Actively Learning Hemimetrics with Applications to Eliciting User Preferences

Adish Singla ADISH.SINGLA@INF.ETHZ.CH
Sebastian Tschiatschek SEBASTIAN.TSCHIATSCHEK@INF.ETHZ.CH
Andreas Krause KRAUSEA@ETHZ.CH
ETH Zurich, Switzerland

Abstract

Motivated by an application of eliciting users' preferences, we investigate the problem of learning hemimetrics, *i.e.*, pairwise distances among a set of n items that satisfy triangle inequalities and non-negativity constraints. In our application, the (asymmetric) distances quantify private costs a user incurs when substituting one item by another. We aim to learn these distances (costs) by asking the users whether they are willing to switch from one item to another for a given incentive offer. Without exploiting structural constraints of the hemimetric polytope, learning the distances between each pair of items requires $\Theta(n^2)$ queries. We propose an active learning algorithm that substantially reduces this sample complexity by exploiting the structural constraints on the version space of hemimetrics. Our proposed algorithm achieves provably-optimal sample complexity for various instances of the task. For example, when the items are embedded into K tight clusters, the sample complexity of our algorithm reduces to $\mathcal{O}(nK)$. Extensive experiments on a restaurant recommendation data set support the conclusions of our theoretical analysis.

1 Introduction

Learning a distance function over a set of items or a data manifold plays a crucial role in many real-world applications. In machine learning algorithms, the distances serve as a notion of similarity (or dissimilarity) between data points and are important for various tasks such as clustering (Xing et al., 2002), object ranking (Lim & Lanckriet, 2014), image retrieval / classification (He et al., 2004; Huang et al., 2015), etc.[1] In economics, the distance function can encode the preferences of users (*e.g.*, buyers or sellers in a marketplace) for different items (*e.g.*, from a catalogue of products) to improve product recommendation and dynamic pricing of goods (Desjardins et al., 2006; Horton & Johari, 2015; Blum et al., 2015; Vázquez-Gallo et al., 2014).

Motivating applications. We are interested in learning the preferences of users across different choices available in a marketplace — these choices are given in the form of n (types of) items. For instance, in a restaurant recommendation system such as *Yelp*, the item types could correspond to restaurants abstracted by attributes such as cuisine, locality, reviews and so on. Consider a user who seeks recommendations from the system and has chosen item i (*e.g.*, "Mexican restaurant in Manhattan with over 50 reviews"). However, to incentivize exploration and maximize its overall utility, the marketplace may consider offering a discount to the user to instead choose item j (*e.g.*, "Newly opened fastfood restaurant in New Jersey with 0 reviews"), *e.g.*, to gather more reviews for item j. The price of the offer would clearly depend on how similar or dissimilar the choices i and j are. In general, a high dissimilarity would require the system to offer higher incentives (larger discounts).

Distance function quantifying private costs. We capture the above-mentioned notion of dissimilarity by a pairwise distance $D_{i,j}$ — the distance $D_{i,j}$ corresponds to the private cost of a user incurred by switching from her default choice of item i to item j. We assume that the distance function D is a *hemimetric*, *i.e.*, a relaxed form of a metric, satisfying only non-negativity constraints and triangle inequalities. The asymmetry in the distances (*i.e.*, $D_{i,j} \neq D_{j,i}$) is naturally required in our application setting — it may arise from various factors such as the underlying quality of the items (*e.g.*, switching between a highly rated and an unreviewed restaurant). Our goal is to efficiently learn the distance function D via interactions with the users, without assuming any knowledge of the underlying attributes that affect the users' preferences.

User query and active learning. In our setting, the interactions with the users take the form of a binary labeling query, *i.e.*, given two items i and j, and a proposed value c, the user

[1] We refer the interested reader to the survey by Bellet et al. (2013) for a detailed discussion of various applications.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

provides a positive response iff value c is at least the underlying distance $D_{i,j}$, and a negative response otherwise. This query is motivated by the *posted-price model* used in marketplaces (Abernethy et al., 2015; Singla & Krause, 2013), where users are offered a take-it-or-leave-it price by the system, and they can accept by providing a positive response, or reject by providing a negative response. However, we are not considering the economic aspects of the query, and each query has unit cost for the algorithm. Such a feedback setting is realistic and often employed by online marketplaces where queries are posted to the users in form of surveys to seek feedback or the users who accept are awarded the actual monetary offer via a lottery. This paper is concerned with the sample complexity, *i.e.*, the number of queries required to learn D.

1.1 Our Approach and Main Contributions

A naive approach to this problem is to learn each of the n^2 pairwise distances independently. However, the sample complexity of this approach is $\Theta(n^2)$. Our goal is to reduce the sample complexity by exploiting the structural constraints on the version space of hemimetrics. The main contributions of this paper are as follows:

Novel metric learning framework. We propose a new learning problem in the framework of metric learning, motivated by the applications to eliciting users' preferences. The key distinctive features of our setting are: *(i)* the specific modality of the user queries (with natural motivation from economics), and *(ii)* the asymmetry of the distances learnt.

Exploiting structural constraints. We develop a novel active learning algorithm LEARNHM that substantially reduces the sample complexity by exploiting the structural constraints of the hemimetric polytope. We provide tight theoretical guarantees on the sample complexity of the proposed algorithm for several natural problem instances.

Practical extensions. Our algorithm extends to various important practical settings, including: *(i)* the online setting where the n items are not known beforehand, and rather appear over time, and *(ii)* the noisy setting that reflects the stochastic nature of acceptance of offers by the users.

2 Problem Statement

We now formalize the problem addressed in this paper.

Items $\mathcal{A}$. There are n items (or types of items), denoted by the set $\mathcal{A} = \{1, 2, \ldots, n\}$. For instance, in a restaurant recommendation system such as *Yelp*, $\mathcal{A}$ could consist of types of restaurants distinguished by high-level attributes such as cuisine, locality, reviews and so on.

Hemimetric D^*. Let $\mathcal{D}$ be the set of bounded *hemimetrics*, *i.e.*, matrices $D \in \mathbb{R}^{n\times n}$ that satisfy

$$D_{i,i} = 0 \quad \forall\, i \in [n], \tag{1}$$

$$D_{i,j} \geq 0 \quad \forall\, i, j \in [n], \tag{2}$$

$$D_{i,j} \leq r \quad \forall\, i, j \in [n], \text{ and} \tag{3}$$

$$D_{i,j} \leq D_{i,k} + D_{k,j} \quad \forall\, i, j, k \in [n], \tag{4}$$

where $[n] = \{1, 2, \ldots, n\}$ and r is the upper bound on the value. We assume that user preferences are represented by an underlying *unknown* hemimetric $D^* \in \mathcal{D}$. The (asymmetric) distance $D^*_{i,j}$ quantifies the private costs of the user for switching from item i to j. Our goal is to learn D^* via interactions with the users, without assuming any knowledge of the underlying attributes that affect the user preferences.

