adapting a Mahalanobis distance with a full matrix W.

We use regularisation trade-off factors that have been determined using a grid-based search for the optimal configuration evaluated by cross-validation. The trade-off factors c were set to $c_{mlr} = 10^{12}$ for MLR, $c_{dmlr} = 10^2$ for the diagonally restricted DMLR (Sect. 3.4.1), and $c_{SC03} = 3$ for the SVM-Light algorithm (Sect. 3.4.4). The RDNN MLP network is set up with two hidden layers, containing 20 and 5 neurons, respectively. The MLP is trained in up to 38 training cycles or until all constraints are satisfied, which was not achieved. We tried longer training, but achieved no improvement of results.

Figure 6 shows the different algorithms using the combined features containing averaged audio and timbre features, Slaney08 features and genre features. This combination was chosen for showing relatively good results for all of the algorithms. Considering the training with the maximum size training sets, both MLR and SVM achieve similar performance on the unknown test set. DMLR and RDNN do not generalise well from the training set onto the test set (see Fig. 7).

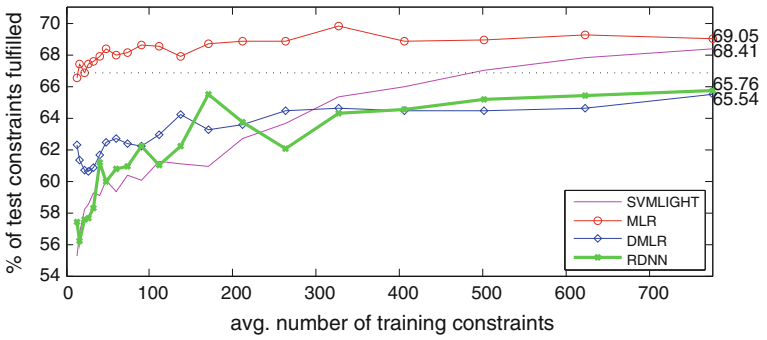

Fig. 6 Overall test set performance for combined features with averaged low-level information: SVM, MLR, DMLR and RDNN performance for full features, with increasing training set size. The dotted line shows the baseline performance of an unweighted Euclidean distance

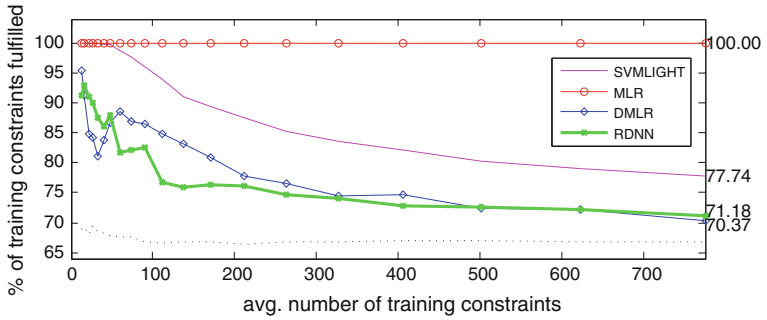

Fig. 7 Overall training performance: SVM, MLR, DMLR and RDNN performance for full features, with increasing training set size. The *dotted line* shows the baseline performance of an unweighted Euclidean distance on the training set

In this experiment the test for the largest subsets results by MLR and SVM-Light are approximately 2 and 1.5 % above the baseline of 66.86 %. At 5 % significance level only the MLR results are significantly better than the Euclidian metric ($p = 0.0007$). Both DMLR and RDNN remain below the baseline performance by 1 % on the test sets.

The generalisation results for small training sets $Q^k_{\text{test}}(p)$ depend highly on the algorithm used, and for SVM-Light , DMLR and RDNN lie considerably below the baseline. For SVM-Light , this is an effect of overfitting on small datasets, as we optimised the parameters for larger training sets. In Wolff et al. (2012), we suggest adaptive regularisation which could improve generalisation on small trainings sets if that is desired. MLR and SVM-Light exhibit different performance over different training set sizes: MLR starts around the baseline and reaches almost maximal performance within the first 100 training examples, while reaching almost 100 % on any training set, which may well be a sign of overfitting. SVM-Light starts with very low generalisation for small training sets and reaches the baseline performance at 500 training constraints. However, the results of SVM-Light continue to improve with the size of the dataset until the full number of training constraints is reached and are still clearly below the test results. This could also indicate overfitting, but again increased regularisation yielded no improvement and more data was not available.

