**Noname manuscript No.**
(will be inserted by the editor)

# Learning Music Similarity from Relative User Ratings

**Daniel Wolff · Tillman Weyde**

Received: date / Accepted: date

**Abstract** Computational modelling of music similarity is an increasingly important task for personalisation and optimisation in Music Information Retrieval and for research in music perception and cognition. Relative similarity ratings provide a new and promising approach to this task as they avoid problems associated with absolute ratings. In this article, we use relative ratings from the MagnaTagATune dataset to develop a complete learning and evaluation process with state-of-the-art algorithms and provide the first comprehensive and rigorous evaluation of this approach. We compare different high and low level audio features, genre data, dimensionality effects, weighted similarity ratings, and different sampling methods. For model adaptation, we compare SVM-based metric learning, Metric-Learning-to-Rank (MLR), including a diagonal and a novel weighted MLR variant, and similarity learning with Neural Networks. Our results show that music similarity measures learnt on relative ratings are significantly better than a standard metric, depending on the choice of learning algorithm, feature set and application scenario. We implemented a testing framework in Matlab®, which we made publicly available[1] to ensure reproducibility of our results.[2]

**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
E-mail: Daniel.Wolff.1@city.ac.uk, t.e.weyde@city.ac.uk

[1] http://mi.soi.city.ac.uk/datasets/ir2012framework

[2] In this paper we extend our previous work on this topic [52,51,53,54]. Compared to the previous papers we provide new experimental results, involving additional algorithms, new analyses of the dataset, and extended and more thorough evaluation. We also present an extended extended rationale and discussion of the proposed approach, providing the first comprehensive study of similarity learning from relative ratings and the MagnaTagATune similarity dataset.

## 1 Introduction

Similarity plays a central role in music retrieval and recommendation as well as musicology. As means of storing recordings and scores of music digitally have become less expensive, increasing amounts data are available for algorithmic music analysis and comparison today. Many applications and a growing number of portable multimedia computing devices encourage the development of elaborate techniques to automatically analyse, classify, index, and retrieve music.

Most commercial approaches to music recommendation use collaborative filtering, the quasi-standard approach for online recommendation. The drawbacks of collaborative filtering are that it relies on user behaviour data, as has been pointed out by Celma [13], and that it fails when there is little data available, as for new or less popular music.

Content-based approaches avoid these issues by directly using 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 [24] or The Freesound Project [2]. Music comparison based on audio content needs to incorporate the extraction of acoustic, psychoacoustic and music theoretic features derived from audio information. The applicability of such features and distance measures is highly dependent on the context of the music, the application, and the user. Learning models can help ensure that the system is appropriate for the users' needs and the designers' intentions.

The users of music exploration and recommendation systems have often been neglected by assuming a general consensus on music similarity perception and retrieval criteria. Besides the disappointment of users who do not fit this assumption, fixed retrieval approaches can also impose a cultural influence, especially where the factors involved in the comparison are not transparent. On the other hand, user-adapted retrieval has the potential to provide personalised search results that are better suited to the user's needs than a standardised suggestion.

For computational musicology, personal ratings or usage data concerning music support the development of new, automatically adapted models of music perception and analysis of cultural characteristics in the use of music. Efforts have been increased recently to adapt retrieval methods to specific contexts and individual users, as in the CompMusic project [41] or the work of Ricci et al. [38]. Context-based and user-adapted retrieval have become popular research goals, following and fostering developments in machine learning to provide algorithms applicable to accumulated user data. Especially in social networks new opportunities are being explored using "games with a purpose" (GWAP), where data is collected while the subjects are playing a game. To optimise distance measures according to data has been tried, mostly using tags or class information, such as genre labels.

This study is part of a project on culture-dependent modelling of similarity of music audio clips. As part of this work, we evaluate modelling approaches for adapting similarity to user ratings based on audio content-based features and genre-tags. In this study we address the question whether and how machine learning can be used to learn optimised similarity measures based on ratings from the only publicly available dataset with audio music similarity data, the MagnaTagATune dataset.

We provide the first comprehensive and rigorous study of the dataset and state-of-the art methods for similarity modelling. The presented methods support the full process chain from feature design up to learning evaluation.

The relative nature of the similarity ratings supports the design of an easy game interface and avoids problems with the consistency in subjective numeric ratings. However, the relative approach complicates the learning of the similarity measure, where problems are the feature definition and the data structure for representing the relative ratings. For the learning itself we have evaluated 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, as well as a non-metric distance measure based on Neural Networks.

We provide an extensive evaluation of the models' performances using cross-validation to assess the training and generalisation success of each modelling and learning approach. This includes the influence of the feature sets representing the music in the model, as well as feature dimensionality. We have developed sampling methods corresponding to different application scenarios. Our experiments show that learning can have a significant positive effect on the performance of systems addressed in this study. The effect depends on the feature set, preprocessing, sampling method, and the learning algorithm which can all produce significant performance differences.

The remainder of this article is organised as follows: Section 2 reports on related work and Section 3 introduces our methods for this study. We present our experiments in Section 5 and discuss the results in Section 6. Section 7 closes this article with conclusions and perspectives for future work.

## 2 Related Work

Our context is Music Information Retrieval, where a standard architecture for adaptive systems as sketched in Figure 1 has become prevalent for information retrieval involving audio [12,35,9]. 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 like dancability. The audio features can be complemented with professional 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. Here, 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 the following general and music specific work on similarity models, methods for collecting similarity data, and computational methods to learn from the data.

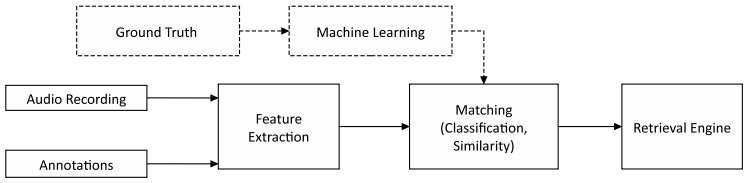



**Fig. 1** Schematic architecture of an adaptive Music Information Retrieval system.

## 2.1 Modelling Similarity

Most similarity models are based on features, as proposed by Tversky [47]. A common mathematical approach is nowadays to view the features as dimensions of a vector space and model dissimilarity as a distance measure, e.g. using the Euclidian or other metrics. Distance measures normally treat the dimensions uniformly, which ignores the different natures of features and their relations, e.g. the aspect of systematicity as pointed out by Gentner and Markman [20]. This can be addressed to some degree by using a Mahalanobis distance [26] (see Section 3.3), which models correlations between features.

Distances in vector spaces are normally symmetric, and metrics are symmetric by definition. However, Tversky [47] already pointed out that similarity perception may be asymmetric. In music perception, asymmetry can be expected, because two comparable clips are presented sequentially and order may play a role. Gentner and Markman [20] relate asymmetry to prototype-instance relationship of objects to compare. Yet, most mathematical and computational similarity models so far are symmetric. This is due to the simplification that symmetry brings to practical and theoretical aspects of the model. Considerations of the mode of data collection and the information available in the data also make a symmetric model a reasonable choice.

## 2.2 Adaptive Similarity Models

There is a considerable variety of computational approaches which can be considered for learning similarity measures. In most cases, the dual problem of a distance measure, which is inversely related to similarity, is addressed using supervised learning methods.

### *2.2.1 Feature Selection and Weighting*

The simplest form of a adapting a distance measure is by applying a feature selection. Feature selection is used in information retrieval for optimising efficiency by only considering relevant data features. E.g. Dash and Liu [14] provided a systematic approach for feature selection in generic classification tasks.

For music information retrieval, Pickens [37] categorised features for use with symbolic score data into “shallow-structure” and “deep-structure” features. In

his paper, most of the features suitable for automatic extraction belong to the shallow-structure group.

In distance learning, a set of features defines the dimensions of vector space, where a measure, normally a metric, is adapted to a set of training examples. Yang [55] has listed a considerable range of distance learning methods, including Linear Discriminant Analysis, nearest-neighbour-based optimisation, and kernelised approaches such as Support Vector Machines (SVM).

#### *2.2.2 Class-based Similarity Learning*

Class information is a standard part of many datasets, so it is interesting to use class information to adapt similarity ratings. The general assumption here is that distances within classes should be smaller than distances between classes. Weinberger et al. [49] present a method for Large Margin Nearest-Neighbour classification (LMNN), using semidefinite programming. A common evaluation method is to test if the $k$ nearest neighbours of a clip are in the same class as the clip. This optimisation maximises a large margin in the trained metric between points belonging to different classes.

Davis et al. [15] developed the Information Theoretic Metric Learning (ITML) algorithm, which optimises a fully parametrised Mahalanobis metric allowing for regularisation with respect to a predefined Mahalanobis metric. An online version of the algorithm is described as well. The results of their experiments with several standard classification datasets show a similar or slightly superior performance of ITML compared to a standard Mahalanobis metric (see Section 3.3), Maximally Collapsing Metric Learning (MCML)[21], and LMNN.

#### *2.2.3 Similarity Learning from User Ratings*

Class labels and data which have been used with the above algorithms are often not similarity-based. Furthermore, depending on the number of classes, class-based data contains relatively little information. Often, genre labels are used in such tasks, and some evidence for a correlation to similarity perception exists as discussed in Section 2.3.1. However, the general considerations of class labels still apply, and there is no openly available dataset containing music similarity class labels which have been assigned by humans.

As increasing amounts of non-class data sets are available from crowd-sourcing and other online resources, distance-learning algorithms using such data have become more popular. Using the hypertext structure of university homepages, Schultz and Joachims [40] presented a method to train a weighted Euclidean distance to relative distance constraints, which we call SVM-Light in the following. As we show in Section 3.4.4, this approach can be also applied to music similarity adaptation. The following section discusses algorithms that have been applied and the data available for music similarity learning.

## 2.3 Modelling Music Similarity from Data

A number of different methods have been used for collecting similarity data. In MIR research, the development of similarity models has been standing alongside the collection of datasets individually fit to the purpose of the task or training algorithm used. unklar The next section will discuss such work where similarity data from subjects has been used for training, stating three typical paradigms for collecting similarity data.

### *2.3.1 Expert and Survey Annotations*

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[3] or in Ferrer and Eerola [17]. Here, it is left to the subject ensure that their similarity statements over time are consistent.

A promising approach is the use of *relative similarity data* that describe the similarity constraints between pairs of clips, specifying one pair to be more similar than the other. A typical setup for collecting relative similarity data is given by the "odd-one-out" scheme. Here, usually three objects are presented to the participants, who are asked to choose the one which least fits into the triplet. This indicates a relatively higher similarity between the two remaining clips than to the selected one. Due to the simplicity of the user interface and the voting task, we decided to develop and evaluate the use of this kind of similarity data as described specifically in Section 3.

Allan et al. [3] discuss the challenges of gathering consistent 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. [34], it is usually infeasible for triplets of music clips in an odd-one-out constellation 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.

Ellis and Whitman [16] use data from a comparative survey on artist similarity to evaluate similarity metrics based on similar artist lists from the All Music Guide[4] to define their ERDÖS distance. Their artist similarity data covers 412 popular musicians, for whom they gathered 16385 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. Furthermore, regularisation does not improve results in most cases.

[3] http://www.music-ir.org/mirex/wiki/2011:Evalutron6000_Walkthrough
[4] http://www.allmusic.com/

McFee et al. [32] introduced a multiple-kernel learning technique for constructing an artist similarity measure given MFCC audio features, based on tags annotated by users and via an auto-tagging algorithm, biographical information, and collaborative filtering data. An artist similarity function is learnt as a weighted combination of several kernels, also using Ellis and Whitman's [16] already mentioned artist similarity data.

For gathering *class-based similarity data*, subjects are asked to classify clips by assigning them to one of a fixed number of unlabelled classes (e.g. [33]). This type of experiment typically requires choosing an appropriate number of classes beforehand. The similarity of clips within one class is then assumed to be higher than between different classes, which is of course only valid as a trend, but not in every instance. This approach has also been applied genre data, as they provide a classification and are widely available. E.g. Novello et al. [34] follow this assumption 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. Similarly, in Pampalk's [36] experiments with different acoustic features' appropriateness for similarity prediction, their performance is evaluated using a metric based on the nearest-neighbour genres as ground truth.

Bade et al. [4] use expert classifications of folk song melodies for training localised similarity measures on folk songs. Pairs of clips from the same and from different classes are used for learning a linear weighting of similarity measures for a folk song database containing symbolic music data and metadata.

### *2.3.2 Crowd-Sourcing Music Data*

Crowd-sourcing makes use of the large numbers of people that can be reached through the Internet. Based on users' playlists, liking data, music purchase history and tag annotations, substantial datasets can be collected and used for machine learning.

Barrington et al. [7] present a method for automatic tagging, based on their model of tag affinity using linear combinations of four SVM kernels relating to different feature similarity measurements. Apart from acoustic and web-mined features, they use crowd-sourced tag data from Last.fm to predict tags from a different dataset. For the individual tag classifiers, they also analyse the contribution of the different feature kernels to the final distance measure.

Bodganov et al., in [8] use preference sets of songs to adapt a content-based relevance measure for music recommendation. After a preliminary feature selection, they use support vector regression to locate songs in a *semantic descriptor space*. By weighting different distance measures within this space, the songs are then compared to retrieve a recommendation distance.

Instead of weighting feature kernels, McFee et al. [27] parametrise a music similarity metric using collaborative filtering data. The distance function allows for a parametrised linear combination of content-based features. Such user listening

and "liking" data has proved highly effective for providing relevant music recommendations. Unfortunately, the availability of the collaborative user data depends on the popularity of the music. Exploiting correlations of the audio feature data and the users listening behaviour, the adapted metric approximates the user data whilst only requiring acoustic feature information and available tag data on the music. They present the MLR algorithm (see Section 3.4.1), which performs the adaptation of a Mahalanobis metric to given ranking data. Post-training analysis of feature weights revealed that tags relating to genre or radio stations were assigned greater weights than those related to music theoretical terms. In our experiments in Section 5, we use MLR to adapt a music similarity metric to user ratings.

Slaney et al. [43] also presented a general method for learning a Mahalanobis distance metric. They adapt similarity on user "like" data covering jazz music from Yahoo! Music. Their experiments evaluate the effectiveness of the similarity metrics by comparing it to songs's nearest neighbours in terms of artist names. They find that the collaborative-filtering based measure outperforms a content-based metric. The authors discuss two drawbacks associated with using artist identity as similarity ground truth: Firstly they note the wide range of musical styles any artist may have. Secondly an imbalance of distribution of collaborative-filtering information in their data with respect to artists and albums is discussed. Users may listen to and "like" all songs of an artist because their playlist is artist-based. The same problem applies to musical genre or any other categories typically used to organise music.

Slaney and White [42] extend the variety of similarity models by comparing six approaches of adapting content-based similarity on the same ground truth (unmodified, whitening, LDA, NCA, LMNN and RCA). As above, the similarity data is derived from metadata classification, but the authors broaden their range of data by adding experiments with the kNN performance measure based on album and blog matches. Here, the content-based features are gathered using the The Echo Nest API.

The Million Song Dataset, containing audio and tag features for 1,000,000 songs, has recently been enhanced by a set of collaborative filtering data for music relevance. This dataset is now used in the Million Song Dataset challenge run by McFee and Bertin-Mahieux [28], a competition for the best prediction of user listening history data given a public training set. McFee and Lanchriet [31] recently also presented a new hypergraph model for playlist generation. This scenario is related to music similarity estimation, as songs played in close succession are often found similar. But, like with collaborative filtering data, other factors also influence the data and model design. The experiments in [31] show that modelling genre sequences also plays a significant role for designing a playlist generator.

#### *2.3.3 Gathering Data via Musical Games With A Purpose*

Games With A Purpose (GWAP) are intended to gather data, often similar to traditional questionnaires, in large quantities from online users. In GWAPs the

users are motivated to participate by an enjoyable game experience and the incentive to provide accurate data lies in rewards for agreement across users. Ellis et al [16] use a game to complement their survey data described in the previous section. The game results somewhat supports the survey, but different sampling methods prohibit a direct comparison. More recently, HerdIt, a GWAP based on the social network Facebook® was presented by Barrington et al. [5]. The tagging data collected with HerdIt was evaluated in [6] but no user data from the game has been published yet.

The TagATune game collects tagging and similarity data for a large number of song excerpts from the Magnatune online label.[5] The resulting data, was published as the MagnaTagATune dataset, which we use in part in this article. In [52,51], we used the MagnaTagATune dataset to adapt similarity measures based on the similarity data and music features contained in this set. For a simplified version of the similarity data, our experiments showed that the similarity data acquired via the human computation game can be modelled to some extent using SVM-based approaches for metric learning. Stober and Nürnberger [44] have worked on the same dataset but with different feature and similarity extraction methods, comparing algorithms for linear and quadratic optimisation of a similarity measure based on feature weighting. They analyse the training methods on two different subsets of the similarity constraints (see Section 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 other, slightly larger set, where not all constraints can be learned, the LIBLINEAR method achieves better results than the other methods. However, in that study only the learning performance is tested, not the generalisation, which is more relevant for most application scenarios.

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 and triangle inequality, made clear that care is necessary when interpreting the results of learning similarity from data, as they depend on the information in the data 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

The last section already mentioned the MagnaTagATune dataset, which we use in this study. It consists in part of data from an odd-one-out game, and in the following we describe data structures and algorithms for using this data to optimise similarity measures.

[5] See http://www.tagatune.org/.

### 3.1 Relative Ratings from Odd-One-Out Games

This section describes how data from an odd-one-out setting can be used and pre-processed to train music similarity models. We consider 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:

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

### 3.2 Similarity Graphs

Relative similarity relations can be represented as edges in a directed weighted graph of pairs of clips (McFee et al. [29], Stober et al. [45]): 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

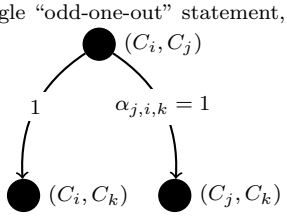
$$V = \{(C_i, C_j) \mid i, j \in I\}$$

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 Equation 2 is shown in Figure 2.

**Fig. 2** Graph induced by a single "odd-one-out" statement, as in Equation 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 a contradiction when considering the transitivity of the induced similarity metric. Contradictions appear as cycles in the graph as shown in Figures 3 and 4. Such cycles can be found and analysed using standard methods for extracting strongly connected components in directed graphs.

**Fig. 3** Graph containing a length-2 cycle

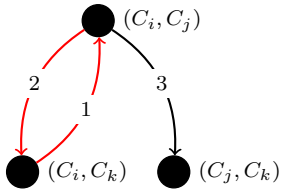
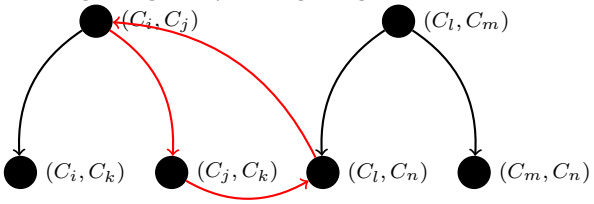



**Fig. 4** Graph containing a length-3 cycle. Edge weights have been hidden.




*3.2.1 Removing Cycles*

The SVM and MLR training algorithms we use here require the similarity data to be consistent. For removing direct contradictions 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 contradicting edges have equal weight, both are deleted, possibly leaving a vertex disconnected from the graph.

Removing cycles of greater length and finding the maximal acyclic subgraph of $G$ is an NP-hard problem [25]. McFee et al. [29] use a randomised algorithm by Aho et al. [1] 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 Section 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 Section 4.1).

## 3.3 Mahalanobis Distance

The MLR algorithm, which we introduce in the next section, adapts a metric that was introduced by Mahalanobis in 1936 [26]. 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 use 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 [48].

As described by Davis et al. [15], 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 [26] of the Mahalanobis distance, $W^{-1}$ represents the covariance of the distribution, $\beta$ represents a normalising factor and $\mu$ 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 for learning a Mahalanobis distance from relative similarity data. (D)MLR and SVM-Light are applicable to a multitude of data sources, with relatively little pre-processing and conversion required. They are both based on Support Vector Machines, and thus work effectively with high-dimensional feature vectors that are commonly used for describing the music clips (see Section 4.2). Implementations of both 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.

Using these algorithms, a parametrised Mahalanobis distance is learnt 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 Section 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 different algorithms used for optimising $W$.

### *3.4.1 Metric Learning to Rank (MLR)*

McFee and Lanckriet[30] describe the MLR algorithm for learning a fully parametrised Mahalanobis distance based on the SVM$^{\text{struct}}$ framework of Tsochantaridis et al. [46]. Specifically well-suited for employment 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 is supposed to 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. In our application 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[6].

