LEARNING SHARED RANKINGS FROM MIXTURES OF NOISY PAIRWISE COMPARISONS

Weicong Ding, Prakash Ishwar, Venkatesh Saligrama

Department of Electrical and Computer Engineering, Boston University, Boston, MA, USA.
{dingwc,pi,srv}@bu.edu

ABSTRACT

We propose a novel model for rank aggregation from pairwise comparisons which accounts for a heterogeneous population of inconsistent users whose preferences are different mixtures of multiple shared ranking schemes. By connecting this problem to recent advances in the non-negative matrix factorization (NMF) literature, we develop an algorithm that can learn the underlying shared rankings with provable statistical and computational efficiency guarantees. We validate the approach using semi-synthetic and real world datasets.

Index Terms— Rank aggregation, nonnegative matrix factorization, extreme point finding, random projection

1. INTRODUCTION

The classical rank aggregation problem aims to generate a single "good" ranking of all items from partial rankings (e.g., pairwise comparisons) provided by a population of users. This type of problem arises in social choice, recommendation systems, meta search, ad placement, etc. [1–13].

The problem of estimating rankings from *pairwise comparisons* data has been extensively studied. A prominent setting is one in which individual user rankings in a homogeneous population are modeled as independent drawings from a probability distribution over all rankings which is centered around a *single* global ranking and decays with some notion of distance from the global ranking. Efficient algorithms have been developed to estimate the global ranking under a variety of probabilistic models [3–7, 11–13].

In order to account for the heterogeneity in the user population, [1, 2] considered models with *multiple* prevalent rankings and proposed consistent combinatorial algorithms for estimating the rankings. The mixture of Mallows model that was recently studied in [14, 15] also considers multiple constituent rankings. In all these multiple rankings settings, however, each user is associated with only one ranking sampled from the mixture model and each user is viewed as being consistent across time in generating all pairwise comparisons. While this model captures shared factors in a user population that may influence user behavior, it does not capture ranking inconsistencies of users across time especially for very similar items. To capture this effect, in Sec. 2 we propose a new model which accounts for a heterogeneous population of inconsistent users whose preferences are different mixtures of multiple shared ranking schemes. The new model subsumes those in [1, 2] as special cases. We develop a computationally and statistically efficient algorithm to consistently estimate the shared rankings in Sec. 3, and demonstrate competitive performance on semi-synthetic and real-world datasets in Sec. 4.

This article is based upon work supported by the U.S. AFOSR and the U.S. NSF under award numbers # FA9550-10-1-0458 (subaward # A1795) and # 1218992 respectively. The views and conclusions contained in this article are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied,of the agencies.

2. MIXTURE OF SHARED RANKINGS

Consider a universe of Q items $\mathcal{U} := \{1, \ldots, Q\}$, e.g., movies from Netflix or products from Amazon. Let $\mathcal{P} = \{\{i, j\} : i < j, i, j \in \mathcal{U}\}$ be the set of all the *unordered* pairs of items. We consider a population of M users in which each user compares $N \geq 2$ pairs of items. [1] We denote the n-th comparison result of user m: $w_{m,n}$, by an *ordered* pair (i, j), if user m compares item i and j and prefers i over j. The choice model for each user, although being distinct, is modeled as arising from a *probabilistic mixture* of K total rankings over the Q items that are *shared* among the M users. Let $\beta_1, \ldots, \beta_K$ denote the K prevailing total rankings as permutations of the Q items, and let the probability vector $\boldsymbol{\theta}_m$ denote the user-specific weights over the K rankings for user m. We adopt the convention that $\beta_k(i)$ is the position of item i in the ranking β_k and item i is preferred over j if $\beta_k(i) < \beta_k(j)$. The generative model for the pairwise comparisons from each user $m = 1, \ldots, M$ is as follows,

1. Sample a K dimensional weight vector $\boldsymbol{\theta}_m$ from a prior distribution $\Pr(\theta)$;
2. For each comparison $n = 1, \ldots, N$:
   (a) Sample a pair of items $\{i, j\}$ from μ.
   (b) Sample $z_{m,n} \in \{1, \ldots, K\} \sim \text{Multinomial}(\boldsymbol{\theta}_m)$.
   (c) If $\beta_{z_{m,n}}(i) < \beta_{z_{m,n}}(j)$, then $w_{m,n} = (i, j)$, otherwise $w_{m,n} = (j, i)$.

