A SYSTEMATIC COMPARISON OF MUSIC SIMILARITY ADAPTATION APPROACHES

Daniel Wolff*, **Tillman Weyde**
MIRG, School of Informatics
City University London, UK
daniel.wolff.1@city.ac.uk

Sebastian Stober*, **Andreas Nürnberger**
Data & Knowledge Engineering Group
Otto-von-Guericke-Universität Magdeburg, DE
stober@ovgu.de

ABSTRACT

In order to support individual user perspectives and different retrieval tasks, music similarity can no longer be considered as a static element of Music Information Retrieval (MIR) systems. Various approaches have been proposed recently that allow dynamic adaptation of music similarity measures. This paper provides a systematic comparison of algorithms for metric learning and higher-level facet distance weighting on the *MagnaTagATune* dataset. A cross-validation variant taking into account clip availability is presented. Applied on user generated similarity data, its effect on adaptation performance is analyzed. Special attention is paid to the amount of training data necessary for making similarity predictions on unknown data, the number of model parameters and the amount of information available about the music itself.

1. INTRODUCTION

Musical similarity is a central issue in MIR and the key to many applications. In the classical retrieval scenario, similarity is used as an estimate for relevance to rank a list of songs or melodies. Further applications comprise the sorting and organization of music collections by grouping similar music clips or generating maps for a collection overview. Finally, music recommender systems that follow the popular "find me more like…"-idea often employ a similarity-based strategy as well. However, music similarity is not a simple concept. In fact there exist various frameworks within musicology, psychology, and cognitive science. For a comparison of music clips, many interrelated features and facets can be considered. Their individual importance and how they should be combined depend very much on the user and her or his specific retrieval task. Users of MIR systems may have various (musical) backgrounds and experience music in different ways. Consequently, when comparing musical clips with each other, opinions may diverge. Apart from considering individual users or user groups, similarity measures also should be tailored to their specific retrieval task to improve the performance of the retrieval system. For instance, when looking for cover versions of a song, the timbre may be less interesting than the lyrics. Various machine learning approaches have recently been proposed for adapting a music similarity measure for a specific purpose. They are briefly reviewed in Section 2. For a systematic comparison of these approaches, a benchmark experiment based on the *MagnaTagATune* dataset has been designed, which is described in Section 3. Section 4 discusses the results of the comparison and Section 5 finally draws conclusions.

*The two leading authors contributed equally to this work.

2. ADAPTATION APPROACHES

The approaches covered in this paper focus on learning a distance measure, which (from a mathematical perspective) can be considered as a dual concept to similarity. The learning process is guided by so-called *relative distance constraints*. A relative distance constraint (s, a, b) demands that the object a is closer to the seed object s than object b, i.e.,

$$d(s, a) < d(s, b) \tag{1}$$

Such constraints can be seen as atomic bits of information fed to the adaptation algorithm. They can be derived from a variety of higher-level application-dependent constraints. For instance, in the context of interactive clustering, assigning a song s to a target cluster with the prototype c_t can be interpreted by the following set of relative distance constraints as proposed by Stober et al. [11]:

$$d(s, c_t) < d(s, c) \qquad \forall c \in C \setminus \{c_t\} \tag{2}$$

where C is the set of cluster prototypes. Bade et al. describe how relative distance constraints can be derived from expert classifications of folk songs [1] or from an existing personal hierarchy of folders with music files [2]. Alternatively, it is also possible to directly ask the users to state the opinion for a triplet of songs as in the bonus round of the *TagATune* game [7]. (Section 3.2 covers this in detail.) McFee et al. [8] use artist similarity triples collected in the web survey described in [5]. They also describe a graph-based technique to detect and remove inconsistencies within sets of constraints such as direct contradictions.

Using relative distance constraints, the task of learning a suitable adaptation of a distance measure can be formulated mathematically as constraint optimization problem.

In the following, the two general approaches covered in this comparison are briefly reviewed.

2.1 Linear Combinations of Facet Distances

Stober et al. model the distance $d(a, b)$ between two songs as weighted sum of *facet distances* $\delta_{f_1}(a, b), \ldots, \delta_{f_l}(a, b)$:

$$d(a, b) = \sum_{i=1}^{l} w_i \delta_{f_i}(a, b) \quad (3)$$

Each facet distance refers to an objective comparison of two music clips with respect to a single facet of music information such as melody, timbre, or rhythm. Here, the facet weights $w_1, \ldots, w_l \in \mathbb{R}^+$ serve as parameters of the distance measure that allow to adapt the importance of each facet to a specific user or retrieval task. These weights obviously have to be non-negative so that the aggregated distance cannot decrease where a single facet distance increases. Furthermore, the sum of the weights should be constant such as

$$\sum_{i=1}^{l} w_i = l \quad (4)$$

to avoid arbitrarily large distance values.

The small number of parameters somewhat limits the expressivity of the distance model. However, at the same time, the weights can easily be understood and directly manipulated by the user. Stober et al. argue that this design choice specifically addresses the users' desire to remain in control and not to be patronized by an intelligent system that "knows better". In [11], they describe various applications and respective adaptation algorithms which they evaluate and compare in [12] using the *MagnaTagATune* dataset. Three of these approaches are covered by the comparison in this paper.

2.1.1 Gradient Descent

Here, if a constraint is violated by the current distance measure, the weighting is updated by trying to maximize

$$obj(s, a, b) = \sum_{i=1}^{l} w_i(\delta_{f_i}(s, b) - \delta_{f_i}(s, a)) \quad (5)$$

which can be directly derived from Equation 1. This leads to the following update rule for the individual weights:

$$w_i = w_i + \eta \Delta w_i, \quad \text{with} \quad (6)$$

$$\Delta w_i = \frac{\partial obj(s, a, b)}{\partial w_i} = \delta_{f_i}(s, b) - \delta_{f_i}(s, a) \quad (7)$$

where the learning rate η defines the step width of each iteration. As in [12], the optimization process is restarted 50 times with random initialization and the best result is chosen to reduce the risk of getting stuck in a local optimum.

2.1.2 Quadratic Programming

Of the various quadratic programming approaches covered in [12], only the one minimizing the quadratic slack is considered here because it was the best performing one in the original comparison. In this approach, an individual slack variable is used for each constraint, which allows violations. As optimization objective, the sum of the squared slack values has to be minimized.

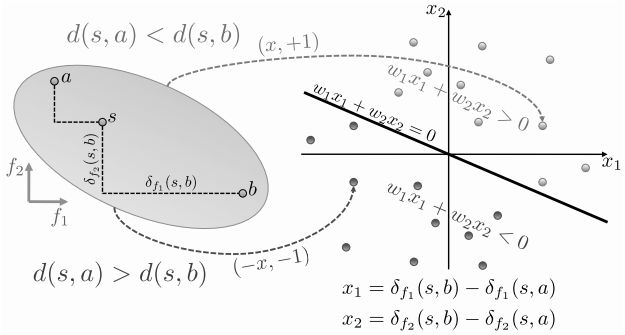

Figure 1. Transformation of a relative distance constraint for linear combination models into two training instances of the corresponding binary classification problem as described by Cheng et al. [3].

2.1.3 Linear Support Vector Machine (LibLinear)

The third approach takes a very different perspective. As described by Cheng et al. [3], the learning task can be reformulated as a binary classification problem, which opens the possibility to apply a wide range of sophisticated classification techniques such as (linear) Support Vector Machines (SVMs). Figure 1 illustrates this idea to rewrite each relative distance constraint $d(s, a) < d(s, b)$ as

$$\sum_{i=1}^{m} w_i(\delta_{f_i}(s, b) - \delta_{f_i}(s, a)) = \sum_{i=1}^{m} w_i x_i = \mathbf{w}^T \mathbf{x} > 0 \quad (8)$$

where x_i is the *distance difference* with respect to facet f_i. The positive training example $(\mathbf{x}, +1)$ then represents the satisfied constraint whereas the negative example $(-\mathbf{x}, -1)$ represents its violation (i.e., inverting the relation sign). For these training examples, the normal vector of the hyperplane that separates the positive and negative instances contains the adapted facet weights. As in [12], the *LibLinear* library is used here, which finds a stable separating hyperplane but still suffers from the so far unresolved problem that the non-negativity of the facet weights cannot be enforced.