The training set performance curves in Fig. 7 exhibit several particular types of learning behaviour. Note that the baseline (dotted line) slightly varies as the training sets grows. In each of the four samplings, the baseline can vary up to 10 % depending on the training subset. Like in earlier studies (Wolff and Weyde 2011b, 2012), MLR learns to fulfil all of the training constraints. The training performance of SVM-Light shows a continuous regularisation tradeoff, allowing for additional constraints to be learnt, whilst preserving good generalisation at the final full training set size. DMLR and the MLP show overfitting to the training examples for small training sets with a consistently inferior performance when compared to SVM-Light and MLR. With these algorithms, no gain is achieved on unknown test sets.

5.1.1 Training speed

We measured running times of the different algorithms as showing in Table 3. Comparison of these absolute runtimes does not necessarily reflect algorithmic efficiency, as SVM-Light is used in a compiled windows executable, while MLR, DMLR and RDNN run

within the MATLAB interpreter. Especially for the large feature spaces used with MLR and SVM-Light, RDNN (see Sect. 3.5) is still by far the slowest of the approaches described in this paper, using large amounts of time even for the small training sets.

5.2 Influence of feature type

As has been shown in Wolff and Weyde (2011b) both feature type and feature dimensionality have an influence on the algorithms' adaptation performances. We now present an evaluation of these parameters on the complete similarity data as described above. To this end, we compare the performances of SVM-Light using

- acoustic-only features
  - single chroma via average or 4 cluster centroids (chroma 1 / 4)
  - single timbre via average or 4 cluster centroids (timbre 1 / 4)
- genre-only features,
- slaney-only features,
- combined acoustic features and
- complete combined features.

The results for the different feature sets should be comparable without changing the algorithm's parametrisation. As we wanted to avoid an additional validation step for selecting c_{mlr} (see discussion in Sect. 7, we use SVM-Light as the most robust method for the examination of feature influence. For MLR the optimal regularisation tradeoff parameter c_{mlr} can vary by several orders of magnitude. We use again the unweighted Euclidean distance metric as baseline for all of the feature configurations.

Table 4 shows the performance of SVM-Light using different parts of the complete feature set available. The combined features achieve the greatest performance, followed by the Slaney08, timbre and genre features. The Slaney08 features (relatively high-level summary information), support particularly good generalisation (difference test vs. training

Table 3 Average training time per dataset in minutes, accumulated over all 20 subset sizes

SVM-Light	MLR	DMLR	RDNN
5	40	30	60

Table 4 SVM Single features test set performance

Features	Chroma(1/4)	Timbre(1/4)	Slaney08	Genre
Test	56.44 / 52.08	64.70 / 65.80	65.80	63.32
Training	61.60 / 59.48	68.97 / 66.27	68.06	68.91
Baseline	56.86 / 56.87	60.84 / 59.33	60.52	47.79
Features	Combined Acoustic(1/4)			Combined All(1/4)
Test	66.03 / 61.50			68.41 / 66.26
Training	71.53 / 76.08			77.74 / 83.92
Baseline	61.07 / 59.44			66.86 / 64.68

Values for single average audio features and 4-cluster audio features are separated by slashes (average/4-cluster)

set only 2.06 %). On the other hand, the chroma features are least effective on test set (difference to training set above 5 %).

Table 5 shows that the differences between the chroma features the others are statistically significant at the 5 % level. Most of the differences between the Slaney08, genre and timbre are not significant. However, the combined feature sets are significantly better than any individual feature set. Clustering vs. averaging makes a significant difference only for chroma but not for timbre or combined features.

Specifically notable is the low baseline of the genre features, which is probably due to the sparsely populated feature space. As each song is assigned 2–3 genres, only a few different distance values actually occur on the binary vectors. Therefore many constraints are not satisfied because of equal distance ($d_W(C_i, C_j) = d(C_i, C_k)$). A number of songs are annotated with exactly the same genres, so training on these constraints is not possible and degrades performance significantly (see Wolff and Weyde 2012).

