A Grouped Ranking Model for Item Preference Parameter

Hideitsu Hino¹, Yu Fujimoto², and Noboru Murata¹
¹School of Science and Engineering, Waseda University.
3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555, Japan
²Department of Integrated Information Technology, Aoyama Gakuin University.
5-10-1-O512b, Fuchinobe, Sagamihara, Kanagawa 229-8558, Japan

Keywords: Probabilistic Models, Ranking Models, em Algorithm, Mixture Models, Data Visualization, Collaborative Filtering

Abstract

Given a set of rating data for a set of items, determining preference levels of items is a matter of importance. Various probability models have been proposed to solve this task. One of such models is the Plackett-Luce model, which parametrizes preference level of each item by a real value. In this paper, the Plackett-Luce model is generalized to cope with grouped ranking observations such as movie or restaurant ratings. Since it is difficult to maximize the likelihood of the proposed model directly, a feasible approximation is derived, and the em algorithm is adopted to find the model parameter by maximizing the approximate likelihood which is easily evaluated. The proposed model is extended to a mixture model, and two applications are proposed. To show the effectiveness of the proposed model, numerical experiments with real-world data are carried out.

1 Introduction

Over the years, there have been a number of efforts to construct models for item rankings or ordered objects evaluated by various judges. These models are classified into two groups: the first group focuses on the permutational order of items, and the second group focuses on the intrinsic preference levels of items. Well known examples of the former models are found in Mallows (1957) and Fligner and Verducci (1986). Probability distributions on permutations are well studied by Diaconis (1988), and still being an active research area (e.g., Murphy and Martin, 2003; Meila et al., 2007; Huang et al., 2007; Lebanon and Mao, 2007; Busse et al., 2007). As an example of the latter, Bradley and Terry (1952) have proposed a model in which each item \( I_i \) has a positive valued parameter \( \theta_i \) and the probability of being chosen item \( I_i \) over item \( I_j \), denoted by \( I_i \succ I_j \), is given by \( P(I_i \succ I_j) = \frac{\theta_i}{\theta_i + \theta_j} \). The Bradley-Terry model is an popular analytical tool in the machine learning community (e.g., Hastie and Tibshirani, 1998; Huang et al., 2006; Takenouchi and Ishii, 2008). A natural extension of the Bradley-Terry model is given by Plackett (1975). He has proposed a sequential ranking model which we refer to as
the Plackett-Luce model henceforth. In the Plackett-Luce model, the probability of observing the ranking \((I_{a(1)} \succ I_{a(2)} \succ \cdots \succ I_{a(N)})\) is defined with the item preference level parameter \(\theta = \{\theta_i\}_{i=1}^N\) for \(N\) items such that \(\sum_{i=1}^N \theta_i = 1\) as

\[
P(I_{a(1)} \succ I_{a(2)} \succ \cdots \succ I_{a(N)}) = \frac{\theta_{a(1)}}{\sum_{j=1}^N \theta_{a(j)}} \frac{\theta_{a(2)}}{\sum_{j=2}^N \theta_{a(j)}} \cdots \frac{\theta_{a(N-1)}}{\theta_{a(N-1)} + \theta_{a(N)}} \\
= \prod_{i=1}^{N-1} \frac{\theta_{a(i)}}{\sum_{j=i}^N \theta_{a(j)}},
\]

where \(a(j)\) denotes the index of the item that occupies the \(j\)-th position in the ranking. This model reflects the assumption that items with high preference parameters tend to be chosen in the early stage of the item selection process. For the Plackett-Luce model, Hunter (2004) gave a lower bound of the log likelihood function and proposed an iterative algorithm to maximize the lower bound.

When we consider the case that a number of judges (users, henceforth) rate several items in finite discrete levels, some items are given the same rating. Recently, the importance of analyzing this kind of data is growing with the prevalence of word-of-mouth data such as movies, books, and restaurants for example. In this paper, we generalize the Plackett-Luce model to cope with grouped ranking observations, where \(M\) rankings are used to evaluate \(N\) items, \(M \leq N\). Our basic idea is that there is a latent order in a set of the same rated items, but we only observe \(M\) groups of items which are divisions of \(N\) items. For example, when 7 items \(I = \{I_1, \ldots, I_7\}\) are rated with \(M = 3\), we get a grouped ranking observation from a user \(u\) such as \(D^u = \{G_u^1, G_u^2, G_u^3\}\), \(G_u^1 = \{3, 5\}\), \(G_u^2 = \{2, 6, 7\}\), \(G_u^3 = \{1, 4\}\), where \(G_u^m\) is an index set of items rated at the \(m\)-th ranking by user \(u\). Those observations are available from \(U\) users, i.e., we get a set of observations \(\{D^u\}_{u=1}^U\). We assume that each of \(N\) items has a preference level \(\theta_i\) such that \(\theta_i > 0\), \(\sum_{i=1}^N \theta_i = 1\), and our goal is to estimate the parameter \(\theta = (\theta_1, \ldots, \theta_N)\) using only a set of the grouped ranking observations \(\{D^u\}_{u=1}^U\).

According to Marden (1995, chap.5), formal modeling of ranking data can be divided into two types:

1. modeling of the ranking process,
2. modeling of the population of rankers.

Since a single Plackett-Luce model will not account for the difference of ranking process of every user, it is natural to consider a mixture model. We formalize a mixture of our proposed models, and propose two applications, i.e., data-user visualization and collaborative filtering.

The rest of this paper is organized as follows. In Section 2, we formally define our grouped ranking model and its log likelihood function. Since it is computationally difficult to directly evaluate the likelihood of this model, we derive its practical approximation. In Section 3, we propose an iterative algorithm to estimate the preference parameter which has a structure of the \(em\) algorithm \(^1\) (Amari, 1995; Amari and Nagaoka, 2000). In Section 4, we consider a mixture of our proposed models. Its applications and two experimental results are shown in Section 5. A short and preliminary version of this paper appeared in PAKDD 2009 (Hino et al., 2009).

2 Grouped Ranking Model

In this section, we define data of interest, i.e., grouped ranking data, and derive a probabilistic model for the grouped ranking data.

\(^1\)The \(em\) algorithm is an information geometric formalization of the EM algorithm (Dempster et al., 1977).
2.1 Model Description and Likelihood Function

Suppose that \( U \) users independently give discrete ratings between 1 and \( M \) to \( N \) items \( I_1, \ldots, I_N \). Let \( G_1^u \) be an index set of the most preferred items by user \( u \), and \( G_2^u \) be that of the next preferred items, and so on. All the index sets for \( M \) rankings by user \( u \) is denoted by \( D^u = \{G_1^u, \ldots, G_M^u\} \), where \( G_m^u = \{i \mid I_i \in m\text{-th group}\} \), and we let \( \gamma_m^u = |G_m^u| \). In this paper, \( D^u \) is called a grouped ranking observation from user \( u \) henceforth. Although we do not know the preference order of the same rated items by user \( u \), we assume that there is a latent ordering within each group. When we need to consider the order of items in a group \( G_m^u \), we use the action of a permutation \( \pi_m^u \) on \( G_m^u \). Let \( \pi_m^u(i) \) be the index of the \( i \)-th item in the ordered group \( \pi_m^u(G_m^u) \). For example, by the action of a permutation \( \pi_m^u = (1, 2, 3, 4) \) on a group \( G_m^u = \{2, 6, 7\} \), we get the ordered set \( \pi_m^u(G_m^u) = (7, 2, 6) \) and \( \pi_m^u(1) = 7 \), \( \pi_m^u(2) = 2 \), \( \pi_m^u(3) = 6 \).

By using the example used in Section 1, we will explain details of the model and derive the likelihood function. We have a grouped ranking observation \( D^u = \{G_1^u = \{3, 5\}, G_2^u = \{2, 6, 7\}, G_3^u = \{1, 4\}\} \) for 7 items. We know that any item in \( G_1^u \) is preferred to any item in \( G_2^u \) by this user, however, there is no information for the order of items in the same groups. Suppose the user gave the ranking \( I_7 \succ I_2 \succ I_6 \) in \( G_2^u \), which means \( \pi_2^u = (4, 3, 2, 1) \). We define \( \Theta_m^u = \sum_{i \in G_m^u} \theta_i \), the sum of the parameters \( \theta_i \) whose indices belong to the group \( G_m^u \). We call \( \Theta_m^u \) a grouped parameter of the group \( G_m^u \) henceforth. Since the items in \( G_1^u \) are already chosen and excluded, the probability that the items in \( G_2^u \) are chosen in this order from remaining items is given by

\[
P(\pi_2^u, G_2^u|G_1^u) = \left( \frac{\theta_7}{\sum_{n=2}^5 \Theta_n^u} \right) \left( \frac{\theta_2}{\sum_{n=2}^5 \Theta_n^u - \theta_7} \right) \left( \frac{\theta_6}{\sum_{n=2}^5 \Theta_n^u - (\theta_7 + \theta_2)} \right) = \frac{\gamma_2^u}{\sum_{i=1}^5 \sum_{n=2}^5 \Theta_n^u - \sum_{j<i} \theta_{\pi_2^u(i)}} \quad (2)
\]

where \( P((\pi_m^u, G_m^u)|G_1^u, \ldots, G_{m-1}^u) \) denotes the conditional probability that the items in the group \( G_m^u \) are chosen in the order specified by \( \pi_m^u \) when the items in groups \( \{G_1^u, \ldots, G_{m-1}^u\} \) have been already chosen. Note that the orderings in the previous groups \( G_1^u, \ldots, G_{m-1}^u \) does not influence the ordering \( \pi_m^u \) in the \( m \)-th group.

Now, let us generalize the above example. When we consider the complete observation \( \{G_1^u, G_2^u, \ldots, G_M^u\} \), the permutations \( \pi_m^u \)'s can be regarded as a latent variables. We can decompose the probability of observing this complete ranking data as

\[
P((\pi_1^u, G_1^u), \ldots, (\pi_M^u, G_M^u)) = P((\pi_1^u, G_1^u)|G_1^u)P((\pi_2^u, G_2^u)|G_1^u) \cdots P((\pi_M^u, G_M^u)|G_1^u, \ldots, G_{M-1}^u). \quad (3)
\]

We define the probability of observing the group \( G_m^u \) with the order \( \pi_m^u \) under the condition that groups \( \{G_1^u, \ldots, G_{m-1}^u\} \) have been already chosen as

\[
P((\pi_m^u, G_m^u)|G_1^u, \ldots, G_{m-1}^u) = \prod_{i=1}^M \frac{\theta_{\pi_m^u(i)}}{\sum_{n=m}^M \Theta_n^u - \sum_{j<i} \theta_{\pi_m^u(j)}}. \quad (4)
\]

Noting that the latent order in each group is expressed by the action of a permutation \( \pi_m^u \) on \( G_m^u \), we let \( S(G_m^u) \) be the set of all the possible \( \gamma_m^u! \) permutations of the items on the group.

---

For simplicity, we assume that all users rate all items. We have to modify the proposed algorithm in Section 3 to deal with unrated items. See Appendix 4 for the parameter estimation algorithm with unrated items.
Then the probability of occurrence of the group $G_m^u$ is a sum of the joint probabilities over possible latent orders

$$P(G_m^u | G_1^u, \ldots, G_{m-1}^u) = \sum_{\pi_m^u \in S(G_m^u)} P((\pi_m^u, G_m^u) | G_1^u, \ldots, G_{m-1}^u), \quad (5)$$

which is nothing but marginalization of permutations. We note that when $G_m^u = \emptyset$, we skip the correspondent index $m$ in summation or product by convention. From this probability for the $m$-th rated group of items, the likelihood of the datum $D^u = \{G_1^u, \ldots, G_M^u\}$ is defined as

$$P(D^u) = \prod_{m=1}^{M} \sum_{\pi_m^u \in S(G_m^u)} P((\pi_m^u, G_m^u) | G_1^u, \ldots, G_{m-1}^u), \quad (6)$$

which we call the grouped ranking model henceforth. Further explanation of the generative process of the grouped ranking observations and validation of the proposed model as a probability model is given in Appendix 1. From (4) and (5), we get a log likelihood of a group $G_m^u$ as

$$l(\theta; m, u) = \log \left( \sum_{\pi_m^u \in S(G_m^u)} \prod_{i=1}^{\gamma_m^u} \frac{\theta_{\pi_m^u(i)}}{\sum_{n=m}^{M} \Theta_n^u - \sum_{j<i} \theta_{\pi_m^u(j)}} \right). \quad (7)$$

By summing up $l(\theta; m, u)$ for all the groups and users, we get the likelihood of the given data $\{D^u\}_{u=1}^{U}$ as

$$L(\theta) = \sum_{u=1}^{U} \sum_{m=1}^{M} \log \left( \sum_{\pi_m^u \in S(G_m^u)} \prod_{i=1}^{\gamma_m^u} \frac{\theta_{\pi_m^u(i)}}{\sum_{n=m}^{M} \Theta_n^u - \sum_{j<i} \theta_{\pi_m^u(j)}} \right). \quad (8)$$

Note that maximization of the likelihood for our model is apparently a hard task, in particular for large numbers of items or users. The main complexity comes from latent orderings in the model. The marginalization of the latent ordering requires enumeration of all the possible permutations within the group, and it is computationally intractable even for a modest group size.

### 2.2 Approximation of Log Likelihood

In this subsection, we approximate the log likelihood function (8) to exclude marginalization or permutations. The denominator in the expression (7) reflects normalization of parameters $\theta_i$’s in the sequential item selection. By replacing the denominator in (7) by $\sum_{n=m}^{M} \Theta_n^u$, we get a lower bound of (7) as

$$l(\theta; m, u) = \log \left( \sum_{\pi_m^u \in S(G_m^u)} \prod_{i=1}^{\gamma_m^u} \frac{\theta_{\pi_m^u(i)}}{\sum_{n=m}^{M} \Theta_n^u} \right) \leq l(\theta; m, u). \quad (9)$$

Note that by this replacement, the marginalization in (7) is reduced to a positive constant factor $\log (\gamma_m^u!)$. The effect of dropping the term $\sum_{j<i} \theta_{\pi_m^u(j)}$ becomes less significant when the number of groups is large, because the number of items in the same group will be small. We will support this intuition by showing a simple experiment in Subsection 3.2.
We next consider the upper bound of $\tilde{l}(\theta; m, u)$. Using the arithmetic-geometric mean inequality

$$\left( \prod_{i \in G_m^u} \theta_i \right)^{1/\gamma_m^u} \leq \frac{1}{\gamma_m^u} \sum_{i \in G_m^u} \theta_i = \frac{1}{\gamma_m^u} \Theta_m^u, \quad (10)$$

we get an inequality

$$\gamma_m^u \prod_{i \in G_m^u} \theta_i \leq \gamma_m^u \left( \sum_{n=m}^M \Theta_n^u \right)^{\gamma_m^u} \leq \gamma_m^u \left( \frac{\Theta_m^u/\gamma_m^u}{\sum_{n=m}^M \Theta_n^u} \right)^{\gamma_m^u}. \quad (11)$$

Then we derive an upper bound of $\tilde{l}(\theta; m, u)$ as

$$\tilde{l}(\theta; m, u) = \log \left\{ \frac{\gamma_m^u}{\sum_{n=m}^M \Theta_n^u} \right\}^{\gamma_m^u} \quad (12)$$

$$= \log \gamma_m^u + \gamma_m^u \left( \log \Theta_m^u - \log \gamma_m^u - \log \left( \sum_{n=m}^M \Theta_n^u \right) \right)$$

$$= \gamma_m^u \left\{ \log \Theta_m^u - \log \left( \sum_{n=m}^M \Theta_n^u \right) \right\} + \log \gamma_m^u - \gamma_m^u \log \gamma_m^u.$$

Although $\tilde{l}(\theta; m, u)$ is not exactly a lower bound of the log likelihood $l(\theta; m, u)$, in our preliminary experiments, it is observed that $\tilde{L}(\theta) = \sum_{m=1}^U \sum_{n=m}^M \tilde{l}(\theta; m, u)$ is smaller than the exact log likelihood $L(\theta)$ with quite high probability. Therefore, we have the following approximate expression

$$\tilde{l}(\theta; m, u) \leq l(\theta; m, u) \leq l(\theta; m, u). \quad (13)$$

An experiment to support this approximation is shown in Subsection 3.2.

By summing up $\tilde{l}(\theta; m, u)$ for all the groups and users, we get an approximate log likelihood of the given data as

$$\tilde{L}(\theta) = \sum_{u=1}^U \sum_{m=1}^M \gamma_m^u \left( \log \frac{\Theta_m^u}{\sum_{n=m}^M \Theta_n^u} \right) + \text{const}. \quad (14)$$

This function $\tilde{L}(\theta)$ does not contain enumeration of possible permutations. As $L(\theta) \geq \tilde{L}(\theta)$ holds, we can expect that the maximization of $\tilde{L}(\theta)$ indirectly leads to the maximization of $L(\theta)$.

We can now apply any non-linear optimizer to maximize $\tilde{L}(\theta)$ with respect to $\theta$. However, still direct optimization of $\tilde{L}(\theta)$ may cause two possible problems. The first problem is computational complexity. As the approximation $\tilde{L}(\theta)$ is a non-linear function of $\theta = (\theta_1, \ldots, \theta_N)$, its complexity increases with the number of items $N$. According to our preliminary numerical experiments, the computational time for direct optimization increases super-linearly with the size of items. The second is more serious when on-line processing is required. When a new user gives a new grouped ranking observation $D^u$, we again need to optimize $\tilde{L}(\theta)$ with both the previous observations and the new observation in a direct optimization approach. In the next section, we will propose an alternative approach to estimate the preference parameter.

At the end of this section, it is worth noting that Huang et al. (2006) have mentioned a similar generalization of the Plackett-Luce model, which they call the multiple team comparison model.
They considered the ranking of the grouped items (teams) instead of items itself. In their model, the group ranking probability is defined as

$$ P(G_1^u > G_2^u > \cdots > G_M^u) = \prod_{m=1}^{M} \frac{\Theta_m^u}{\sum_{n=m}^{M} \Theta_n^u}. $$

This reduces to the original case (1) when each $G_m^u$ contains only one item and $M$ equals to $N$.

The log likelihood of this model is given by

$$ L_{\text{Huang}}(\theta) = \sum_{u=1}^{U} \sum_{m=1}^{M} \log \left( \frac{\Theta_m^u}{\sum_{n=m}^{M} \Theta_n^u} \right) = \sum_{u=1}^{U} \sum_{m=1}^{M} \left( \log \Theta_m^u - \log \sum_{n=m}^{M} \Theta_n^u \right), $$

which looks similar to our approximate likelihood (14). The difference between this model and our proposed model is whether the model considers the orderings in the groups. Their model assumes that groups of items are somehow predefined and users only give rankings of those groups. It may suit for estimating individual player’s skill from scores of teams such as football or baseball games where each player plays many times, and possibly sometimes plays in different teams. On the contrary, our model assumes that each user gives both ranking and grouping to a set of items. We intend to model the way users give ratings to movies or restaurants, for example. To avoid enumeration of permutations in groups, we derived the lower bound $l(\theta; m, u)$, and to get closer to the exact likelihood, we gave an upper bound of $\tilde{l}(\theta; m, u)$. Intuitively speaking, the expression (14) for $\tilde{L}(\theta)$ corresponds to the log likelihood of the probability that each user sequentially chooses the $m$-th group $\gamma_m^u$ times like the Plackett-Luce model according to the grouped parameter $\{\Theta_m^u\}_{m=1}^{M}$.

3  Algorithm for Parameter Estimation

In this section, we devise an algorithm for parameter estimation from the viewpoint of information geometry (Amari and Nagaoka, 2000), and show the effectiveness of the algorithm by simple experiments using synthetic data.

3.1  Algorithm Derivation

Our motivation of maximizing the likelihood is to find the preference parameter $\theta$ which is consistent with all the observations as much as possible. For this purpose, we decompose the first term of (14) to $U$ independent optimization problems with respect to group parameters $\Theta_m^u$, which correspond to one grouped ranking observation $D^u = \{G_1^u, \ldots, G_M^u\}$:

$$ \max_{\Theta_m^u} \sum_{m=1}^{M} \gamma_m^u \log \left( \frac{\Theta_m^u}{\sum_{n=m}^{M} \Theta_n^u} \right), \quad \text{subject to} \quad \Theta_m^u > 0, \sum_{m=1}^{M} \Theta_m^u = 1. \tag{15} $$

These problems are relatively small-sized problems with linear constraints, and efficiently solved with arbitrary optimizers. In order to impose the positivity constraint $\Theta_m^u > 0$, we add a log barrier term $\sum_{m=1}^{M} \frac{1}{M} \log \Theta_m^u$ to the objective of (15) and solve

$$ \max_{\Theta_m^u} \sum_{m=1}^{M} \gamma_m^u \log \left( \frac{\Theta_m^u}{\sum_{n=m}^{M} \Theta_n^u} \right) + \sum_{m=1}^{M} \frac{1}{M} \log \Theta_m^u, \quad \text{subject to} \quad \sum_{m=1}^{M} \Theta_m^u = 1. \tag{16} $$

6
instead of (15). This barrier term corresponds to the uniform prior to the objective, because the uniform distribution \( \{ \Theta_m^u \}_{m=1}^M = \{ \frac{1}{M}, \ldots, \frac{1}{M} \} \) maximizes the barrier term. See Hunter (2004) and Huang et al. (2007) for detailed discussion of identifiability and regularization.

Now our problem is finding the optimal parameter \( \theta \) which is the most consistent with \( U \) sets of grouped parameters \( \{ \Theta_m^u \}_{m=1}^M, u = 1, \ldots, U, \) and we will show that it is solved by the \( em \) algorithm in the information geometry literature (Amari, 1995). The algorithm finds a local maximum of likelihood by iterating projections between a probability model space and observation spaces.

The solutions of the optimization problems \( (16) \) can be seen as incomplete observations of the preference parameter. As the preference parameters \( \theta \) have constraints \( \forall i, \theta_i > 0 \) and \( \sum_{i=1}^N \theta_i = 1, \) it forms a manifold known as the standard \((N-1)\)-simplex denoted by \( \Delta_{N-1} \). The solution \( \{ \hat{\Theta}_m^u \}_{m=1}^M \) of \( (16) \) defines a submanifold of \( \Delta_{N-1} \) (Figure 1a):

\[
\mathcal{D}_u = \{ \theta \mid \sum_{i \in G^u_m} \theta_i = \hat{\Theta}_m^u, m = 1, \ldots, M \} \subset \Delta_{N-1}.
\] (17)

By identifying the preference parameter \( \theta = \{ \theta_i \} \) with a discrete probability measure, the optimal parameter of \( \theta \) can be defined as a point in \( \Delta_{N-1} \) that is the nearest to all the submanifolds \( \{ \mathcal{D}_u \}_{u=1}^U \) in terms of Kullback-Leibler (KL) divergence \( KL(\theta, \theta') = \sum_{i=1}^N \theta_i \log(\theta_i/\bar{\theta}_i) \). That is, we minimize an objective function

\[
L_{em}(\theta) = \sum_{u=1}^U KL(\mathcal{D}_u, \theta) = \sum_{u=1}^U \min_{\theta' \in \mathcal{D}_u} KL(\theta', \theta)
\] (18)

with respect to \( \theta \in \Delta_{N-1} \). The \( em \) algorithm gradually minimizes this objective function by repeating the \( e \)-step and the \( m \)-step alternately.

Suppose we have an estimated parameter \( \theta(t) \) after the \( t \)-th iteration. In the \( e \)-step, we find a point \( \hat{\theta}_m^u(t) \) on the submanifold \( \mathcal{D}_u \) which is the nearest in terms of the KL divergence from the previous estimate \( \theta(t) \) (Figure 1b), that is,

\[
\hat{\theta}_m^u(t) = \arg \min_{\theta \in \mathcal{D}_u} KL(\theta, \theta(t)).
\] (19)

This procedure is called the \( e \)-projection. Using the Lagrangian multiplier method, the \( e \)-projection is explicitly written as

\[
\hat{\theta}_m^u(t) = \frac{\theta_i(t)}{\sum_{j \in G^u_m} \theta_j(t)} \hat{\Theta}_m^u, \quad i = 1, \ldots, N, \quad u = 1, \ldots, U,
\] (20)

where \( G^u_m \) denotes the group to which \( I_i \) belongs, and \( \hat{\Theta}_m^u \) denotes the corresponding group parameter.

In the \( m \)-step, we find a point \( \theta(t+1) \) on \( \Delta_{N-1} \) which minimizes the sum of the KL divergences from the points \( \hat{\theta}_m^u(t) \) on the submanifolds \( \mathcal{D}_u \) (Figure 1c), that is,

\[
\theta(t+1) = \arg \min_{\theta} \sum_{u=1}^U KL(\hat{\theta}_m^u(t), \theta).
\] (21)

This procedure is called the \( m \)-projection and explicitly written as

\[
\theta_i(t+1) = \frac{1}{U} \sum_{u=1}^U \hat{\theta}_i^u(t), \quad i = 1, \ldots, N.
\] (22)
Figure 1: (a): Each observation defines submanifold $D_u$ of $\Delta_{N-1}$. (b): Find the nearest point $\hat{\theta}_u(t)$ from the previous estimate $\theta(t)$ to each submanifold ($e$-step). (c): Find the nearest point $\theta(t+1)$ from each the points $\hat{\theta}_u(t)$ of all the submanifolds ($m$-step).

Iterating these $e$- and $m$-steps until neither $\theta(t)$ nor $\hat{\theta}_u(t)$ change, we get a (locally) optimal parameter $\theta(t)$. The derivation of the $e$- and $m$-projection formulas (20) and (22) are given in Appendix 3. Figure 1 shows these procedures in the case of $N = 3$, $U = 3$, $M = 2$. The em algorithm for our model is summarized in Figure 2.

Before showing experimental results, we will briefly summarize our approximation approach. An evaluation of the likelihood function $L(\theta)$ is prohibitive because of the computational cost for enumerating all the possible permutations. Therefore, we derived an approximate likelihood $\tilde{L}(\theta)$ without marginalization and permutation. For large numbers of items $N$ or users $U$, the maximization of $\tilde{L}(\theta)$ with respect to $\theta$ may still be computationally demanding. As an approximation approach, instead of maximizing $\tilde{L}(\theta)$, we considered the small-sized optimization problems (16) with respect to the grouped parameters $\{\Theta_u^m\}_{m=1}^M$ for individual users. The solutions of the problems (16) are used to define observation submanifolds, and the most consistent preference parameter $\theta$ with the observations is estimated by the em algorithm.

3.2 Experiments with Synthetic Data

To show how the proposed algorithm works, we apply it to synthetically generated grouped ranking observations.

Proposed algorithm increases exact likelihood
We first check that the maximization of the approximate likelihood leads to the maximization of the exact likelihood. We fixed the number of groups to $M = 3$. The number of items is $N = 7$, which is the maximum number that the exact likelihood can be efficiently calculated with our computational resource. The number of users is set to $U = 100$, and complete ratings are generated using the Plackett-Luce model and then randomly divided into three groups. We estimated the parameter by the proposed algorithm, and calculated the values of $L(\theta)$ and $\tilde{L}(\theta)$ 100 times with data sets generated from different true preference parameters. Figure 3 (left) depicts the average of exact and approximate likelihoods where the horizontal axis shows the number of iterations of the algorithm. To show the values of $L(\theta)$ and $\tilde{L}(\theta)$ in the same figure, a constant positive value is added to $\tilde{L}(\theta)$. We also show the average of the KL divergence of estimated parameters from true parameters. The figure shows that monotonic increasing of $\tilde{L}(\theta)$ leads to monotonic increasing of $L(\theta)$. Also, the proposed algorithm monotonically decreases the KL divergence between true and estimated parameters.
Algorithm 1: The $em$ algorithm for the grouped ranking model.

**input:** grouped ranking observations $\{D^u = \{G^u_m\}_{m=1}^M\}_{u=1}^U$.

**initialize:** choose initial parameter $\theta(0)$, and solve the optimization problems

$$\max_{\{\Theta^u_m\}} \sum_{m=1}^M \gamma^u_m \log \left( \frac{\Theta^u_m}{\sum_{n=m}^M \Theta^u_n} \right) + \sum_{m=1}^M \frac{1}{M} \log \Theta^u_m, \quad \text{subject to} \quad \sum_{m=1}^M \Theta^u_m = 1$$

to get $U$ sets of grouped parameter values $\{\hat{\Theta}^u_m\}_{m=1}^M$.

**repeat:** from $t = 0$, until convergence

**e-step:** update $\hat{\theta}^u(t)$ by the $e$-projection:

$$\hat{\theta}^u_i(t) := \frac{\theta_i(t)}{\sum_{j \in G^u_{mi}} \theta_j(t)} \hat{\Theta}^u_{m|i}, \quad i = 1, \ldots, N, \ u = 1, \ldots, U.$$ 

**m-step:** update $\theta(t)$ by the $m$-projection:

$$\theta_i(t+1) := \frac{1}{U} \sum_{u=1}^U \hat{\theta}^u_i(t), \quad i = 1, \ldots, N.$$ 

**output:** converged parameter $\theta$.

Figure 2: The $em$ algorithm for the grouped ranking model. The algorithm finds a preference parameter which is the most consistent with observations. The consistency is measured by the KL divergence between the parameter and submanifolds which is defined by the solutions of the optimization problem in the initialization step. The $e$-projection formula is replaced by (57) and (58) in the Appendix 4 for the data with unrated items.

**Validity of approximate likelihood**

We next consider in what situation, the gap between the approximate likelihood $\tilde{l}(\theta; m, u)$ and the exact likelihood $l(\theta; m, u)$ becomes small. To derive $l(\theta; m, u)$, we used two inequalities. The first inequality comes from omitting the term $\sum_{j < i} \theta_{\pi^u_{m}(j)}$ in denominator of (7). When the group size $\gamma^u_m$ is small enough, the number of possible permutations in the group is also small, so the effect of omitting the term $\sum_{j < i} \theta_{\pi^u_{m}(j)}$ is small. The second is the arithmetic-geometric mean inequality, where equality holds if and only if all $\theta_{\pi^u_{m}(i)}$’s in the group $G^u_m$ are the same value. It is unlikely that all $\theta_{\pi^u_{m}(i)}$’s in a group are the same, except the case that $\gamma^u_m$ is very small. When $M$ is large, most of groups will contain only a small number of items, and consequently the arithmetic-geometric mean inequality almost holds as equality. To check the above argument, we show absolute values of the ratio $|L(\theta) / \tilde{L}(\theta)|$ with varying the number of groups. Since $L(\theta)$ and $\tilde{L}(\theta)$ are always negative, $|L(\theta) / \tilde{L}(\theta)| \leq 1$ holds if $L(\theta) \geq \tilde{L}(\theta)$. Setting the number of items to $N = 7$ and the number of users to $U = 100$, we varied $M$ from two to seven in increments of one. We calculated the ratio $|L(\theta) / \tilde{L}(\theta)|$ 100 times using randomly.
chosen parameters $\theta$ for each setting of $M$. Figure 3 (right) shows the average of the ratio $|L(\theta)/\hat{L}(\theta)|$ with one standard deviation error bars. We can see that $L(\theta)$ is bounded below by $\hat{L}(\theta)$, and the ratio approaches to one as the number of groups increases. Therefore, the exact likelihood $L(\theta)$ is bounded below by the approximate likelihood $\hat{L}(\theta)$, and we can expect maximization of $\hat{L}(\theta)$ to obtain a good estimate of parameter $\theta$ when the number of items in a group is relatively small.

The above discussion considers an ideal case where the approximate likelihood gets close to the exact likelihood. In Appendix 2, we will analytically consider the gap between the exact and approximate likelihoods.

**Computational costs**

Finally, we see the computational costs of the proposed algorithm with respect to the number of items $N$ and the number of groups $M$. Since it is clear from its formulation that the computational time depends on the number of users $U$ linearly, we fixed $U = 100$ and only consider $N$ and $M$.

We measured the time for estimating the parameter by the proposed method, and Figure 4 shows the average processing time. The error bars are omitted because the standard deviation of processing time is small. The algorithm is implemented with R-language version 2.9.1 (R Development Core Team, 2009) and processed on an Intel machine$^3$, and the time spent for I/O is excluded. The initialization part of the algorithm involves solving $U$ optimization problems (16). For this optimization, we used the “L-BFGS-B” method by using “optim” routine equipped in R-language. From Figure 4, we can see that the algorithm scales almost linearly with the increase of the numbers of both items and groups.

---

$^3$ with 2.4 GHz dual processors, 4,096 MB memory, and the operation system is Mac OS X version 10.5.8.
4 Mixture of Grouped Ranking Models

In this section, we consider a mixture of ranking models to express various populations of users. A mixture of $K$ models is written as a form of

$$P(x) = \sum_{k=1}^{K} \omega_k P(x; \theta^k),$$

(23)

where $x$ is a datum, $\omega_k$ is a prior probability that a datum is generated from the $k$-th element model, and $\theta^k$ is the preference parameter of the $k$-th model. In principle, the parameters of the mixture model (23), $\{\omega_k, \theta^k\}_{k=1}^{K}$ can be estimated by the EM algorithm (Dempster et al., 1977). When we consider a mixture of grouped ranking models, however, there is a difficulty mentioned in Subsection 4.2. To avoid this difficulty, we introduce the entropy regularized K-means soft clustering algorithm (Sahbi and Boujemaa, 2005) in the parameter space of the grouped ranking model.

4.1 Entropy Regularized Soft Clustering

The soft clustering, also called the fuzzy clustering, divides data elements $x^u, u = 1, \ldots, U$ into clusters, and each datum $x^u$ is associated with a set of membership levels $g_{uk}, k = 1, \ldots, K$, which indicates how degree the datum $x^u$ belongs to the $k$-th cluster. We define a membership vector for a datum $x^u$ as $g_u = (g_{u1}, \ldots, g_{uK}) \in \mathbb{R}^K$ which is normalized as $\sum_{k=1}^{K} g_{uk} = 1, \quad u = 1, \ldots, U$. Each cluster has a cluster representative $\xi^k, k = 1, \ldots, K$, and a distance $d_{uk}$ between the datum $x^u$ and the representative $\xi^k$ of the $k$-th cluster is defined with an appropriate distance measure $d_{uk} = d(x^u, \xi^k)$. There are many soft clustering methods, and the entropy regularized soft clustering is known to be one of the best methods. We define the entropy of a membership vector as $H(g_u) = -\sum_{k=1}^{K} g_{uk} \log g_{uk}$. In the entropy regularized soft clustering, the objective
\[
J_\lambda(\{g_{uk}\}, \{\xi^k\}) = \sum_{u=1}^U \sum_{k=1}^K g_{uk}d_{uk} - \lambda \sum_{u=1}^U H(g_u)
\]

\[
= \sum_{u=1}^U \sum_{k=1}^K g_{uk}d_{uk} + \lambda \sum_{u=1}^U \sum_{k=1}^K g_{uk} \log g_{uk},
\]  

(24)

is minimized with respect to the membership \(g_{uk}\) and the cluster representatives \(\xi^k\), subject to the constraints \(\sum_{k=1}^K g_{uk} = 1, \ u = 1, \ldots, U\). The first term of (24) represents a cost for cluster assignment to each datum \(x^u\). If a datum \(x^u\) is far from a cluster center \(\xi^k\), membership \(g_{uk}\) to the cluster becomes small. The second term is the entropy regularization to avoid hard partitioning, that is, this term penalizes data for belonging to only one cluster. This optimization problem can be solved by an iterative update similar to the EM algorithm. We denote the value of the membership \(g_{uk}\), the cluster representative \(\xi^k\), and the distance \(d_{uk}\) at the \(t\)-th iteration by \(g_{uk}(t)\), \(\xi^k(t)\) and \(d_{uk}(t)\) respectively. With the constraints \(\sum_{k=1}^K g_{uk} = 1, \ u = 1, \ldots, U\), the update formula for the memberships are easily derived using the Lagrangian multiplier method as

\[
g_{uk}(t) = \frac{\exp \left(\frac{-d_{uk}(t-1)}{\lambda}\right)}{\sum_{l=1}^K \exp \left(\frac{-d_{ul}(t-1)}{\lambda}\right)}. \tag{25}
\]

We then describe a parameter estimation algorithm for the mixture of grouped ranking models. In our setting, the distance \(d_{uk}\) between the datum \(D^u\) and the cluster representative \(\xi^k\) is defined as

\[
d_{uk} = KL(D^u, \xi^k) = \min_{\theta \in D_u} KL(\theta, \xi^k). \tag{26}
\]

The membership update procedure is the same as (25). The cluster representative update is not straightforward as that of membership, and needs the EM algorithm again.

**e-step**: e-projection from a cluster representative \(\xi^k\) to the data submanifolds \(D_u\):

\[
\hat{\theta}^{uk} = \arg \min_{\hat{\theta}^u \in D_u} KL(\hat{\theta}^u, \xi^k), \ u = 1, \ldots, U.
\]

**m-step**: m-projection from data submanifolds \(D_u\’s\) to the cluster representatives \(\xi^k\):

\[
\hat{\xi}^k = \arg \min_{\xi^k \in \Delta_{N-1}} \sum_{u=1}^U g_{uk}KL(\hat{\theta}^{uk}, \xi^k), \ k = 1, \ldots, K.
\]

We can derive more tangible formula for updates using the Lagrangian multipliers in each step. On a submanifold \(D_u\), the \(i\)-th component of the point closest to the cluster representative \(\xi^k\) is given by

\[
\hat{\theta}_{ik} = \frac{\xi_{ik}^k}{\sum_{j \in G_{m|i}} \xi_{jk}^k \Theta_{m|i}^u},
\]

and the \(i\)-th component of the cluster representative \(\xi^k\) is updated as

\[
\hat{\xi}_i = \frac{\sum_{u=1}^U g_{uk} \hat{\theta}_{ik}}{\sum_{u=1}^U g_{uk}}.
\]

We summarize the entropy regularized soft clustering algorithm for the mixture of grouped ranking models in Figure 5.
Algorithm 2: The soft clustering algorithm for the mixture of grouped ranking models.

**input:** grouped ranking observations \( \{D^u = \{G^u_m\}_{m=1}^M\}_{u=1}^U \), the number of clusters \( K \), and an entropy regularizing parameter \( \lambda \).

**initialize:** solve the optimization problems

\[
\max_{\{\Theta^u_m\}} \sum_{m=1}^M \gamma^u_m \log \left( \frac{\Theta^u_m}{\sum_{n=m}^M \Theta^u_n} \right) + \sum_{m=1}^M \frac{1}{M} \log \Theta^u_m, \quad \text{subject to} \quad \sum_{m=1}^M \Theta^u_m = 1
\]

to get \( U \) sets of grouped parameter values \( \{\hat{\Theta}^u_m\}_{m=1}^M \). set the initial membership to \( g_{uk}(0) = \frac{1}{K}, k = 1, \ldots, K, u = 1, \ldots, U \). select \( \xi^k(0), k = 1, \ldots, K \) at random from \( N - 1 \) simplex.

**repeat:** from \( t = 0 \), until convergence

**Membership Update:** update memberships \( g_{uk}(t) \):

\[
g_{uk}(t) := \frac{\exp(-d_{uk}(t-1)/\lambda)}{\sum_{l=1}^K \exp(-d_{ul}(t-1)/\lambda)}, \quad d_{uk} = KL(D^u, \xi^k) = \min_{\theta \in D^u} KL(\theta, \xi^k).
\]

**Cluster Representatives Update:** update cluster representatives \( \xi^k \);

introduce a loop index \( s \) for the iteration of the \( em \) algorithm.

\( \xi^k(t_0) := \xi^k(t-1) \), randomly select \( \hat{\theta}^{uk}(t_0) \) from \( D^u \).

**repeat:** from \( s = 0 \) until convergence

**e-step:** update \( \hat{\theta}^{uk}(t_s) \) by the \( e \)-projections:

\[
\hat{\theta}^{uk}(t_s) := \frac{\xi^k(t_{s-1}) \hat{\theta}^u_{m|i}}{\sum_{j \in G^u_{m|i}} \xi^k(t_{s-1})}, \quad i = 1, \ldots, N, \ u = 1, \ldots, U, \ k = 1, \ldots, K.
\]

**m-step:** update \( \xi^k(t_s) \) by the \( m \)-projection:

\[
\xi^k(t_s) := \frac{\sum_{u=1}^U g_{uk}(t) \hat{\theta}^{uk}(t_s)}{\sum_{u=1}^U g_{uk}(t)}, \quad i = 1, \ldots, N, \ k = 1, \ldots, K.
\]

\( \xi^k(t) := \xi^k(t_{s^*}) \), where \( s^* \) is the converged index \( s \).

**output:** converged parameters \( \{g_{uk}\}, \{\xi^k\} \).

Figure 5: The soft clustering algorithm for the mixture of the grouped ranking model. The \( e \)-projection formula is modified according to (57) and (58) in the Appendix 4 for the data with unrated items.
4.2 Difficulty in Applying the EM Algorithm

In this subsection, we will show that the EM algorithm, which is a standard approach to estimate parameters of a mixture model, will not work for our problem. Let us try to adopt the EM algorithm to estimate parameters of the mixture of grouped ranking models:

\[
P(D^u) = \sum_{k=1}^{K} \omega_k P(D^u; \theta^k). \tag{27}\]

Letting a joint probability that the k-th element model is chosen and then generates the observation \(D^u\) be

\[
P(D^u, k; \theta) = \omega_k P(D^u; \theta^k) = \omega_k \prod_{m=1}^{M} P(G^u_m | G^u_1, \ldots, G^u_{m-1}; \theta^k), \tag{28}\]

the \(Q\) function in the EM algorithm is written as

\[
Q(\theta | \theta(t)) = \sum_{u=1}^{U} \sum_{k=1}^{K} P(k|D^u; \theta(t)) \log P(D^u, k; \theta), \tag{29}\]

where \(\theta\) denotes all parameters \(\{\theta^k\}_{k=1}^{K}, \{\omega_k\}_{k=1}^{K}\) and \(\theta(t)\) is the value of the parameters estimated in the \(t\)-th iteration of the EM algorithm.

Given a datum \(D^u\) by a user \(u\), the probability that \(D^u\) is generated from the \(k\)-th model is calculated as

\[
P(k|D^u; \theta(t)) = \frac{P(D^u, k; \theta(t))}{\sum_{l=1}^{K} P(D^u, l; \theta(t))} = \frac{\omega_k P(D^u; \theta^k(t))}{\sum_{l=1}^{K} \omega_l P(D^u; \theta^l(t))}. \tag{30}\]

Substituting (28) and (30) into the function (29), we maximize (29) under the constraint \(\sum_{k=1}^{K} \omega_k = 1\) by iterating the E- and the M-steps.

We can easily find the formula to update \(\omega_k\)’s, though, in the update formula for \(\theta^k, k = 1, \ldots, K\), the same computational difficulty arises as mentioned in Section 2. In the single grouped ranking model, we avoid this difficulty by approximating the likelihood. Similarly, we get a modified \(Q\) function by replacing \(P(D^u; \theta^k)\) with \(\tilde{P}(D^u; \theta^k)\) as

\[
\hat{Q}(\theta | \theta(t)) = \sum_{u=1}^{U} \sum_{k=1}^{K} \frac{\omega_k(t) \tilde{P}(D^u; \theta^k(t))}{\sum_{l=1}^{K} \omega_l(t) \tilde{P}(D^u; \theta^l(t))} \left( \log \omega_k + \log \tilde{P}(D^u; \theta^k) \right), \tag{31}\]

where

\[
\tilde{P}(D^u; \theta^k) = \prod_{m=1}^{M} \frac{\gamma_m^{u_m} \left( \frac{\Theta_{mk}}{\gamma_m} \right)^{\gamma_m^{u_m}}}{\left( \sum_{n=1}^{M} \Theta_{nk} \right)^{\gamma_m^{u_m}}}.
\]

Since direct maximization of (31) with respect to \(\theta^k\) is difficult, we again apply the \(em\) algorithm to this update step by dividing \(\hat{Q}(\theta | \theta(t))\) into each user’s term and iterating the e- and the m-projections between the model manifold and data submanifolds. In the mixture case, however, this approximating approach doesn’t work properly, because the inequality \(Q(\theta | \theta(t)) \geq \hat{Q}(\theta | \theta(t))\) does not hold in many cases even though \(P(D^u; \theta^k) \geq \tilde{P}(D^u; \theta^k)\) holds. The modified \(Q\) function (31) contains the term

\[
\tilde{P}(k|D^u; \theta(t)) = \frac{\omega_k(t) \tilde{P}(D^u; \theta^k(t))}{\sum_{l=1}^{K} \omega_l(t) \tilde{P}(D^u; \theta^l(t))}. \tag{32}\]
and this term often becomes larger than the exact value

$$ P(k|D^u; \theta(t)) = \frac{\omega_k(t)P(D^u; \theta^k(t))}{\sum_{l=1}^K \omega_l(t)P(D^u; \theta^l(t))}. \quad (33) $$

and maximizing $\tilde{Q}$ does not always guarantee $Q$ to be maximized. This throws out the approximating approach. In our preliminary experiments, the estimated probabilities (30) by the EM algorithm almost always become 1 for only one label of $k$, and 0 for others, which means that the conditional probability of the class becomes deterministic. There are at least two possible reasons for this failure. One is overfitting to training data. The other is due to the likelihood approximation, and it is accountable by considering the effect of approximation of the likelihood

$$ P(D^u; \theta) \not\leq P(D^u; \theta'). $$

For simplicity, we omit prior factor $\{\omega_k\}_{k=1}^K$, and let $\alpha_k = P(D^u; \theta^k)$ and $\varepsilon_k = P(D^u; \theta^k) - \tilde{P}(D^u; \theta^k)$. Then, the gap between (33) and (32) is written as

$$ \frac{\sum_{l=1}^K P(D^u; \theta^l(t))}{\sum_{l=1}^K P(D^u; \theta^l(t))} - \frac{\tilde{\sum}_{l=1}^K \tilde{P}(D^u; \theta^l(t))}{\sum_{l=1}^K \tilde{P}(D^u; \theta^l(t))} = \sum_{l=1}^K (\alpha_l - \alpha_k - \varepsilon_l) \sum_{l=1}^K (\alpha_l - \varepsilon_l). \quad (34) $$

The gap (34) tends to be positive when $\alpha_k$ is relatively small, and negative when $\alpha_k$ is large. Since $0 \leq P(k|D^u; \theta) \leq 1$, $P(k|D^u; \theta)$ is apt to be overestimated for a large $P(k|D^u; \theta)$, and to be underestimated for a small $P(k|D^u; \theta)$. This effect will contribute to the deterministic attribution of the estimated conditional distributions.

## 5 Applications of the Mixture Model

In this section, we propose two applications of the mixture of grouped ranking models. The first is as a tool for data analysis and visualization, and the second is as a collaborative filtering system. We show simple experimental results with movie and book rating data.

### 5.1 Data Visualization

In the field of market research, it is very important to model and interpret a relationship between users (consumers) and items (products). That is, apart from the apparent attributes of items such as prices or colors, it is important to know which items are similar (similarly accepted by the same kind of users), and which items are dissimilar (accepted by completely different kinds of users), and the same is true for the relationship between users.

When we account for each user’s preference by the mixture of $K$ different ranking models, each model has the meaning of “typical user”. In other words, there are $K$ ideal users and their preference patterns are explained by $\theta^k = (\theta^k_1, \ldots, \theta^k_N), \ k = 1, \ldots, K$. Accordingly, each item can be characterized by $K$ ideal users’ preference and can be seen as a point in $K$ dimensional space as

$$ (\theta^1_1, \ldots, \theta^K_1) \in \mathbb{R}^K. \quad (35) $$

On the other hand, each user has its own membership probability $P(k|D^u)$ to the $k$-th cluster, and this can be seen as coordinates of a $K$ dimensional vector. This vector

$$ u = (P(1|D^u), \ldots, P(K|D^u)) \quad (36) $$

15
can be regarded as a directional vector for a user \( u \), where the \( k \)-th dimension of \( u \) represents the probability that the user \( u \) belongs to the \( k \)-th “typical user” group. We note that in the case of the mixture of grouped ranking models, the probability \( P(k|D^u) \) is replaced by the membership \( g_{uk} \).

These correspondences (35) and (36) lead us to a mapping of items to points in the first quadrant of \( \mathbb{R}^K \), and users to axes (half lines) in \( \mathbb{R}^K \). The mapped items in \( \mathbb{R}^K \) are projected onto a user’s axis naturally by

\[
I_i \mapsto \theta_i^u, \quad \theta_i^u = \sum_{k=1}^{K} \theta_{k}^i P(k|D^u),
\]

where \( \theta^u = \{\theta_i^u\}_{i=1}^{N} \) is interpreted as a user specific preference parameter. This preference parameter for a user \( u \) leads us to an application of the mixture of grouped ranking models to a collaborative filtering system, as shown in the next subsection. This simultaneous mapping reveals the relation between items and users. The axes of similar users will be drawn closely, and similar items will be located closely each other. There are some researches on the preference visualization. In Zenebe and Norcio (2007), demographic information such as gender and age are attached to each user, and items (movies in the literature) are vaguely divided into clusters of genres, such as “action”, “horror”, “comedy” using fuzzy set theory. The expected preference level of the movie for each user in each genre is mapped on \( \mathbb{R}^3 \), where user’s age and gender correspond to \((X, Y)\) axes and a preference level to the movie in the genre corresponds to \( Z \) axis. In Mei and Shelton (2006), users and items are mapped on the same Euclidean space, and ratings are considered as outputs of a rating function \( f(||u - I_i||) \) which is a function of the Euclidean distance between user \( u \) and item \( I_i \). This rating function is learned with a rating data set, then used to embed items and users in the Euclidean space so as to minimize the difference between true ratings and the estimates from the rating function. The most notable difference between these works and ours is the space in which items and users are embedded. In our method, items are mapped into \( \mathbb{R}^K \), whereas users are mapped into the space of mappings on the item set \(^4 \) defined by (37) which are represented as half lines in \( \mathbb{R}^K \). For some data such as movie ratings, items are the objects to be rated by users, so the users and the items are not the same kind of data. For such kind of data, it seems to be natural to differently represent items as points and users as lines.

**Experiments of the item-user visualization**

We will demonstrate the visualization method explained above with the MovieLens rating data set (Riedl and Konstan, 2000). This data set consists of 100,000 ratings ranging from 1 to 5 for 1,682 movies by 943 users. For our demonstration purpose, we only used a small subset of the data. We first selected \( N = 100 \) most frequently rated movies, and then extracted \( U = 554 \) users who rated more than 20 of the those popular 100 movies. We adopted a mixture model (27) of \( K = 2 \),

\[
P(x) = \omega P(x; \theta^1) + (1 - \omega) P(x; \theta^2),
\]

and the parameters \( \{\omega, \theta^1, \theta^2\} \) are estimated by using the algorithm described in Figure 5. Figure 6 is an example of the item visualization. We mapped 5 items with their titles which have high parameter values and other movies are mapped without titles on two dimensional space. Each item \( I_i \) has its coordinate values defined by \( \theta_{1}^i, \theta_{2}^i \). Using this visualization scheme, we can see that, for example, the movie “Star Wars” and “Titanic” are preferred by many users, but populations of users who like these movies are rather different. Figure 7 shows the mappings

\(^4 \)Note that it is not a dual space, which is defined as a vector space consists of all linear functions.
of users into the two dimensional space. Each user is expressed by a line in this space defined by $u = (P(1|D^u), P(2|D^u))$ as a directional vector, and we can interpret this line represents user’s evaluation standard. From this figure, we can infer that users 4 and 6 share a similar taste, while user 1 will enjoy movies quite differently from them. We next show the mapping of both items and a user at the same time in Figure 8. Movies already rated by user 114 are depicted as circles, and unrated movies are depicted as diamonds. Using the projection defined by (37), items are projected on the user axis. From this figure, we can expect that this user will enjoy “The Shawshank Redemption” and will not enjoy “Alien”. With this visualization technique,

![Image](image.png)

**Figure 6**: An example of item mapping. Items are mapped into 2D space. Movies with high parameter value are mapped with their titles.

![Image](image.png)

**Figure 7**: An example of user mapping. Three users are mapped into 2D space as lines defined by (36).

we can easily see the relationships between items, between users, and between items and users.

### 5.2 Collaborative Filtering

Collaborative filtering systems attempt to present items which are likely of interest to the user automatically. For this purpose, the rating of item $I_i$ for user $u$ is estimated by ratings already assigned by other users to the item. Similarities between users are calculated based on ratings which they give to the same items. See Adomavicius and Tuzhilin (2005) for a comprehensive survey. There is a huge amount of research on collaborative filtering systems, and each technique has its own advantages and drawbacks. In addition, each technique must be finely tuned for each specific occasions and data sets to be used. Here we do not attempt to make thorough comparison, and only show that our proposal works comparably with existing methods for two data sets.

Usually, collaborative filtering is formulated as a problem of predicting the rating $r_{u,i}$ for an unknown item $I_i$ by the target (active) user $u$. A large number of collaborative filtering systems employ the rate prediction formula

$$r_{u,i} = \bar{r}_u + \frac{1}{C} \sum_{v \in ne(i)} \text{sim}(u, v)(r_{v,i} - \bar{r}_v),$$  

**Equation (38)**

---

17
where $\bar{r}_u$ is the average value of ratings given by the active user $u$, and the neighborhood $ne(i)$ is a set of users who evaluated item $I_i$. $C$ is a normalization factor defined as $C = \sum_{v \in ne(i)} |\text{sim}(u, v)|$. This prediction formula highly depends on the similarity measure between users. One of the most famous collaborative filtering systems is the GroupLens (Resnick et al., 1994), and they used Pearson’s correlation as the similarity measure:

$$\text{sim}_{P_e}(u, v) = \frac{\sum_{i \in S_{u,v}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in S_{u,v}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in S_{u,v}} (r_{v,i} - \bar{r}_v)^2}}.$$  \hspace{1cm} (39)

where $S_{u,v}$ is a subset of items both users $u$ and $v$ have already rated. For some data sets, it is reported that the cosine similarity performs better than Pearson’s correlation:

$$\text{sim}_{\text{cos}}(u, v) = \frac{\sum_{i \in S_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in S_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in S_{u,v}} r_{v,i}^2}}.$$  \hspace{1cm} (40)

We devised some similarity measures using the mixture of grouped ranking models and conducted preliminary experiments, and found a similarity measure that shows a comparable result to conventional measures. To define our similarity measure, we use the approximation of the grouped ranking model $\tilde{P}(D^u; \theta^u)$ with the user specific parameter $\theta^u = (\theta^u_1, \ldots, \theta^u_N)$, where the parameters

$$\theta^u_i = \sum_{k=1}^{K} \theta^u_i P(k|D^u) \sim \sum_{k=1}^{K} \theta^u_i \beta_{uk}, \quad i = 1, \ldots, N$$

are acquired by a mixture model. Now that we have a generative model for each user, it is natural to define the similarity between users based on their generative models. It is known that Fisher
kernel (Jaakkola and Haussler, 1999) provides a principled similarity measure that takes into account an underlying probability distribution. We use the Fisher score \( s_u = \partial_{\theta} \log \tilde{P}(D^u; \theta^u) \) as a feature vector for user \( u \) (Hofmann, 2000), and define the similarity of users by cosine of the Fisher scores\(^5\) based on the grouped ranking model:

\[
sim_{GR}(u, v) = \frac{s_u^T s_v}{\|s_u\| \cdot \|s_v\|}.
\]

(41)

**Experiments with real-world data**

We conclude this section with simple experiments of collaborative filtering with real-world data. We use the MovieLens and the BookCrossing data sets (Ziegler et al., 2005). The GroupLens research team kindly provides the 5-level movie rating data divided for 80% training subset (that is, 80,000 rating data) and 20% test subset (20,000 rating data) for 5-fold cross validation. We implemented the similarity based collaborative filtering system with Pearson’s similarity (39), cosine similarity (40), and cosine of the Fisher score similarity (41). For each method, 80% of the data are used to calculate the similarity between users and 20% are used to evaluate the rate prediction accuracy. The BookCrossing data set contains 1,149,780 ratings (i.e., 433,681 explicit ratings on 1 to 10 scale and 716,109 implicit ratings expressed by zero scale) by 278,858 users on 271,379 books. Since the size of data is uncontrollably large, we first removed the data with scale zero because it is out of our scope to analyze the implicit ratings. Then we picked out a smaller data set, which consists of users who rated more than 8 items and items which are rated by at least 12 users. The resulting data set contains 48,484 ratings by 663 users for 1,191 items. We divide this down-sized data set for 5-fold cross validation in the same manner as the GroupLens data set. The number of components in a mixture of grouped ranking models should be optimized by cross validation or other model selection methods, though, we determined it by a preliminary experiment and fixed it to five for the sake of simplicity.

We adopt the Mean Absolute Error (MAE) as an evaluation criterion for the prediction accuracy, which is defined as follows:

\[
MAE = \frac{1}{|Te|} \sum_{(u,i) \in Te} |r_{u,i}^* - r_{u,i}|,
\]

where \( Te \) is a set of test samples consists of user and item pairs saved for the accuracy evaluation, and \( r_{u,i}^* \) is the true rating by user \( u \) for item \( I_i \). Table 1 shows the evaluation result of two data sets with three similarity measures. The figures in the table are means and standard deviations of five trials. We see that the Pearson’s correlation similarity yields the best performance for the MovieLens data set, while the Fisher cosine similarity with the grouped ranking models yields the better performance for the BookCrossing data set. We can conclude that the proposed similarity measure defined by the grouped ranking model is comparable to the existing collaborative filtering methods in rating prediction accuracy.

---

\(^5\)which is also known as the normalized Fisher kernel (Shawe-Taylor and Cristianini, 2004).
Conclusion

In this paper, we proposed a probability model for grouped ranking observations parametrized by an item preference level parameter. To the best of our knowledge, this is the first proposal of an item ranking model which considers permutations in grouped items with an efficient parameter estimation algorithm. We derived the likelihood of the parameter, and gave its practical approximation. We also proposed the *em* algorithm to find the preference parameter. From the result of numerical experiments, we can conclude that our proposed algorithm reduces the KL divergence from the true parameters by iterative optimization and increases both the approximate and exact likelihood.

The main advantages of our algorithm are two-fold. The first is that the computational time of our algorithm scales almost linearly with the numbers of users and items, and it is exemplified by an experiment. The second and prominent advantage is applicability to on-line processing systems. If a new user joins and gives a new datum, a direct maximization approach requires full maximization of the likelihood again. In our proposed algorithm, however, we only have to solve a small maximization problem (16) with respect to the new datum, and we can obtain a revised estimate $\theta$ by iterating (20) and (22) of the algorithm with $U := U + 1$. For real-world applications such as collaborative filtering systems, we need to handle a vast amount of data which increase hour to hour. The scalability of our algorithm will enable us to fit our model to such applications.

Lebanon and Mao (2007) studied a non-parametric distance-based partial ranking model from the viewpoint of the group theory and derived an efficient estimation method. It is interesting to investigate under what circumstances, parametric models explain real data better than non-parametric models, and vice versa. Recently, Guiver and Snelson (2009) cast the Plackett-Luce model into the Bayesian framework using the Gumbel distribution as a prior distribution for preference parameters. A Bayesian extension of our proposed ranking model is another direction of our future work.

We also considered mixtures of grouped ranking models, and proposed two applications: data visualization and collaborative filtering. There are several researches which consider mixtures of ranking models to explain heterogeneity among users. Croon and Luijkx (1993) considered the mixture of the Bradley-Terry models, while Murphy and Martin (2003) and Busse et al. (2007) considered the mixture of distance-based models such as Mallow’s model. Kamishima and Akaho (2006) also studied an efficient clustering method of order data and applied it to collaborative filtering systems. Our proposal for use of a mixture model is apparently applicable to mixtures of any models that assign each item a preference level (Fujimoto et al., 2009).

The accuracy of the recommendation based on Fisher scores of our proposed model is comparable with the conventional methods. We believe that the research of recommender systems are shifting to the direction of developing methods to provide additional information to users such as reasoning for recommendations. The recommendation followed by visualization might be very persuasive, that is, we can recommend an item predicted as high rating and present visually similar items and users that are the basis of the recommendation. Recently, Stern et al. (2009) proposed a framework to predict ratings based on Bayesian inference, and it also provides user and item visualization method. The way to show the reasoning of the recommendation effectively with an appropriate visualization technique will be explored in our future works.
Acknowledgements

The authors would like to thank the GroupLens Research Group for providing the MovieLens ratings data set, and would like to thank Cai-Nicolas Ziegler for providing the BookCrossing rating data set. The authors are also grateful to Yutaro Seki for developing programs used for our user-item visualization experiment. Lastly, the authors would like to express our special thanks to anonymous reviewers whose comments led to valuable improvements in the manuscripts.

Appendix 1: Details of proposed probability model

In this appendix section, we will provide more details of data generation process. We first note that in the study of ranking data, there are at least two kinds of ‘incomplete ranking’ data. The first incompleteness is containing unrated items. Almost all ranking models must address this unrated items problem, and we will show how to treat such kinds of data in Appendix 4. The second incompleteness is that rankings are given in the form of grouped ranking, the principal subject of this paper.

The grouped ranking observation $D^u$ is composed of groups of items $\{G^u_m\}_{m=1}^M$. The number of items in each group $G^u_m$ reflects each user’s evaluation tendency. For example, a rigorous user will put only a few items into the first rated group, and a permissive user will put a lot of items into the first rated group. To express this notion of user’s tendency, we introduce the concept of composition (Lebanon and Mao, 2007).

**Definition.** A composition of $N$ into $M$ is a sequence $\gamma^u = (\gamma^u_1, \ldots, \gamma^u_M)$ of positive integers whose sum is $N$.

A composition $\gamma^u = (\gamma^u_1, \ldots, \gamma^u_M)$ corresponds to a grouped ranking with $\gamma^u_1$ items in the first group $G^u_1$, $\gamma^u_2$ items in the second group $G^u_2$ and so on. Then, we suppose a user makes a grouped ranking observation according to the following two steps.

1. Give a full ranking to all $N$ items:
   
   We denote the ranking data given by a user $u$ as $O^u = (I^u_{u(1)} \succ I^u_{u(2)} \succ \cdots \succ I^u_{u(N)})$, where $u(i)$ denotes the index of the $i$-th ranked item.

2. Divide $N$ items into $M$ groups without changing item ordering:
   
   We denote the divided ranking data according to the user’s composition $\gamma^u$ as
   
   $$(O^u, \gamma^u) = ((I^u_{u(1)} \succ \cdots \succ I^u_{u(\gamma^u_1)}), \ldots, (I^u_{u(\sum_{m=1}^{M-1} \gamma^u_m + 1)} \succ \cdots \succ I^u_{u(N)})), \quad (42)$$
   
   and identify the pair $(O^u, \gamma^u)$ to a grouped ranking data $D^u = \{G^u_1, \ldots, G^u_M\}$.

In other words, we suppose that each user has a complete ranking of all the items, but for some reason, the ranking is only partially observed as the grouped ranking observation $D^u = \{G_1, \ldots, G_M\}$. It may be because one is asked to make a rough evaluation, or one does not have the time to perform a detailed ranking and only reports ratings without ordering in the same rated items. The data $O^u$ is the same with those of the original Plackett-Luce model. In our grouped ranking model, though, the given data set is composed of grouped ranking observations $\{D^u = \{G^u_m\}_{m=1}^M\}_{u=1}^U$, and the ordering within each group is not observable. When $N = M$ and each group contains only one item, this model is reduced to the original Plackett-Luce model. In this sense, our model is a generalization of the Plackett-Luce model.

---

6This step is assumed to be independent of the first ranking step.
The proposed ranking models is certainly a probability model. Noting that enumeration of all possible configurations in groups \( \{G_m^u\}_{m=1}^{M} \) for all possible compositions \( \gamma^u \) for \( N \) items is equivalent to enumeration of all possible orderings of \( N \) items in the Plackett-Luce model. That is,

\[
\sum_{\gamma^u} \sum_{\{G_m^u\}_{m=1}^{M}} P(\{G_m^u\}_{m=1}^{M}) = \sum_{\gamma^u} \sum_{\{G_m^u\}_{m=1}^{M}} P((O^u, \gamma^u)) = \sum_u P(I_{u(1)} \succ I_{u(2)} \succ \cdots \succ I_{u(N)}) = 1,
\]

where summation with respect to \( \gamma^u \) is taken over all the possible compositions of \( N \) items, summation with respect to \( \{G_m^u\}_{m=1}^{M} \) is taken over all the configurations of items in groups with a fixed composition \( \gamma^u \), and summation with respect to \( u \) for the Plackett-Luce model is take over all the permutations of \( N \) items.

**Appendix 2: Further analysis of approximate likelihood**

In this appendix section, we will derive an upper bound of the gap between the likelihood \( l(\theta; m, u) \) and its approximation \( \tilde{l}(\theta; m, u) \). From \( l(\theta; m, u) \), we eliminate summation with respect to possible permutations \( \sum_{\pi_m \in S(G_m^u)} \) by fixing \( \pi_m^u \) to a permutation \( \pi_m^{u*} \) which maximizes \( l(\theta; m, u) \), and denote the log likelihood with the permutation \( \pi_m^{u*} \) as \( l^*(\theta; m, u) \), which satisfies \( l^*(\theta; m, u) \geq l(\theta; m, u) \). Then, the gap between \( l(\theta; m, u) \) and \( \tilde{l}(\theta; m, u) \) is bounded by

\[
l(\theta; m, u) - \tilde{l}(\theta; m, u) \leq l^*(\theta; m, u) - \tilde{l}(\theta; m, u)
\]

\[
= \log \gamma_m^u! + \sum_i \log \theta_i - \sum_i \log \left( \sum_n \Theta_n^u - \sum_{j<i} \theta_{\pi_n^{u*}(j)} \right)
\]

\[
- \left[ \gamma_m^u \left( \log \Theta_m^u - \log \left( \sum_n \Theta_n^u \right) \right) + \log \gamma_m^u! - \gamma_m^u \log \gamma_m^u \right]
\]

\[
= \gamma_m^u \log \gamma_m^u + \log \left( \prod_{i \in G_m^u} \frac{\theta_i}{\Theta_m^u \gamma_m^u} \right) + \log \left( \frac{\sum_{n=m}^{M} \Theta_n^u}{\prod_{i=1}^{M} (\sum_{n=m}^{M} \Theta_n^u - \sum_{j<i} \theta_{\pi_m^{u*}(j)})} \right)
\]

\[
\leq \gamma_m^u \log \gamma_m^u + \prod_{i \in G_m^u} \frac{\theta_i}{\Theta_m^u \gamma_m^u} + \frac{\sum_{n=m}^{M} \Theta_n^u}{\prod_{i=1}^{M} (\sum_{n=m}^{M} \Theta_n^u - \sum_{j<i} \theta_{\pi_m^{u*}(j)})} - 2
\]

\[
\leq \gamma_m^u \log \gamma_m^u + \left( \frac{\sum_{n=m}^{M} \Theta_n^u}{\sum_{n=m+1}^{M} \Theta_n^u} \right) \gamma_m^u - 1,
\]

where we used an inequality \( \log x \leq x - 1 \), \( (x > 0) \). Therefore, the gap \( l(\theta; m, u) - \tilde{l}(\theta; m, u) \) is bounded by an increasing function of the group size \( \gamma_m^u \).

The expression \( \tilde{l}(\theta; m, u) \) is originally motivated to obtain the approximate maximum likelihood estimate with a tractable calculation, and \( \tilde{l}(\theta; m, u) \) may be a rough approximation. In real applications, users rate only a small subset of total items, and consequently, the gap \( l(\theta; m, u) - \tilde{l}(\theta; m, u) \) is not significantly large and the proposed algorithm based on the approximate likelihood will find a good estimate of the parameter. Further investigation of tighter approximation is our important future work.
Appendix 3: Derivation of the em algorithm

In our setting, the e-projection in the em algorithm is a procedure that finds the parameters \( \hat{\theta}^u(t) \) on each submanifold \( \mathcal{D}_u = \{ \theta \mid \sum_{i \in G^u_m} \theta_i = \hat{\Theta}^u_m \} \) as

\[
\hat{\theta}^u(t) = \arg \min_{\theta \in \mathcal{D}_u} KL(\theta, \theta(t)) , \quad u = 1, \ldots, U,
\]

where \( KL(\theta, \theta(t)) = \sum_{i=1}^N \theta_i \log \frac{\theta_i}{\theta_i(t)} = \sum_{m=1}^M \sum_{i \in G^u_m} \theta_i \log \frac{\theta_i}{\theta_i(t)} \). Since the sum of \( \theta_i \) in the group \( G^u_m \) are constrained to \( \hat{\Theta}^u_m \), it is sufficient to consider minimization of \( \sum_{i \in G^u_m} \theta_i \log \frac{\theta_i}{\theta_i(t)} \) for a group \( G^u_m \) of a fixed user \( u \). Then, the minimization problem is formulated as

\[
\min_{\theta} \sum_{i \in G^u_m} \theta_i \log \frac{\theta_i}{\theta_i(t)} , \quad \text{subject to} \quad \sum_{i \in G^u_m} \theta_i = \hat{\Theta}^u_m, \quad \theta_i > 0.
\]

We can solve the optimization problem (46) by introducing the Lagrange multiplier \( \lambda \) and the Lagrangian \( F(\theta, \lambda) \) as

\[
F(\theta, \lambda) = \sum_{i \in G^u_m} \theta_i \log \frac{\theta_i}{\theta_i(t)} + \lambda \left( \hat{\Theta}^u_m - \sum_{i \in G^u_m} \theta_i \right).
\]

Differentiating this Lagrangian with respect to \( \theta_i, \ i \in G^u_m \) and equating to 0, we get

\[
\frac{\partial F}{\partial \theta_i} = \log \frac{\theta_i}{\theta_i(t)} + 1 - \lambda = 0.
\]

By the constraint in (46), we get

\[
\lambda = \log \left( \frac{\hat{\Theta}^u_m}{\sum_{i \in G^u_m} \theta_i(t)} \right) + 1,
\]

and the e-projection is proved to be

\[
\hat{\theta}^u_i(t) = \frac{\theta_i(t)}{\sum_{j \in G^u_m} \theta_j(t)} \hat{\Theta}^u_m, \quad i = 1, \ldots, N, \quad u = 1, \ldots, U,
\]

where \( G^u_m \) denotes the group to which \( I_i \) belongs, and \( \hat{\Theta}^u_m \) denotes the corresponding group parameter.

In the m-projection, we find a point \( \hat{\theta}(t+1) \) in the manifold \( \Delta_{N-1} \) which minimizes the sum of the KL divergences from the points \( \hat{\theta}^u(t) \) on the submanifolds \( \{ \mathcal{D}_u \}_{u=1}^U \). That is,

\[
\theta(t+1) = \arg \min_{\theta} \frac{1}{U} \sum_{u=1}^U KL(\hat{\theta}^u(t), \theta)
\]

\[
= \arg \min_{\theta} \frac{1}{U} \sum_{u=1}^U \sum_{i=1}^N \left( \hat{\theta}^u_i(t) \log \hat{\theta}^u_i(t) - \hat{\theta}^u_i(t) \log \theta_i \right)
\]

\[
= \arg \max_{\theta} \frac{1}{U} \sum_{u=1}^U \sum_{i=1}^N \hat{\theta}^u_i(t) \log \theta_i
\]

\[
= \arg \max_{\theta} \sum_{i=1}^N p_i \log \theta_i,
\]
where \( p_i = \frac{1}{U} \sum_{u=1}^{U} \hat{\theta}_i^u(t) \). