

# HETEROGENEOUS EMBEDDING FOR SUBJECTIVE ARTIST SIMILARITY

**Brian McFee**

Computer Science and Engineering  
University of California, San Diego  
bmcfee@cs.ucsd.edu

**Gert Lanckriet**

Electrical and Computer Engineering  
University of California, San Diego  
gert@ece.ucsd.edu

## ABSTRACT

We describe an artist recommendation system which integrates several heterogeneous data sources to form a holistic similarity space. Using social, semantic, and acoustic features, we learn a low-dimensional feature transformation which is optimized to reproduce human-derived measurements of subjective similarity between artists. By producing low-dimensional representations of artists, our system is suitable for visualization and recommendation tasks.

## 1. INTRODUCTION

A proper notion of similarity can dramatically impact performance in a variety of music applications, such as search and retrieval, content-based tagging engines, and song or artist recommendation. When designing such a system, practitioners must choose an appropriate measure of similarity for the task at hand. Often, this involves selecting among multiple heterogeneous feature types, which may not be directly comparable, e.g., social network connectivity and probabilistic models of keywords. Integration of diverse features must be conducted carefully to ensure that the resulting similarity measure sufficiently captures the qualities desired for the application.

In music applications, the problem of selecting an optimal similarity measure is exacerbated by *subjectivity*: people may not consistently agree upon whether or to what degree a pair of songs or artists are similar. Even more flexible notions of similarity, such as ranking, may suffer from the effects of inconsistency, which must be understood and counteracted.

In this work, our goal is to construct artist-level similarity measures, adhering to two key principles. First, a similarity measure should integrate heterogeneous features in a principled way, emphasizing relevant features while being robust against irrelevant features. Second, instead of relying solely on features, the measure should learn from people and be optimized for the task at hand, i.e., predicting human perception of similarity. Using recently developed

algorithms, we demonstrate how to learn optimal metrics for subjective similarity while seamlessly integrating multiple feature modalities. We do not mean to imply that there exists a fully consistent *ground truth* in musical similarity. Rather, we seek to construct similarity measures which are maximally consistent with human perception.

### 1.1 Related work

There has been a considerable amount of research devoted to the topic of musical similarity, primarily in the realms of playlist generation and recommendation [3, 14, 18]. The present work is perhaps most similar to that of Slaney, et al. [21], in which convex optimization techniques were applied to learn metric embeddings optimized according to side information. Our work differs in that we focus on artist similarity, rather than classification, and we use direct measurements of human perception to guide the optimization.

Barrington, et al. applied multiple-kernel learning to a classification task [4]. Our approach uses a different formulation of multiple-kernel learning which allows greater flexibility in assigning weights to the features and training set, and produces a metric space instead of a linear separator.

Ellis, et al. and Berenzweig, et al. studied the issue of consistency in human perception of artist similarity, and evaluated several acoustic- and socially-driven similarity measures against human survey data [5, 9]. Their work focused on the comparison of existing measures of similarity (e.g., playlist co-occurrence), rather than learning an optimal measure.

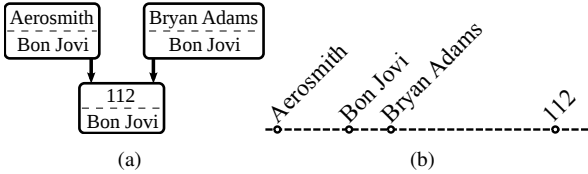
## 2. EMBEDDING ALGORITHMS

Our approach to the artist recommendation task is to embed each artist from a set  $\mathcal{X}$  into a Euclidean space so that distances correspond to human perception of dissimilarity. Although it has been documented that notions of similarity between artists can vary dramatically from person to person, *rankings* of similarity between pairs of artists are comparatively more robust [9].

One simple ranking method involves comparisons of artists  $j$  and  $k$  relative to a fixed reference artist  $i$ . This yields similarity *triplets*  $(i, j, k)$ , indicating that the pair  $(i, j)$  are more similar to each-other than the pair  $(i, k)$ . Data of this variety are becoming increasingly common

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© 2009 International Society for Music Information Retrieval.