5.2.1 PCA and impact of dimensionality

A common approach in MIR is to reduce the feature space dimensionality, which can help to make the learning quicker and more effective. For this experiment we use Principal Component Analysis (PCA) to reduce feature vectors to the same dimensionality. This serves also to explore whether the performance differences of the feature types are dependent on the dimensionality of the features. E.g. the combined features might give best performance, because the input feature vector has more dimensions.

We compare two sets of dimension-reduced features to explore the effect of dimensionality on learning: PCA12 and PCA52. PCA12 reduces the PCA-transformed information to the 12 dimensions carrying most of the variance. In PCA12 we used for single chroma mean features, timbre mean features, Slaney08 features, audio features combined, and all features combined. The chroma and timbre mean features already have 12 dimensions, the others are reduced. In the same manner, PCA52 features are built from 4-cluster chroma and timbre features, genre features, audio features combined, and all features combined. The 4-cluster chroma and timbre already have 52 dimensions (4

Table 5 Significance of performance differences between feature types (Wilcoxon signed rank p values)

Features	Chroma(1/4)	Timbre(1/4)	Slaney08	Genre	Acoustic (1/4)
Comb. All(4)	**0.000/0.000**	**0.001/0.000**	**0.000**	**0.000**	**0.000/0.002**
Comb. All(1)	**0.000/0.000**	**0.015/0.002**	**0.008**	**0.000**	**0.000/0.013**
Acoustic(4)	**0.000/0.008**	**0.002/0.006**	**0.000**	0.145	**0.000**/–
Acoustic(1)	**0.000/0.000**	0.753/0.179	0.823	0.116	–/**0.000**
Genre	**0.000/0.000**	0.076/0.244	**0.037**	–	
Slaney08	**0.000/0.000**	0.751/0.505	**0.000 / 0.000**	–	
Timbre(4)	**0.000/0.000**	0.251/–			
Timbre(1)	**0.000/0.000**				
Chroma(4)	**0.000**/–				

Features	Comb. All (1/4)
Comb. All(4)	0.086/–

Significant values at the 5 % level are set in bold type

12-dimensional chroma or timbre vectors with 1 weight value each). The Slaney08 features do not have enough dimensions to build a single high-dimensional PCA feature, but they are still included in the combined audio and combined all features. As above, SVM-Light is used for comparing the effectiveness of the different feature types and the results are shown in Table 6.

Table 6 shows that learning on the PCA12 chroma features did not improve generalisation results. The Slaney08 and timbre features both provide significant performance increase over chroma data. The combined features further improve the performance, with PCA12 all-features-combined reaching better result than the original features (see Fig. 6).

All pairwise differences in test performance between feature types are significant at $p\%$, except timbre versus Slaney08 and Slaney08 vs. genre. indicating that the reduced dimensionality makes learning more effective, at least with SVM-Light . It also provides evidence that the combination of different feature types is still effective, even when the dimensionality is reduced. As above, most of the training success is achieved with small training set sizes, up to 100 constraints.

The test set results of PCA52 features are mostly similar to PCA12, but the performance is generally lower for the single features. Interestingly the performance of timbre features drops by 7 % in comparison to both the raw and the PCA12 features. Similar to the 12-dimensional case, all pairwise differences are significant except timbre vs. genre.

The training performance, as in Table 6, indicates that the bad generalisation of 52-dimensional features is a result of overfitting: The training performance of 52-dimensional PCA features is considerably (3–5 %) higher than the performance of 12-dimensional PCA feature, while the baseline of the 52-dimensional features is much lower (−5 % for all except genre features). Thus, the performance gained for training data is far greater than for the 12-dimensional features. This indicates increased learning capacity of the model based on the 52-dimensional data. With increasing dimensionality, maximal performance needs more data. The generalisation does not improve, indicating that quantity or quality of the MagnaTagATune similarity data is not sufficient: The increased number of parameters allows for more specific optimisation whilst delaying the generalisation resulting from larger training sets. So the higher dimensional data might lead to better results if more data were available. However, the generalisation performance between PCA12, PCA52 and unreduced all-combined features on the maximal training set is not significantly different. The combined features achieve a very similar performance to the raw features in Table 4.