User query. A query $x \in \mathcal{X} := \{(i, j, c) \mid i, j \in [n], c \in [0, r]\}$ to the user is characterized by the tuple (i, j, c), where item i denotes the choice of the user, item j denotes the alternative suggested by the algorithm, and c denotes the monetary incentives offer. Note that the notion of *user* as used in this paper is rather generic, and could correspond to one single user or a crowd / cohort of users.

User response. The response to a query (also called label) is denoted as $y = Y(x)$, where $Y\colon \mathcal{X} \mapsto \{0, 1\}$. A label $y = 1$ indicates acceptance, while $y = 0$ indicates rejection. We denote a labeled datapoint by $z = (x, y)$, where $x \in \mathcal{X}$, and $y = Y(x)$. In our setting the acceptance function is stochastic. It is characterized by $\mathbb{P}(Y(x) = y)$ and is required to satisfy the following two mild conditions related to the decision boundary at $D^*_{i,j}$ and monotonicity:

$$\mathbb{P}(Y((i, j, c)) = 1) \geq 0.5 \quad \text{iff } c \geq D^*_{i,j}, \text{ and}$$

$$\mathbb{P}(Y((i, j, c)) = 1) \geq \mathbb{P}(Y((i, j, c')) = 1) \quad \text{for } c \geq c'.$$

For ease of exposition of our main results, we focus on a deterministic noise-free setting in the main paper — treatment of the (more realistic) stochastic acceptance function is presented in the extended version of this paper (Singla et al., 2016). In this noise-free setting, the acceptance function reduces to the threshold function

$$Y((i, j, c)) = \begin{cases} 1 & \text{if } c \geq D^*_{i,j}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Objective. This paper is concerned with the sample complexity, *i.e.*, the number of queries required for learning the unknown hemimetric D^*. We consider a PAC-style setting, *i.e.*, we aim to design an algorithm that, given positive constants (ϵ, δ), determines a hemimetric $\hat{D} \in \mathcal{D}$, such that with probability at least $1 - \delta$ it holds that

$$\|\hat{D} - D^*\|_\infty \leq \epsilon, \tag{6}$$

i.e., $|\hat{D}_{i,j} - D^*_{i,j}| \leq \epsilon \quad \forall\, i, j \in [n]$.

Our objective is to achieve the desired (ϵ, δ)-PAC guarantee while minimizing the number of user queries.

3 Warmup: Overview of our Approach

We now present the high-level ideas behind our approach.

3.1 Independent Learning: INDGREEDY

One possible way to tackle our learning problem is to learn each of the n^2 pairwise distances independently. Let us fix a particular pair of items $(i, j) \in [n]^2$. Given the query modality considered in our framework, the goal of learning the distance $D^*_{i,j}$ up to precsion ϵ is equivalent to learning a threshold function in the active-learning setting (Castro & Nowak, 2006; Settles, 2012). In terms of sample complexity, this is most effectively achieved by perfoming a binary-search over the range $[0, r]$. More formally, at iteration $t = 0$, we initialize a lower bound of $D^*_{i,j}$ to $L^t_{i,j} = 0$ and an upper bound to $U^t_{i,j} = r$. At any $t > 0$, we pick a value $c^t = \frac{1}{2}(L^{t-1}_{i,j} + U^{t-1}_{i,j})$ and issue the query $x^t = (i, j, c^t)$. Then, based on the returned label y^t, we update $U^t_{i,j} = c^t$ if $y^t = 1$, otherwise we update $L^t_{i,j} = c^t$ if $y^t = 0$. We continue querying until $(U^t_{i,j} - L^t_{i,j}) \leq \epsilon$, and then output any $\hat{D}_{i,j} \in [L^t_{i,j}, U^t_{i,j}]$ as the estimated distance. The number of queries required in the noise-free setting is given by $\lceil \log(\frac{r}{\epsilon}) \rceil$. As there are n^2 pairwise distance learning problems, the total sample complexity of this approach is $n^2 \lceil \log(\frac{r}{\epsilon}) \rceil$.

Such an algorithm, based on independently learning the pairwise distances, also needs to pick a pair (i^t, j^t) to query at iteration t. One policy inspired by uncertainty sampling (Settles, 2012) is to pick the pair (i^t, j^t) with maximum uncertainty quantified by $(U^{t-1}_{i,j} - L^{t-1}_{i,j})$. This policy can also be seen as to greedily minimize our objective stated in Equation 6. We call this query policy QGREEDY. At any iteration, it issues the query $x^t = (i^t, j^t, c^t)$ according to

$$(i^t, j^t) = \arg\max_{(i,j)\in[n]^2} (U^{t-1}_{i,j} - L^{t-1}_{i,j}), \text{ and} \tag{7}$$

$$c^t = \tfrac{1}{2}(L^{t-1}_{i^t,j^t} + U^{t-1}_{i^t,j^t}). \tag{8}$$

We refer to the independent learning algorithm employing query policy QGREEDY as INDGREEDY.

3.2 Exploiting Structural Constraints: LEARNHM

We now present our algorithm LEARNHM in Algorithm 1[2]. Our algorithm depends on three functions LU-PROJ, QCLIQUE and GETUSERRESPONSE. A high-level description of these functions is given as follows:

LU-PROJ shrinks the search space of hemimetrics by updating the lower and upper bounds from $\widetilde{L}^t, \widetilde{U}^t$ to L^t, U^t. Details are provided in Section 4.

QCLIQUE is the query policy that determines the next query $x^t = (i^t, j^t, c^t)$ at iteration t given the current state of the learning process as determined by L^{t-1} and U^{t-1}. Details are provided in Section 5.

GETUSERRESPONSE returns the label y^t for query x^t. In the deterministic noise-free setting, this label is determined by Equation 5. We also develop a robust noise-tolerant variant of GETUSERRESPONSE for the stochastic setting in the extended version of this paper (Singla et al., 2016).

4 LU-PROJ: Updating Bounds

We now present the details of our function LU-PROJ. All proofs are given in the extended version of this paper (Singla et al., 2016).

[2] Algorithm 1 reduces to INDGREEDY if QCLIQUE is replaced by QGREEDY and LU-PROJ$(\widetilde{L}^t, \widetilde{U}^t)$ simply returns $\widetilde{L}^t, \widetilde{U}^t$.

