
Simultaneous Preference and Metric Learning from Paired Comparisons

Austin Xu

Georgia Institute of Technology
Atlanta, GA 30332
axu@gatech.edu

Mark Davenport

Georgia Institute of Technology
Atlanta, GA 30332
mdav@gatech.edu

Abstract

A popular model of preference in the context of recommendation systems is the so-called *ideal point* model. In this model, a user is represented as a vector \mathbf{u} together with a collection of items $\mathbf{x}_1, \dots, \mathbf{x}_N$ in a common low-dimensional space. The vector \mathbf{u} represents the user’s “ideal point,” or the ideal combination of features that represents a hypothesized most preferred item. The underlying assumption in this model is that a smaller distance between \mathbf{u} and an item \mathbf{x}_j indicates a stronger preference for \mathbf{x}_j . In the vast majority of the existing work on learning ideal point models, the underlying distance has been assumed to be Euclidean. However, this eliminates any possibility of interactions between features and a user’s underlying preferences. In this paper, we consider the problem of learning an ideal point representation of a user’s preferences when the distance metric is an unknown Mahalanobis metric. Specifically, we present a novel approach to estimate the user’s ideal point \mathbf{u} and the Mahalanobis metric from paired comparisons of the form “item \mathbf{x}_i is preferred to item \mathbf{x}_j .” This can be viewed as a special case of a more general metric learning problem where the location of some points are unknown *a priori*. We conduct extensive experiments on synthetic and real-world datasets to exhibit the effectiveness of our algorithm.

1 Introduction

Personalized recommendation and ranking algorithms have become increasingly important in recent years, influencing not only the items a user buys and movies he or she watches, but also potentially influencing which job candidates are interviewed, which college applicants are admitted, and even the matching behavior of online dating services. While there are a number of approaches to developing personalized recommendation systems, a particularly common approach uses a classical model for user preference known as the *ideal point model* [1]. In this model a user’s preferences are represented as a point $\mathbf{u} \in \mathbb{R}^D$ that is embedded in the same space as a set of items $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^D$ (movies, shoes, food, etc). The key model assumption is that the closer an item \mathbf{x}_j is to \mathbf{u} , the more the user will prefer item \mathbf{x}_j . We note that the ideal point is not necessarily a specific item \mathbf{x}_i , but rather represents the combination of features that the user most prefers. The ideal point model is an intuitive and interpretable way to model preferences and has been empirically shown to exhibit superior performance compared to other models of preference [2, 3].

In a practical system, the main challenge is to learn the latent \mathbf{u} that represents a particular user’s preferences. Given a precise quantification of a user’s preferences for a number of items, one could infer the distances from \mathbf{u} to those items and then easily estimate a good embedding \mathbf{u} . In practice, however, users find *paired comparison* queries of the form “do you prefer item \mathbf{x}_i or item \mathbf{x}_j ” to be far easier to answer [4, 5]. As a result, a number of approaches to learning to rank from such paired comparisons have been proposed in recent years [6–16]. In the specific context of ideal point models, such queries allow the user to reveal which of the two items is closer to their ideal point. There is

now a range of both practical algorithms for estimating \mathbf{u} from such queries as well as theoretical treatments analyzing the performance of these algorithms in terms of error bounds and/or sample complexity guarantees [17, 9, 18-26].

While the problem of learning from paired comparisons in the ideal point setting is now well-understood, the vast majority of past work has only examined the case where the user makes judgements under the standard Euclidean distance metric. Assuming a Euclidean metric imposes two main limitations. First, it does not allow for features to interact. In practice, features often complement or compensate for each other. For example, consider the process of purchasing shoes. Each shoe can be described in terms of features such as color, price, materials, etc. An individual may prefer a cost of \$50 and a particular material. However, if the price was set instead to \$200, the user’s preferred material may change to reflect the change in price – an effect that cannot be accommodated by a Euclidean (isotropic) metric. Second, the Euclidean metric assumes that all features are of equal importance to the user, which is often not the case. In the shoe purchasing example, a price conscious consumer may prioritize finding the best “bang for their buck,” in which case a lower price and higher quality of material would be prioritized over aesthetic features such as color.

To overcome these limitations, we consider the case where the user makes comparisons between items under a *Mahalanobis distance*. Specifically, let $\mathbf{M} \in \mathbb{R}^{D \times D}$ be a symmetric positive definite matrix and set $\|\mathbf{x}\|_{\mathbf{M}} = \sqrt{\mathbf{x}^T \mathbf{M} \mathbf{x}}$. Then $\|\mathbf{x} - \mathbf{y}\|_{\mathbf{M}}$ defines a Mahalanobis distance between \mathbf{x} and \mathbf{y} . This metric captures both feature interactions and the relative significance of those feature interactions via the eigenvalue decomposition $\mathbf{M} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$. The eigenvectors specify how features can interact to jointly affect preferences, and the eigenvalues allow for different combinations of features to play a larger or smaller role. See Section 4.2 below for a concrete example.

While a Mahalanobis metric allows for more flexible and powerful models of preference, the appropriate choice of \mathbf{M} will in general be unknown *a priori*. In this paper, we develop a novel method to jointly learn both the ideal point and Mahalanobis metric from paired comparisons, which to the best of our knowledge represents the first approach for solving these problems simultaneously. By leveraging the structure of paired comparisons, we develop a simple convex optimization program that estimates \mathbf{M} and can then directly solve for \mathbf{u} . In the process, we also effectively learn the user’s ranking of the items. We also explore the possible benefits of a more sophisticated alternating scheme that iteratively refines the estimates of \mathbf{M} and \mathbf{u} . We demonstrate the effectiveness of our approach through experiments on both synthetic and real-world datasets.