Table 6 SVM Single features test and training performance

Features	Chroma	Timbre	Slaney08	Genre	Audio Comb.	Combined
Test12	55.54	64.22	62.00	60.20	66.65	69.73
Training12	59.43	66.74	63.03	62.77	69.324	71.18
Baseline12	55.81	61.40	59.42	60.12	58.37	66.86
Gain12	−0.27	2.82	2.58	0.08	8.28	2.87
Test52	51.71	57.41	/	61.46	63.73	69.50
Training52	64.41	68.03	/	65.43	71.50	75.78
Baseline52	50.70	51.28	/	58.26	53.02	55.93
Gain52	1.01	6.13	/	3.20	10.71	13.57

The Slaney08 features are not available to 52-dimensional PCA features

Finally, the differences between the different feature types are all significant, indicating that the choice of features is important. In particular combining information sources can lead to improved performance.

5.3 Sampling: effects of transductive learning

We have compared TD-sampling, as introduced in Sect. 4.1.3), with ID-sampling that was used in the experiments so far, In TD-sampling pairs and individual clips (but not constraints) can appear in both training and test set. Figure 8 shows the results for the SVM-Light, MLR and DMLR algorithms. The performance of an unweighted Euclidean distance measure for the test sets is again our baseline. During cross-validation, baseline results are averaged over all test sets and the average performance is calculated for the whole dataset. With TD-sampling, both MLR and SVM-Light performance are significantly better than the baseline (both $p < 0.001$).

The training performance of all algorithms displayed is similar to the performance with ID-sampling as plotted in Figure 7. In contrast, the performance on the test sets, as in Figure 8, shows a considerable increase of performance (6 %) for MLR and a slight increase for SVM-Light . This reproduces the findings of Wolff et al. (2012). The explanation for the positive effect seems to be that involving almost all the feature vectors of the test set in training allows for MLR to make better decisions when the separation oracle selects the instances of the constraints to involve in the optimisation process (see Sect.3.4.1), while for the Support Vector Machine SVM-Light , the set of possible support vectors is increased with the number of feature vectors, increasing by 10 % (93 clips, see Sect.4.1.3) due to the TD-sampling referencing more feature vectors during training.

5.4 Weighting constraints by vote differences

As described in Sect. 4.1, the 860 unique similarity constraints represent differences of 6,98 votes after cancellation in the similarity graph. The vote difference for each edge can be used as an indicator for the reliability of the constraints. In the following experiment each constraint (i, j, k) is weighted in proportion to its weight $\alpha_{i,j,k} > 0$, using the weighted MLR training introduced in Sect. 3.4.3 and weighted SVM-Light (see Sect. 3.4.4).

Instead of using the unweighted evaluation considering the unique constraints satisfied, as used above, we measure the *weighted performance* of a metric as sum of the weights $\sum$

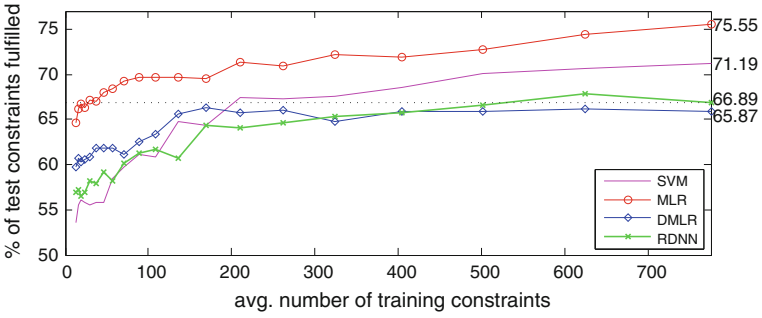

Fig. 8 Transductive sampling: SVM, MLR, DMLR and RDNN test set performance for full features. The training set size increases from left to right

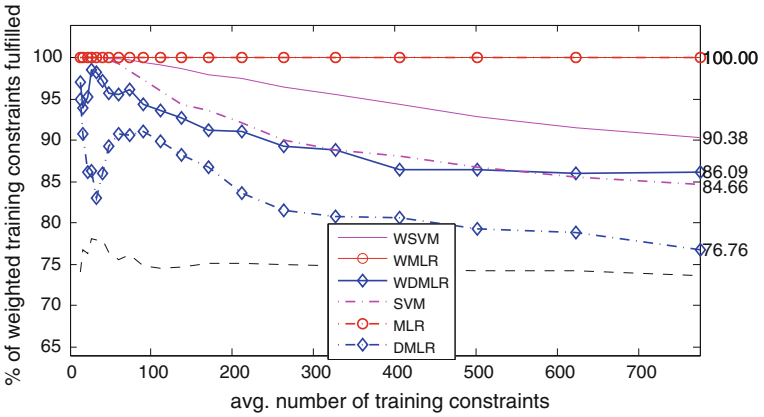

Fig. 9 Overall training performance, weighted evaluation (E:W) for training with: SVM, weighted SVM (WSVM), MLR, DMLR, WMLR and WDMLR. The bottom dashed curve displays the weighted baseline performance