Algorithm 1 Our Algorithm: LEARNHM

Input: set $\mathcal{A}$ of n items, range r, error parameters (ϵ, δ)
Output: hemimetric $\hat{D}$
Initialize:
  iteration $t = 0$; labeled data $\mathcal{Z}^t = \emptyset$
  lower bounds: $L^t_{i,j} = 0 \,\forall\, i, j \in [n]$
  upper bounds: $U^t_{i,j} = r \,\forall\, i, j \in [n]$
  $U^t_{i,i} = 0 \,\forall\, i \in [n]$
while $\exists i, j : (U^t_{i,j} - L^t_{i,j}) > \epsilon$ **do**
  $t = t + 1$
  $x^t = (i^t, j^t, c^t) \leftarrow$ QCLIQUE(L^{t-1}, U^{t-1})
  $z^t = ((i^t, j^t, \bar{c}^t), y^t) \leftarrow$ GETUSERRESPONSE(x^t)
    // $\bar{c}^t = c^t$ in the noise-free setting
  $\widetilde{U}^t = U^{t-1}, \widetilde{L}^t = L^{t-1}$
  if $y^t = 1$ **then**
    update $\widetilde{U}^t_{i^t,j^t} = \bar{c}^t$
  else
    update $\widetilde{L}^t_{i^t,j^t} = \bar{c}^t$
  $L^t, U^t \leftarrow$ LU-PROJ$(\widetilde{L}^t, \widetilde{U}^t)$
  $\mathcal{Z}^t = \mathcal{Z}^{t-1} \cup \{z^t\}$
$\hat{D} \leftarrow U^t$
Return: hemimetric $\hat{D}$

4.1 Valid Bounds

We begin by defining minimal conditions for the lower and upper bounds returned by LU-PROJ to be *valid* in terms of the version space. Let us start by formally defining the version space for our setting. In Algorithm 1, the labeled data at iteration t is given by $\mathcal{Z}^t = \{z^1, \ldots, z^t\}$ where $z^l = (x^l, y^l)$ for $l \in [t]$. Then, the version space at iteration t is defined as

$$\mathcal{D}^t := \{D \in \mathcal{D} \mid \forall\, l \in [t] \colon D(x^l) = y^l\}, \tag{9}$$

where $D(x^l) = \mathbf{1}(c^l \geq D_{i^l,j^l})$; here $\mathbf{1}(\cdot)$ denotes the indicator function. That is, $\mathcal{D}^t \subseteq \mathcal{D}$ is the set of hemimetrics at iteration t that are consistent with the labeled data $\mathcal{Z}^t$.

Also, for given lower bounds L and upper bounds U, we define the set of hemimetrics satisfying these bounds as

$$\mathcal{D}(L, U) := \{D \in \mathcal{D} \mid L \leq D \leq U\}, \tag{10}$$

where the inequalities are understood component-wise, *i.e.*, $L_{i,j} \leq D_{i,j} \leq U_{i,j} \;\forall\, i, j \in [n]$. The bounds L^t, U^t at iteration t are *valid*, iff $\mathcal{D}(L^t, U^t) \supseteq \mathcal{D}^t$. Validity of the bounds ensures that D^* is always contained in $\mathcal{D}(L^t, U^t)$.

4.2 Updating Bounds via Projection

We formalize the problem of obtaining the bounds L^t, U^t as the solution of the following optimization problem:

$$\min_{U,L} \quad \|U - L\|_1 \tag{P1}$$
$$\text{s.t.} \quad \mathcal{D}(L, U) \supseteq \mathcal{D}^t,$$

where the entry-wise ℓ_1-norm of a matrix is defined as $\|M\|_1 = \sum_{i,j} |M_{i,j}|$. The intuitive idea behind this problem is to decrease the gap between upper and lower bounds as much as possible while ensuring that the resulting bounds are valid.

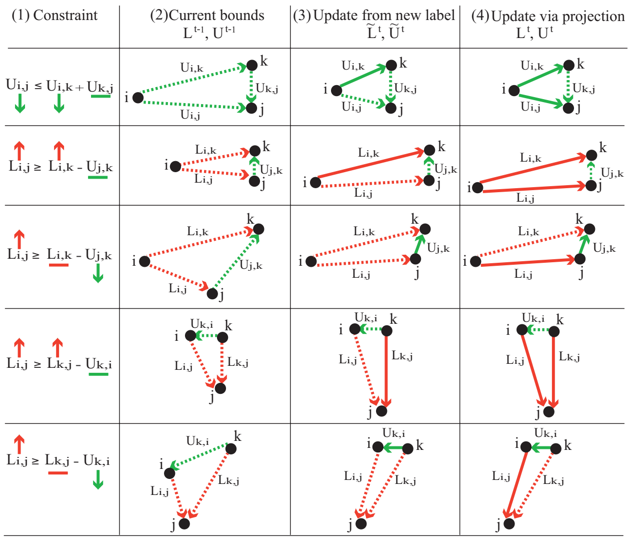


Figure 1. Geometric interpretation of the effect of exploiting the constraints in L-PROJ and U-PROJ— each row illustrates this for a particular constraint from the definitions of the sets $\mathcal{D}$ (Equation 4) and $\mathcal{L}$ (Equation 11). Column (1) shows the constraint, column (2) shows the current lower and upper bounds (L^{t-1}, U^{t-1} in Algorithm 1), column (3) shows an update from the new labeled datapoint ($\widetilde{L}^t, \widetilde{U}^t$ in Algorithm 1), and column (4) shows the effect of the corresponding constraint (L^t, U^t in Algorithm 1). For instance, in row 3, we consider the constraint $L_{i,j} \geq L_{i,k} - U_{j,k}$; after receiving a new label, $U_{j,k}$ decreases; this in turn can lead to an increase of $L_{i,j}$.

It turns out, *cf.* Theorem 1, that Problem P1 can be solved in a two step process by solving the following two problems:

$$U^{*t} = \underset{U \in \mathcal{D} \text{ s.t. } U \leq \widetilde{U}^t}{\arg\min} \|U - \widetilde{U}^t\|_1 \tag{P2}$$

$$L^{*t} = \underset{L \in \mathcal{L}(U^{*t}) \text{ s.t. } L \geq \widetilde{L}^t}{\arg\min} \|L - \widetilde{L}^t\|_1, \tag{P3}$$

where $\widetilde{L}^t, \widetilde{U}^t$ are obtained by Algorithm 1 in lines 8–12. Here, the set $\mathcal{L}$ parameterized by the upper bound matrix $U \in \mathcal{D}$ is defined as

$$\begin{aligned} \mathcal{L}(U) := \{L \mid \ \forall\, i,j,k \in [n]\colon L_{i,i} = 0,\ 0 \leq L_{i,j} \leq U_{i,j} \\ L_{i,j} \geq \max\left(L_{i,k} - U_{j,k}, L_{k,j} - U_{k,i}\right)\}. \end{aligned} \tag{11}$$