**Figure 1.** Similarity triplets can be interpreted as a directed graph over pairs of artists: an edge  $(i, j) \rightarrow (i, k)$  indicates that  $i$  and  $j$  are more similar than  $i$  and  $k$ . (a) The graph representation of two triplets:  $(Bon\ Jovi, Aerosmith, 112)$  and  $(Bon\ Jovi, Bryan\ Adams, 112)$ . (b) An example of a 1-dimensional embedding that satisfies these triplets.

for general ranking and human perception modeling tasks, such as the Tag-a-Tune bonus round [12].

In this setting, we seek a Euclidean embedding function  $g : \mathcal{X} \rightarrow \mathbb{R}^D$  such that each given triplet  $(i, j, k)$  yields

$$\|g(i) - g(j)\|^2 + 1 < \|g(i) - g(k)\|^2, \quad (1)$$

where the unit margin is enforced for numerical stability. In other words, distance in the embedding space corresponds to perceived similarity. This framework eliminates the need to normalize quantitative similarity scores (as in multi-dimensional scaling), and does not over-simplify the description language to a binary problem (e.g., *same* versus *different*).

Several algorithms have been proposed to solve embedding problems in this framework [1, 16, 20]. Here, we briefly summarize the partial order embedding (POE) algorithm of [16].

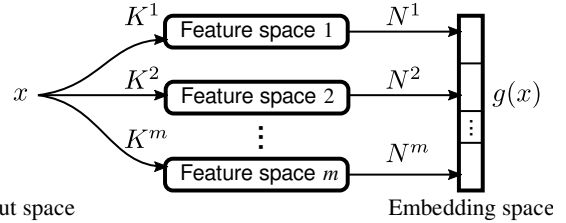
## 2.1 Partial order constraints

A collection of similarity triplets can be equivalently represented as a directed graph in which each vertex represents a pair of artists, and a directed edge indicates a comparison of pairwise similarities (see Figure 1). Interpreting the similarity triplets as a graph allows us to simplify the embedding problem by pruning edges which may be redundant or inconsistent.

If the triplets give rise to a *directed acyclic graph* (DAG), this defines a partial order over distances, which implies the existence of some similarity space which is consistent with the measured triplets. If the graph contains cycles, then no similarity function can satisfy all of the triplets, and we say that the triplets are *inconsistent*. In practice, there are always inconsistencies in human similarity perception, but the graph representation provides a direct way to locate and quantify these inconsistencies. Section 4.1 describes an experiment and methodology to analyze inconsistencies in a collection of similarity measurements.

## 2.2 Multi-kernel embedding

Since our eventual goal is to recommend similar artists when presented with a previously unseen artist, we will need to provide a means to map unseen artists into the embedding space after training, without requiring any similarity measurements for the new artist. POE achieves this



**Figure 2.** The embedding procedure first maps a point  $x$  into  $m$  different non-linear spaces (encoded by  $m$  different kernel matrices), and then learns a set of projections  $N^p$  ( $p = 1 \dots m$ ) which form the embedding space.

by restricting the choice of embedding functions to linear projections from a given feature space. This readily generalizes to non-linear embeddings through the use of *kernel functions* [19]. Artists are first mapped into a high-dimensional inner-product space by a feature transform defined by a kernel function  $k(\cdot, \cdot)$ . POE then learns a projection from this feature space into a low-dimensional Euclidean space. This leads to the parameterization

$$g(x) = NK_x,$$

where  $N$  is a linear projection matrix, and  $K_x$  is the vector formed by evaluating a kernel function  $k(x, i)$  against all points  $i$  in the training set.

Since formulating the problem in terms of  $N$  would lead to a non-convex optimization problem — with perhaps infinitely many parameters — POE instead optimizes over a positive semidefinite matrix  $W = N^T N \succeq 0$  [6].  $N$  may be infinite-dimensional (as is the case in Gaussian kernels), but an approximation to  $N$  can be recovered from  $W$  by spectral decomposition:

$$\begin{aligned} N^T N = W &= V \Lambda V^T = V \Lambda^{1/2} \Lambda^{1/2} V^T \\ &= \left( \Lambda^{1/2} V^T \right)^T \left( \Lambda^{1/2} V^T \right) = \tilde{N}^T \tilde{N} \end{aligned} \quad (2)$$

where  $V$  and  $\Lambda$  contain the eigenvectors and eigenvalues of  $W$ .