#### *3.4.2 DMLR*

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

For MLR, to our knowledge, no experiments or methods for weighted training 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. During slack aggregation, performed by averaging error along the training constraints, the repeated constraints gain their respective weight. This approach is obviously not efficient, but for the MagnaTagATune similarity dataset it is feasible. Experiments with quantised constraint weights showed similar performance with using only fractions (10%) of data overhead, which improves the scalability to larger datasets. The effects of weighted learning with MLR and DMLR are explored in Section 5.4.

#### *3.4.4 Metric Learning with SVM-Light*

In [40], Schultz and Joachims present a metric learning strategy based on their SVM-Light framework[7]. Here, the matrix $W$, as introduced in Equation 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.

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

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

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: $\Delta^{x_i,x_j} = \left((x_{i_1} - x_{j_1})^2, ..., (x_{i_N} - x_{j_N})^2\right)$. 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:

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

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 $\|W\|_F^2 = tr(W^T \cdot W)$. 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

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

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 et al. [45]. 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 Equation 6.

### 3.5 Distance Learning using Neural Networks

Unlike the previous models, Neural Networks, specifically Multi Layer Perceptrons (MLP), are capable of approximating arbitrary functions (cf. Hornik et al.[23]). 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 Section 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 [22], based on earlier work by Braun et al. [11], for making a neural network learn an absolute

rating from relative information. This strategy is based on a combined network sketched in Figure 5 with two MLP networks, $net1$ and $net2$, 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, $net1$ gets the vector of the most similar pair, and should thus output a higher distance value than $net2$. The outputs of $net1$ and $net2$ are connected to a comparator neuron $c$ with negative fixed weight $-/+v$ for $net1/net2$ 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.

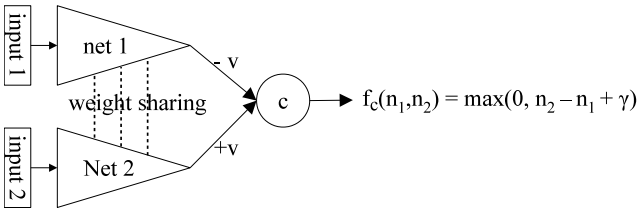



**Fig. 5** Scheme for neural network learning from relative ratings.

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 [10] is the use of a semi-linear activation function $f_c$ for the comparator neuron, which we use as indicated in Figure 5. We also introduce a margin between the higher and the lower ratings with a variable $\gamma$.

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 $net1$ and $net2$ for input $p$ are

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

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

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

We used this to implement training on a single network with $\gamma = 0.5$ with resilient backpropagation (cf. Riedmiller and Braun [39]) with regularisation. The procedure is described in listing 1.

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:

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

**Algorithm 1** Training of an MLP with relative constraints

**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** $T := \left\{(t_{i,j}, t_{i,k}) \mid \exists (i,j,k) \in Q^*\right\}$ ▷ training targets
  MLP = 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
  cycles = 0
  **while** 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 $\gamma$
      $t_{i,k} = \bar{d}_{i,j,k} + \gamma$ ▷ add margin for less similar pair
      MLP = trainRp(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**

## 4 The MagnaTagATune Dataset

As mentioned in Section 2.3.1, existing datasets for similarity statements of users are small and rarely accessible. The MagnaTagATune dataset is to our knowledge the only similarity dataset that is freely available[8] with the corresponding music data. Our experiments are therefore 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, both of the users' votings are saved into a histogram for this triplet. The MagnaTagATune dataset contains 7650 such votings for a total 346 of triplets, referring to 1019 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 listening. On average, each instance of a triplet permutation counts 14 votings. 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 Equation 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, thereby removing 8402 weight points, resolves all cycles existing in the initial graph. The resulting directed acyclic weighted graph consists of 337 connected subgraphs $G^i_{\text{sub}}$, each containing three vertices or clip pairs. The 6898 weight points for 860 unique connections contain the remaining similarity information $Q$. 27 vertices are isolated by the above process, indicating equal vote

[8] http://www.tagatune.org/Magnatagatune.html

counts for contradictory statements including 27 songs. 26 of those songs are not referenced by any remaining similarity constraints, thus reducing the number of referenced clips to 993.

A retrieval of the connected components in the graph shows the largest connected subgraph to be containing 3 vertices. In fact, when excluding the 27 isolated vertices with no associated similarity information, the remaining triplets correspond to 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 Section 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. To give an impression of the dataset's structure, we divided the genre groups using the most frequently annotated genres:

**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 |

#### *4.1.2 Similarity Weights*

For the MagnaTagATune dataset, the numbers of votes (see Section 4.1) per constraint varies. Since the weights of the edges are determined as the differences of conflicting votes as in Stober et al.[45], there is a compensation between total vote number and vote proportion: constraints with a small proportional 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 could be interesting, e.g. in a probabilistic model, but is not explored in this study.

#### *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.

Since our training data consist of three 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 Section 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 $Q^k_{\text{test}}$ of 86 constraints, while the remaining 9 subsets are combined to the training set $Q^k_{\text{train}}$ 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.[19]). As our results in Section 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, another method is needed. In Wolff et al. [50] we introduced and tested a sampling method, which separates similarity data the clip pair level. Rather than defining the subsets on the basis of constraints $(i, j, j) \in Q$, we use the disjoint subgraphs $G^i_{\text{sub}}$ of the full similarity graph $G$ (see Section 3.2). 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 $G$). We call this method *inductive sampling* (ID-sampling). In the MagnaTagATune dataset, after removing contradicting edges, the subgraphs are also disjoint in terms of clips.

The $G^i_{\text{sub}}$ 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 Section 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: 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 our initial experiments in [52,51], 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 [24]). 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 Section 5.2.

*Aggregation by Averaging* In most of our experiments, we aggregate this information to the 30 seconds time scale of a clip. Like in [44], 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 Section 5.2 we only evaluate features based on the means of chroma and timbre values, i.e. for each clip $C_i,\ i \in \{1, \cdots 1019\}$, 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* In the previous experiments [52,51,54], we 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 $C_i,\ i \in \{1, \cdots 1019\}$. 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 [52, 51] we restricted the set of features to the easily extractable low-level features mentioned above. Slaney et al. [42] 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 PhD thesis [24].

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. [42], as far as the required information is available in the MagnaTagATune dataset. Features where this was not the case have been omitted to ensure reproducibility of the experiments. Table 2 shows a list of the features used in this study.

**Table 2** Features from [42] used in our experiments.

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

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 [42], as we are interested in keeping

the features' original associations to properties in music theory. Note that some of the features allocate only a small number of actual values. For example, the *timeSignature* feature uses only the values $\{\frac{0}{7}, \frac{1}{7}, \cdots, \frac{7}{7}\}$.

### 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[9]. It contains descriptions of the songs present the MagnaTagATune dataset's clips: Each song is annotated with 2 to 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. This includes training on All performances are evaluated with cross-validation based on the percentage of unique distance constraints being satisfied by the learnt distance function. The distance constraints used below are extracted as described in Section 4.1.3. Following the strategy from [50], we start from a set of 13 constraints on average and increase the training set size $|Q^v_{\text{train}(p)}|$ 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 $4 \cdot 10$ 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 Section 4.2. Section 5.4 explores the use of weight information in the similarity graph. Finally, Section 5.3 compares the ID-sampling, which was used in all other experiments, to TD-sampling.

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

### 5.1 Algorithms Compared

We compare MLR, DMLR, SVM-Light and MLP neural network. 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 (Section 3.4.1), and $c_{SC03} = 3$ for the SVM-Light algorithm (Section 3.4.4). The MLP 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.

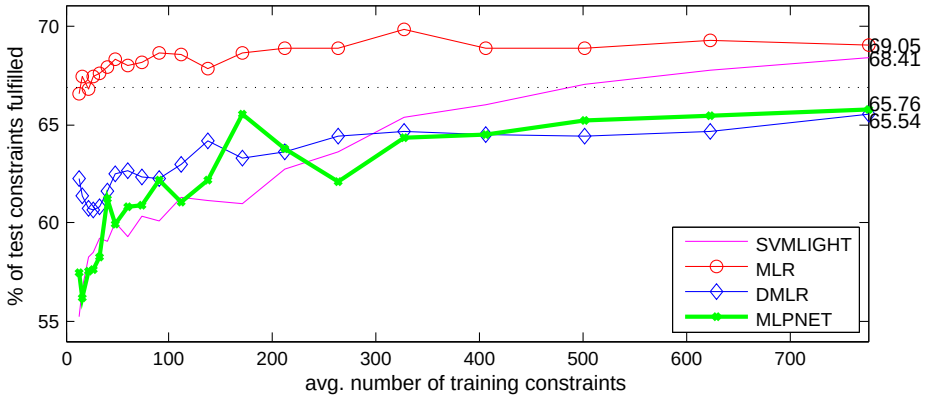



**Fig. 6** Overall test set performance for combined features with averaged low-level information: SVM, MLR, DMLR and MLP performance for full features, with increasing training set size. The dotted line shows the baseline performance of an unweighted Euclidean distance.

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 MLP do not generalise well from the training set onto the test set (see Figure 7) .

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 the MLP network remain below the baseline performance by 1% on the test sets.

The generalisation results for small training sets $Q^v_{\text{train}(p)}$ depend highly on the algorithm used, and for SVM-Light, DMLR and MLP lie considerably below the

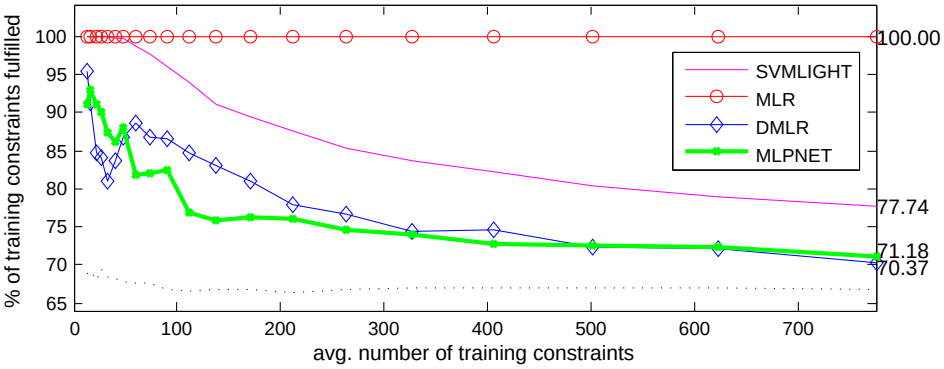



**Fig. 7** Overall training performance: SVM, MLR, DMLR and MLP performance for full features, with increasing training set size. The dotted line shows the baseline performance of an unweighted Euclidean distance on the training set.

baseline. For SVM-Light, this is an effect of overfitting on small datasets, as we optimised the parameters for larger training sets. In [50] 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. While 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 Figure 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 [52, 54], 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 and efficiency*

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 the MLP net run within the MATLAB interpreter. Especially for the large feature spaces used with MLR and SVM-Light, the MLP method (see 1) is still by far the slowest of the approaches described in this paper, using large amounts of time even for the small training sets.

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

| SVM | MLR | DMLR | MLP |
|---|---|---|---|
| 5 | 40 | 30 | 60 |

## 5.2 Influence of Feature Type

As has been shown in [52] 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
  - single timbre via average or 4 cluster centroids
- 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 Section 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, the baseline values are plotted on the left vertical axis in Figure 8 and 9.

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 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,

**Table 4** SVM Single features test set performance. Values for single average audio features and 4-cluster audio features are separated by slashes (average / 4-cluster).

| 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 |

**Table 5** Significance of performance differences between feature types (Wilcoxon signed rank $p$ values). Significant values at 5% level are set in bold type.

| 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 / – |

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 [54]).

### *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 task simpler and more tractable. 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 is 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 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 Figure 8 and in Figure 9.

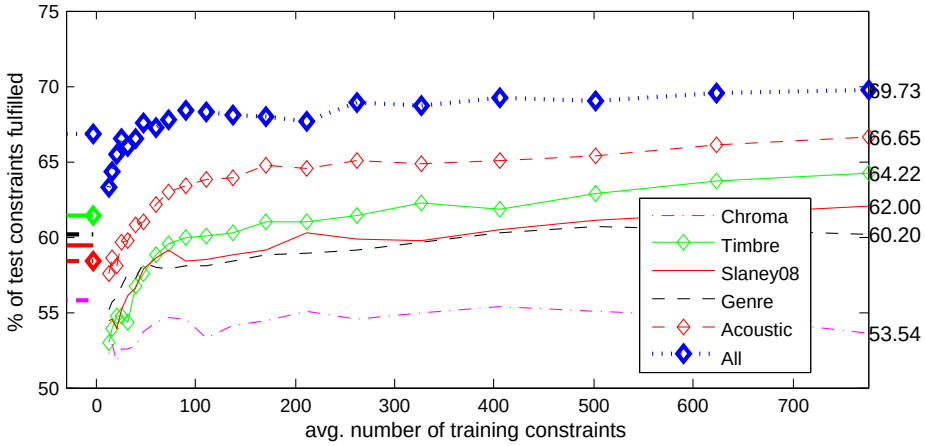



**Fig. 8** SVM performance with 12-dimensional features: chroma (mean), timbre (mean), Slaney08, genre, combined features with increasing training set size.

Figure 8 shows that learning on the PCA12 chroma features has very little effect. 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 Figure 6).

All pairwise differences in test performance between feature types are significant at $p < 5\%$, except timbre vs. 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.

PCA52 features are compared in Figure 9. The results 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

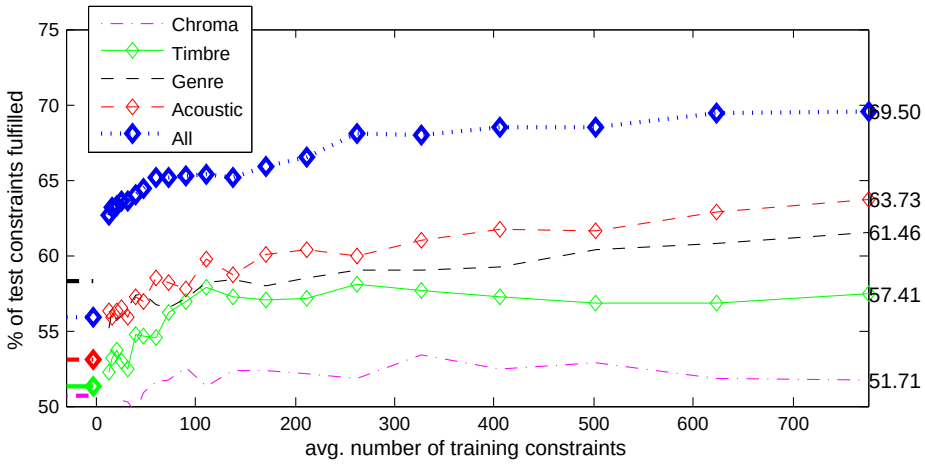



**Fig. 9** SVM performance with 52-dimensional features: chroma (4 clusters), timbre (4 clusters), combined audio, genre, combined features with increasing training set size.

the PCA12 features. Similar to the 12-dimenstional case, all pairwise differences are significant except timbre vs. genre.

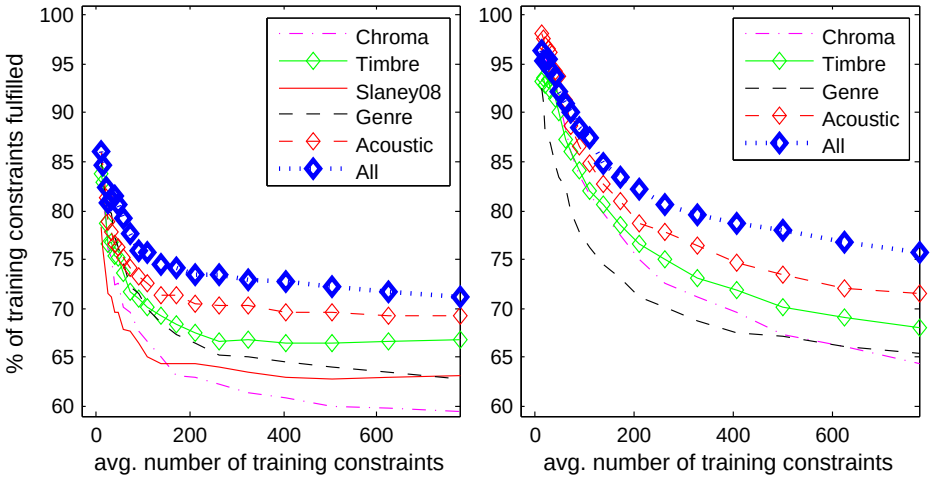



**Fig. 10** SVM feature training performance at 12(l) and 52 (r) dimensions: Increasing training set size.

The training performance, as depicted in Figure 10, indicates that the bad generalisation of 52-dimensional features is a result of overfitting: The training performance of 52-dimensional PCA features, also presented in Table 6, 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 is thus far greater (by factor 2-3) than for the

12-dimensional features. This indicates increased learning capacity of the model based on the 52-dimensional data. The generalisation does not improve, however, indicating that quantity or quality of the MagnaTagATune similarity data is not sufficient to support generalisation with more flexible models.

For both PCA12 and PCA52, the combined features achieve a very similar performance to the raw features in Table 4. It has been suggested that results, especially generalisation for SVM-Light can be increased using appropriate dimension reduction. However, the generalisation performance between PCA12, PCS52 and unreduced all-combined features on the maximal training set is not significantly different. With increasing dimensionality, maximal performance needs more data. 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. On the other hand, 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.

**Table 6** SVM Single features training set performance. The Slaney08 features are not available to 52-dimensional PCA features.

| Features | Chroma | Timbre | Slaney08 | Genre | Audio Comb. | Combined |
|---|---|---|---|---|---|---|
| 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 | 3.61 | 5.35 | 3.61 | 2.65 | 10.94 | 4.32 |
| Training52 | 64.41 | 68.03 | / | 65.43 | 71.50 | 75.78 |
| Baseline52 | 50.70 | 51.28 | / | 58.26 | 53.02 | 55.93 |
| Gain52 | 13.71 | 16.75 | / | 07.18 | 18.48 | 19.85 |

### 5.3 Sampling: Effects of Transductive Learning

As detailed in Section 4.1.3, sampling for cross-validation can be realised as ID-sampling, like in the experiments so far, or as TD-sampling, where pairs and individual clips (but not constraints) can appear in both training and test set. Figure 11 shows the results for the SVM-Light, MLR and DMLR algorithms. The baseline shows the performance of an unweighted Euclidean distance measure for the test sets. 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 11, shows a considerable increase of performance (6%) for MLR and a slight increase for SVM-Light. This reproduces the findings of Wolff et al. [50]. 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 Section 3.4.1). For

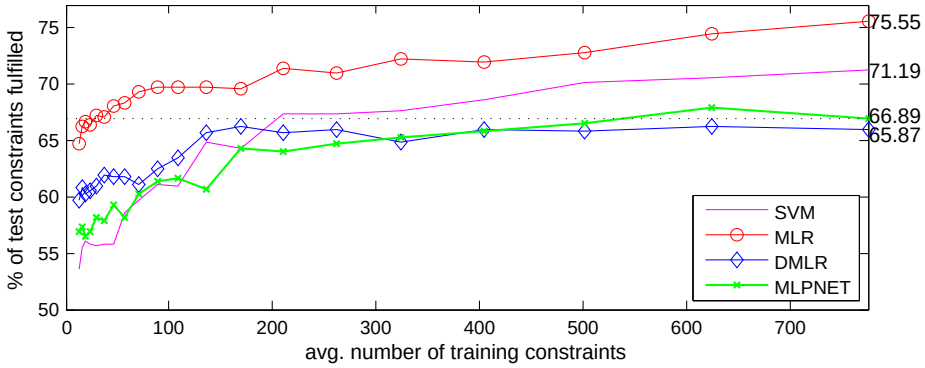



**Fig. 11** Transductive sampling: SVM, MLR, DMLR and MLP test set performance for full features. The training set size increases from left to right.

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 Section 4.1.3) due to the TD-sampling referencing more feature vectors during training.

## 5.4 Weighting Constraints by Vote Differences

As described in Section 4.1, the 860 unique similarity constraints represent differences of 6898 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 Section 3.4.3 and weighted SVM-Light (see Section 3.4.4).

In the figures below, two methods of evaluation are used:

- (E:UW) refers to the unweighted evaluation considering the unique constraints satisfied, as used above.
- (E:W) measures the *weighted performance* of a metric as sum of the weights $\sum \alpha_{i,j,k}$ 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 12 shows the weighted performances (E:W) on the training sets of weighted training (T:W) with MLR, DMLR, and SVM-Light. We compare these to weighted performance (E:W) of the unweighted training and an Euclidean metric. For the Euclidian metric, the weighted evaluation (E:W) is yields about 6% better performance than (E:UW), indicating a correlation of the weighted constraints with the Euclidean distance in feature space. For MLR, 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.

Figure 13 shows the weighted (E:W) and unweighted (E:UW) test results of weighted training (T:W) for MLR, DMLR, and SVM-Light. At the top of the

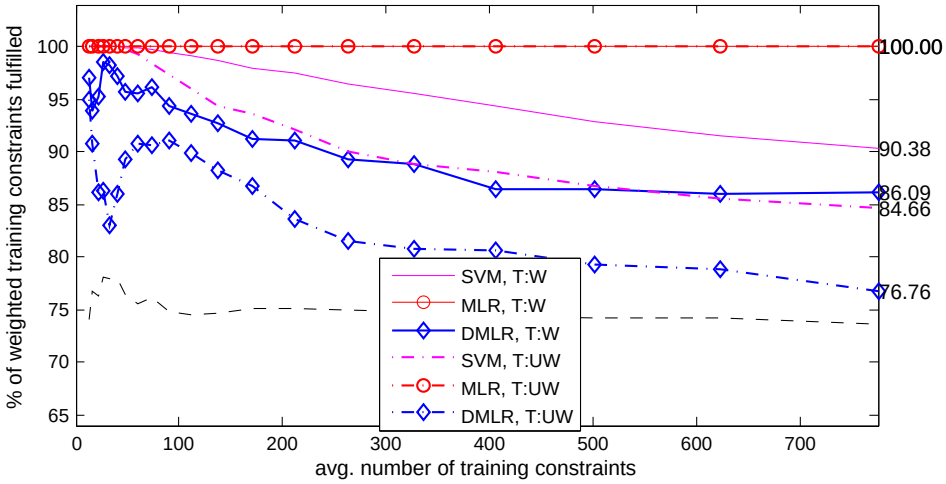



**Fig. 12** Overall training performance, weighted evaluation (E:W) for weighted (T:W,–) and unweighted (T:UW,· – ·) training: SVM, MLR, DMLR. The bottom dashed curve displays the weighted baseline performance.

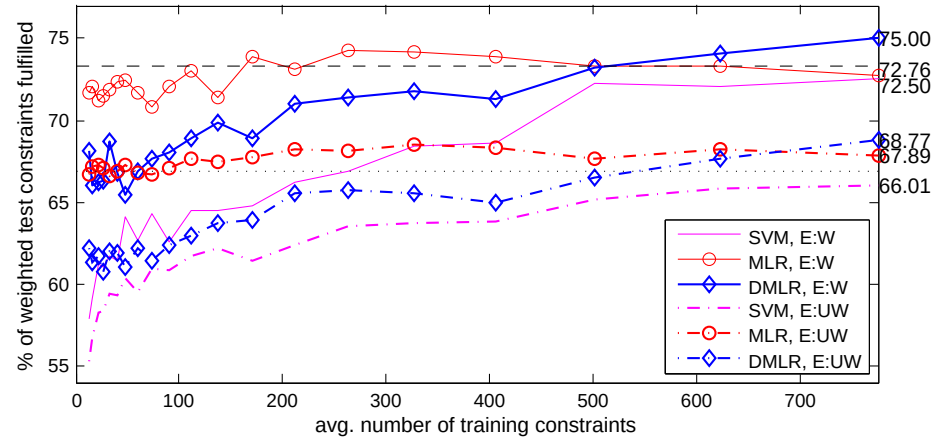



**Fig. 13** Overall weighted(E:W, –) / unweighted (E:UW, · – ·) generalisation performance for weighted training: SVM, MLR, DMLR. The dotted and dashed horizontal lines shows the unweighted and weighted baseline, respectively.

figure, the three continuous lines represent the weighted test-set performance of the algorithms. Here, only DMLR exceeds the baseline performance for weighted evaluation (E:W), 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 DMLR allows more effective learning on this dataset.

The unweighted performance (E:UW) of the models learnt from weighted constraints (T:W) is plotted in the lower part of Figure 13 as dotted (· – ·) lines. Results for MLR and DMLR are slightly lower compared to those in Figure 6

obtained with unweighted training, but the still signnificantly better than the baseline.

We also compared the generalisation results of unweighted (T:UW) and weighted (T:W) training using weighted evaluation (E:W). Interestingly, MLR reaches 75.5% and SVM-Light 74.2% performance in the (T:UW)(E:W) case, slightly exceeding the (T:W)(E:W) case (see Figure 13), and the difference is significant only for SVM-Light ($p = 0.0248$). Conversely for DMLR the weighted training performs slightly but significantly better ($p = 0.0042$).

Overall, the weighted training is effective on the training data but on test sets only DMLR can reach significant improvement above the baseline. The raised performance of the Euclidean baseline shows that the features chosen for our tests correspond well to the weightings. However, as the distribution of weights depends on both the number of votes and the ratio of conflicting vote (see Section 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. However, the size of the improvement is small and the accuracy on test constraints remains below 70%. 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.

The results are in a similar range as in earlier studies by ourselves [54, 52, 51], by Stober and Nürnberger [44] and a joint study [54]. The method of Stober et al. differs from ours in that it applies early fusion or feature data into intermediate similarity measures and then applies learning of a linear combinations of those, with the SVM-Light method. This approach can support better user understanding and interaction, but it yields no improvement of the learning result.

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, but SVM-Light 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. DMLR is the most effective when using weighted training, but performs much worse than MLR and SVM-Light in all other tasks.

The experiments with Multi Layer Perceptrons (MLP) 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 parameterisations 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 (Section 5.2.1) has no significant effect on the generalisation with either the 12- or the 52-dimensional feature sets, although the training results differ considerably. These results show that the SVM-Light algorithm is robust and extracts relevant information from input data in high and low dimensions.

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. [42] 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 Section 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 second 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 Section 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 user 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 [30, 18, 40] 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 Section 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

