ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9747038

DELTA DISTANCING: A LIFTING APPROACH TO LOCALIZING ITEMS FROM USER COMPARISONS

Andrew D. McRae[g], Austin Xu[g], Jihui Jin[g], Namrata Nadagouda[g], Nauman Ahad[g], Peimeng Guan[g], Santhosh Karnik[m], Mark A. Davenport[g]

[g] Electrical & Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA
[m] Department of Computational Mathematics, Science, & Engineering, Michigan State University, East Lansing, MI, USA

ABSTRACT

A common problem in recommendation systems is to learn a model of user preferences based only on comparisons of the relative attractiveness of different items. We consider this problem in the context of an ideal point model of user preference, where each user can be represented as a point in a low-dimensional space together with a set of items. In this model, the closer an item is to a user's ideal point, the more that user prefers the item. When an embedding of items is known *a priori*, the problem of localizing a user's ideal point from comparisons amongst items is well studied. However, relatively little work exists on learning embeddings for new items based only on such comparisons. In this paper, we consider the problem of embedding a set of items using paired comparisons from a set of known users. Specifically, we present a novel convex lifted method of learning the embedding representation $p_1, \ldots, p_n \in \mathbf{R}^d$ of n items given noisy responses of the form "user u_k prefers item p_i to item p_j" for an arbitrary set of users $\{u_k\}$ in $\mathbf{R}^d$. We provide a range of simulations that validate the efficacy of our approach.

***Index Terms*—** lifting, paired comparisons, generalized nonmetric multidimensional scaling, matrix completion, delta distancing

1. INTRODUCTION

Personalized recommendation systems have become an essential tool in many modern applications, most prominently in online shopping, media consumption, and information retrieval. As a result, a number of methods have been developed in recent years to learn models of the preferences of the users of these systems. One prominent approach utilizes the *ideal point model* [1], a classical model for human preference where the user is represented as an ideal point in some low-dimensional space along with a set of items. The closer an item's embedding is to a user's ideal point under the Euclidean distance, the more strongly preferred that item is. We emphasize that the ideal point of a user is not necessarily an instantiated item, but rather the most preferred combination of features. In order to collect preference information from the user, we will query the users with *paired comparisons* [2] of the form "do you prefer item p_i or item p_j?", which users typically find easier to answer than providing precise quantifications of preference [3].

While most of the methods developed in the context of the ideal point model have focused on learning user ideal point representations [4]–[7] or user item rankings [8]–[13], relatively little work exists in learning how to embed *new items* that are introduced into the recommendation system. This lack of ability to embed new items is surprising given that in prominent recommendation system settings, such as online shopping or streaming platforms, new items are constantly being added to the content catalogue. However, in order to recommend these new items to new or existing users, the items must first be accurately embedded into the low-dimensional feature space. While it may be possible to construct these embeddings using item meta-data, such data may not capture the perceptually relevant criteria, in which case it would be preferable to be able to embed the items via user preference judgements.

Towards this end, we consider the problem of learning embeddings for a set of items $p_1, \ldots, p_n \in \mathbf{R}^d$ based on outcomes of paired comparisons of user preferences from a set of known user ideal points $u_1, \ldots u_\ell$. In particular, we will utilize paired comparisons of the form "user u_k prefers item p_i to item p_j." In this setting, if we let D denote the $n \times \ell$ matrix of all possible (squared) item-user distances (i.e., the matrix whose $(i,k)^{\text{th}}$ entry $D_{ik} = \|p_i - u_k\|^2$), then each paired comparison reveals comparative information about the difference in distances

$$\delta_{ijk} = D_{ik} - D_{jk}. \tag{1}$$

Specifically, in the simplest (noise-free) model, the response that "item p_i is preferred to item p_j for user u_k" could be

This work was supported, in part, by the NSF grant CCF-2107455 and a generous gift from the Adam S. Charles Foundation. The student authors are listed in alphabetical order by first name. Email: {admcrae, axu, jihui, namrata.nadagouda, nauman.ahad, guanpeimeng, mdav}@gatech.edu, karniksa@msu.edu

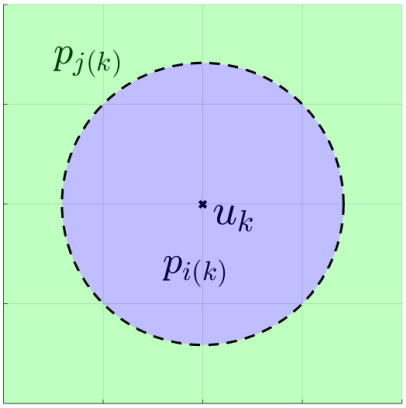


Fig. 1. An illustrative example of paired comparison response "$p_{i(k)}$ is preferred to $p_{j(k)}$ for a known user u_k." Note that if $p_{j(k)}$ were also known, the feasible region for $p_{i(k)}$ is a disk centered around u_k (shown in indigo blue) with radius $\|p_{j(k)} - u_k\|_2/2$. However, note that if $p_{i(k)}$ were known instead, the feasible region for $p_{j(k)}$, shown in lime green, is a non-convex region.

modeled as revealing the sign of δ_{ijk}. Our task here is to learn the p_i from such observations in the setting where the u_k are known.