$\alpha_{i,j,k}$ of $(i,j,k) \in Q_{test}$ or $(i,j,k) \in Q_{train}$ satisfied by the metric divided by the total sum of weights in the respective set.

Figure 9 shows the weighted performances on the training sets of weighted training with WMLR, WDMLR, and SVM-Light . We compare these to weighted performance (E:W) of the unweighted training with MLR, DMLR, SVM and an Euclidean metric. For the Euclidian metric, the weighted evaluation yields about 6 % better performance than using unweighted evaluation, indicating a correlation of the weighted constraints with the Euclidean distance in feature space. For WMLR and MLR, noth satisfying 100 % of the unique training constraints, the weighting makes no performance difference. The results of the other algorithms improve by similar amounts as the baseline. This shows the weighted learning approach described in Sect.3.4.3 succeeds in improving results towards the weighting of the constraints supplied during training.

When considering the test results of weighted training with WMLR, WDMLR, and SVM-Light , only WDMLR exceeds the baseline performance for weighted evaluation, which is also the only significant result on test sets in this comparison. Given that the DMLR training performance was lower than for the other algorithms, this seems to indicate that the lower model complexity of WDMLR allows more effective learning on the weighted dataset.

When considering the unweighted performance of the models learnt from weighted constraints, they perform worse than in Fig.7, but still significantly better than the baseline. Overall, the use for weighted data from MagnaTagATune seems not to improve the generalisation of learnt models. The weighted training is effective on the training data but on test sets only WDMLR can reach significant improvement above the baseline. However, as the distribution of weights depends on both the number of votes and the ratio of conflicting vote (see Sect. 4.1.2), there is no straightforward interpretation of these results.

6 Discussion

In this section we discuss and contextualise the results of the dataset analysis and experiments.

6.1 Learning results

The experiments presented here have shown, that learning similarity measures from relative user ratings can achieve significant improvements over a standard Euclidian metric, yielding a accuracy of almost 70 % on test constraints. The results are better when transductive learning is included by using TD-sampling, reaching 75.5 %. TD-sampling can be useful, e.g. in a closed database scenario, but depends on the training set covering a large proportion of the clips in the database.

These results leave room for improvement, and we discuss possible potential options for further development. A relevant question is whether we can expect better results from improving the algorithms and procedures, acquiring more or better data, or from changes in the approach.

6.2 Choice of algorithms

The tested algorithms show different behaviour, on different features and different similarity data. The choice of algorithm clearly depends on the scenario: for ID-sampling both MLR and SVM-Light achieve significant improvements over the Euclidian metric. MLR results are better, especially in training, but SVM-Light , reaching similar generalisation results, is more efficient in the implementation we used and thus the resulting metric can be calculated more efficiently. For TD-sampling, only MLR achieves significantly better results than the Euclidian metric and the improvement is smaller than for ID-sampling. WDMLR is the most effective when using weighted training, but DMLR performs much worse than MLR and SVM-Light in all other tasks.

The experiments with RDNN show low performance in all tasks despite the potentially higher flexibility of the model. However, the near perfect training performance of the MLR shows that the flexibility of the Mahalanobis matrix is already sufficient. There are alternatives for network architectures and parametrisations that we have not yet explored, so that there may be potential for improvement.

All algorithms showed high differences in performance between training and test sets, even with optimised regularisation. This indicates that improving the amount of data may lead to either improved results or to a high level of noise in the data.

6.3 Input features and preprocessing

The reduction of the input dimensionality with PCA (Sect. 5.2.1) has no significant effect on the generalisation with either the 12- or the 52-dimensional feature sets, although the training results improve considerably with larger feature dimensionality. This is likely the result of the number of parameters increasing with the feature dimensionality. The generalisation results show that the SVM-Light algorithm is robust and extracts relevant information from input data in high and low dimensions. Higher dimensional features might improve in generalisation given a greater amount of training data.