2.2 Metric Learning

Alternative approaches to weighting predefined facet distance measures include direct manipulation of parametrized vector distance measures. All features are concatenated to a single combined feature vector per clip. We model a clip's feature vector by $g(a) : \mathbb{N} \mapsto \mathbb{R}^N$. This corresponds to assigning a single facet to each feature dimension. Frequently, the mathematical form of Mahalanobis metrics is used to specify a parametrized vector distance measure. In contrast to the approaches described in the previous section, adaptation is performed in the (combined) feature space itself: Given two feature vectors $\mathbf{a} = g(a)$, $\mathbf{b} = g(b) \in \mathbb{R}^N$, the family of Mahalanobis distance measures can be expressed by

$$d_{\mathbf{W}}(\mathbf{a}, \mathbf{b}) = \sqrt{(\mathbf{a} - \mathbf{b})^T \mathbf{W} (\mathbf{a} - \mathbf{b})}, \quad (9)$$

where $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a positive semidefinite matrix, parametrizing the distance function. Generic variants of the

Euclidean metric, Mahalanobis metrics allow for linear transformation of the feature space when accessing distance. An important property of this approach is that the number of adjustable parameters directly depends on the dimensionality N of the feature space. As this number grows quadratically with N, many approaches restrict training to the N parameters of a diagonal matrix $\mathbf{W}$, only permitting a weighting of the individual feature dimensions.

2.2.1 Linear Support Vector Machine (SVMLight)

The SVM approach explained in Section 2.1.3 has been shown as well suited to learning a Mahalanobis distance measure: Schultz et al. [10] adapted a weighted kernelized metric towards relative distance constraints. We follow the approach of Wolff et al. [13], where a linear kernel is used. This simplifies the approach of Schultz et al. to learning a diagonally restricted Mahalanobis distance (Equation 9).

Like the SVM for the facet distances, a large margin classifier is optimized to the distance constraints. Here, for each constraint (s, a, b), we replace the facet *distance difference* vector $\mathbf{x}$ in Equation 8 with the difference of the pointwise squared [1] feature difference vectors $\mathbf{x} = (\mathbf{s} - \mathbf{b})^2 - (\mathbf{s} - \mathbf{a})^2$.

Given the vector $\mathbf{w} = \text{diag}(\mathbf{W}), \mathbf{w}_i \geq 0$ and slack variables $\xi_{(s,a,b)} \geq 0$, optimization is performed as follows:

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + c \cdot \sum_{(s,a,b)} \xi_{(s,a,b)} \qquad (10)$$

$$\text{s.t.} \; \forall\,(s,a,b) \qquad \mathbf{w}^T\mathbf{x}_{(s,a,b)} \geq 1 - \xi_{(s,a,b)}$$

Here, c determines a trade-off between regularization and the enforcement of constraints. For the experiments below, the SVM^{light} framework [2] is used to optimize the weights $\mathbf{w}_i$. As for *LibLinear*, $\mathbf{w}_i \geq 0$ cannot be guaranteed.

2.2.2 Metric Learning to Rank

McFee et al. [9] developed an algorithm for learning a Mahalanobis distance from rankings. [3] Using the constrained regularization of Structural SVM, the matrix $\mathbf{W}$ is optimized to an input of clip rankings and their feature vectors. Given a relative distance constraint (s, a, b) (see Equation 1), the corresponding ranking assigns a higher ranking score to a than to b, when querying clip s. For a set X of training query feature vectors $\mathbf{q} \in X \subset \mathbb{R}^N$ and associated training rankings y_q^*, Metric Learning to Rank (MLR) minimizes

$$\min_{\mathbf{W},\xi} \quad tr(\mathbf{W}^T\mathbf{W}) + c\frac{1}{n}\sum_{q\in X}\xi_q, \qquad (11)$$