2 Related Work

Our work naturally builds on the existing literature on learning from paired comparisons, taking particular inspiration from the convex optimization approach to non-metric multidimensional scaling of [17] and the approaches in [9, 18, 21, 26] to developing algorithms for the ideal point setting. We also build on the extensive prior work on *metric learning*. Learning a metric from paired comparisons was introduced in [27], where the authors assume the distance is parametrized by a known matrix \mathbf{A} and a weighting matrix \mathbf{W} with non-negative diagonal entries. \mathbf{W} is learned via a convex program by manipulating the form of diagonal matrix multiplication. Setting \mathbf{A} to \mathbf{I} does not allow for feature interactions, whereas picking more complex \mathbf{A} without overfitting the training data is non-trivial. Similarly in [28], the authors minimize the squared-hinge loss of differences in distances of pairs of items. However, the user is presented with two pairs of items must pick which pair of items is more similar, which is a more complex querying scheme for the user to answer. The same query type is used to learn a metric for images in [29]. The problem of learning low-rank/sparse metrics where all points are known was studied in [30], which established bounds for sample complexity, generalization error, and model identification. The performance of nearest-neighbor-based classifiers have also benefited from learning a Mahalanobis metric that enhances class separation [31], where here class membership is used to inform the learning process.

Metric learning has also been explored in prior work on recommendation systems. For example, [32], [33], and [34] all learn Mahalanobis metrics for ranking given a known reference point and sets of similar and dissimilar items. Using sets of positive and negative items for each user, [35] learns a personalized projection operator for each user and estimates user preference in the learned latent space. In a similar setting, [36] learns transformed ideal points and items directly before learning a metric. Finally, [37] assumes each user has an ideal feature vector where user preference is measured by the inner product of this ideal feature vector with an item’s feature vector and develops a feature

selection scheme to account for intransitivity in noisy comparison outcomes while learning an ideal feature vector.

Our work differs from the above in that it uniquely assumes that *both* the metric and ideal point are unknown. Thus, it can be viewed as a more generalized metric learning problem where some of the data are missing. Most existing metric learning papers avoid the problem of knowing a user’s preference by assuming a known reference point or utilizing more difficult queries (asking the user to compare two pairs of items). Based on this prior work, it might be unclear if simultaneous recovery of an unknown metric and ideal point is even feasible, but we show that it is indeed possible.

3 Observation Model and Estimation Strategy

3.1 Observation model

For simplicity, we will begin by considering the noiseless observation model where the user always prefers the item closest to the user’s ideal point \mathbf{u} under the Mahalanobis distance metric induced by M , where M is a symmetric positive definite matrix. For a discussion on how to accommodate noisy paired comparison outcomes, please see the supplementary material. To be more concrete, we let $\mathbf{d} \in \mathbb{R}^N$ be the vector with entries $d_i = \|\mathbf{x}_i - \mathbf{u}\|_M^2$. We let $\mathbf{y} \in \mathbb{R}^P$ denote our observations, where the k^{th} element of \mathbf{y} denotes the outcome of the k^{th} comparison (between items \mathbf{x}_{i_k} and \mathbf{x}_{j_k}) and is given by

$$y_k = \text{sign}(d_{i_k} - d_{j_k}). \quad (1)$$

For now we assume that the set of indices $\Omega = \{(i_1, j_1), \dots, (i_P, j_P)\}$ corresponding to the items compared contains each pair of indices at most once, although our methods could easily be adapted to the case where Ω is a multiset. We will assume throughout our treatment that the embedding of the items $\mathbf{x}_1, \dots, \mathbf{x}_N$ is fixed and known, as in a mature recommendation system. This embedding may correspond to known and interpretable features or be learned from other side information (or even paired comparisons, following a strategy along the lines of [38]).

Before we describe our estimation strategy, a few observations are in order. First, note that \mathbf{y} consists of (1-bit) quantized samples of the $N \times N$ matrix $\Delta = \mathbf{d}\mathbf{1}_N^T - \mathbf{1}_N\mathbf{d}^T$, where $\mathbf{1}_N \in \mathbb{R}^N$ is the vector of all ones. It will also be useful to work with the vectorized version of Δ , which we will denote by δ and can be written as¹

$$\delta = (\mathbf{1}_N \otimes \mathbf{I}_N - \mathbf{I}_N \otimes \mathbf{1}_N)\mathbf{d},$$

where \mathbf{I}_N denotes the $N \times N$ identity matrix and \otimes denotes the Kronecker product. For conciseness, we will let $\mathbf{Q} = \mathbf{1}_N \otimes \mathbf{I}_N - \mathbf{I}_N \otimes \mathbf{1}_N$.

To index into δ , we map every $(i_k, j_k) \in \Omega$ to a linear index between 1 and N^2 defined as $\Gamma = \{(i_k - 1)N + j_k : (i_k, j_k) \in \Omega\}$. We can equivalently write our observation model in (1) as

$$\mathbf{y} = \text{sign}(\delta_\Gamma) = \text{sign}(\mathbf{Q}_\Gamma \mathbf{d}), \quad (2)$$

where the notation δ_Γ and \mathbf{Q}_Γ indicates the vector or matrix obtained by selecting only the indices/rows indexed by Γ .

3.2 Estimation from unquantized observations

To gain some insight into this problem, we will temporarily ignore the quantization and suppose that we have direct access to δ_Γ – in this case, how might we go about estimating M and \mathbf{u} ?

Consider $d_{i_k} = \|\mathbf{x}_{i_k} - \mathbf{u}\|_M^2$ and $d_{j_k} = \|\mathbf{x}_{j_k} - \mathbf{u}\|_M^2$ for any $(i_k, j_k) \in \Omega$. Then, for the linear index p corresponding to (i_k, j_k) , $\delta_p = \|\mathbf{x}_{i_k} - \mathbf{u}\|_M^2 - \|\mathbf{x}_{j_k} - \mathbf{u}\|_M^2$. Observe that when we expand these terms we can cancel the coupled term $\mathbf{u}^T M \mathbf{u}$, greatly simplifying our subsequent analysis:

$$\begin{aligned} \delta_p &= \|\mathbf{x}_{i_k} - \mathbf{u}\|_M^2 - \|\mathbf{x}_{j_k} - \mathbf{u}\|_M^2 \\ &= \mathbf{x}_{i_k}^T M \mathbf{x}_{i_k} - 2\mathbf{x}_{i_k}^T M \mathbf{u} + \mathbf{u}^T M \mathbf{u} - (\mathbf{x}_{j_k}^T M \mathbf{x}_{j_k} - 2\mathbf{x}_{j_k}^T M \mathbf{u} + \mathbf{u}^T M \mathbf{u}) \\ &= \mathbf{x}_{i_k}^T M \mathbf{x}_{i_k} - \mathbf{x}_{j_k}^T M \mathbf{x}_{j_k} - 2(\mathbf{x}_{i_k} - \mathbf{x}_{j_k})^T M \mathbf{u} \end{aligned} \quad (3)$$

¹Note that this follows from the general identity $\text{vec}(\mathbf{A}\mathbf{B}\mathbf{C}) = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}(\mathbf{B})$.