The problem of simultaneously embedding a set of new items can equivalently be viewed as one of *matrix completion*. Specifically, given (potentially noisy) one-bit quantized observations of δ_{ijk} as in (1), the task of estimating the p_i is equivalent to recovering the (low-rank) matrix of item-user distances D. In this work we assume that the u_k are known *a priori* (as in a mature recommendation system), which considerably simplifies our task. In this context, a natural approach might be to formulate an optimization problem for the p_i by leveraging our observations as constraints (or through a simple loss function).

Unfortunately, as noted in [14], the most straightforward formulation of the constraints imposed by user-item comparisons can be non-convex, as illustrated in Fig. 1. In our approach, which we dub *delta distancing*, we "lift" the non-convex problem into a higher dimensional space of $(d+1) \times (d+1)$ symmetric positive semi-definite (PSD) matrices. In this lifted space our non-convex constraints become convex, enabling a natural approach to estimating the p_i. As we will show below, this provides a powerful and effective method for embedding items from user comparisons.

2. RELATED WORK

Our method builds on existing work in learning embeddings from paired comparisons. In particular, the problem is closely related to the *generalized non-metric multidimensional scaling* problem of [15], where no distinction is made between "users" and "items", greatly expanding the set of potential queries. Assuming a probabilistic item comparison response model (where no user ideal points are utilized), [16] applies gradient descent to a non-convex optimization problem.

Our approach is perhaps most closely related to [14], which localizes one new item at a time by forming a convex linear relaxation of non-convex constraints. In doing so, the feasible region resulting relaxation does not capture the full feasible region of the original constraints. The primary difference between our developed method and established item embedding methods from paired comparisons is that we take an alternative approach to addressing the non-convexity of the constraints, which allows us to simultaneously embed multiple new items at once. The previous work of [14] assumed that only a single new item was being embedded and that all other reference items were known *a priori*. In contrast, our approach is valid even if *none* of the items in consideration have a known embedding.

The key to our approach to avoiding non-convex constraints is "lifting" our problem to a higher dimensional space. The fundamental idea of lifting is that when our observations depend in a *nonlinear* way on parameters we are interested in, we can convert the problem to a *linear* model by mapping the parameters to a larger space. This is the idea behind the "kernel trick" behind support vector machines widely used in regression and classification [17]. It has also been used in nonlinear matrix completion [18]. Our particular method of making a *quadratic* measurement linear by lifting to a space positive semidefinite matrices is widely used in problems in optimization, statistics, and telecommunications (see, e.g., [19]–[21]; see [22] for a more comprehensive survey and introduction).

3. PROBLEM STATEMENT

In this work, we assume that the users provide noisy responses to paired comparisons which follow the well established Bradley-Terry model [23]. This model has not only been analyzed extensively in both the social science and machine learning communities, but versions of it have also been implemented in real-world ranking settings, such as the World Chess Federation player rankings [6], [24] or the MSR TrueSkill online video game player rankings [13]. Under this model, when user u_k is presented items $p_{i(k)}$ and $p_{j(k)}$, the paired comparison outcome $y_k \in \{-1, +1\}$ is $2b_k - 1$, where b_k distributed as a Bernoulli random variable with parameter μ_k, where

$$\mu_k = \frac{1}{1 + \exp\left(-\delta_{i(k)j(k)k}\right)}. \tag{2}$$

Given m paired comparison outcomes $y_k \in \{-1, +1\}$, $k = 1, \ldots, m$ generated in accordance to the Bradley-Terry model and known vectors $u_1, \ldots, u_m \in \mathbf{R}^d$ (possibly repeated if a