In MIR tasks, it is becoming common to combine data descriptions from multiple feature modalities, e.g., social tags and spectral features [4]. POE accomplishes this by learning a separate transformation  $N^p$  for each of  $m$  kernel matrices  $K^p$  ( $p = 1 \dots m$ ), and concatenating the resulting vectors (see Figure 2). This formulation allows (squared) distance computations in the embedding space to be decomposed as

$$d(i, j) = \sum_{p=1}^m (K_i^p - K_j^p)^T W^p (K_i^p - K_j^p). \quad (3)$$

The multi-kernel POE algorithm is given as Algorithm 1. The objective function has three components: the first term,  $\sum_{i,j} d(i, j)$  maximizes the variance of the embedded points, which has been demonstrated to be effective for reducing dimensionality in manifold data [23]. In the present application, variance maximization diminishes erroneous recommendations by pushing all artists far away from each-

other, except where prevented from doing so by similarity ordering constraints.

The second term,  $-\beta \sum_{\mathcal{C}} \xi_{ijk}$ , incurs hinge-loss penalties for violations of similarity constraints, scaled according to a free parameter  $\beta$ . The last term,  $-\gamma \sum \text{Tr}(W^p K^p)$ , regularizes the solution and enforces sparsity in the solution, again scaled by a free parameter  $\gamma$ . Parameters  $\beta$  and  $\gamma$  are tuned by cross-validation, similar to the  $C$  parameter in support vector machines [7].

There are four types of constraints in Algorithm 1. The first,  $d(i, j) \leq \Delta_c$ , bounds the diameter of the embedding to resolve scale invariance.<sup>1</sup> The second set of constraints,  $d(i, j) + 1 - \xi_{ijk} \leq d(i, k)$  enforces consistency between the learned distances and similarity triplets in the training set, as in Equation 1. The slack terms  $\xi_{ijk} \geq 0$  allow similarity constraints to be violated, provided it yields an overall increase in the value of the objective function. Finally,  $W^p \succeq 0$  forces each  $W^p$  to be positive semidefinite, so that the  $N^p$  matrices can be recovered as in Equation 2.

The optimal solution  $(W^1, W^2, \dots, W^m)$  is computed by gradient ascent, and then each matrix is decomposed to produce the embedding function

$$g(x) = (N^p K_x^p)_{p=1}^m, \quad (4)$$

where  $(N^p K_x^p)_{p=1}^m$  denotes the concatenation over all  $m$  vectors  $N^p K_x^p$ .

---

**Algorithm 1** Multi-kernel partial order embedding [16].  $d(i, j)$  is defined as in Equation 3, and  $(W^1, W^2, \dots, W^m)$  are optimized by gradient ascent.

---

**Input:** kernel matrices  $K^1, K^2, \dots, K^m$ , triplets  $\mathcal{C} = \{(i, j, k) : (i, j) \text{ more similar than } (i, k)\}$   
**Output:** matrices  $W^1, W^2, \dots, W^m \succeq 0$ .

$$\begin{aligned} & \max_{W^p, \xi} \sum_{i, j} d(i, j) - \beta \sum_{\mathcal{C}} \xi_{ijk} - \gamma \sum_p \text{Tr}(W^p K^p) \\ & \text{s. t.} \\ & \forall i, j \in \mathcal{X} \quad d(i, j) \leq \Delta_c \\ & \forall (i, j, k) \in \mathcal{C} \quad d(i, j) + 1 - \xi_{ijk} \leq d(i, k) \\ & \quad \quad \quad \xi_{ijk} \geq 0 \\ & \forall p \in 1, 2, \dots, m \quad W^p \succeq 0 \end{aligned}$$


---

### 3. DATA

To evaluate our system, we designed experiments around the *aset400* data set of Ellis, et al [9]. The data consists of 412 popular artists, and similarity triplets collected with a web survey in 2002. We augmented the data set with several types of features, both human-derived (tags and text), and purely content-driven, as described below.

#### 3.1 Text features

Our text-based features were collected from Last.FM between January and May of 2009. To standardize the list

<sup>1</sup>  $\Delta_c$  is computed from the structure of the similarity triplets graph, and is not a free parameter. See [16] for details.

of artist names, we used the *search\_artists* method of the Echo Nest API [17].