[1] We assume that pairs of items i, j for comparison are independently drawn from a distribution μ on $\mathcal{P}$ and $\mu_{i,j} > 0$ for all i, j pairs.

For convenience, we represent the K rankings by a $W \times K$ nonnegative ranking matrix $\boldsymbol{\beta}$ whose $W = Q(Q-1)$ rows are indexed by all the ordered pairs (i, j). The entries of $\boldsymbol{\beta}$ are determined as $\beta_{(i,j),k} = \mathbb{I}(\beta_k(i) < \beta_k(j))$. The k-th column of $\boldsymbol{\beta}$ is therefore an equivalent representation of the ranking β_k. We denote by $\mathbf{P}$ a $W \times W$ diagonal matrix with the (i, j)-th diagonal component $P_{(i,j),(i,j)} = \mu_{i,j}$. We then denote by $\boldsymbol{\theta}$ the $K \times M$ dimensional weight matrix whose columns are the mixing weights $\boldsymbol{\theta}_m$ of each user over the K rankings. Finally, we denote by $\mathbf{X}$ the $W \times M$ empirical comparisons-by-user matrix where $X_{(i,j),m}$ denotes the number of times that user m compares pair $\{i, j\}$ and prefers item i over j. Then, given $\mathbf{X}$ and K, our primary goal is to estimate the ranking matrix $\boldsymbol{\beta}$, i.e., the rankings $\beta_1, \ldots, \beta_K$.

3. AN NMF APPROACH TO LEARNING

We connect the problem of estimating the ranking matrix $\boldsymbol{\beta}$ from the empirical aggregate comparisons matrix $\mathbf{X}$ to an NMF problem by examining the asymptotic structure of the empirical second-order moments of the columns of $\mathbf{X}$, i.e., a co-occurrence matrix of pairwise comparisons. Specifically, let $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{X}}'$ be obtained from $\mathbf{X}$ by first randomly partitioning the set of all comparisons of each user into two disjoint equal-sized subsets (which are then independent by construction) and then re-scaling the rows to make them row-stochastic. Using the results in [16], it can be shown that

$$M\widetilde{\mathbf{X}}'\widetilde{\mathbf{X}}^\top \xrightarrow[\text{almost surely}]{M\to\infty} \bar{\boldsymbol{B}}\bar{\mathbf{R}}\bar{\boldsymbol{B}}^\top =: \mathbf{E}, \qquad (1)$$

where $\bar{\boldsymbol{B}} = \operatorname{diag}^{-1}(\boldsymbol{B}\mathbf{a})\boldsymbol{B}\operatorname{diag}(\mathbf{a})$, $\boldsymbol{B} = \mathbf{P}\boldsymbol{\beta}$, $\bar{\mathbf{R}} = \operatorname{diag}^{-1}(\mathbf{a})\mathbf{R}\operatorname{diag}^{-1}(\mathbf{a})$, and $\mathbf{a}$ and $\mathbf{R}$ are, respectively, the $K \times 1$ expectation and $K \times K$ correlation matrix of the weight vector $\boldsymbol{\theta}_m$. To exclude redundant rankings and ensure unique identifiability, $\bar{\mathbf{R}}$ is assumed to have full rank.

A new approach to efficiently and consistently estimate $\boldsymbol{B}$ from a consistent estimate of the $W \times W$ dimensional matrix $\mathbf{E}$ has emerged in a recent line of work on nonnegative matrix factorization [16–19]. A key ingredient of this approach is the so-called *separability condition* which in our context of rankings translates to the condition that for each $k = 1, \ldots, K$, there exists some row, i.e., some ordered pair (i, j), such that $\beta_{(i,j),k} > 0$ and $\beta_{(i,j),l} = 0, \forall\, l \neq k$. In other words, for each ranking, there exist at least one "novel" pair of items $\{i, j\}$ such that i is uniquely preferred over j in that ranking while j is ranked higher than i in all the other rankings. When this property holds, we say that the $W \times K$ ranking matrix $\boldsymbol{\beta}$ is separable. The separability of $\boldsymbol{\beta}$ is equivalent to the separability of $\boldsymbol{B}$. Figure 1 shows an example of a separable ranking matrix $\boldsymbol{\beta}$ with $Q = 3, W = 6$ and $K = 3$ rankings. The ordered pair $(1, 3)$ is novel to ranking $\boldsymbol{\beta}_1$, the pair $(2, 1)$ to $\boldsymbol{\beta}_2$, and the pair $(3, 2)$ to $\boldsymbol{\beta}_3$. The separability condition has also appeared, albeit implicitly in a different form, in [1, 2] within the context of rank aggregation. Similar to the results in [1, 2] it can be shown that the separability condition will be satisfied with high probability for all $K \ll Q$ when the K underlying rankings are sampled uniformly from the set of all $Q!$ permutations. We have observed such conditions to hold approximately in our experiments.