$$\text{s.t.} \quad \forall \mathbf{q} \in X,\ \forall y \in Y \setminus \{y_q^*\}:$$

$$H_{\mathbf{W}}\left(\mathbf{q}, y_q^*\right) \geq H_{\mathbf{W}}\left(\mathbf{q}, y\right) + \Delta(y_q^*, y) - \xi_q,$$

with $W_{i,j} \geq 0$ and $\xi_q \geq 0$. Here, the matrix $\mathbf{W}$ is regularized using the trace. Optimization is subject to the constraints creating a minimal slack penalty of ξ_q. c determines the trade-off between regularization and the slack penalty for the constraints below. $H_{\mathbf{W}}(q, y)$ [4] assigns a score to the validity of ranking y given the query $\mathbf{q}$ with regard to the Mahalanobis matrix $\mathbf{W}$. This enforces $\mathbf{W}$ to fulfill the training rankings y_q^*. The additional ranking-loss term $\Delta(y_q^*, y)$ assures a margin between the scores of given training rankings y_q^* and incorrect rankings y. The method is kept efficient by selecting only a few possible alternative rankings $y \in Y$ for comparison with the training rankings: A separation oracle is used for predicting the worst violated constraints (see [6]). In our experiments, an MLR variant *DMLR* restricts $\mathbf{W}$ to a diagonal shape.

[1] $(\mathbf{a}^2)_i := (\mathbf{a}_i)^2$

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

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

[4] For simplification, $H_{\mathbf{W}}(q, y)$ substitutes the Frobenius product $\langle W, \psi(q, y)\rangle_F$ in [9].

3. EXPERIMENT DESIGN

3.1 The *MagnaTagATune* Dataset

MagnaTagATune is a dataset combining mp3 audio, acoustic feature data, user votings for music similarity, and tag data for a set of 25863 clips of about 30 seconds taken from 5405 songs provided by the Magnatune [5] label. The bundled acoustic features have been extracted using version 1.0 of the *EchoNest API* [6] . The tag and similarity data has been collected using the TagATune game [7]. TagATune is a typical instance of an online "Game With A Purpose". While users are playing the game mainly for recreational purposes, they annotate the presented music clips. The tag data is collected during the main mode of the game, where two players have to agree on whether they listen to identical clips. Their communication is saved as tag data. The bonus mode of the game involves a typical odd one out survey asking two players to independently select the same outlier out of three clips presented to them. The triplets of clips presented to them vary widely in genre, containing material from ambient and electronica, classical, alternative, and rock.

3.2 Similarity Data

The comparative similarity data in *MagnaTagATune* can be represented in a constraint multigraph with pairs of clips as nodes [8, 12]. The vote for an outlier k in the clip triplet (i, j, k) is transformed into two relative distance constraints: (i, j, k) and (j, i, k). Each constraint (s, a, b) is represented by an edge from the clip pair (s, a) to (s, b). This results in 15300 edges of which 1598 are unique. In order to adapt similarity measures to this data, the multigraph has to be acyclic, as cycles correspond to inconsistencies in the similarity data. The *MagnaTagATune* similarity data only contains cycles of length 2, corresponding to contradictive user statements regarding the same triplet. In order to remove these cycles, the contradicting multigraph edge numbers are consolidated by subtracting the number of edges connecting the same vertices in opposite directions. The remaining 6898 edges corresponding to 860 unique relative distance constraints constitute the similarity data we work with. [7]

[5] http://magnatune.com/

[6] http://developer.echonest.com/

[7] In [12], the authors report that the number of consistent constraints is 674. This differing number was caused by a software bug in the filtering algorithm, which led to the removal of more constraints than necessary.

3.3 Data Partitioning

In order to assess the training performance of the approaches described in Section 2, we compare two cross-validation variants to specify independent test and training sets.