We then collected for each artist two types of textual features from Last.FM: biography summaries and the top 100 tags [11]. The tags were filtered by a small set of regular expressions to resolve common spelling variations. For example, *r-n-b*, *r&b*, *r-and-b* were all mapped to *rnb*, and the merged tag *rnb* received a score equal to the sum of scores of its constituent tags.

The tags and biographies were filtered by stop-word removal and stemming, resulting in dictionaries of 7737 unique tag words, and 16753 biography words. Each artist was summarized as two bags of words (one for tags and one for biographies), which were then re-weighted by TF-IDF. Finally, to compare similarity between artists, we constructed kernels  $K^{\text{tag}}$  and  $K^{\text{bio}}$  defined by the cosine similarity between word vectors.

#### 3.2 Acoustic features

For each artist, we selected between one and ten songs at random (depending on availability), with an average of 3.8 songs per artist. From these songs, we extracted a variety of content-based features. Since content-based features relate to songs and not directly to artists, we do not expect them to perform as well the textual features described above. We are primarily interested in integrating heterogeneous features, and quantifying the improvements achieved by optimizing for artist similarity.

##### 3.2.1 MFCC

Mel-frequency cepstral coefficients (MFCCs) have been demonstrated to capture timbral or textural qualities, and perform well in a variety of MIR applications [13, 15]. For each song, we compute the first 13 MFCCs for up to 10000 half-overlapping short-time segments (23 msec), along with the first and second instantaneous derivatives. This results in a collection of 39-dimensional delta-MFCC vectors for each song.

Each artist was summarized by modeling the distribution of delta-MFCC vectors in all songs belonging to that artist, using a Gaussian mixture model (GMM) of 8 components and diagonal covariances. Then, to compare models between artists, we construct a probability product kernel (PPK) between the GMMs:

$$K_{ij}^{\text{MFCC}} = \int \sqrt{p(x; \theta_i^{\text{MFCC}}) p(x; \theta_j^{\text{MFCC}})} dx,$$

where  $\theta_i^{\text{MFCC}}$  and  $\theta_j^{\text{MFCC}}$  are the GMM model parameters for artists  $i$  and  $j$  [10]. Unlike kernels derived from Kullback Leibler divergence, PPK can be computed in closed-form for mixtures of Gaussians.

##### 3.2.2 Chroma

For each song in our database, we modeled the distribution of spectral energy present in frequencies corresponding to the chromatic scale, resulting in a 12-dimensional vector for every 250 msec of audio. Although chroma features are not specifically suited to the artist similarity task, they

have been shown to work well in other applications when combined with other features, such as MFCCs [8]. We summarized each artist by collecting chroma features for each of the artist’s songs, which were then modeled with a single full-covariance Gaussian distribution ( $\theta^{\text{ch}}$ ). From these chroma models, we construct an artist similarity kernel<sup>2</sup> from symmetrized KL-divergence:

$$D_{KL}(i, j) = \int p(x; \theta_i^{\text{ch}}) \log \frac{p(x; \theta_i^{\text{ch}})}{p(x; \theta_j^{\text{ch}})} dx$$

$$K_{ij}^{\text{ch}} = \exp\left(-\frac{D_{KL}(i, j) + D_{KL}(j, i)}{\mu}\right),$$

where  $\mu$  is the mean KL-divergence over all pairs  $i, j$ . Since we are not using mixture models here, this can be computed in closed form.

### 3.2.3 Content-based auto-tagging

In contrast to the low-level acoustic features, we also evaluate high-level conceptual features which were automatically synthesized from audio content. To achieve this, we computed semantic multinomial distributions using the system described in [22]. For each song, the auto-tagger examines the acoustic content and produces a multinomial distribution over a vocabulary  $\mathcal{V}$  of 149 words, e.g., *mel-low*, *dance pop*, *horn section*, etc. The semantic model parameters  $\theta_i^{\text{SM}}$  for an artist  $i$  were computed by averaging the parameters of each of that artist’s song-level models. (We also tested a version using the point-wise maximum of song-level models, but it yielded little quantitative difference.) To compare models between artists, we construct a semantic multinomial kernel using the multinomial PPK:

$$K_{ij}^{\text{SM}} = \left( \sum_{x \in \mathcal{V}} \sqrt{p(x; \theta_i^{\text{SM}}) p(x; \theta_j^{\text{SM}})} \right)^s.$$

This is equivalent to a homogeneous polynomial kernel of degree  $s$  over the model parameter vectors. For our experiments, setting  $s = 75$  yielded reasonable results.

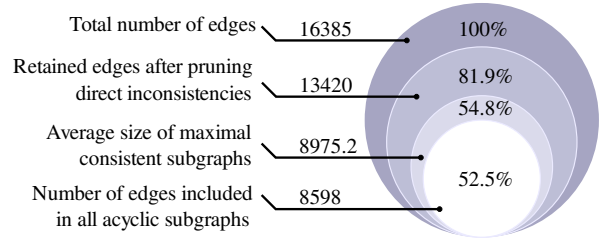
## 4. EXPERIMENTS

### 4.1 Quantifying inconsistency

The aset400 data set consists of 412 popular artists, and similarity triplets gathered from a web-based survey. In the survey, an informant was presented with a query artist  $i$ , and was asked to select, from a list of ten artists, the response  $j$  most similar to the query artist. Then, for each of the remaining responses  $k$  which were not selected, measurements  $(i, j, k)$  were recorded. Note that in a list of ten potential responses, there may be several “good” choices. Being forced to choose a single best response therefore results in numerous inconsistencies in the triplets, which we set out to quantify.

The survey data contains 98964 triplets, generated from 10997 queries to 713 human informants. We analyze the *filtered* version of the data, which has been reduced to

<sup>2</sup> Symmetrized KL-divergence does not generally produce a PSD kernel matrix, but the POE algorithm is still correct for indefinite kernels.



**Figure 3.** Quantitative summary of consistency within the aset400 filtered triplets. *Directly inconsistent* triplets are those where both  $(i, j, k)$  and  $(i, k, j)$  are present.

16385 measurements wherein the informant was likely to be familiar with the artists in question. Although this greatly reduces the amount of noise present in the full set, the filtered set still contains numerous inconsistencies.

Consistency in the similarity measurements can be quantified by analyzing their graph representation. As a first step, we filter out all measurements  $(i, j, k)$  if  $(i, k, j)$  is also present, i.e., those artist-pairs which the informants could not consistently rank. We refer to these triplets as *directly inconsistent*. Removing these triplets decreases the number of edges by 18.1% to 13420.

However, simply removing all length-2 cycles from the graph does not ensure consistency: all cycles must be removed. Finding a maximum acyclic subgraph is NP-hard, but we can find an approximate solution by Algorithm 2. Since the algorithm is randomized, we repeat it several times to compute an estimate of the average maximal acyclic subgraph. With 10 trials, we find consistent subsets of average size 8975.2.

To evaluate the stability of these subgraphs, we count the number of edges present in all solutions, i.e., those measurements which are never pruned. Over 10 trials, 8598 edges (95.8%) were common to all solutions, leaving 4.2% variation across trials. Our results are summarized in Figure 3.

---

#### Algorithm 2 Approximate maximum acyclic subgraph

---

**Input:** Directed graph  $G = (V, E)$

**Output:** Acyclic graph  $G'$

$E' \leftarrow \emptyset$

**for** each  $(u, v) \in E$  in random order **do**

**if**  $E' \cup \{(u, v)\}$  is acyclic **then**

$E' \leftarrow E' \cup \{(u, v)\}$

**end if**

**end for**

$G' \leftarrow (V, E')$

---

### 4.2 Order prediction

The goal of our system is to recommend similar artists in response to a query. To evaluate the system, we test its ability to predict for artists  $i, j$  and  $k$  (where  $i$  is unseen), the ordering of similarity between  $(i, j)$  and  $(i, k)$ , i.e., which of the artists  $j$  or  $k$  is more similar to artist  $i$ .

### 4.2.1 Experimental Setup

We split the data for 10-fold cross validation, resulting in 370 training and 42 test artists for each fold. All directly inconsistent triplets were removed from both training and test sets, as described in Section 4.1. For each training set, we filtered the triplets to produce a maximal acyclic subgraph, retaining only those measurements which were included in all of 10 trials of Algorithm 2. The acyclic subgraphs were then pruned down to their *transitive reductions*, i.e., minimal graphs with equivalent transitivity properties [2]. This effectively removes the measurements which could be deduced from others, thereby reducing the complexity of the embedding problem with no loss of quality. The resulting training sets have an average of 6252.7 similarity measurements. The corresponding graphs have average diameter 30.2, indicating the longest contiguous chain of comparisons which can be followed in the training sets.