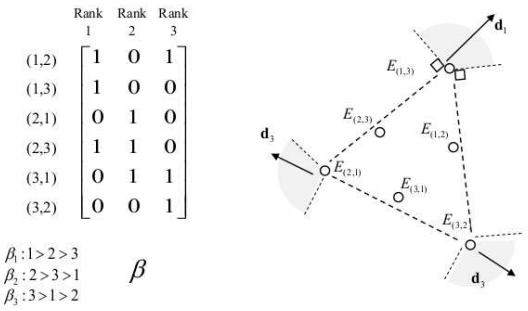

Fig. 1. A separable ranking matrix and the underlying geometric structure for the row vectors of $\mathbf{E}$.

If the separability condition is satisfied and $\bar{\mathbf{R}}$ has full rank, then the row vectors of $\mathbf{E}$ have an intriguing geometric property which is illustrated in Fig. 1. The rows of $\mathbf{E}$ that correspond to novel pairs are the extreme points of the convex hull formed by all the row vectors of $\mathbf{E}$. Once the novel pairs of K distinct rankings are detected, the ranking matrix $\boldsymbol{\beta}$ can be estimated in a straightforward manner by expressing the non-novel rows of $\mathbf{E}$ as convex combinations of the novel rows via least squares [16, 17]. We adopt the approach proposed in [16, 19] to efficiently detect all the novel pairs using random projections. The main steps are outlined in Algorithm 1 and expanded in detail in Algorithms 2, 3, and 4.

As in [16, 19], Algorithms 2 and 3 produce estimates of $\boldsymbol{B}$ as $\widehat{\boldsymbol{B}}$. To obtain an estimate of the binary ranking matrix $\boldsymbol{\beta}$, we note that since $\boldsymbol{B} = \mathbf{P}\boldsymbol{\beta}$, $\mu_{i,j} = \mu_{j,i}$, and $\beta_{(i,j),k} + \beta_{(j,i),k} = 1$, therefore

$$\beta_{(i,j),k} = \frac{B_{(i,j),k}}{B_{(i,j),k} + B_{(j,i),k}}. \qquad (2)$$

This motivates Algorithm 4 that produces $\widehat{\boldsymbol{\beta}}$ as an estimate of the ranking matrix $\boldsymbol{\beta}$. For a finite number of users M, the estimate $\widehat{\boldsymbol{\beta}}$ is not guaranteed to be a binary matrix. We obtain a binary matrix by simply rounding each element to 0 or 1. The proposed approach inherits the asymptotic consistency and statistical and computational efficiency properties as in [16, 20]. Formally, Suppose that $\boldsymbol{\beta}$ is separable and $\bar{\mathbf{R}}$ is full rank. Then Algorithm 1 is polynomial in all model parameters, and consistently recovers $\boldsymbol{\beta}$ element-wise up to a column permutation as the number of users $M \to \infty$.

4. EXPERIMENTAL RESULTS

4.1. Semi-synthetic simulation

We first use a semi-synthetic dataset to validate the performance of the proposed approach. We focus the collaborative filtering application where the mixture models have demonstrated superior performance [21]. In order to match the dimensionality and the other characteristics that are representative of real-world examples, we generate the semi-synthetic

Algorithm 1 Ranking Recovery (Main Steps)

Input: Pairwise comparisons $\widetilde{\mathbf{X}}$, $\widetilde{\mathbf{X}}'(W \times M)$; Number of rankings K; Number of projections P; Tolerance parameters $\zeta, \epsilon > 0$.
Output: Ranking matrix estimate $\widehat{\boldsymbol{\beta}}$.
Set of Novel Pairs $\mathcal{I} \leftarrow$NovelPairDetect($\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}', K, P, \zeta$)
$\hat{\boldsymbol{B}} \leftarrow$EstimateRankings($\mathcal{I}, \mathbf{X}, \epsilon$)
$\hat{\boldsymbol{\beta}} \leftarrow$PostProcess($\hat{\boldsymbol{B}}$)