A straightforward method, randomly sampling the constraints into cross-validation bins and therefore into combinations of test and training sets has been used on the dataset before by Wolff et al. [13]. We use this standard method (sampling A) to perform 10-fold cross validation, sampling the data into non-overlapping test and training sets of 86 and 774 constraints respectively

For the second sampling, it is considered that two constraints were derived from each user voting, as such are related to the same clips. Assigning one of such two constraints to training and the remaining one to a test set might introduce bias by referring to common information. In our second validation approach, (sampling B) it is assured that the test and training sets also perfectly separate on the clip set. The 860 edges of the *MagnaTagATune* similarity multigraph connect 337 components of three vertices each. These correspond to the initial setup of clip triplets presented to the players during the *TagATune* game.

As the removal of one clip causes the loss of all similarity information (maximally 3 constraints) within its triplet, the sampling of the test data is based on the triplets rather than the constraints. On the 337 triplets, we use 10-fold cross validation for dividing these into bins of 33 or 34 triplets. Due to the varying number of 2-3 constraints per triplet, the training set sizes vary from 770-779 constraints, leaving the test sets at 81-90 constraints.

For evaluation of generalization and general performance trends, the training sets are analyzed in an expanding subset manner. We start with individual training sets of either 13 constraints (sampling A) or 5 triplets (sampling B), corresponding to 11-15 constraints. The size of the training sets is then increased exponentially, including all the smaller training sets' constraints in the larger ones. Constraints remaining unused for each of the smaller training set sizes are used for further validation, and referred to as *unused training constraints*. For both sampling methods, all test and training sets are fixed, and referred to as *sampling A* and *sampling B*.

3.4 Features and Facets

As features, we use those defined in [12] plus the genre features used by Wolff et al. [13]. This results in the set of features shown in Table 1.

Of the 7 global features, "danceability" and "energy" were not contained in the original clip analysis information of the dataset but have become available with a newer version of the *EchoNest API*. Furthermore, the segment-based features describing pitch ("chroma") and timbre have been aggregated (per dimension) resulting in 12-dimensional vectors with the mean and standard deviation values. This has been done according to the approach described in [4] for the same dataset. The 99 tags were derived from annotations collected through the *TagATune* game [7] by applying the preprocessing steps described in [12]. The resulting binary tag vectors are more dense than for the original 188 tags but still very sparse. The genre labels were obtained from the *Magnatune* label as described by Wolff et al. [13]. A total of 42 genres was assigned to the clips in the test set with 1-3 genre labels per clip. This also results in very sparse binary vectors.

feature	dim.	value description
key	1	0 to 11 (one of the 12 keys) or -1 (none)
mode	1	0 (minor), 1 (major) or -1 (none)
loudness	1	overall value in decibel (dB)
tempo	1	in beats per minute (bpm)
time signature	1	3 to 7 ($\frac{3}{4}$ to $\frac{7}{4}$), 1 (complex), or -1 (none)
danceability	1	between 0 (low) and 1 (high)
energy	1	between 0 (low) and 1 (high)
pitch mean	12	dimensions correspond to pitch classes
pitch std. dev.	12	dimensions correspond to pitch classes
timbre mean	12	normalized timbre PCA coefficients
timbre std. dev.	12	normalized timbre PCA coefficients
tags	99	binary vector (very sparse)
genres	44	binary vector (very sparse)

Table 1. Features for the *MagnaTagATune* dataset. Top rows: Globally extracted *EchoNest* features. Middle rows: Aggregation of *EchoNest* features extracted per segment. Bottom row: Manual annotations from *TagATune* game and the *Magnatune* label respectively.

For the facet-based approaches described in Section 2.1, two different sets of facets are considered consisting of 26 and 155 facets respectively. In both sets, the 7 global features are represented as individual facets (using the distance measures described in [12]). As the genre labels are very sparse, they are combined in a single facet using the Jaccard distance measure. The set of 155 facets is obtained by adding 99 tag facets (as in [12]) and a single facet for each dimension of the 12-dimensional pitch and timbre features. For the set of 26 facets, the pitch and timbre feature are represented as a single facet each (combining all 12 dimensions). Furthermore, 14 tag-based facets are added of which 9 refer to aggregated tags that are less sparse (solo, instrumental, voice present, male, female, noise, silence, repetitive, beat) and 5 compare binary vectors for groups of related tags (tempo, genre, location, instruments, perception / mood). This results in a realistic similarity model of reasonable complexity that could still be adapted manually by a user. The more complex model with almost six times as many facet weight parameters serves as the upper bound of the adaptability using a linear approach for the given set of features.