On the other hand, the choice of input features has significant effects in almost all experiments, even if the input dimensionality is normalised as in the PCA12 and PCA52 datasets. Chroma features generally perform poorly, while genre, timbre and the music-structural features defined by Slaney et al. (2008) provide useful additional information. The calculation of clusters for chroma and timbre features provides additional information to the system. Although earlier experiments with MLR show small improvements for 4-cluster features, the simpler averaging features show more stable results while there was

no significant difference in the overall performance. The single most effective way to improve the performance is to combine different types of features, which yields significant improvements over all individual features, regardless of whether clustering or dimension reduction is applied or not.

6.4 Data quality and quantity

The MagnaTagATune similarity dataset is the only available dataset of its kind and therefore worth studying. However, the analysis reveals that there several issues that impede effective learning and interpretation of results. When compared to psychological studies, the weighting data does not fulfil criteria of balancedness to allow for any conclusions. Even for the general MagnaTagATune similarity dataset, we found that the data has an unsystematic distribution of genres over the test triplets. In informal tests on the MagnaTagATune dataset, subjects found it difficult to make a decision in the odd-one-out scenario, because each of the clips came from a different genre. The lack of reappearance of songs in between triplets (see Sect. 4.1) also prevents the study of learning transitivity.

The results consistently support the interpretation that the learning performance is limited by the size and the quality of the dataset. Thus, collecting more data in a more balanced way is a promising way to potentially improve results.

6.5 Approaches for improvement

One possible approach for improvement is the selection of the stimuli and feature extraction process. The 30 s clips may introduce artefacts or uncertainties that might prevent reliable similarity judgements. However, subjects in informal tests reported no issues with the length of the stimuli. The features tested here are already of different types, but it seems interesting to develop new features that model more aspects of musical structure. However, the low ratings of chroma values, which are associated with the distribution of pitch classes, suggests this is not a straightforward task.

Another approach is the use of user data and more cultural context information. As discussed in Sect. 2.1, perceived similarity can depend on context of the objects and the subject, especially cultural terms of reference. Both music metadata and user related information could help improve the learning results by enabling selective training set for multiple models or incorporating contextual information into the model. In addition to user information, multiple models or contextual models will require more and more balanced data than currently available. Both approaches can enable personalised and contextualised music information retrieval, providing not only improved machine learning, but also improved services for users. In addition, such models could provide information to researchers on cultural aspects of music perception.

7 Conclusions and future work

In this study we addressed learning music similarity measures from relative user ratings. To this end we analysed the MagnaTagATune similarity dataset and applied a number feature extraction and machine learning techniques. We evaluated the learning success in relation to a number of choices regarding features, algorithms and scenarios. The main findings can be summarised as follows:

- Learning of metrics based on relative user ratings is possible with the tested features and algorithms. The performance on unseen test data can be significantly improved, depending on the application, the choice of algorithm, and features used.
- Mahalanobis metrics, and often weighted Euclidian metrics, are sufficiently flexible to model similarity relations in the given data, as the more flexible model.
- For SVM learning on the given dataset, chroma features are least effective, and combinations of different feature types are most effective, independent of dimensionality reduction and clustering vs. averaging of timbre and chroma data.
- The test performance leaves considerable room for improvement, which we attribute mostly to the dataset used.

As the results show, using machine learning is a good choice on a static dataset. For a dynamic MIR scenario and a small data set like the MagnaTagATune for training, the results are not yet on the level needed for many applications.

Given the successful application of the MLR and SVM-Light algorithms in other contexts (Galleguillos et al. 2011; Mcfee and Lanckriet 2010; Schultz and Joachims 2003) the main areas for work towards improved performance on new data are the quantity and quality of the training data. Another approach is the extraction of features that capture more of the musical structure. Generally, a better understanding of music perception and cognition and its cultural dimensions can help improve the development of MIR systems that meet user needs.

7.1 Future work

As discussed in Sect. 5, setting the regularisation parameters is a difficult but crucial step for reaching optimal training performance. Particularly for computationally expensive algorithms like MLR, optimisation can be very costly. For learning with growing training sets, plans are to adapt regularisation dynamically, proportional to the number of training examples.