Algorithm 2 NovelPairDetect (via Random Projections)

Input: $\widetilde{\mathbf{X}}$, $\widetilde{\mathbf{X}}'$; number of rankings K; number of projections P; tolerance ζ;
Output: The set of all novel pairs of K distinct rankings $\mathcal{I}$.
$\widehat{\mathbf{E}} \leftarrow M\widetilde{\mathbf{X}}'\widetilde{\mathbf{X}}^\top$
$\forall (i,j)$, $\mathcal{J}_{(i,j)} \leftarrow \{(s,t) : \widehat{E}_{(i,j),(i,j)} - 2\widehat{E}_{(i,j),(s,t)} + \widehat{E}_{(s,t),(s,t)} \geq \zeta/2\}$,
for $r = 1, \ldots, P$ **do**
Sample $\mathbf{u}_r \in \mathbb{R}^W$ from an isotropic prior
$\hat{q}_{(i,j),r} \leftarrow \mathbb{I}\{\forall (s,t) \in \mathcal{J}_{(i,j)}, \widehat{\mathbf{E}}_{(s,t)}\mathbf{u}_r \leq \widehat{\mathbf{E}}_{(i,j)}\mathbf{u}_r\}$, $\forall (i,j)$
end for
$\hat{q}_{(i,j)} \leftarrow \frac{1}{P}\sum_{r=1}^{P} \hat{q}_{(i,j),r}, \forall (i,j)$
$k \leftarrow 0$,$l \leftarrow 1$, and $\mathcal{I} \leftarrow \emptyset$
while $k \leq K$ **do**
$(s,t) \leftarrow$ index of the l^{th} largest value among $\hat{q}_{(i,j)}$'s
if $(s,t) \in \bigcap_{(i,j)\in\mathcal{I}} \mathcal{J}_{(i,j)}$ **then**
$\mathcal{I} \leftarrow \mathcal{I} \cup \{(s,t)\}$, $k \leftarrow k+1$
end if
$l \leftarrow l + 1$
end while

Algorithm 3 EstimateRankings

Input: $\mathcal{I} = \{(i_1, j_1), \ldots, (i_K, j_K)\}$ the set of novel pairs of K topics; $\mathbf{X}$, $\mathbf{X}'$; precision ϵ
Output: $\widehat{\boldsymbol{B}}$: the estimate of $\boldsymbol{B}$.
$\mathbf{Y} = (\widetilde{\mathbf{X}}_{(i_1,j_1)}^\top, \ldots, \widetilde{\mathbf{X}}_{(i_K,j_K)}^\top)^\top$,
$\mathbf{Y}' = (\widetilde{\mathbf{X}}_{(i_1,j_1)}'^\top, \ldots, \widetilde{\mathbf{X}}_{(i_K,j_K)}'^\top)^\top$
for all (i,j) pairs **do**
Solve $\widehat{\boldsymbol{\beta}}_{(i,j)} \leftarrow (\frac{1}{M}\mathbf{X}_{(i,j)}\mathbf{1}) \arg\min_{\mathbf{b}} M(\widetilde{\mathbf{X}}_{(I,J)} - \mathbf{b}\mathbf{Y})(\widetilde{\mathbf{X}}'_{(I,J)} - \mathbf{b}\mathbf{Y}')^\top$
Subject to $b_k \geq 0, \sum_{k=1}^{K} b_k = 1$, With precision ϵ
end for
$\widehat{\boldsymbol{B}} \leftarrow$column normalize $\widehat{\boldsymbol{\beta}}$

Algorithm 4 Post Processing

Input: $\widehat{\boldsymbol{B}}$
Output: $\widehat{\boldsymbol{\beta}}$ as the estimate of $\boldsymbol{\beta}$
$\widehat{\boldsymbol{\beta}}_{(i,j),k} \leftarrow \frac{\widehat{\boldsymbol{B}}_{(i,j),k}}{\widehat{\boldsymbol{B}}_{(i,j),k} + \widehat{\boldsymbol{B}}_{(j,i),k}}$
$\widehat{\boldsymbol{\beta}}_{(i,j),k} \leftarrow$ Round$[\widehat{\boldsymbol{\beta}}_{(i,j),k}]$

pairwise comparisons dataset using a benchmark movie star-ratings dataset, Movielens [22] which has approximately 1 million ratings for 3952 movies from $M = 6040$ users.

We follow [4] and [14] and generate the semi-synthetic dataset as follows. We consider the $Q = 100$ most frequently rated movies and train a latent factor model on the star-ratings data using a state-of-the-art probabilistic matrix factorization algorithm [21, 23]. This approach is selected for its state-of-the-art performance on many real world collaborative filtering tasks. This procedure learns a $Q \times K$ movie-factor matrix whose columns are interpreted as scores of Q movies over the K latent factors[4, 21]. By sorting the scores of each column of the movie-factor matrix, we obtain K rankings for generating the semi-synthetic dataset. We set $K = 10$ as suggested by [14, 21]. We note that the resulting ranking matrix $\boldsymbol{\beta}$ satisfies the separability condition.

To generate the semi-synthetic data, we simply set $\mu_{i,j} = 1/\binom{Q}{2}$ so that all the pairs are equally likely. We adopt the Dirichlet prior for $\boldsymbol{\theta}_m$ as suggested by [14]. The correlation matrix $\mathbf{R}$ of the Dirichlet distribution has full rank [16, 18]. The parameters α_k's of the Dirichlet distribution $\Pr(\boldsymbol{\theta}_m|\boldsymbol{\alpha}) = \frac{1}{C}\prod_{k=1}^{K} \theta_k^{\alpha_k}$, are determined by $\alpha_k = \alpha_0 a_k$. The probability vector $\mathbf{a} = [a_1, \ldots, a_K]^\top$ is the expectation of $\boldsymbol{\theta}$ and $\alpha_0 > 0$ controls the sparsity of $\boldsymbol{\theta}_m$. We set $\alpha_0 = 0.1$ and sample $\mathbf{a}$ uniformly from the $K = 10$ dimensional simplex for each random realization. We fix $N = 300$ comparisons per user to approximate the observed average pairwise comparisons in the Movielens dataset and vary M.

We measure the performance by the standard ℓ_1 error between the ground truth ranking matrix $\boldsymbol{\beta}$ and the estimate $\widehat{\boldsymbol{\beta}}$. Since the output of the proposed approach is determined only up to a column permutation, we use bipartite matching based on ℓ_1 distance to match the columns of $\boldsymbol{\beta}$ and $\widehat{\boldsymbol{\beta}}$. We note that due to the way $\boldsymbol{\beta}$ is defined, the ℓ_1 error metric is equivalent to the widely-used *Kendall's tau distance* between two rankings which is proportional to the number of pairs in which two ranking schemes differ. We further normalize the ℓ_1 error by $W = Q \times (Q-1)$ so that the error measure for each column is a number between $[0, 1]$.

We compared our proposed algorithm (denoted by RP) against the algorithm proposed in [1, 2] (denoted by FJS) for estimating the ranking matrix. This is the most recent algorithm with consistency guarantees for $K > 1$. We compared how the estimation error varies with the number of users M. For each setting, we average over 10 Monte Carlo runs. The estimation errors as a function of the number of users M are depicted in Fig. 2. Evidently, our algorithm shows superior performance over FJS. More specifically, since our ground truth ranking matrix is separable, as M increases, the estimation error of RP converges to zero, and the convergence is

much faster than FJS. We note that only for $M \geq 100,000$ does the error of the FJS algorithm eventually start approaching 0.

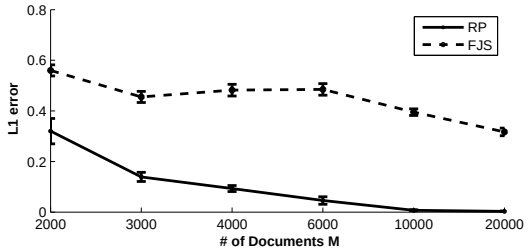

Fig. 2. The normalized ℓ_1 errors of the ranking matrices, as functions of M, estimated by RP and FJS from the semi-synthetic dataset with $Q = 100, N = 300, K = 10$.

4.2. Movielens dataset

In this section, we apply the proposed algorithm to the real-world Movielens dataset introduced in Sec. 4.1. We consider two tasks: (1) new rating prediction, and (2) new user prediction. We focus on the $Q = 100$ most frequently rated movies as in Sec. 4.1 and obtain a subset of $183,000$ star-ratings from $M = 5940$ users. To generate pair-wise comparisons from the star-rating data, for each user m, we **select** pairs of movies i, j that user m rated, and **compare** the star-ratings of the two movies to generate comparisons, as motivated by [4, 14].

To **select** pairs of items to compare, we consider two methods: (a)(Full) all pairs of movies that a user has rated, or (b)(Partial) randomly select $5N_{star,m}$ pairs of items where $N_{start,m}$ is the number of movies user m has rated.

To **compare** a pair of movies for a user, $w_{m,n} = (i, j)$ if the star-rating of i is higher than j. To deal with the case when the two ratings tie, we consider three strategies: (i)(Both) generate both $w_{m,1} = (i, j)$ and $w_{m,2} = (j, i)$, (ii) (Ignore) generate no comparison, and (iii) (Random) randomly generate one of $w_{m,1} = (i, j)$ and $w_{m,2} = (j, i)$ with equal probability.

New comparison prediction: this task is to predict new comparisons for users whose comparisons have been observed in the training set. We followed the training/testing split as in [21, 23]. In such a split, both training and testing data contain ratings from all the $M = 5940$ users. We convert both the training and testing star-rating data to pairwise comparisons independently.

We evaluate the performance by the predictive log-likelihood of the testing data, i.e., $\Pr(\mathbf{w}_{test}|\mathbf{w}_{train}, \widehat{\boldsymbol{\beta}})$. Given the estimate $\widehat{\boldsymbol{\beta}}$, we follow [16, 18] to learn a Dirichlet prior model. We then apply the Gibbs Sampling based approximation proposed in [24, 25] to calculate the prediction log-likelihood.

We compare against the rankings estimated by the FJS algorithm. Figure 3(upper) summarizes the results for different strategies in generating the pairwise comparisons with $K = 10$ held fixed. The log-likelihood is normalized by the total number of pairwise comparisons in the corresponding testing set. As depicted in Fig. 3 (upper), the log-likelihood produced by the proposed algorithm RP is higher, by a large margin, compared to FJS. The predictive accuracy is robust to how the comparison data is constructed.

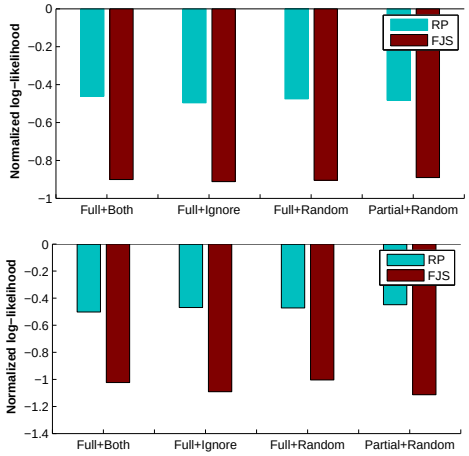

Fig. 3. The Normalized log-likelihood for $K = 10$ under different settings for (upper) new comparison prediction and (lower) new user prediction on the Movielens dataset.

We also consider the performance as function of K (Fig. 4). The results validate the superior performance and suggest $K = 10$ is a reasonable parameter choice.

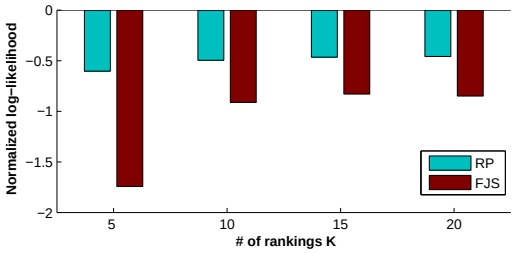

Fig. 4. The normalized log-likelihood for Full+Ignore construction strategy for different settings of K for new comparison prediction in the Movielens dataset.

New user prediction: this task is to predict the comparisons of new users. Following [14], we split the first 4000 users in the Movielens dataset for training, and the remaining 2040 users for testing. We focus on the $Q = 100$ most frequently rated movies and obtain 3986 users in training and 1994 in testing that each has rated at least two movies. We use the held-out log-likelihood, i.e., $\Pr(\mathbf{w}_{test}|\widehat{\boldsymbol{\beta}})$ to measure the performance. The log-likelihoods are calculated using the approximation method proposed in [24]. We compare our algorithm RP with the FJS algorithm as in the previous task. The log-likelihoods are then normalized by the total number of comparisons in the testing phase. Motivated by the results in the previous task, we fix the number of rankings at $K = 10$. The results which are summarized in Fig. 3 (lower) are consistent with the results for the previous task.

References

[1] S. Jagabathula and D. Shah. Inferring rankings under constrained sensing. In *Advances in Neural Information Processing Systems*, pages 753–760. Vancouver, Canada, Dec. 2008.

[2] V. Farias, S. Jagabathula, and D. Shah. A data-driven approach to modeling choice. In *Advances in Neural Information Processing Systems*, pages 504–512. Vancouver, Canada, Dec. 2009.

[3] A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proc. of the 31st International Conference on Machine Learning*, pages 118–126, Beijing, China, Jun. 2014.

[4] M. Volkovs and R. Zemel. New learning methods for supervised and unsupervised preference aggregation. *Journal of Machine Learning Research*, 15:1135–1176, 2014.

[5] D. F. Gleich and L.-H. Lim. Rank aggregation via nuclear norm minimization. In *Proc. of the 17th ACM International Conference on Knowledge Discovery and Data Mining*, pages 60–68, San Diego, CA, USA, 2011.

[6] H. Azari Soufiani., W. Chen, D. Parkes, and L. Xia. Generalized method-of-moments for rank aggregation. In *Advances in Neural Information Processing Systems*, pages 2706–2714. Lake Tahoe, NV, USA, Dec. 2013.

[7] B. Osting, C. Brune, and S. Osher. Enhanced statistical rankings via targested data collection. In *Proc. of the 30th International Conference on Machine Learning*, pages 489–497, Atlanta, GA, USA, Jun. 2013.

[8] I. Mitliagkas, A. Gopalan, C. Caramanis, and S. Vishwanath. User rankings from comparisons: Learning permutations in high dimensions. In *Communication, Control, and Computing, 2011 49th Annual Allerton Conference on*, pages 1143–1150, Sept 2011.

[9] J. I. Marden. *Analyzing and modeling rank data*. Chapman and Hall, 1995.

[10] E. Zermelo. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1929.

[11] C. L Mallows. Non-null ranking models. i. *Biometrika*, pages 114–130, 1957.

[12] L. Ford. Solution of a ranking problem from binary comparisons. *American Mathematical Monthly*, pages 28–33, 1957.

[13] R. Bradley and M. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, pages 324–345, 1952.

[14] T. Lu and C. Boutilier. Effective sampling and learning for Mallows models with pairwise preference data. *Journal of Machine Learning Research*, 2014.

[15] P. Awasthi, A. Blum, O. Sheffet, and A .Vijayaraghavan. Learning mixtures of ranking models. In *Advances in Neural Information Processing Systems*, pages 2609–2617. Montreal, Canada, Dec. 2014.

[16] W. Ding, M. H. Rohban, P. Ishwar, and V. Saligrama. Efficient distributed topic modeling with provable guarantees. In *Proc. ot the 17th International Conference on Artificial Intelligence and Statistics*, pages 167–175, Reykjavik, Iceland, Apr. 2014.

[17] S. Arora, R. Ge, and A. Moitra. Learning topic models – going beyond SVD. In *Proc. of the IEEE 53rd Annual Symposium on Foundations of Computer Science*, New Brunswick, NJ, USA, Oct. 2012.

[18] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M.l Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proc. of the 30th International Conference on Machine Learning*, pages 280–288, Atlanta, GA, USA, Jun. 2013.

[19] W. Ding, M. H. Rohban, P. Ishwar, and V. Saligrama. Topic discovery through data dependent and random projections. In *Proc. of the 30th International Conference on Machine Learning*, pages 1202–1210, Atlanta, GA, USA, Jun. 2013.

[20] W. Ding, P. Ishwar, M. H. Rohban, and V. Saligrama. Necessary and sufficient conditions for novel word detection in separable topic models. In *Advances in on Neural Information Processing Systems (NIPS), Workshop on Topic Models: Computation, Application*, Lake Tahoe, NV, USA, Dec. 2013.

[21] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proc. of the 25th International Conference on Machine Learning*, pages 880–887, Helsinki, Finland, Jun. 2008.

[22] Movielen dataset. `http://grouplens.org/datasets/movielens/`.

[23] `http://www.cs.toronto.edu/~rsalakhu/BPMF.html`.

[24] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proc. of the 26th International Conference on Machine Learning*, pages 1105–1112, Montreal, Canada, Jun. 2009.

[25] `http://people.cs.umass.edu/~wallach/code/etm/`.