For each test set, we included only those triplets  $(i, j, k)$  where  $i$  is in the test set and  $j, k$  are in the training set, resulting in an average of 1149.6 triplets per test set. Aside from pruning directly inconsistent triplets, no further processing was done to enforce consistency in the test set. Therefore, we cannot expect 100% prediction accuracy on the test set. As shown in Figure 3, we can expect a lower-bound on the achievable accuracy of 67% (8975.2/13420). This is consistent with the upper-bound of 85% constructed in [9].

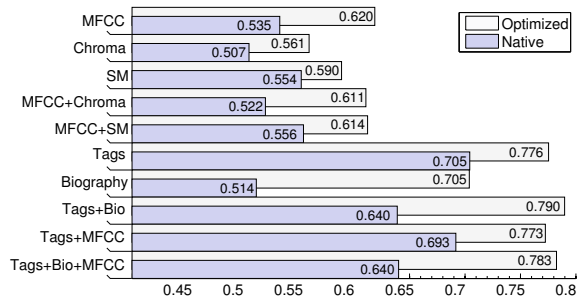
### 4.2.2 Results

We tested the embedding method on each of the kernels described in Sections 3.1 and 3.2 independently, and then combined. The free parameters  $\beta$  and  $\gamma$  were tuned by sweeping over  $\beta \in [100, 10000]$  and  $\gamma \in [100, 1000]$ . After learning, performance was evaluated by counting the number of test-triplets correctly predicted by Euclidean distance in the embedding space.

Figure 5 illustrates two regions of an embedding produced by the combination of tags and biography features, including several query points which were mapped in after learning. The nearest neighbors of the query points provide reasonable recommendations, and the neighborhoods are generally consistent. Moreover, neighborhoods which are largely dissimilar (e.g., *female vocals* and *punk*) have been pushed to opposite extremes of the space by the variance maximization objective.

For comparison purposes, we also evaluated the prediction accuracy of distance-based ranking in the native feature spaces. Native multi-kernel results were computed by concatenating the kernels together to form feature vectors, which is equivalent to setting each  $N^p = I$ . This provides an intuitive and consistent way to compute distances to neighbors in one or more feature spaces.

Figure 4 lists the quantitative results of our experiments. In all cases, prediction accuracy improves significantly after learning the optimal embedding. Moreover, the improvement is more significant than it may at first seem,



**Figure 4.** Triplet prediction accuracy for each feature and combinations, before and after learning.

since the maximum achievable performance is less than 100% due to inconsistencies in the test set.

It is not surprising that textual features give the best performance, and there are two main factors which explain this effect. First, only textual features were attributed directly to artists and not songs. Second, textual features derive from natural language, which is well-suited to describing subtle differences. We achieve significant improvements by optimizing the similarity metric, with gains of 7% for tags and 19% for biographies. Moreover, combining both types of textual features results in better performance than either feature on its own.

As expected, embeddings based on acoustic features perform significantly worse than those derived from text. We believe this is primarily due to the fact that acoustic features relate directly to songs, and variation across an artist’s songs introduces noise to the artist-level models. Note that combining a kernel which performs poorly (e.g., chroma) does not significantly degrade the overall performance, indicating that the algorithm correctly selects the most relevant features available.