4. RESULTS

For the similarity data sampling A, Figure 2 shows all of the algorithms to improve the baseline of 63% satisfied unknown constraints by 7% to 10%. Plotted are the performance averages over 10-fold cross-validation as described in Section 3.3. Except for SVM^{light}, most of the final generalization success is achieved within the first 250 training constraints. Only diagonal MLR shows notable non-monotonic behaviour for larger training sets >200 con-

straints. Further tests on the unused training data reproduce the results on the static test sets shown here. As shown in Figure 3, all algorithms are able to satisfy the initial training constraints. With the exception of MLR, (see Section 4.2), the training performance decreases for growing training sets, asymptotically approaching the test set performance. Such effects have been shown in [13] not to contradict good generalization results.

4.1 Impact of Model Complexity

For the facet-based linear approaches (Figure 3, left), a strong impact of the number of facet weight parameters can be observed. Whilst the performance for the model with 155 facets is significantly superior on the training data, it is generally worse on the test data. Only for a high number of training constraints, the simpler model with 26 facets can be matched or slightly outperformed. This is a strong indicator for model overfitting. With its many parameters, the complex model adapts too much to the training data at the cost of a reduced ability to generalize. In contrast, the simple model is able to generalize much quicker. This is especially remarkable for the quadratic programming approach with the quickest generalization of all approaches. Its adaptation performance on the test data also comes closest to the training performance, which can be seen as an upper bound. It appears as if this limit is increased by about 5%, if 155 facets are used instead, but more training examples would be needed to get closer to this value. Here lies great potential for future research: By adapting the model complexity (i.e., the number of parameters) depending on the number of training examples and the performance on some unseen constraints, the ability of simple models to quickly generalize could be combined with the superior adaptability of more complex ones.

4.2 Effects of Similarity Sampling

For most of the algorithms tested, the effect of choosing sampling A or sampling B is small. Best performing are MLR (samling A) and quadratic programming(m = 155) for sampling B. Except for MLR, decrease in test set performance is limited to 1% when trained with the clip-separating sampling B. In the right column of Figure 3, the metric learning algorithms are compared. The bottom black curves represent the test set results for sampling A (dashed, · – ·) and sampling B (solid, —). The training performances for these samplings are plotted on the top of the graphs. While SVM^{light} (d) and *DMLR* (f) only loose 2-3% in performance, MLR (e) drops by more than 6%. Exclusively among the algorithms tested, the fully parametrized MLR(e) variant shows a 100% training performance for all training sizes. In line with results from Wolff et al. [13,14], the algorithm generalizes well on the similarity data with sampling A. Even with further permutations of the data, this capability to generalize reduces significantly when using MLR with our sampling method B, possibly caused by the lack of feature reoccurence in the training data.

5. CONCLUSIONS

The results of the experiment show that all approaches can adapt a similarity model to training data and generalize the learned information to unknown test data. Training performance curves can be used as an indicator for the maximal generalization outcome to expect, which depends on the number of facets and the features used. Sensitivity with respect to the sampling method of the test data was observed for MLR, which requires further investigation. Another promising direction for future work is to dynamically adapt the model complexity, e.g., by regularization. The feature data and sampling information are available online [8] for benchmarking of approaches developed in the future.

6. REFERENCES

[1] K. Bade, J. Garbers, S. Stober, F. Wiering, and A. Nürnberger. Supporting folk-song research by automatic metric learning and ranking. In *Proc. of ISMIR'09*, 2009.

[2] K. Bade, A. Nürnberger, and S. Stober. Everything in its right place? learning a user's view of a music collection. In *Proc. of NAG/DAGA'09, Int. Conf. on Acoustics*, 2009.

[3] W. Cheng and E. Hüllermeier. Learning similarity functions from qualitative feedback. In *Proc. of ECCBR'08*, 2008.