If we define

$$\mathbf{R} = \begin{bmatrix} - & (\mathbf{x}_{i_1} - \mathbf{x}_{j_1})^T & - \\ - & (\mathbf{x}_{i_2} - \mathbf{x}_{j_2})^T & - \\ & \vdots & \\ - & (\mathbf{x}_{i_P} - \mathbf{x}_{j_P})^T & - \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} - & (\mathbf{x}_{i_1} + \mathbf{x}_{j_1})^T & - \\ - & (\mathbf{x}_{i_2} + \mathbf{x}_{j_2})^T & - \\ & \vdots & \\ - & (\mathbf{x}_{i_P} + \mathbf{x}_{j_P})^T & - \end{bmatrix}$$

then we can write (3) more concisely as

$$\delta_\Gamma = \text{diag}(\mathbf{S}\mathbf{M}\mathbf{R}^T) - 2\mathbf{R}\mathbf{M}\mathbf{u}, \quad (4)$$

where $\text{diag}(\mathbf{A})$ returns the diagonal of $\mathbf{A} \in \mathbb{R}^{N \times N}$ as a column vector. For brevity, let $\mathbf{a}_M = \text{diag}(\mathbf{S}\mathbf{M}\mathbf{R}^T)$. We now observe that if we observed δ_Γ directly *and* already knew \mathbf{M} , then we could estimate \mathbf{u} by solving a standard least squares problem, resulting in the estimate

$$\hat{\mathbf{u}} = \frac{1}{2}\mathbf{M}^\dagger\mathbf{R}^\dagger(\mathbf{a}_M - \delta_\Gamma). \quad (5)$$

Plugging this estimate into (4), we obtain a simple system of equations that is linear in \mathbf{M} :

$$\begin{aligned} \delta_\Gamma &= \mathbf{a}_M - 2\mathbf{R}\mathbf{M}\left(\frac{1}{2}\mathbf{M}^\dagger\mathbf{R}^\dagger\right)(\mathbf{a}_M - \delta_\Gamma) \\ &= \mathbf{a}_M - \mathbf{R}\mathbf{M}\mathbf{M}^\dagger\mathbf{R}^\dagger(\mathbf{a}_M - \delta_\Gamma) \\ &= \mathbf{a}_M - \mathbf{R}\mathbf{R}^\dagger(\mathbf{a}_M - \delta_\Gamma), \end{aligned}$$

where the last equality follows from the fact that \mathbf{M} is assumed to be positive definite (and hence full-rank). Rearranging terms, we obtain the more convenient expression

$$\mathbf{0} = (\mathbf{I} - \mathbf{R}\mathbf{R}^\dagger)(\mathbf{a}_M - \delta_\Gamma). \quad (6)$$

3.3 Single-step estimation from quantized observations

Given direct observations of δ_Γ , we could immediately estimate \mathbf{M} using (6). However, our observations as in (2) are (1-bit) quantized. In this case, using (6), we can instead formulate a constrained optimization problem to jointly estimate a \mathbf{M} and a set of distances \mathbf{d} (and hence δ_Γ) that are consistent with both our observations \mathbf{y} and (6). Specifically, we will aim to find a solution that satisfies (6) while minimizing $\ell(\mathbf{d})$, where $\ell(\mathbf{d})$ is a loss function that encourages \mathbf{d} to be such that the signs of entries of $\delta_\Gamma = \mathbf{Q}_\Gamma\hat{\mathbf{d}}$ are consistent with the observed comparisons. For example, one could set $\ell(\mathbf{d})$ to be the *hinge loss*:

$$\ell(\mathbf{d}) = \sum_{k=1}^P \max(0, 1 - y_k(\mathbf{Q}_\Gamma\mathbf{d})_k), \quad (7)$$

where $(\mathbf{Q}_\Gamma\mathbf{d})_k$ denotes the k^{th} element in the vector $\mathbf{Q}_\Gamma\mathbf{d}$. In the remainder of this paper and in our experiments, we use the hinge loss, but our framework could easily be extended to accommodate any convex loss. Finally, in our proposed approach we also introduce slack variables $\zeta \in \mathbb{R}^P$ to loosen the constraint (6) to improve stability and robustness to noise, and also introduce terms to the objective function to allow for a small amount of regularization on both \mathbf{M} and \mathbf{d} :

$$\begin{aligned} (\widehat{\mathbf{M}}, \widehat{\mathbf{d}}, \widehat{\zeta}) &= \arg \min_{\mathbf{M}, \mathbf{d}, \zeta} \ell(\mathbf{d}) + \gamma_1\|\zeta\|_1 + \gamma_2\|\mathbf{M}\|_F^2 + \gamma_3\|\mathbf{d}\|_2^2 \\ \text{s.t.} \quad & -\zeta \leq (\mathbf{I} - \mathbf{R}\mathbf{R}^\dagger)(\text{diag}(\mathbf{S}\mathbf{M}\mathbf{R}^T) - \mathbf{Q}_\Gamma\mathbf{d}) \leq \zeta \\ & \zeta \geq \mathbf{0}, \quad \mathbf{M} \succcurlyeq \mathbf{0}. \end{aligned} \quad (8)$$

The first two constraints aim to enforce (6), while the final constraint enforces that $\widehat{\mathbf{M}}$ is symmetric positive semi-definite. The constants $\gamma_1, \gamma_2, \gamma_3$ are parameters set by the user to control the amount of regularization. The above formulation is a convex (semi-definite) program and can be solved by standard tools such as CVX [39, 40].