Additional details and a formal development of these sets are given in the extended version of this paper (Singla et al., 2016). While the set of upper bound matrices corresponds to the set $\mathcal{D}$ of bounded hemimetrics, the set $\mathcal{L}(U)$ represents more complex dependencies. It turns out that the set $\mathcal{L}$ cannot be constrained to contain only hemimetrics. In fact, we provide a counter-example in the extended version of this paper (Singla et al., 2016). We can now state one of our main theoretical results:

Theorem 1. *The optimal solution of Problem P1 is unique and is given by L^{*t}, U^{*t} (defined in Problems P3 and P2).*

4.3 Function LU-PROJ: Tightening Bounds

We now present an efficient solver for the optimization Problem P1 given by the function LU-PROJ in Algorithm 2. The algorithm is invoked with inputs $\widetilde{L}^t$, $\widetilde{U}^t$ — its optimality is ensured by the following theorem.

Algorithm 2 Updating Lower & Upper Bounds: LU-PROJ

1: **Input:** $\widetilde{L}^t, \widetilde{U}^t$; **Output:** L^t, U^t
2: $U^t \leftarrow$ U-PROJ$(\widetilde{U}^t)$
3: $L^t \leftarrow$ L-PROJ$(\widetilde{L}^t, U^t)$
4: **Return:** L^t, U^t

Algorithm 3 Updating Upper Bounds: U-PROJ

1: **Input:** $\widetilde{U}^t$; **Output:** U^t
2: **Initialize:** $U^t = \widetilde{U}^t$
3: **for** $k = 1$ **to** n **do**
4: **for** $i = 1$ **to** n **do**
5: **for** $j = 1$ **to** n **do**
6: $U^t_{i,j} = \min\left(U^t_{i,j}, U^t_{i,k} + U^t_{k,j}\right)$
7: **Return:** U^t

Algorithm 4 Updating Lower Bounds: L-PROJ

1: **Input:** $\widetilde{L}^t, U^t$; **Output:** L^t
2: **Initialize:** $L^t = \widetilde{L}^t$
3: **for** $k = 1$ **to** n **do**
4: **for** $i = 1$ **to** n **do**
5: **for** $j = 1$ **to** n **do**
6: $L^t_{i,j} = \max\left(L^t_{i,j}, L^t_{i,k} - U^t_{j,k}, L^t_{k,j} - U^t_{k,i}\right)$
7: **Return:** L^t

Theorem 2. *The lower and upper bounds L^t, U^t returned by* LU-PROJ *is the unique optimal solution of Problem P1.*

We now briefly describe the function LU-PROJ. It first invokes the function U-PROJ shown in Algorithm 3 to compute U^t — which in fact equals the optimal solution of Problem P2 (refer to the proof of the theorem). Then, it invokes the function L-PROJ shown in Algorithm 4 to compute L^t — which in fact equals the optimal solution of Problem P3 (again, refer to the proof of the theorem). Both U-PROJ and L-PROJ can be seen as iterating over the constraints of the sets $\mathcal{D}$ and $\mathcal{L}$, updating variables that violate constraints. U-PROJ is in fact equivalent to the Floyd-Warshall algorithm for solving the all-pair shortest paths problem in a graph (Floyd, 1962). Similar equivalence has been shown by Brickell et al. (2008) while studying the problem of projecting a non-metric matrix to a metric via decrease-only projections. The function L-PROJ operates in similar fashion as U-PROJ. However, additional challenges in the interpretation and analysis of the solution of L-PROJ arise from the fact that the class $\mathcal{L}$ is not a set of hemimetrics and has more complex dependencies.

Figure 1 provides a geometric interpretation of the constraints imposed by the sets $\mathcal{D}$ (Equation 4) and $\mathcal{L}$ (Equation 11) exploited by U-PROJ and L-PROJ in line 6.

4.4 Geometric Interpretation of L^{*t} and U^{*t}

In this section we provide a geometric interpretation of the optimal solution to Problem P1. Consider the space $\mathbb{R}^{n^2}$. Let us define π^0 to be the set of inequalities

$$\{D_{i,i} = 0, 0 \le D_{i,j} \le r,\\ D_{i,j} \le D_{i,k} + D_{k,j} \quad \forall\, i, j, k \in [n]\}.$$

Thus, the subset of $\mathbb{R}^{n^2}$ described by π^0 corresponds to the set of bounded hemimetrics. At iteration t, we get a new labeled datapoint $z^t = ((i^t, j^t, c^t), y^t)$ and update the set of inequalities as follows:

$$\pi^t = \begin{cases} \pi^{t-1} \cup \{c^t \ge D_{i^t,j^t}\} & \text{if } y^t = 1, \\ \pi^{t-1} \cup \{c^t < D_{i^t,j^t}\} & \text{if } y^t = 0. \end{cases}$$

Now, consider the polytope Λ^t defined by π^t in $\mathbb{R}^{n^2}$. Furthermore, consider the hypercube in $\mathbb{R}^{n^2}$ described by any lower and upper bounds L, U — the extent of the hypercube in dimension (i, j) is given by $[L_{i,j}, U_{i,j}]$. Then, the optimal solution to Problem P1 describes the *unique* tightest hypercube containing the polytope Λ^t.

5 QCLIQUE: Proposing Queries

We first show the limitations of a greedy myopic policy QGREEDY for proposing queries, and then design our query policy QCLIQUE that overcomes these limitations.

5.1 Myopic Policy QGREEDY

The policy QGREEDY is inspired by the idea of shrinking the gap $(U^{t-1}_{i,j} - L^{t-1}_{i,j})$ of pair (i^t, j^t) with maximum uncertainty. As already mentioned, this policy can be seen as myopic (greedy) in terms of minimizing the objective $\|\hat{D} - D^*\|_\infty$. However, this policy may turn out to be suboptimal in terms of exploiting the structural constraints of hemimetrics.

In particular, consider a simple instance with n items belonging to a single tight cluster, *i.e.*, $\forall i, j \in [n]\colon D^*_{i,j} = 0$. Clearly, given the $2n$ distances $D^*_{1,j}, D^*_{j,1}$ for $j \in [n]$, one can infer all other distances because of the tightness of the triangle inequalities. For this example, INDGREEDY will, in every iteration t, half the largest upper bound $\max_{i,j} U^t_{i,j}$. Thus, in iteration t, all upper bounds $U^t_{i,j}$ are in the range $[\alpha/2, \alpha]$ where $\alpha = r \cdot 2^{-\lfloor t/(n^2 - n) \rfloor}$ and all lower bounds $L^t_{i,j} = 0$. Here, invoking LU-PROJ cannot exploit the structural constraints, *i.e.*, it would simply return $(\widetilde{L}^t, \widetilde{U}^t)$.