[4] J. Donaldson and P. Lamere. Using visualizations for music discovery. Tutorial at *ISMIR'09*, 2009.

[5] D. P. W. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proc. of ISMIR'02*, 2002.

[6] T. Joachims, T. Finley, and C.-N.J. Yu. Cutting-plane training of structural SVMs. In *Machine Learning*, 2009.

[7] E. Law and L. von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proc. of CHI '09*, 2009.

[8] B. McFee and G. Lanckriet. Heterogeneous embedding for subjective artist similarity. In *Proc. of ISMIR'09*, 2009.

[9] B. McFee and G. Lanckriet. Metric learning to rank. In *Proc. of ICML'10*, 2010.

[10] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2003.

[11] S. Stober. Adaptive distance measures for exploration and structuring of music collections. In *Proc. of AES 42nd Conference on Semantic Audio*, 2011.

[12] S. Stober and A. Nürnberger. An experimental comparison of similarity adaptation approaches. In *Proc. of AMR'11*, 2011.

[13] D. Wolff and T. Weyde. Adapting metrics for music similarity using comparative judgements. In *Proc. of ISMIR 2011*, 2011.

[14] D. Wolff and T. Weyde. Adapting Similarity on the MagnaTagATune Database In *ACM Proc. of WWW'12*, 2012.

[8] http://mi.soi.city.ac.uk/datasets/ismir2012/

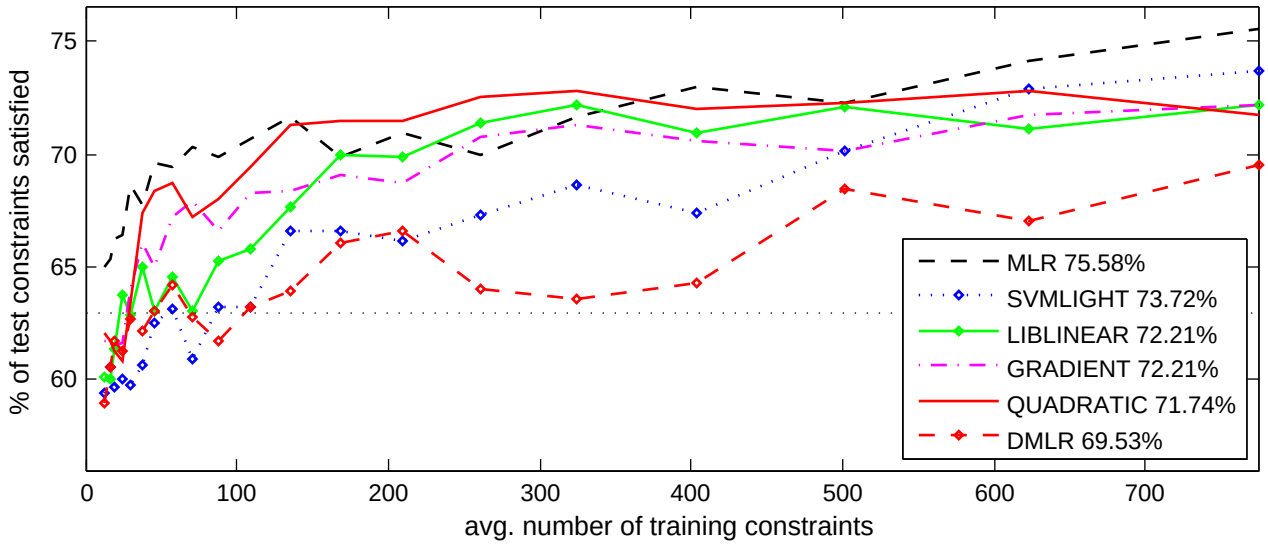

Figure 2. Performance comparison of facet-based approaches (with 26 facets) and metric learning. Values are averaged over all 20 folds of sampling A. The baseline at 63% refers to the mean performance of random facet weights ($n = 1000$).