With $\widehat{\mathbf{M}}$ in hand, we can then immediately solve for $\hat{\mathbf{u}}$ via (5). However, since we do not expect our estimate \mathbf{M} to be perfect, we instead use the regularized estimate:

$$\hat{\mathbf{u}} = \frac{1}{2}(\widehat{\mathbf{M}}\mathbf{R}^T\mathbf{R}\widehat{\mathbf{M}} + \alpha\mathbf{I})^{-1}\widehat{\mathbf{M}}\mathbf{R}^T(\mathbf{a}_{\widehat{\mathbf{M}}} - \mathbf{Q}_\Gamma\widehat{\mathbf{d}}), \quad (9)$$

where α is a regularization parameter set by the user. We will see in Section 4 that this simple single-step estimation procedure of estimating \mathbf{M} followed by \mathbf{u} is surprisingly effective.

3.4 Noise considerations

A common noise setting is when paired comparison outcomes are made with respect to differences in distances corrupted by additive iid noise. Such noise may arise from an imperfectly learned embedding or as a way of modeling response errors. While we pose the problem in a noiseless environment for simplicity, the estimation strategy outlined above can be adapted to accommodate such noise by replacing the loss function in (7) with the negative log-likelihood of observing comparison outcomes given a noise model, provided that the log-likelihood function is concave. For example, suppose we assume paired comparison outcomes follow the *Bradley-Terry model* [41], i.e.,

$$\mathbb{P}(\mathbf{x}_i \succ \mathbf{x}_j) = \frac{e^{-\|\mathbf{x}_i - \mathbf{u}\|_{\widehat{M}}^2}}{e^{-\|\mathbf{x}_i - \mathbf{u}\|_{\widehat{M}}^2} + e^{-\|\mathbf{x}_j - \mathbf{u}\|_{\widehat{M}}^2}} \quad (10)$$

where $\mathbf{x}_i \succ \mathbf{x}_j$ denotes that item \mathbf{x}_i is preferred to item \mathbf{x}_j and y_k is the outcome of the k^{th} comparison (with $y_k = -1$ if $\mathbf{x}_i \succ \mathbf{x}_j$ and $y_k = +1$ if $\mathbf{x}_j \succ \mathbf{x}_i$). Then, we can replace the loss function in (7) with the negative log-likelihood of observing the P paired comparison outcomes:

$$\ell(\mathbf{d}) = \sum_{k=1}^P \log(1 + e^{-y_k (\mathbf{Q}_\Gamma \mathbf{d})_k}), \quad (11)$$

where $(\mathbf{Q}_\Gamma \mathbf{d})_k$ denotes the k^{th} entry of the vector $\mathbf{Q}_\Gamma \mathbf{d}$

3.5 Alternating estimation

While the single-step approach described above is appealing due to its simplicity, the process of dividing the problem into first estimating \mathbf{M} and then estimating \mathbf{u} suggests a natural extension of then taking our estimate of \mathbf{u} and refining our estimate of \mathbf{M} , and then iteratively alternating between these two problems to (potentially) improve our estimates. Specifically, after obtaining $\widehat{\mathbf{M}}$ and $\widehat{\mathbf{u}}$ using (8) and (9), we can set $\widehat{\mathbf{M}}^{(0)} = \widehat{\mathbf{M}}$ and $\widehat{\mathbf{u}}^{(0)} = \widehat{\mathbf{u}}$. Then, for iteration $k > 0$, we re-estimate \mathbf{M} by solving the following optimization problem that replaces the constraint from (6) with the constraint from (4) to allow us to directly incorporate our previous estimate of \mathbf{u} :

$$\begin{aligned} (\widehat{\mathbf{M}}^{(k)}, \widehat{\mathbf{d}}^{(k)}, \widehat{\boldsymbol{\zeta}}^{(k)}) &= \arg \min_{\mathbf{M}, \mathbf{d}, \boldsymbol{\zeta}} \ell(\mathbf{d}) + \gamma_1^{(k)} \|\boldsymbol{\zeta}\|_1 + \gamma_2^{(k)} \|\mathbf{M}\|_F^2 + \gamma_3^{(k)} \|\mathbf{d}\|_2^2 \\ \text{s.t.} \quad & -\boldsymbol{\zeta} \leq \text{diag}(\mathbf{S}\mathbf{M}\mathbf{R}^T) - \mathbf{Q}_\Gamma \mathbf{d} - 2\mathbf{R}\mathbf{M}\widehat{\mathbf{u}}^{(k-1)} \leq \boldsymbol{\zeta} \\ & \boldsymbol{\zeta} \geq \mathbf{0}, \quad \mathbf{M} \succcurlyeq \mathbf{0}. \end{aligned} \quad (12)$$

We can then update our estimate of \mathbf{u} as before:

$$\widehat{\mathbf{u}}^{(k)} = \frac{1}{2} (\widehat{\mathbf{M}}^{(k)} \mathbf{R}^T \mathbf{R} \widehat{\mathbf{M}}^{(k)} + \alpha^{(k)} \mathbf{I})^{-1} \widehat{\mathbf{M}}^{(k)} \mathbf{R}^T (\mathbf{a}_{\widehat{\mathbf{M}}^{(k)}} - \mathbf{Q}_\Gamma \widehat{\mathbf{d}}^{(k)}).$$

Note that we allow the regularization parameters to change across iterations. In practice we fix $\gamma_1^{(k)}$, $\gamma_2^{(k)}$, $\gamma_3^{(k)}$, and $\alpha^{(k)}$ for all iterations $k \geq 1$, but we do consider an alternative set of parameters for the initialization ($k = 0$). This is somewhat natural since the initialization step actually involves solving a slightly different optimization problem.

3.6 Identifiability of the metric and ideal point

We conclude our description of our approach with a brief discussion of the degree to which the ideal point \mathbf{u} and metric \mathbf{M} are potentially identifiable.

Proposition 1. *For a fixed $\mathbf{M} \in \mathbb{R}^{D \times D}$, the ideal point \mathbf{u} is identifiable if and only if \mathbf{M} is (strictly) positive definite.*

The proof is provided in the supplementary material and is similar to the proof of Proposition 3 in [37]. This result is not surprising, as if \mathbf{M} is rank-deficient, any part of $\mathbf{u} \in \ker(\mathbf{M})$ will be not recoverable. We note that, since we desire our constraint set to be closed, our estimation procedure enforces a positive *semi*-definite constraint. In practice, if \mathbf{M} is ill-conditioned, we may estimate a solution which is rank-deficient (which ignores the eigenvectors corresponding to relatively small eigenvalues), and thus the portion of \mathbf{u} in the span of these eigenvectors may be extremely difficult to

estimate. Note, however, that due to the influence of M , the unidentifiable portion of \mathbf{u} also plays little role in determining the underlying preferences. In recognition of this, we typically quantify our estimation performance in terms of $\|\hat{\mathbf{u}} - \mathbf{u}\|_M$.