The drawbacks of MagnaTagATune dataset are being addressed in a similarity data collection framework which is currently being tested at City University. It allows for a controlled presentation of same and different-genre triplets as well as for a balancing of triplet permutation and recurrence of songs across different triplets. Ultimately, we are interested in researching and modelling the impact of cultural factors on reported clip similarity. To this end, the user similarity votes are being annotated with user-provided information, the cultural indicators. By correlating these indicators with parameterisations of learnt similarity models we hope to establish better user models. These user models can then be used for further research and should enable better learning success to support group-specific or personalised music recommendation and retrieval.

Acknowledgements We thank Brian McFee for providing and maintaining the MLR code and Thorsten Joachims for providing the SVM-Light software and his support with using the solver. We would also like to thank Andrew Macfarlane and Gregory Slabaugh for their helpful comments on this work.

References

Aho, A. V., Garey, M. R., & Ullman, J. D. (1972). The transitive reduction of a directed graph. *SIAM Journal on Computing, 1*(2), 131–137.

Akkermans, V., Font, F., Funollet, J., De Jong, B., Roma, G., Togias, S., et al. (2011). Freesound 2: An improved platform for sharing audio clips. In *International Society for Music Information Retrieval Conference (ISMIR 2011), Late-breaking Demo Session*. Miami, Florida, USA.

Allan, H., Müllensiefen, D., & Wiggins, G. (2007). Methodological considerations in studies of musical similarity. In *8th International conference on music information retrieval*, pp. 473–478.

Bogdanov, D., Serrà, J., Wack, N., & Herrera, P. (2009). From low-level to high-level: Comparative study of music similarity measures. In *IEEE International symposium on multimedia. Workshop on Advances in Music Information Research* (*AdMIRe*).

Bosma, M., Veltkamp, R. C., & Wiering, F. (2006). Muugle: A modular music information retrieval framework. In *International symposium on music information retrieval*.

Braun, H. (1997). *Neuronale Netze—Optimierung durch Lernen und Evolution*. Springer, Berlin.

Braun, H., Feulner, J., & Ullrich, V. (1991). Learning strategies for solving the planning problem using backpropagation. In *Proceedings of NEURO-Nimes 91, 4th international conference on neural networks and their applications*.

Casey, M., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., Slaney, M. (2008). Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE, 96*(4), 668–696.

Celma, O. (2008). *Music recommendation and discovery in the long tail*. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona.

Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007). Information-theoretic metric learning. In *Proceedings of the 24th international conference on machine learning, ICML '07* (pp. 209–216). New York, NY, USA: ACM.

Ellis, D. P. W., & Whitman, B. (2002). The quest for ground truth in musical artist similarity. In *Proceedings of the international symposium on music information retrieval (ISMIR)* (pp. 170–177).

Ferrer, R., & Eerola, T. (2010). Timbral qualities of semantic structures of music. In *Proceedings of the 11th International Society for Music* (pp. 571–576).

Galleguillos, C., McFee, B., Belongie, S., & Lanckriet, G. R. G. (2011). From region similarity to category discovery. In*IEEE conference in computer vision and patter recognition* (*CVPR*) (pp. 2665–2672).

Gammerman, A., Vovk, V., & Vapnik, V. (1998). Learning by transduction. In G. Cooper & S. Moral (Eds.), *Uncertainty in artificial intelligence* (pp. 148–155). San Francisco, CA: Morgan Kaufmann.

Gentner, D., & Markman, A. (1997) Structure mapping in analogy and similarity. *American Psychologist, 52*(1), 45–56.

Hörnel, D. (2004). Chordnet: Learning and producing voice leading with neural networks and dynamic programming. *Journal of New Music Research, 33*(4), 387–397.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359–366. doi:10.1016/0893-6080(89)90020-8.

Jehan, T. (2005). *Creating music by listening*. Ph.D. thesis, Massachusetts Institute of Technology, MA, USA.

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller & J. W. Thatcher (Eds.), *Complexity of computer computations* (pp. 85–103). New York: Plenum Press.

Mahalanobis, P. C. (1936). On the generalised distance in statistics. In *Proceedings of the National Institute of Sciences of India 2* (pp. 49–55). MIT Press.