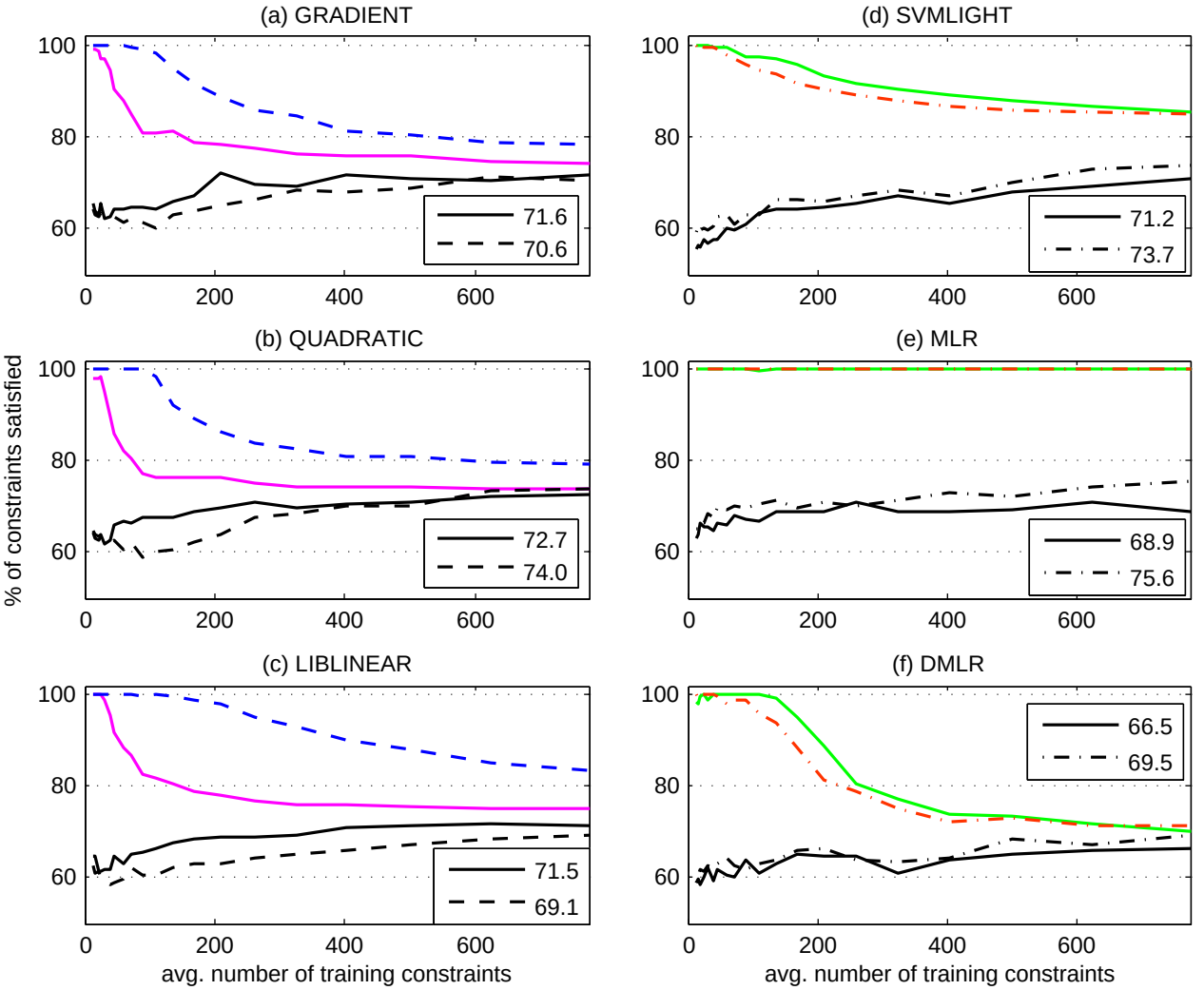

Figure 3. Detailed performance of the individual approaches under different conditions. Top curves show training performance, bottom curves and legend show test set performance. **Left column (a, b, c):** Performance of the facet-based approaches using 26 facets (—) and 155 facets (– –). Comparison based on sampling B. **Right column (d, e, f):** Performance of the metric-based approaches. Effects of sampling A(· – ·) and B(—) are compared.