We also note that, at least in the noise-free setting considered in this paper, even if the metric is fully identifiable, it will only be so up to a constant scaling factor. However, note that a constant scaling factor does not change the learned ideal point or ranking of items. To see why this is true, note that for any M and \mathbf{d} satisfying the constraint in (4), rescaling M and \mathbf{d} (and hence δ_Γ) by an arbitrary constant $c > 0$ will yield another valid solution. However, it is relatively easy to show that for arbitrary scaling of M and δ_Γ , the estimate provided by (5) of $\hat{\mathbf{u}}$, as well as the resulting ranking of the items, remains unchanged.

Finally, we also note that when recovering M , if a subset of the eigenvalues of M are equal or relatively close, it becomes impossible, or at least more difficult, to distinguish among the specific eigenvectors. In this case, our estimate may swap the order of the eigenvectors or learn different eigenvectors that span a similar space to the original, but can be quite different. As with scaling, this has little impact on estimating \mathbf{u} or in terms of the resulting rankings, but plays an important factor in determining the appropriate evaluation metrics.

4 Experiments

4.1 Synthetic experiments

In this section, we demonstrate the effectiveness of the joint estimation on synthetically generated data². We assume *a priori* knowledge of an existing embedding of items and estimate \mathbf{u} and M . In each simulation, N items $\mathbf{x}_1, \dots, \mathbf{x}_N$ are generated uniformly on the hypercube $[-2, 2]^D$ and one user \mathbf{u} is generated uniformly on $[-1, 1]^D$. A positive definite matrix M is generated by $M = \mathbf{L}^T \mathbf{L}$, where the entries of $\mathbf{L} \in \mathbb{R}^{D \times D}$ are drawn from the standard normal distribution. Noiseless comparisons are chosen uniformly without repetition and used for joint estimation.

Certain conditions are imposed on the matrix M : 1) The Frobenius norm of M exceeds a small chosen threshold ϵ_F , 2) The smallest singular value of M is larger than a small chosen threshold ϵ_S , and 3) The fraction $\|M\mathbf{u}\|_2/\|\mathbf{u}\|_2$ exceeds a small chosen threshold ϵ_P . ϵ_F and ϵ_S are imposed to guard against numerical instabilities while ϵ_P is necessary to ensure that \mathbf{u} is identifiable. For all synthetic experiments, the chosen values were $\epsilon_F = 0.5$, $\epsilon_S = 0.25$, and $\epsilon_P = 0.2$.

We define the user’s ideal point reconstruction error (UR error) as $\|\hat{\mathbf{u}} - \mathbf{u}\|_M^2/\|\mathbf{u}\|_M^2$. Letting the eigendecompositions of M and \hat{M} be $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ and $\hat{\mathbf{V}}\hat{\mathbf{\Lambda}}\hat{\mathbf{V}}^T$, respectively, we define the weighted eigenstructure reconstruction error (WER error) as $\|\mathbf{\Lambda} \odot |\mathbf{V}^T \hat{\mathbf{V}}| - \hat{\mathbf{\Lambda}}\|_F^2/\|\mathbf{\Lambda}\|_F^2$, where \odot denotes element-wise multiplication and $|\mathbf{A}|$ takes the element-wise absolute value of \mathbf{A} . When \hat{M} is recovered to be a scaled version of M , we expect the diagonal elements of $|\mathbf{V}^T \hat{\mathbf{V}}|$ to be 1. In all cases when the WER error is small, M is recovered well. However, there are instances in which a high value of the WER error does not imply a poor estimate of M . For example, (large) repeated eigenvalues in M would result in a large WER error if the eigenvectors in $\hat{\mathbf{V}}$ differed, but spanned the same space. Our synthetic data avoids this, but care is needed to quantify performance in general.

Single-step estimation In the first simulation, we show the improvement in estimation as the number of comparisons increases. For a fixed number of comparisons, we perform 100 trials and report the median UR error and WER error, and interpolated median of the fraction of the top 10 closest items to \mathbf{u} identified for $D = 2, 5$, and 10. Since the fraction of the top 10 items is discrete, we utilize the interpolated median in place of the usual median. In all cases, we include the 25% and 75% quantiles. For each trial, we generate a new metric, ideal point, and $N = 100$ items.

As shown in Fig. 1, when a small number of comparisons are used for joint estimation, the UR and WER error are large, while the fraction of top 10 items correctly identified is small. As the number of comparisons increases from 10 to 500, the UR and WER errors decrease rapidly, while the fraction of top 10 items increases rapidly.

In the second simulation, we compare the performance of our algorithm against two algorithms that assume Euclidean distance to estimate the ideal point. **Euclidean Algorithm 1** is an adaptation of

²Code available at <https://github.com/austinxu87/IdealPointMetric>

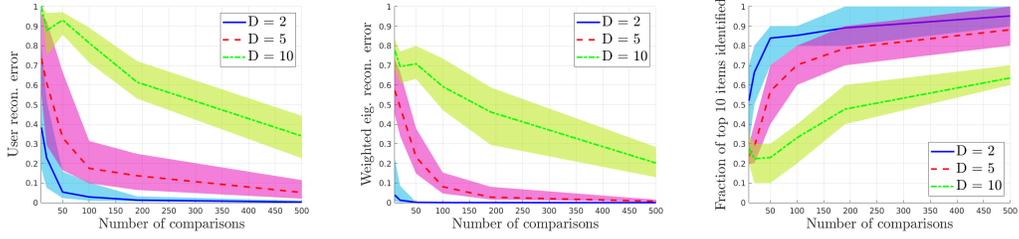


Figure 1: Median UR error, WER error, and interpolated median fraction of top 10 items identified over 100 trials, plotted with 25% and 75% quantiles. As the number of comparisons grows, both UR and WER error decrease to 0 as the fraction of top 10 items increases to 1 for all D . Regularization parameters: $\gamma_1 = 2, \gamma_2 = 0.002, \gamma_3 = 0.001, \alpha = 1$.