McFee, B., Barrington, L., & Lanckriet, G. (2010). Learning similarity from collaborative filters. In *Proceedings of the International Society for Music Information Retrieval Conference* (pp. 345–350).

McFee, B., & Lanckriet, G. (2009). Heterogeneous embedding for subjective artist similarity. In *Proceedings of the international symposium on music information retrieval* (*ISMIR*).

Mcfee, B., & Lanckriet, G. (2010). Metric learning to rank. In *Proceedings of the 27th annual International conference on machine learning* (*ICML*).

McFee, B., & Lanckriet, G. (2012). Hypergraph models of playlist dialects. In *13th International symposium for music information retrieval* (*ISMIR2012*)).

Musil, J., El-Nusairi, B., & Müllensiefen, D. (2012). Perceptual dimensions of short audio clips and corresponding timbre features. In *Proceedings of the 9th international symposium on computer music modelling and retrieval* (*CMMR 2012*).

Novello, A., Mckinney, M. F., & Kohlrausch, A. (2006). Perceptual evaluation of music similarity. In *Proceedings of the 7th international conference on music information retrieval* (*ISMIR*).

Page, K., Fields, B., De Roure, D., Crawford, T., & Downie, J. S. (2012). Reuse, remix, repeat: The workflows of mir. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*. Porto, Portugal.

Ricci, F. (2012). Context-aware music recommender systems: workshop keynote abstract. In *Proceedings of the 21st world wide web conference*, *WWW 2012* (pp. 865–866). Lyon.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the IEEE international conference on neural networks* (pp. 586–591). San Francisco, CA.

Schultz, M., & Joachims, T. (2003). Learning a distance metric from relative comparisons. In *Advances in neural information processing systems* (*NIPS*). MIT Press.

Serra, X. (2012). Data gathering for a culture specific approach in mir. In *Proceedings of the 21st World Wide Web Conference*, *WWW 2012*, Lyon, pp. 867–868.

Slaney, M., Weinberger, K. Q., & White, W. (2008). Learning a metric for music similarity. In J. P. Bello, E. Chew, D. Turnbull (eds.) *International Society for Music Information Retrieval* (*ISMIR*) *2008* (pp. 313–318).

Slaney, M., & White, W. (2007). Similarity based on rating data. In *Proceedings of the 2007 International Society for Music Information Retrieval* (*ISMIR*) (pp. 479–484).

Stober, S., & Nürnberger, A. (2010). Similarity adaptation in an exploratory retrieval scenario. In *Proceedings of 8th international workshop on adaptive multimedia retrieval* (*AMR'10*). Linz, Austria (To appear).

Stober, S., & Nürnberger, A. (2011). An experimental comparison of similarity adaptation approaches. In *Proceedings of 9th international workshop on adaptive multimedia retrieval* (*AMR*). Barcelona, Spain (To appear).

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the international conference on machine learning* (*ICML*).

Tversky, A. (1977). Features of similarity. *Psychological Review, 84*, 327–352.

Weinberger, K., & Saul, L. (2009). Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research, 10*, 207–244.

Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research, 10*, 207–244.

Wolff, D., Stober, S., Nürnberger, A., & Weyde, T. (2012). A systematic comparison of music similarity adaptation approaches. In *Proceedings of international symposium on music information retrieval* (*ISMIR*) (To appear).

Wolff, D., & Weyde, T. (2011a). Adapting metrics for music similarity using comparative judgements. In *Proceedings of international symposium on music information retrieval* (*ISMIR*).

Wolff, D., & Weyde, T. (2011b). Combining sources of description for approximating music similarity ratings. In *Proceedings of 9th international workshop on adaptive multimedia retrieval* (*AMR*). Barcelona, Spain.

Wolff, D., & Weyde, T. (2011c). On culture-dependent modelling of music similarity. In *Proceedings of fourth international conference of students of systematic musicology sysmus*. Cologne, Germany.

Wolff, D., & Weyde, T. (2012). Adapting similarity on the magnatagatune database: effects of model and feature choices. In *Proceedings of the 21st international conference companion on world wide web*, *WWW '12 Companion* (pp. 931–936). New York, NY, USA: ACM.

Yang, L. (2006). Distance metric learning: A comprehensive survey. Michigan State Universiy pp. 1–51.