5.2 Non-myopic Policy QCLIQUE

We now present an alternative policy QCLIQUE that proposes pairs (i^t, j^t) in a non-myopic way in Algorithm 5. Instead of greedily minimizing $\|\hat{D} - D^*\|_\infty$, we aim to learn distances in a systematic way such that we can exploit the structural constraints of the hemimetrics more effectively. Note that in this section, we assume a fixed ordering of the items indexed by $1, \ldots, n$.

The high level idea behind QCLIQUE is to maintain a clique of items for which all pairwise distances are already learnt. It proposes queries in a systematic way to grow the clique item by item according to the assumed ordering. In more detail, the policy works as follows:

Algorithm 5 Query Policy: QCLIQUE

1: **Input:** L^{t-1}, U^{t-1}; **Output:** query (i^t, j^t, c^t)
2: $\mathcal{C} = \{i \mid \forall j < i\colon U^{t-1}_{i,j} - L^{t-1}_{i,j} \le \epsilon \wedge U^{t-1}_{j,i} - L^{t-1}_{j,i} \le \epsilon\}$
3: $a = \max \mathcal{C} + 1$
4: $b =$ minimal $i \in \mathcal{C}$ s.t. $(U^{t-1}_{a,i} - L^{t-1}_{a,i} > \epsilon \vee U^{t-1}_{i,a} - L^{t-1}_{i,a} > \epsilon)$
5: **if** $U^{t-1}_{a,b} - L^{t-1}_{a,b} > \epsilon$ **then**
6:     **Return:** $(a, b, \frac{1}{2}(L_{a,b} + U_{a,b}))$
7: **else**
8:     **Return:** $(b, a, \frac{1}{2}(L_{b,a} + U_{b,a}))$

1. At iteration t, the policy maintains a clique $\mathcal{C} = \{1, \ldots, |\mathcal{C}|\} \subseteq \mathcal{A}$ of items — see line 2 of Algorithm 5. For any pair of items $(i, j) \in \mathcal{C}$ it holds that $U^{t-1}_{i,j} - L^{t-1}_{i,j} \le \epsilon$.
2. It then identifies the next item to be added to the clique, denoted as a in line 3.
3. Now, it picks an item $b \in \mathcal{C}$ for which the distance to a is not learnt up to precision ϵ in line 4, and returns the next query.

Online model. In many practical scenarios, the n items are not known beforehand, and rather appear over time. Let us denote by $\mathcal{A}^m = \{1, \ldots, m\} \subseteq \mathcal{A}$ the set of items present at time m. When a new item arrives at time $m + 1$ we could learn a hemimetric solution from scratch. However, it would be desirable to make use of the hemimetric solution for $\mathcal{A}^m$ and extend it. In fact, LEARNHM together with QCLIQUE can be readily applied to this scenario. We can identify the clique $\mathcal{C}$ with the itemset $\mathcal{A}^m$, where $m = |\mathcal{C}|$, for which the hemimetric solution is known. The idea of adding a to $\mathcal{C}$ in Algorithm 5, is then equivalent to extending the hemimetric solution for $\mathcal{A}^m$ to $\mathcal{A}^{m+1}$. By this equivalence, the sample complexity of growing the hemimetric solution item by item up to size n is the same as that of computing the hemimetric solution for all n items at once.

6 Performance Analysis

In this section we analyze the sample complexity and runtime of our proposed algorithm LEARNHM. All proofs are provided in the extended version of this paper (Singla et al., 2016).

6.1 Sample Complexity

Motivated by our preference elicitation application (see Section 7), we analyze sample complexity under a clusteredness assumption. In particular, we say the hypothesis D^* is (r_{in}, K)-clustered, if the following condition holds: The items are partitioned into K clusters, such that for any pair of items (i, j) their distance is $D^*_{i,j} \in [0, r_{\text{in}}]$ if i and j are from the same cluster and $D^*_{i,j} \in [r_{\text{in}}, r]$ otherwise. Note that K and r_{in} are *unknown* to the algorithm. For this setting, the sample complexity of LEARNHM is bounded by the following theorem.

Theorem 3. *If D^* is (r_{in}, K)-clustered, the sample complexity of* LEARNHM *is upper bounded by*

$$2nK\Big\lceil \log\Big(\frac{r}{\epsilon}\Big)\Big\rceil + n^2\Big\lceil \log\Big(\frac{2r_{\text{in}}+3\epsilon}{\epsilon}\Big)\Big\rceil.$$

In real-world applications, the distances $D^*_{i,j}$ might correspond to monetary incentives and are, therefore, naturally quantized to some precision Δ (monetary incentives are multiples of the smallest currency unit, *e.g.*, one cent). In this setting, the learning algorithms can learn D^* exactly, *i.e.*, $\hat{D} = D^*$, with a bounded number of queries. The idea is that both, INDGREEDY and LEARNHM, can collapse the gap $U^t_{i,j} - L^t_{i,j}$ to zero whenever $U^t_{i,j} - L^t_{i,j} < \Delta$. Hence, by invoking these algorithms with any $\epsilon < \Delta$, we learn D^* exactly. We then obtain the following corollary for this interesting special case.

Corollary 1. *If D^* is $(0, K)$-clustered, and assuming all distances $D^*_{i,j}$ are quantized to precision $\Delta > 0$, the sample complexity of* LEARNHM *to exactly learn D^* is upper bounded by $2nK\lceil \log\big(\frac{r}{\Delta}\big)\rceil$. This matches the lower bound of $\Omega(nK)$.*

Note that our algorithm LEARNHM does not perform more queries than INDGREEDY for any instance. In fact, the hardest instance for our algorithm is given by D^* where all distances $D^*_{i,j} = r/2$. In this case, LU-PROJ cannot exploit any structural constraints — the number of queries performed by LEARNHM exactly equals that of INDGREEDY (equal to $n^2\lceil\log(\frac{r}{\epsilon})\rceil$).

Stochastic responses. We can also bound the sample complexity for the more realistic case in which query responses are stochastic. Here, we briefly introduce our noise model — a more detailed description is given in the extended version of this paper (Singla et al., 2016). Our noise model is parametrized by the variance matrix $\sigma \in \mathbb{R}^{n,n}_+$ *unknown* to the algorithm. The acceptance function $\mathbb{P}(Y((i,j,c)) = 1)$ is given by the CDF of a normal distribution $\mathcal{N}(D^*_{i,j}, \sigma_{i,j})$ truncated to $[D^*_{i,j} - \beta_{i,j}, D^*_{i,j} + \beta_{i,j}]$ where $\beta_{i,j} = \min\{D^*_{i,j}, r - D^*_{i,j}\}$. Note that in our model the noise is unbounded, *i.e.*, at $c = D^*_{i,j}$ the acceptance function $\mathbb{P}(Y((i,j,c)) = 1) = 0.5$. In order to deal with this, we develop a robust noise-tolerant variant of GETUSERRESPONSE which ensures that the maximum noise experienced by the algorithm is bounded by

$$\eta_{max} = \frac{1}{2} - \frac{\int_0^{\epsilon/3} e^{-\frac{w^2}{2\max_{i,j}\sigma^2_{i,j}}}\mathrm{d}w}{\int_{-r/2}^{r/2} e^{-\frac{w^2}{2\max_{i,j}\sigma^2_{i,j}}}\mathrm{d}w}.$$

The sample complexity bounds are characterized by the quantity $\gamma = \big(\frac{3\ln(3n^2/\delta)}{(0.5-\eta_{\max})^2}\big)$ — the theoretical results corresponding to the settings in Theorem 3 and Corollary 1 are given as follows.

Theorem 4. *With probability $1-\delta$,* LEARNHM *learns D^* with precision ϵ and the sample complexity is upper bounded by*[3]

$$\widetilde{\mathcal{O}}\Big(\gamma\Big(2nK\Big\lceil \log\Big(\frac{3r}{\epsilon}\Big)\Big\rceil + n^2\Big\lceil \log\Big(\frac{6r_{\text{in}}+9\epsilon}{\epsilon}\Big)\Big\rceil\Big)\Big).$$

Corollary 2. *Consider the case D^* is $(0, K)$-clustered, and assume all distances $D^*_{i,j}$ are quantized to precision $\Delta > 0$. With probability $1-\delta$,* LEARNHM *learns D^* exactly and the sample complexity is upper bounded by*

$$\widetilde{\mathcal{O}}\Big(\gamma 2nK\Big\lceil \log\Big(\frac{3r}{\Delta}\Big)\Big\rceil\Big).$$

6.2 Runtime Analysis and Speeding Up LEARNHM

We now begin by analyzing the runtime of LEARNHM. The algorithm LEARNHM invokes LU-PROJ after every query — there are $\Theta(n^2\log(\frac{r}{\epsilon}))$ calls to LU-PROJ in the worst case. The runtime of LU-PROJ is $\Theta(n^3)$, resulting in a total runtime $\Theta(n^5\log(\frac{r}{\epsilon}))$ — this is prohibitively expensive for most realistic problem instances.

The key idea for speeding up LEARNHM is that we can choose the constraints that should be exploited instead of exploiting all the constraints after every query (line 6 in U-PROJ & L-PROJ). If the constraints to be exploited are selected carefully, we can still get reasonable benefits from tightening lower and upper bounds. For LEARNHM, this can be achieved as follows:

- First, we do not need to invoke LU-PROJ after every query (this is equivalent to not exploiting any violated constraint). At any iteration t, LEARNHM invokes LU-PROJ only if $\widetilde{U}^t_{i^t,j^t} - \widetilde{L}^t_{i^t,j^t} \leq \epsilon$. Otherwise, it simply sets $(L^t, U^t) \leftarrow (\widetilde{L}^t, \widetilde{U}^t)$.
- Second, when LEARNHM invokes LU-PROJ at iteration t we only consider the $2n$ constraints which involve i^t, j^t in lines 4 and 5 of Algorithms 3 and 4.

Details of this idea are presented in the extended version of this paper (Singla et al., 2016). LEARNHM implementing this idea invokes LU-PROJ at most n^2 times each with a runtime of $4n$. Hence, the total runtime of the speeded up LEARNHM is $\Theta(n^3)$. Most importantly, the sample complexity bounds from the previous section still apply.

7 Experimental Evaluation

7.1 Benchmarks

We compare the performance of the speeded up LEARNHM against the baseline INDGREEDY. We also compare against a second baseline INDGREEDY-SIT (INDGREEDY with side information of triplet comparisons). This baseline utilizes a low-dimensional embedding of the items as a preprocessing step. Following the work of Jamieson & Nowak (2011), using $n^2\log n$ triplet queries, *i.e.*, for a triplet (i,j,k) such a query returns $\mathbf{1}(D^*_{i,j} \leq D^*_{i,k})$, one can

[3]The $\widetilde{\mathcal{O}}(\cdot)$ notation is used to omit factors logarithmic in the factors present explicitly.

compute a low-dimensional embedding of the items. Using this embedding, one can infer the response to all possible n^3 triplet queries — this is the side information that we supply to INDGREEDY-SIT.

This side information can be exploited by INDGREEDY-SIT as follows. For a given triplet, $\mathbf{1}(D^*_{i,j} \leq D^*_{i,k}) = 1$ implies two constraints on the lower and the upper bounds — $U_{i,j} \leq U_{i,k}$ and $L_{i,k} \geq L_{i,j}$. After every query, INDGREEDY-SIT first updates an upper or lower bound according to the response. Then it exploits these two constraints for n^3 triplets to tighten the bounds on every pair of items. We will report the sample complexity of INDGREEDY-SIT as the total number of queries for computing the low-dimensional embedding and for learning all distances up to precision ϵ.

7.2 Experimental Setup

Yelp dataset. We use the recently proposed *Yelp Dataset Challenge (round 7)* data for our experiments.[4] This data contains information about 77K businesses located across 10 cities around the world. We looked into businesses belonging to the category *Restaurants* and being located in the city of *Pittsburgh, PA*. In particular, we extracted information for all 290 restaurants offering food from the cuisines Mexican (50), Thai (26), Chinese (53), Mediterranean (75), Italian (86). For each of these restaurants we also collected the review count and coordinates (longitude and latitude). We discretized the review count into *High* (popular, 166 restaurants) when there were more than 25 reviews and into *Low* (unpopular, 124 restaurants) otherwise. The collected data is visualized in Figure 3.

User preference models. We simulate user preference models from this data by creating the underlying hemimetric D^* as follows. For notational ease, we use the shorthands cuisine$_i$, review$_i$, lat$_i$, long$_i$ to refer to the above mentioned attributes for item i. We quantify the distance between item i and j by

$$W_{i,j} = w_1 W^{\text{cuisine}}_{i,j} + w_2 W^{\text{review}}_{i,j} + w_3 W^{\text{geo}}_{i,j} + w_4 W^{\text{random}}_{i,j},$$

where

$$W^{\text{cuisine}}_{i,j} = r\mathbf{1}(\text{cuisine}_i \neq \text{cuisine}_j),$$
$$W^{\text{review}}_{i,j} = r\mathbf{1}(\text{review}_i > \text{review}_j),$$

$W^{\text{geo}}_{i,j}$ is the great-circle distance based on the latitude and longitude coordinates normalized to lie in $[0, r]$, and $W^{\text{random}}_{i,j}$ is drawn uniformly at random from $[0, r]$. Recall, r is the upper bound on the distance. The weights $w_1, \ldots, w_4 \in \mathbb{R}_+$ sum up to 1. We compute D^* as the closest metric to W according to Brickell et al. (2008).

For different weights $w_1, \ldots, w_4$, we can instantiate different user preference models D^*. In particular, we instantiate the following two models. In the first model (*YelpM*^{1}), we use $w_1 = 0.9, w_4 = 0.1$ and $w_2 = w_3 = 0$ — this corresponds to the setting we considered in Theorem 3 with $K = 5$ and intra cluster distance $r_{\text{in}} \leq w_4 \cdot r$. The second model (*YelpM*^{2}) is more generic, with weights given by $w_1 = 0.5, w_2 = w_3 = 0.2, w_4 = 0.1$.

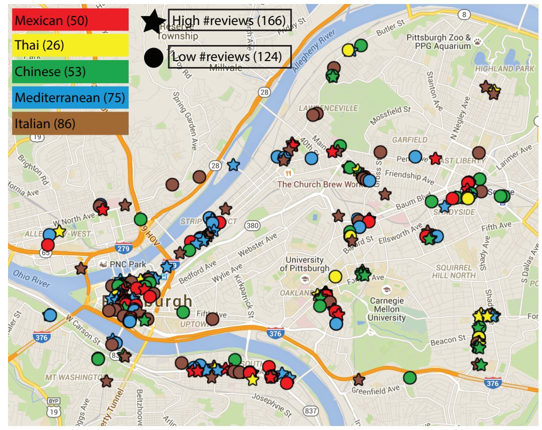


Figure 3. Visualization of the 290 restaurants from the *Yelp* data in Pittsburgh. The restaurants are distinguished by cuisine (color), popularity (star) and location (coordinates).

7.3 Results

We now present our results. We focus on the noise-free setting here, results for the stochastic setting are provided in the extended version of this paper (Singla et al., 2016). As the metric for comparing LEARNHM, INDGREEDY and INDGREEDY-SIT, we use the sample complexity of learning the unknown hemimetric D^* up to precision ϵ. In the experiments we used $r = 1$, and varied ϵ and n. When varying n (by sub-sampling), we used $\epsilon = 0.01$. Results are shown in Figures 2 (a-c).

In Figure 2a, for the first model *YelpM*^{1}, we can observe that the sample complexity of LEARNHM, *cf.* Theorem 3, is almost an order of magnitude smaller than that of INDGREEDY. Note that INDGREEDY-SIT also has lower sample complexity than INDGREEDY. However, because of the large cost for computing an embedding, INDGREEDY-SIT performs worse than LEARNHM.

The results for the more generic model *YelpM*^{2} without specific clusters are shown in Figure 2b. We observe that LEARNHM and INDGREEDY-SIT have higher sample complexity compared to the first model *YelpM*^{1}. The sample complexity of LEARNHM is still better by a factor of $\sim$2 than that of the baseline algorithms — although the assumptions in Theorem 3 do not hold.

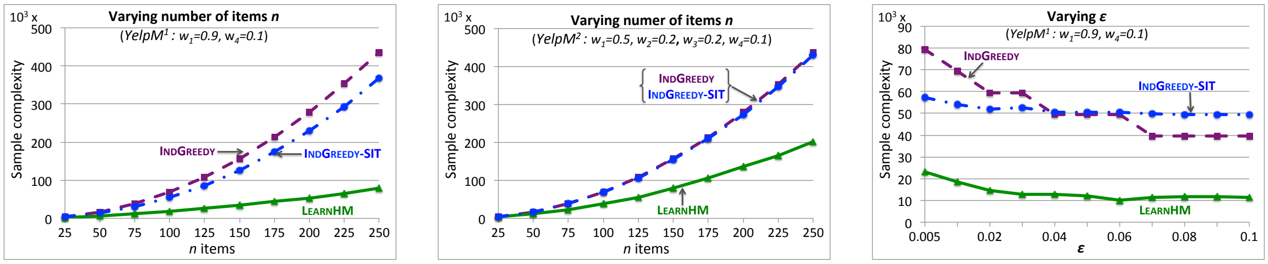
Finally, Figure 2c shows results for varying ϵ for $n = 100$ items. As expected, with increasing ϵ the sample complexity of all algorithms increases. For larger values of ϵ, INDGREEDY-SIT performs worse than INDGREEDY as the sample complexity for computing the embedding dominates the learning. Most importantly, LEARNHM has the lowest sample complexity of all three algorithms.

8 Related Work

8.1 Metric Learning & Low-dimensional Embeddings

Learning distances to capture notions of similarity or dissimilarity plays a central role in many machine learning

[4] https://www.yelp.com/dataset_challenge/



(a) Varying n (*YelpM*^{1}) (b) Varying n (*YelpM*^{2}) (c) Varying ϵ (*YelpM*^{1})

Figure 2. Sample complexity results for two different user preference models defined via the *Yelp* dataset. For the first model corresponding to the $K = 5$ cluster setting, *cf.* Theorem 3, the sample complexity of LEARNHM is about an order of magnitude smaller than that of INDGREEDY. Even for the generic model, LEARNHM substantially outperforms the two baselines.

applications. We refer the reader to the detailed surveys by Yang & Jin (2006); van der Maaten et al. (2009); Bellet et al. (2013). Some of the key distinctions of our formulation from the existing research are described in the following.

Supervised metric learning. In their seminal work, Xing et al. (2002) introduced the supervised metric learning framework for learning Mahalanobis distance functions via a convex formulation. In contrast to our setting, they assume that the algorithm has access to the feature space of the input data. Furthermore, this framework and its variants are restricted to recover symmetric distance functions. Another line of research considers learning asymmetric distances, for instance by learning local invariant Mahalanobis distances (Fetaya & Ullman, 2015) or by learning general bilinear similarity matrices (Liu et al., 2015). However, this line of work is not directly applicable to our setting because it also requires access to the feature space of the input data.

Learning embeddings. Another line of research is that of learning low-dimensional embeddings for a set of items respecting the observed geometric relations between these items (Amid & Ukkonen, 2015; Tamuz et al., 2011; Cox & Cox, 2000; Jamieson & Nowak, 2011). For applications involving human subjects, a triplet-based queries framework — for a triplet (i, j, k) it queries $\mathbf{1}(D_{i,j} \leq D_{i,k})$ — has been employed. However the distances recovered from these approaches are merely optimized to respect the observed relations seen in the data from query responses — they are symmetric and importantly *do not* have an actual quantitative (economic) interpretation as we seek in our formulation.

8.2 Exploiting Structural Constraints

Our approach of exploiting the structural constraints of the hemimetrics polytope is in part inspired by Elkan (2003), who accelerates K-means by exploiting the triangle inequality. By maintaining bounds on the distances, he efficiently reduces the number of distance computations. Brickell et al. (2008) study the problem of projecting a non-metric matrix to a metric matrix, and consider a specific class of decrease-only projections. Our approach towards updating upper bounds via decrease-only projections is similar in spirit. However, the main technical difficulties arise in maintaining and updating the lower bounds, for which we develop novel techniques. The active-learning approach proposed by Jamieson & Nowak (2011) exploits the geometry of the embedding space to minimize the sample complexity using triplet-based queries. While similar in spirit, their approach is based on triplet-based queries, and differs from our methodology of exploiting the structural constraints.

8.3 Learning User Preferences

Another relevant line of research is concerned with eliciting user preferences. Specifically, we seek to learn private costs of a user for switching from her default choice of item i to instead choose item j. This type of preferences can be used in marketing applications, *e.g.*, for persuading users to change their decisions (Kamenica & Gentzkow, 2009). Singla et al. (2015) considered similar preferences in the context of balancing a bike-sharing system by incentivizing users to go to alternate stations for pickup or dropoff of bikes. Abernethy et al. (2015) considered the application of purchasing data from users, and quantified the prices that should be offered to them. One key difference of our approach is that we are interested in jointly learning the preferences between n items, *i.e.*, tackling n^2 learning problems jointly.

8.4 Active Learning

Our problem formulation shares the goal of reducing sample complexity with other instances of active learning (Settles, 2012). Our goal to specifically exploit the structural constraints of the hemimetric polytope is along the lines of research in structured active learning where the hypotheses or the output label space have some inherent structure that can be utilized, *e.g.*, the structure of part-of-speech tagging of the sentence (Roth & Small, 2006).

9 Conclusions

We investigated the novel problem of actively learning hemimetrics. The two key techniques used in the construction of our algorithm LEARNHM are novel projection techniques for tightening the lower and upper bounds on the solution space and a non-myopic (non-greedy) query policy. Our algorithm can be readily applied to the online setting allowing one to extend the hemimetric solution over time. We provided a thorough analysis of the sample complexity and runtime of our algorithm. Our experiments on *Yelp* data showed substantial improvements over baseline algorithms in line with our theoretical findings.

Acknowledgements

This research is supported in part by SNSF grant 200020_159557 and the Nano-Tera.ch program as part of the Opensense II project.

References

Abernethy, J., Chen, Y., Ho, C., and Waggoner, B. Low-cost learning via active data procurement. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pp. 619–636, 2015.

Amid, E. and Ukkonen, A. Multiview triplet embedding: Learning attributes in multiple maps. In *ICML*, pp. 1472–1480, 2015.

Bellet, A., Habrard, A., and Sebban, M. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

Blum, A., Mansour, Y., and Morgenstern, J. Learning what's going on: Reconstructing preferences and priorities from opaque transactions. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pp. 601–618, 2015.

Brickell, J., Dhillon, I. S., Sra, S., and Tropp, J. A. The metric nearness problem. *SIAM Journal on Matrix Analysis and Applications*, 30(1):375–396, 2008.

Castro, Rui M and Nowak, Robert D. Upper and lower error bounds for active learning. In *Allerton*, 2006.

Cox, T. F. and Cox, M. AA. *Multidimensional scaling*. CRC Press, 2000.

Desjardins, M., Eaton, E., and Wagstaff, K. L. Learning user preferences for sets of objects. In *ICML*, pp. 273–280, 2006.

Elkan, C. Using the triangle inequality to accelerate k-means. In *ICML*, volume 3, pp. 147–153, 2003.

Fetaya, E. and Ullman, S. Learning local invariant mahalanobis distances. In *ICML*, 2015.

Floyd, Robert W. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

He, J., Li, M., Zhang, H., Tong, H., and Zhang, C. Manifold-ranking based image retrieval. In *Proceedings of the 12th annual ACM International Conference on Multimedia*, pp. 9–16, 2004.

Horton, J. J. and Johari, R. At what quality and what price?: Eliciting buyer preferences as a market design problem. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pp. 507–507, 2015.

Huang, Z., Wang, R., Shan, S., Li, X., and Chen, X. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, 2015.

Jamieson, K. G. and Nowak, R. D. Low-dimensional embedding using adaptively selected ordinal data. In *Allerton*, pp. 1077–1084, 2011.

Kamenica, E. and Gentzkow, M. Bayesian persuasion. Technical report, National Bureau of Economic Research, 2009.

Lim, D. and Lanckriet, G. Efficient learning of mahalanobis metrics for ranking. In *ICML*, pp. 1980–1988, 2014.

Liu, Kuan, Bellet, Aurélien, and Sha, Fei. Similarity learning for high-dimensional sparse data. In *AISTATS*, pp. 653–662, 2015.

Roth, Dan and Small, Kevin. Margin-based active learning for structured output spaces. In *ECML*, pp. 413–424. Springer, 2006.

Settles, B. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

Singla, A. and Krause, A. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *WWW*, pp. 1167–1178, 2013.

Singla, A., Santoni, M., Bartók, G., Mukerji, P., Meenen, M., and Krause, A. Incentivizing users for balancing bike sharing systems. In *AAAI*, 2015.

Singla, A., Tschiatschek, S., and Krause, A. Actively learning hemimetrics with applications to eliciting user preferences (extended version). *CoRR*, abs/1605.07144, 2016. URL http://arxiv.org/abs/1605.07144.

Tamuz, O., Liu, C., Belongie, S., Shamir, O., and Kalai, A. T. Adaptively learning the crowd kernel. In *ICML*, 2011.

van der Maaten, Laurens JP, Postma, Eric O, and van den Herik, H Jaap. Dimensionality reduction: A comparative review. *JMLR*, 10(1-41):66–71, 2009.

Vázquez-Gallo, M., Estévez, M., and Egido, S. Active learning and dynamic pricing policies. *American Journal of Operations Research*, 2014, 2014.

Xing, E. P., Jordan, M. I., Russell, S., and Ng, A. Y. Distance metric learning with application to clustering with side-information. In *NIPS*, pp. 505–512, 2002.

Yang, Liu and Jin, Rong. Distance metric learning: A comprehensive survey. *Michigan State Universiy*, 2, 2006.