1. Aho, A.V., Garey, M.R., Ullman, J.D.: The Transitive Reduction of a Directed Graph. SIAM Journal on Computing **1**(2), 131–137 (1972)
2. Akkermans, V., Font, F., Funollet, J., De Jong, B., Roma, G., Togias, S., Serra, X.: 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 (2011)
3. Allan, H., Müllensiefen, D., Wiggins, G.: Methodological considerations in studies of musical similarity. In: 8th International Conference on Music Information Retrieval, pp. 473–478 (2007)
4. Bade, K., Garbers, J., Stober, S., Wiering, F., N"urnberger, A.: Supporting folk-song research by automatic metric learning and ranking. In: Proceedings of the 10th International Conference on Music Information Retrieval, ISMIR, pp. 741–746. Kobe, Japan (2009)
5. Barrington, L., O'Malley, D., Turnbull, D., Lanckriet, G.: User-centered design of a social game to tag music. In: Proceedings of the ACM SIGKDD Workshop on Human Computation, HCOMP '09, pp. 7–10. ACM, New York, NY, USA (2009). DOI http://doi.acm.org/10.1145/1600150.1600152
6. Barrington, L., Turnbull, D., Lanckriet, G.: Game-powered machine learning. Proceedings of the National Academy of Sciences **109**(17), 6411–6416 (2012)
7. Barrington, L., Yazdani, M., Turnbull, D., Lanckriet, G.: Combining feature kernels for semantic music retrieval. In: Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR), pp. 614–619 (2008)
8. Bogdanov, D., Serrà, J., Wack, N., Herrera, P.: 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) (2009)
9. Bosma, M., Veltkamp, R.C., Wiering, F.: Muugle: A modular music information retrieval framework. In: In International Symposium on Music Information Retrieval (2006)
10. Braun, H.: Neuronale Netze - Optimierung durch Lernen und Evolution. Springer (1997)
11. Braun, H., Feulner, J., Ullrich, V.: Learning strategies for solving the planning problem using backpropagation. In: Proceedings of NEURO-Nimes 91, 4th International Conference on Neural Networks and their Applications (1991)
12. Casey, M., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., Slaney, M.: Content-based music information retrieval: Current directions and future challenges. Proceedings of the IEEE **96**(4), 668 –696 (2008)
13. Celma, O.: Music recommendation and discovery in the long tail. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona (2008)
14. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis **1**(1-4), 131–156 (1997). DOI http://dx.doi.org/10.1016/S1088-467X(97)00008-5
15. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th international conference on Machine learning, ICML '07, pp. 209–216. ACM, New York, NY, USA (2007)
16. Ellis, D.P.W., Whitman, B.: The quest for ground truth in musical artist similarity. In: Proc. International Symposium on Music Information Retrieval (ISMIR), pp. 170–177 (2002)
17. Ferrer, R., Eerola, T.: Timbral qualities of semantic structures of music. In: Proceedings of the 11th International Society for Music, pp. 571–576 (2010)
18. Galleguillos, C., McFee, B., Belongie, S., Lanckriet, G.R.G.: From region similarity to category discovery. In: IEEE Conference in Computer Vision and Patter Recognition (CVPR), pp. 2665–2672 (2011)

19. Gammerman, A., Vovk, V., Vapnik, V.: Learning by transduction. In: In Uncertainty in Artificial Intelligence, pp. 148–155. Morgan Kaufmann (1998)
20. Gentner, D., Markman, A.: Structure mapping in analogy and similarity. American Psychologist **52**(1), 45 –56 (1997)
21. Globerson, A., Roweis, S.T.: Metric learning by collapsing classes. In: Advances in Neural Information Processing Systems 18 (2005)
22. Hörnel, D.: Chordnet: Learning and producing voice leading with neural networks and dynamic programming. Journal of New Music Research **33**(4), 387?397 (2004)
23. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2**(5), 359 – 366 (1989). DOI 10.1016/0893-6080(89)90020-8
24. Jehan, T.: Creating music by listening. Ph.D. thesis, Massachusetts Institute of Technology, MA, USA (2005)
25. Karp, R.M.: Reducibility Among Combinatorial Problems. In: R.E. Miller, J.W. Thatcher (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press (1972)
26. Mahalanobis, P.C.: On the generalised distance in statistics. In: Proceedings of the National Institute of Sciences of India 2, p. 4955. MIT Press (1936)
27. McFee, B., Barrington, L., Lanckriet, G.: Learning similarity from collaborative filters. In: Proceedings of the International Society for Music Information Retrieval Conference, pp. 345–350 (2010)
28. McFee, B., Bertin-Mahieux, T., Ellis, D., Lanckriet, G.: The million song dataset challenge. In: Proc. of the 4th International Workshop on Advances in Music Information Research (AdMIRe '12) (2012)
29. McFee, B., Lanckriet, G.: Heterogeneous embedding for subjective artist similarity. In: Proc. International Symposium on Music Information Retrieval (ISMIR) (2009)
30. Mcfee, B., Lanckriet, G.: Metric learning to rank. In: Proceedings of the 27th annual International Conference on Machine Learning (ICML) (2010)
31. McFee, B., Lanckriet, G.: Hypergraph models of playlist dialects. In: 13th International Symposium for Music Information Retrieval (ISMIR2012)) (2012)
32. McFee, B., Lanckriet, G.R.G.: Learning multi-modal similarity. Journal of Machine Learning Research **12**, 491–523 (2011)
33. Musil, J., El-Nusairi, B., Mllensiefen, D.: 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) (2012)
34. Novello, A., Mckinney, M.F., Kohlrausch, A.: Perceptual evaluation of music similarity. In: Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR) (2006)
35. Page, K., Fields, B., De Roure, D., Crawford, T., Downie, J.S.: Reuse, remix, repeat: the workflows of mir. In: Proceedings of the 13th International Society for Music Information Retrieval Conference. Porto, Portugal (2012)
36. Pampalk, E.: Computational models of music similarity and their application in music information retrieval. Ph.D. thesis, Vienna University of Technology, Vienna, Austria (2006)
37. Pickens, J.: A survey of feature selection techniques for music information retrieval. In: Proceedings of the 2nd International Symposium on Music Information Retrieval (ISMIR) (2001)
38. Ricci, F.: Context-aware music recommender systems: workshop keynote abstract. In: Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, pp. 865–866 (2012)
39. Riedmiller, M., Braun, H.: 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 (1993)
40. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. In: Advances in Neural Information Processing Systems (NIPS). MIT Press (2003)
41. Serra, X.: Data gathering for a culture specific approach in mir. In: Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, pp. 867–868 (2012)
42. Slaney, M., Weinberger, K.Q., White, W.: 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 (2008)
43. Slaney, M., White, W.: Similarity based on rating data. In: Proceedings of the 2007 International Society for Music Information Retrieval (ISMIR), pp. 479–484 (2007)

44. Stober, S., Nürnberger, A.: Similarity adaptation in an exploratory retrieval scenario. In: Proceedings of 8th International Workshop on Adaptive Multimedia Retrieval (AMR'10). Linz, Austria (2010). To appear
45. Stober, S., Nürnberger, A.: An experimental comparison of similarity adaptation approaches. In: Proc. of 9th International Workshop on Adaptive Multimedia Retrieval (AMR). Barcelona, Spain (2011). To appear
46. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proceedings of the International Conference on Machine Learning (ICML) (2004)
47. Tversky, A.: Features of similarity. Psychological Review **84**, 327–352 (1977)
48. Weinberger, K., Saul, L.: Distance metric learning for large margin nearest neighbor classification. The Journal of Machine Learning Research **10**, 207–244 (2009)
49. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**, 207–244 (2009)
50. Wolff, D., Stober, S., Nürnberger, A., Weyde, T.: A systematic comparison of music similarity adaptation approaches. In: Proc. International Symposium on Music Information Retrieval (ISMIR) (2012). To appear
51. Wolff, D., Weyde, T.: Adapting metrics for music similarity using comparative judgements. In: Proc. International Symposium on Music Information Retrieval (ISMIR) (2011)
52. Wolff, D., Weyde, T.: Combining sources of description for approximating music similarity ratings. In: Proc. of 9th International Workshop on Adaptive Multimedia Retrieval (AMR). Barcelona, Spain (2011)
53. Wolff, D., Weyde, T.: On culture-dependent modelling of music similarity. In: Proc. of Fourth International Conference of students of Systematic Musicology Sysmus. Cologne, Germany (2011)
54. Wolff, D., Weyde, T.: 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. ACM, New York, NY, USA (2012)
55. Yang, L.: Distance metric learning: A comprehensive survey. Michigan State Universiy pp. 1–51 (2006)