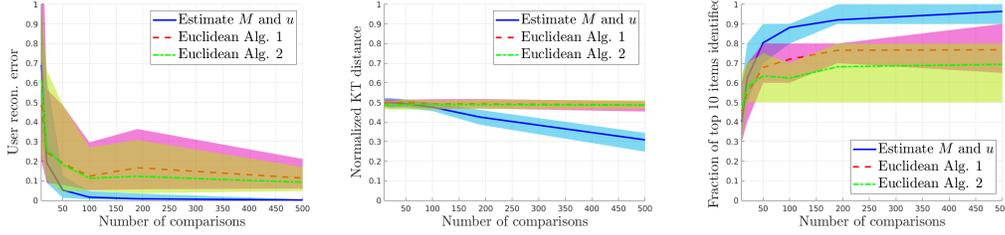


Figure 2: Comparison of single-step estimation against Euclidean Algorithms 1 and 2 when the true distance metric is $M \neq I$. Regularization parameters: $\gamma_1 = 2, \gamma_2 = 0.002, \gamma_3 = 0.001, \alpha = 1$.

our single-step algorithm to solve for only the distances d_e between items and the ideal point:

$$\begin{aligned}
 (\hat{d}_e, \hat{\zeta}) &= \arg \min_{d, \zeta} \ell(d_e) + \gamma_1 \|\zeta\|_1 + \gamma_2 \|d_e\|_2^2 & (13) \\
 \text{s.t.} \quad & -\zeta \leq (I - RR^\dagger)(\text{diag}(SR^T) - Q_\Gamma d_e) \leq \zeta, \quad \zeta \geq 0.
 \end{aligned}$$

From here, we can solve for \hat{u} by replacing replacing \hat{M} with I in (9). **Euclidean Algorithm 2** is the approach in [21], which directly solves a convex program for u from the paired comparisons.

We sweep the performance for all three algorithms for $D = 2$ over different numbers of comparisons between 10 and 500. For a fixed number of comparisons, we perform 100 trials and report the median (or interpolated median) and 25% and 75% quantile for UR error, normalized Kendall’s Tau distance, and the fraction of top 10 items identified. For each trial, we generate a new metric and ideal point and $N = 100$ new items. As seen in Fig. 2, our algorithm outperforms both algorithms that assume a Euclidean distance metric by recovering a more accurate ideal point, ranking of items, and fraction of top K items. The same experiment was performed when $M = I$ for all trials with very little loss in performance by using our algorithm (see the supplementary material for further details).

Alternating estimation We now explore the potential improvements that can be attained by our alternating estimation procedure. For $D = 5$, we fix an ideal point, metric, and a set of $N = 100$ items, and vary the number of comparisons. For a fixed number of comparisons P , we run 100 trials, where we select P new comparisons at random. We then run the alternating descent until the difference in the user reconstruction error between successive iterations is less than 10^{-3} , with a maximum number of iterations set to 100. We report the median and 25% and 75% quantiles for the initial and final UR error in Fig. 3. We observe that alternating estimation does not improve the estimate of u much when the number of comparisons is small (< 40) or large

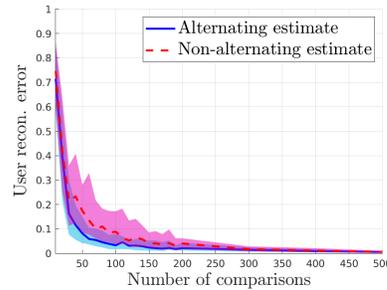


Figure 3: UR error for single-step and alternating estimation. Regularization parameters: $\gamma_1^{(0)} = 2, \gamma_2^{(0)} = 0.002, \gamma_3^{(0)} = 0.0001, \alpha^{(0)} = 1; \gamma_1^{(k)} = \frac{2}{3}, \gamma_2^{(k)} = \frac{1}{15}, \gamma_3^{(k)} = \frac{7}{1500}, \alpha^{(k)} = \frac{1}{2}$ for $k \geq 1$.

(> 200). In the first regime, the comparisons do not reveal enough information to reliably recover \mathbf{u} , while in the second regime, the number of comparisons is sufficient to make the single-step estimation very accurate. The alternating method offers steady improvement in the intermediate regime, and is able to successfully reduce the error nearly 60%.

4.2 Graduate admissions dataset

We now apply our models to two PhD program admissions datasets from Georgia Tech School of Electrical and Computer Engineering. The *Unranked Candidates* dataset consists of over 3,000 applicants in three categories: 1) admitted with fellowship, 2) admitted, and 3) denied admission. The applicants are not ranked, so the only paired comparisons we can form are across categories. We assume that fellowship recipients are preferred to admitted candidates, who are preferred to denied candidates, so for N_F fellowship recipients, N_A admitted candidates, and N_D denied candidates, we can form at most $N_F(N_A + N_D) + N_A N_D$ comparisons. For each applicant, we have access to five features: GPA, GRE quantitative, verbal, and analytical writing scores, and a letter of recommendation (LoR) score. Each candidate’s GPA is normalized to a 4.0 scale. The GRE verbal and quantitative scores are integers between 130 and 170, inclusive, while the GRE writing score is from 0 to 6 in 0.5 increments. Each candidate submitted at most three letters of recommendation, each of which is scored on a scale of 0 to 3. The scores are averaged and then exponentiated to obtain a LoR score between 1 and $e^3 \approx 20.09$.

The *Ranked Candidates* dataset consists of 88 applicants who are scored on a scale of 1 to 10, with 1 being the most preferred and 10 being least preferred. The top 11 candidates are uniquely rank ordered, and the rest of the candidates are sorted into 8 bins of various sizes. We only form comparisons between candidates with different scores, so two candidates with the same score are not compared. For each applicant, we have access to the same features except for letter of recommendation scores.

Unranked Candidates We begin by noting that the features being used in this model are inherently restrictive. Applicants are evaluated on many criteria beyond the features included, which can lead to occasional unexpected results. For instance, there exist large subsets of denied candidates whose average GRE scores are higher than those of a some fellowship recipients, which might indicate that a lower GRE score is more favorable, occasionally leading to rather unusual ideal point placement. Furthermore, the learned ideal point may fall outside of appropriate ranges (e.g., GPA > 4), which indicates an extremely strong preference for that feature. However, one can interpret values outside of appropriate ranges as a preference for the maximum/minimum appropriate score. In reality, we would expect that the optimal set of features should be the maximum value for all possible features. Furthermore, of the five features, we suspect that the GRE verbal score should likely matter the least, followed by the GRE quantitative score, as applicants from across the categories score similarly on these two GRE sections. Our expectation is that the most significant features should be some combination of GRE writing, GPA, and LoR. With this in mind, we use our algorithm to learn relevant feature interactions and confirm our hypothesized ordering of the importance of features via the learned metric. We take $N_F = 33$, $N_A = 33$, and $N_D = 34$, form all 3333 possible comparisons, and learn the ideal point \mathbf{u} and metric \mathbf{M} using a subset of all features.

When all five features are used to learn $\hat{\mathbf{u}}$ and $\hat{\mathbf{M}}$, our hypothesized ordering of importance for the features is correct. The three most significant features are a weighted difference between GPA and GRE writing, a weighted sum of GPA and GRE writing, and the LoR score. The learned ideal scores are 158.08 GRE verbal, 162.50 GRE quantitative, 4.68 GRE writing, 4.06 GPA, 15.28 LoR score. As seen in Fig. 4, when the GRE verbal and quantitative scores features are used, the learned feature interactions are a weighted difference (eigenvector 1) and sum (eigenvector 2) of GRE verbal and quantitative scores. The structure of the learned metric seems to make intuitive sense, indicating that in order to compensate for a slightly lower GRE quantitative score, one must score significantly higher on the GRE verbal section.

Ranked Candidates Since ranking information is partially available in this dataset, we record the fraction of top $K = 11, 17$, and 22 candidates correctly identified as the number of comparisons increases using all four features. For a fixed number of comparisons, we perform 20 trials and report the mean and standard deviation of the fraction of the top K candidates correctly identified in Fig. 5. The fraction of the top K candidates correctly identified for $K = 11, 17$, and 22 increases rapidly as the number of comparisons increases. With less than 20% of the total number of comparisons, we can identify over 90% of the top 22 and 17 candidates and over 80% of the top 11 candidates correctly.

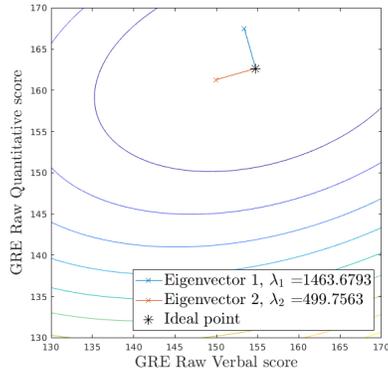


Figure 4: Level sets for learned metric for *Unranked Candidates* GRE verbal and quantitative scores. Regularization parameters: $\gamma_1 = \frac{1}{650}$, $\gamma_2 = \frac{1}{6500}$, $\gamma_3 = \frac{2}{65} \cdot 10^{-6}$, $\alpha = 1$.

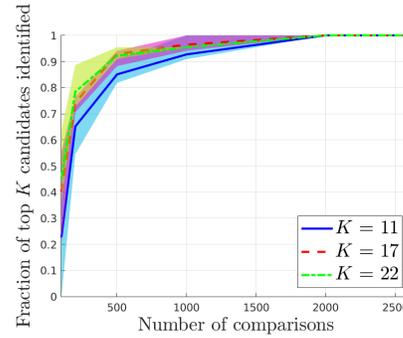


Figure 5: Fraction of top 11, 17, and 22 of ranked candidates identified. Regularization parameters: $\gamma_1 = \frac{3}{800}$, $\gamma_2 = \frac{1}{8000}$, $\gamma_3 = \frac{5}{8} \cdot 10^{-11}$, $\alpha = 1$.

5 Discussion

In this paper, we develop a method for jointly learning a user’s ideal point and an underlying distance metric from paired comparisons. The metric captures feature interactions and their relative significance to users, neither of which are captured by the traditional Euclidean metric. We demonstrate our algorithm can correctly identify the ideal point and metric and can correctly rank graduate admission candidates and determine feature interactions on real-world data. We conclude by noting that in the Euclidean setting, adaptive querying schemes have been shown to enable dramatic reductions in the required number of comparisons [9, 42]. We expect similar gains are possible in our setting. Developing novel methods for adaptively selecting comparisons to maximize the amount of information collected about both u as well as M is an important avenue for future research.

Broader Impact

In an increasingly diverse set of contexts, automated ranking systems are playing an increasing role in society. We naturally have an interest in ensuring that these systems are as accurate as possible and not undermined by poor modeling assumptions. If used to augment such systems, our research could be used to develop improved models for how users implicitly process the features of the ranked items by learning a metric. In particular, in addition to providing a more powerful and flexible model of preference, the metric illuminates how features are combined and the ordering of importance of the combined features, providing both additional insight as well as helping to better identify user desires and rankings.

In certain applications, such as in hiring or admissions committees, there is also the potential to apply our techniques to discover if the metric of an evaluator is significantly influenced by certain factors (e.g., race, gender, sexual orientation, religion, etc.) which we do not wish to influence our decisions. This information may be useful as feedback to the evaluator or alternatively, to explicitly compensate for the existence of such influence (e.g., by re-ranking candidates using a modified metric that eliminates dependence on those factors).

However, this optimistic assessment of the potential of our approach should be tempered by two important caveats. First, in the absence of any strong theoretical guarantees, we are unable to provide any notion of confidence intervals or statistical significance in the learned parameters, and so one must evaluate the learned models with a degree of caution and awareness that the learned parameters may be the result of chance variation in a small dataset. More generally, there is no notion of causal influence in our model, and it is always possible that the observed dependence on the features is the result of correlations with some unobserved latent factors. Second, due to both the potential for inaccuracies in the learned parameters as well as the personal nature of the revealed preferences, as with any system involving personal data, privacy remains a significant concern.

Acknowledgments and Disclosure of Funding

This work was supported, in part, by the Alfred P. Sloan Foundation.