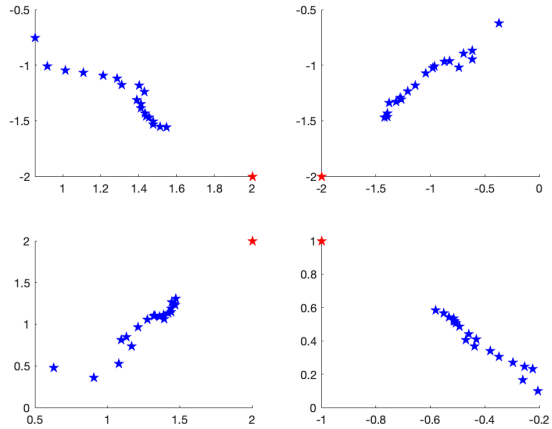


Fig. 2. Item estimate trajectories observed in a sample experiment. The red stars indicate true locations and the path of blue stars indicate the estimates provided by our method.

single user makes multiple comparisons, e.g., these could be choices from the set of ℓ distinct users discussed above), we want to estimate $p_1, \ldots, p_n \in \mathbf{R}^d$. We can "lift" this problem to the space of PSD matrices by noting that for any user u_k and item p_i,

$$\begin{aligned} D_{ik} &= \|p_i - u_k\|_2^2 \\ &= \|p_i\|_2^2 - 2\langle p_i, u_k\rangle + \|u_k\|_2^2 \\ &= \left\langle \begin{bmatrix} p_i p_i^T & p_i \\ p_i^T & 1 \end{bmatrix}, \begin{bmatrix} I_d & -u_k \\ -u_k^T & \|u_k\|_2^2 \end{bmatrix} \right\rangle \\ &=: \langle P_i, U_k\rangle, \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ is the trace matrix inner product.

This is a standard method to transform a quadratic function of vectors into a linear function of a matrix (again, see [22] for a comprehensive introduction to this class of techniques). We have effectively transformed our problem from estimating n vectors in $\mathbf{R}^d$ to estimating n symmetric positive semi-definite matrices in $\mathbf{R}^{(d+1)\times(d+1)}$. Although this would seem to make the problem more difficult, the key benefit is that the D_{ik}'s are now linear in the matrices P_i, and maximum likelihood is now a convex program. Estimates for the matrices P_i can be obtained using commercial solvers such as CVX [25], [26]. Note that each matrix P_i contains additional exploitable structure in that it is rank-one, and its bottom-right entry is 1. In order to estimate $\{P_i\}_{i=1}^n$, we utilize a maximum-likelihood motivated optimization program:

$$\min_{P_1,\ldots,P_n} \sum_{k=1}^{m} \log\left(1 + \exp\left(y_k \langle P_{i(k)} - P_{j(k)}, U_k\rangle\right)\right) \quad (3)$$
$$+ \lambda \sum_{i=1}^{n} \mathrm{tr}(P_i)$$
$$\text{subject to } P_i \succeq 0, \quad (4)$$
$$(P_i)_{d+1,d+1} = 1,\ i = 1, \ldots, n.$$

In order to encourage our solutions to be low-rank, we use trace regularization with $\lambda > 0$ being a user specified parameter. From the resulting solution $(\widehat{P}_1, \ldots, \widehat{P}_n)$, we can, for each i, recover a vector estimate $\hat{p}_i$ from the first d elements of the last column of $\widehat{P}_i$.

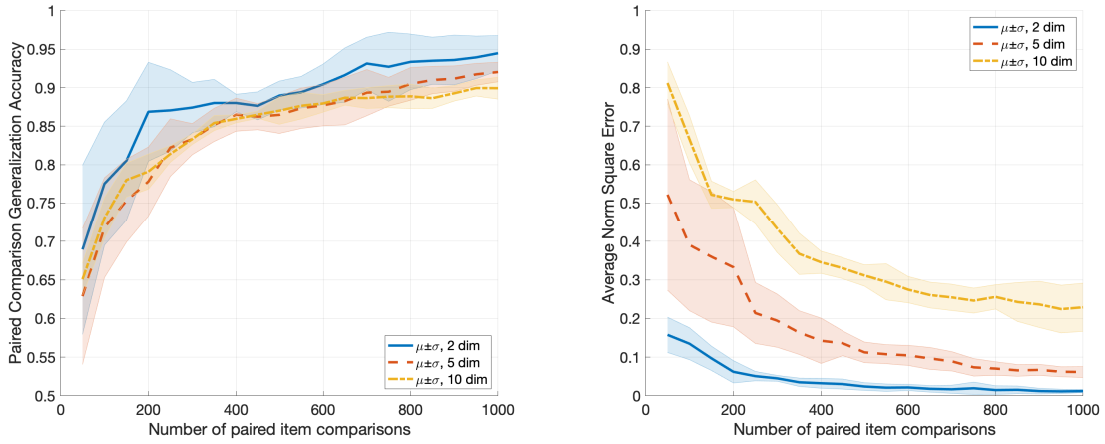
4. EXPERIMENTS

We demonstrate the effectiveness of our recovery algorithm on synthetically generated datasets with known ground truth.

To begin, we illustrate the convergence behavior of our algorithm to the true solution. For $d = 2$, we generate 5 items and 100 distinct users and track the estimate for a particular item as the number of paired comparison increases. Note that because of the quantization of our problem, we can only estimate the item embeddings within a certain convex region around the true value, as described in [27]. However, we hope to observe that as the number of paired comparisons increases, the estimates for the embeddings converges to the location of the true embedded item vectors. As seen in Fig. 2, this is in fact the case.

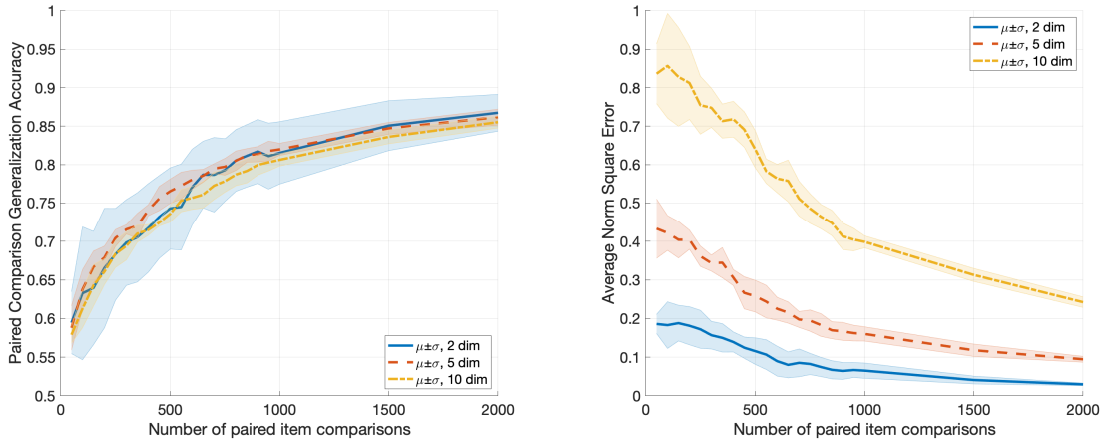
Next, we empirically evaluate the performance of our algorithm using two error metrics. The first is *paired comparison generalization accuracy (PCGA)*, which is the fraction of paired comparisons induced by the learned items which are consistent with the ground truth paired comparisons. This takes value between 0 and 1, with a higher value being more desirable. The second is *averaged norm-square error (ANSE)*: $\frac{1}{n}\sum_{i=1}^{n} \|\hat{p}_i - p_i\|_2^2$. For $d = 2, 5, 10$, we generate 5 and 20 items and 100 distinct users uniformly on $[-0.5, 0.5]^d$ and $[-1, 1]^d$, respectively. With 5 items, the total number of available paired comparisons is 1000, while with 20 items, the total number of available paired comparisons is 19000. We sweep the number of comparisons (choosing them randomly without repetition), record both the PCGA and ANSE, and report the mean and $\pm$ one standard deviation over five trials. For all experiments, CVX [25], [26] was used to solve the optimization problem (3), with $\lambda = 2$. Plots of the performance of our recovery method can be seen in Fig. 3

Our algorithm demonstrates strong performance under both error metrics even with noisy measurements. In particular, as the number of paired comparisons increases, the PCGA increases rapidly, while the ANSE decays rapidly, indicating



(a) PCGA with 5 items

(b) ANSE with 5 items



(c) PCGA with 20 items

(d) ANSE with 20 items

Fig. 3. PCGA (left column) and ANSE (right column) for 5 items (top row) and 20 items (bottom row). As the number of comparisons increases, PCGA increases steadily and ANSE decreases rapidly, indicating that our algorithm is capable of estimating accurate embedding points and predicting paired comparison outcomes well. All plots report mean over five trials with $\pm$ one standard deviation for multiple item dimensions.

strong performance both in estimation of unknown embedded points and predicting paired comparison outcomes. It is notable that for 20 items, only a fraction of the 19,000 available paired comparisons are needed to accurately estimate the embedded points.

5. CONCLUSION

In this paper, we described a method for learning an embedding of a set of items given paired comparisons from users whose embedding location is known. Our method works by "lifting" the problem from the space of d-dimensional vectors to the space of $(d+1) \times (d+1)$ symmetric PSD matrices, where our problem becomes convex. We performed simulations on a synthetic dataset to demonstrate the effectiveness of our proposed optimization program.

There are several interesting directions for potential future work. While an off-the-shelf convex solver worked well for validating our approach, we could likely scale the algorithm up to much larger problem sizes with a problem-specific implementation (e.g., of an interior-point or proximal algorithm). Another interesting extension would be to consider nonmetric multidimensional scaling where we know neither the items or users *a priori*. Similar to [14], we could perform alternating minimization over both users and items (in each step performing an optimization similar to ours).

6. REFERENCES

[1] C. H. Coombs, "Psychological scaling without a unit of measurement.," *Psychol. Rev.*, vol. 57, no. 3, pp. 145–158, 1950.

[2] H. A. David, *The method of paired comparisons*. New York: Hafner, 1963.

[3] G. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information.," *Psychol. Rev.*, vol. 63, no. 2, p. 81, 1956.

[4] A. Xu and M. A. Davenport, "Simultaneous preference and metric learning from paired comparisons," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, Virtual conference, Dec. 2020.

[5] G. Canal, M. Connor, J. Jin, N. Nadagouda, M. O'Shaughnessy, C. J. Rozell, and M. A. Davenport, "The PICASSO algorithm for Bayesian localization via paired comparisons in a union of subspaces model," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Virtual conference, May 2020.

[6] A. Bower and L. Balzano, "Preference modeling with context-dependent salient features," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Virtual conference, 2020.

[7] G. Canal, A. Massimino, M. A. Davenport, and C. Rozell, "Active embedding search via noisy paired comparisons," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, Jun. 2019, pp. 902–911.

[8] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais, "Here or there: Preference judgments for relevance," in *Proc. European Conf. on Inf. Retrieval (ECIR)*, Glasgow, Scotland, Apr. 2008, pp. 16–27.

[9] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," *Artif. Intell.*, vol. 172, no. 16-17, pp. 1897–1916, 2008.

[10] N. Ailon, "Active learning ranking from pairwise preferences with almost optimal query complexity," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Grenada, Spain, Dec. 2011.

[11] K. G. Jamieson and R. Nowak, "Active ranking using pairwise comparisons," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Granada, Spain, Dec. 2011.

[12] S. Negahban, S. Oh, and D. Shah, "Rank centrality: Ranking from pairwise comparisons," *Operations Res.*, vol. 65, no. 1, pp. 266–287, 2017.

[13] ——, "Iterative ranking from pair-wise comparisons," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Lake Tahoe, CA, Dec. 2012.

[14] M. R. O'Shaughnessy and M. A. Davenport, "Localizing users and items from paired comparisons," in *Proc. IEEE Int. Work. on Machine Learning for Signal Processing (MLSP)*, Vietri sul Mare, Italy, Sep. 2016.

[15] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. Kriegman, and S. Belongie, "Generalized non-metric multidimensional scaling," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, San Juan, Puerto Rico, 2007.

[16] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. Kalai, "Adaptively learning the crowd kernel," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Bellevue, WA, Jun.–Jul. 2011.

[17] I. Steinwart and A. Christmann, *Support Vector Machines*. New York: Springer, 2008.

[18] G. Ongie, D. Pimentel-Alarcón, L. Balzano, R. Willett, and R. D. Nowak, "Tensor methods for nonlinear matrix completion," *SIAM J. Math. Data Sci.*, vol. 3, no. 1, pp. 253–279, 2021.

[19] A. Nemirovski, C. Roos, and T. Terlaky, "On maximization of quadratic form over intersection of ellipsoids with common center," *Math. Program.*, vol. 86, pp. 463–473, 1999.

[20] E. J. Candès, T. Strohmer, and V. Voroninski, "PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming," *Commun. Pure Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, 2012.

[21] W.-K. Ma, P.-C. Ching, and Z. Ding, "Semidefinite relaxation based multiuser detection for M-ary PSK multiuser systems," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 2862–2872, 2004.

[22] Z.-Q. Luo, W.-K. Ma, A. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, 2010.

[23] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs I: The method of paired comparisons," *Biometrika*, vol. 39, no. 3-4, pp. 324–345, 1952.

[24] J. E. Menke and T. R. Martinez, "A bradley–terry artificial neural network model for individual ratings in group competitions," *Neural Comput. Appl.*, vol. 17, pp. 175–186, 2008.

[25] M. C. Grant and S. P. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.1*, `http://cvxr.com/cvx`, Mar. 2014.

[26] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. D. Blondel, S. P. Boyd, and H. Kimura, Eds., Springer, 2008, pp. 95–110.

[27] A. K. Massimino and M. A. Davenport, "As you like it: Localization via paired comparisons," *J. Mach. Learn. Res.*, vol. 22, 2021.