### 4.2.3 Comparison

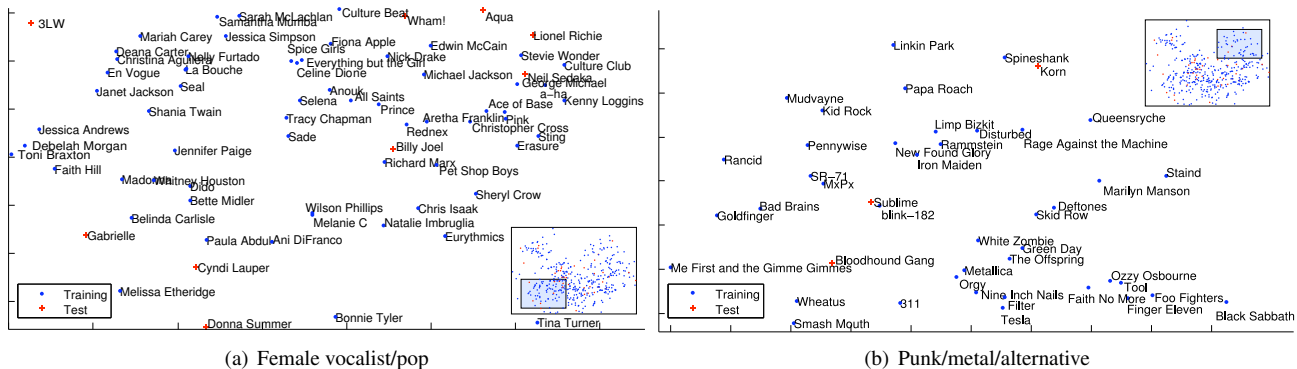
Our results can be directly compared to the “unweighted agreement” score measurements of [9]. Particularly of interest is the comparison of our biography-based embedding, which is analogous to the text-based measures in [9]. Our biography features natively achieve 51.4% accuracy, compared to 57.4% for the web documents in [9]. However, the optimized embedding improves prediction accuracy to 70.5%.

## 5. CONCLUSION

In this paper, we demonstrated a method for optimizing multi-modal musical similarity measures to match human perception data. We believe that the techniques illustrated here could be applicable in other subjective similarity tasks, particularly at the song level, and this will be the focus of future work.

## 6. ACKNOWLEDGEMENTS

We thank Luke Barrington and Douglas Turnbull for their advice. Gert Lanckriet is supported by NSF grant DMS-MSPA 0625409.



**Figure 5.** Two neighborhoods at opposite extremes of an optimized text-based embedding.

## 7. REFERENCES

- [1] Sameer Agarwal, Joshua Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multi-dimensional scaling. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2007.
- [2] A. V. Aho, M. R. Garey, and J. D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.
- [3] Jean-Julien Aucouturier and François Pachet. Music similarity measures: What’s the use? In *International Symposium on Music Information Retrieval (ISMIR2002)*, pages 157–163, 2002.
- [4] Luke Barrington, Mehrdad Yazdani, Douglas Turnbull, and Gert Lanckriet. Combining feature kernels for semantic music retrieval. In *International Symposium on Music Information Retrieval (ISMIR2008)*, pages 614–619, September 2008.
- [5] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.
- [6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [7] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [8] D. Ellis. Classifying music audio with timbral and chroma features. In *International Symposium on Music Information Retrieval (ISMIR2007)*, pages 339–340, 2007.
- [9] D. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR2002)*, pages 170–177, October 2002.
- [10] Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.
- [11] Last.FM, 2009. <http://www.last.fm/>.
- [12] Edith Law and Luis von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1197–1206, New York, NY, USA, 2009. ACM.
- [13] B. Logan. Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information Retrieval (ISMIR2000)*, 2000.
- [14] B. Logan. Music recommendation from song sets. In *International Symposium on Music Information Retrieval (ISMIR2004)*, 2004.
- [15] M. Mandel and D. Ellis. Song-level features and support vector machines for music classification. In *International Symposium on Music Information Retrieval (ISMIR2005)*, pages 594–599, 2005.
- [16] Brian McFee and Gert Lanckriet. Partial order embedding with multiple kernels. In *Proceedings of the Twenty-sixth International Conference on Machine Learning*, 2009.
- [17] The Echo Nest, 2009. <http://www.echonest.com/>.
- [18] E. Pampalk, A. Rauber, and D. Merkl. Content-based Organization and Visualization of Music Archives. In *Proceedings of the ACM Multimedia*, pages 570–579, Juan les Pins, France, December 1-6 2002. ACM.
- [19] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
- [20] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.
- [21] M. Slaney, K. Weinberger, and W. White. Learning a metric for music similarity. In *International Symposium on Music Information Retrieval (ISMIR2008)*, pages 313–318, September 2008.
- [22] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):467–476, February 2008.
- [23] Kilian Q. Weinberger, Fei Sha, and Lawrence K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty-first International Conference on Machine Learning*, pages 839–846, 2004.