References

- [1] Clyde H Coombs. Psychological scaling without a unit of measurement. *Psychological review*, 57(3):145, 1950.
- [2] Bernard Dubois. Ideal point versus attribute models of brand preference: a comparison of predictive validity. *ACR North American Advances*, 1975.
- [3] Alberto Maydeu-Olivares and Ulf Böckenholt. Modeling preference data. In R. Millsap and A. Maydeu-Olivares, editors, *The SAGE Handbook of Quantitative Methods in Psychology*, chapter 12, pages 264–282. SAGE Publications Ltd., London, UK, 2009.
- [4] George Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psych. Rev.*, 63(2):81, 1956.
- [5] Herbert Aron David. *The method of paired comparisons*, volume 12. London, 1963.
- [6] Ben Carterette, Paul N Bennett, David M Chickering, and Susan T Dumais. Here or there: Preference judgments for relevance. In *Proc. European Conf. on Inf. Retrieval (ECIR)*, Glasgow, Scotland, Apr. 2008.
- [7] Eyke Hullermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artif. Intell.*, 172(16-17):1897–1916, 2008.
- [8] Nir Ailon. Active learning ranking from pairwise preferences with almost optimal query complexity. In *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Grenada, Spain, Dec. 2011.
- [9] Kevin G Jamieson and Robert Nowak. Active ranking using pairwise comparisons. In *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*. Grenada, Spain, 2011.

- [10] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: Ranking from pair-wise comparisons. *Operations Research*, 65, 09 2012.
- [11] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Lake Tahoe, California, Dec. 2012.
- [12] Fabian Wauthier, Michael Jordan, and Nebojsa Jojic. Efficient ranking from pairwise comparisons. In *Proc. Int. Conf. Mach. Learn. (ICML)*, Atlanta, Georgia, 2013.
- [13] Brian Eriksson. Learning to top-k search using pairwise comparisons. In *Proc. Int. Conf. Art. Intell. Stat. (AISTats)*, Scottsdale, Arizona, 2013.
- [14] Yuxin Chen and Changho Suh. Spectral mle: Top-k rank aggregation from pairwise comparisons. 2015.
- [15] Nihar B Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramchandran, and Martin J. Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *J. Mach. Learn. Res.*, 17(58):1–47, 2016.
- [16] Nihar B Shah and Martin J. Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *J. Mach. Learn. Res.*, 18(199):1–38, 2018.
- [17] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *Proc. Int. Conf. Art. Intell. Stat. (AISTats)*, San Juan, Puerto Rico, Mar. 2007.
- [18] Kevin Jamieson and Robert Nowak. Low-dimensional embedding using adaptively selected ordinal data. In *Proc. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Sept. 2011.
- [19] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Kalai. Adaptively learning the crowd kernel. In *Proc. Int. Conf. Mach. Learn. (ICML)*, Bellevue, Washington, Jul. 2011.
- [20] Laurens Van Der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *Proc. IEEE Int. Work. Machine Learning for Signal Processing (MLSP)*, Santander, Spain, Sept. 2012.
- [21] Mark Davenport. Lost without a compass: Nonmetric triangulation and landmark multidimensional scaling. In *Proc. IEEE Int. Work. Comput. Adv. Multi-Sensor Adaptive Processing (CAMSAP)*, Saint Martin, French West Indies, France, 2013.
- [22] Matthäus Kleindessner and Ulrike Luxburg. Uniqueness of ordinal embedding. In *Conf. Learning Theory (COLT)*, Princeton, New Jersey, Jun. 2014.
- [23] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Proc. Int. Work. on Similarity-Based Pattern Recognition (SIMBAD)*, Copenhagen, Denmark, Oct. 2015.
- [24] Lalit Jain, Kevin Jamieson, and Rob Nowak. Finite sample prediction and recovery bounds for ordinal embedding. In *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Barcelona, Spain, Dec. 2016.
- [25] Ery Arias-Castro. Some theory for ordinal embedding. *Bernoulli*, 23(3):1663–1693, 2017.
- [26] Andrew K Massimino and Mark Davenport. As you like it: Localization via paired comparisons. *arXiv preprint arXiv:1802.10489*, 2018.
- [27] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Vancouver, British Columbia, 2004.
- [28] Eric Yi Liu, Zhishan Guo, Xiang Zhang, Vladimir Jojic, and Wei Wang. Metric learning from relative comparisons by minimizing squared residual. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 2012.
- [29] Marc T Law, Nicolas Thome, and Matthieu Cord. Learning a distance metric from relative comparisons between quadruplets of images. *Int. J. Computer Vision*, 121(1):65–94, 2017.

- [30] Blake Mason, Lalit Jain, and Robert Nowak. Learning low-dimensional metrics. In *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Long Beach, California, 2017.
- [31] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10(2), 2009.
- [32] Brian McFee and Gert R Lanckriet. Metric learning to rank. In *Proc. Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010.
- [33] Daryl Lim and Gert Lanckriet. Efficient learning of mahalanobis metrics for ranking. In *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014.
- [34] Cijo Jose and François Fleuret. Scalable metric learning via weighted approximate rank component analysis. In *Proc. European Conf. Comp. Vision (ECCV)*, pages 875–890. Springer, 2016.
- [35] Tong Zhao, Julian McAuley, and Irwin King. Improving latent factor models via personalized feature projection for one class recommendation. In *Proc. ACM Int. Conf. on Information and Knowledge Management*, 2015.
- [36] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *Proc. Int. Conf. on World Wide Web*, Perth, Australia, 2017.
- [37] Amanda Bower and Laura Balzano. Preference modeling with context-dependent salient features. In *Proc. Int. Conf. Mach. Learn. (ICML)*, Online, 2020.
- [38] Matthew R O’Shaughnessy and Mark Davenport. Localizing users and items from paired comparisons. In *Proc. IEEE Int. Work. Machine Learning for Signal Processing (MLSP)*, Salerno, Italy, 2016. IEEE.
- [39] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, Mar. 2014.
- [40] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.
- [41] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- [42] Gregory Canal, Andrew Massimino, Mark Davenport, and Christopher Rozell. Active embedding search via noisy paired comparisons. In *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, California, 2019.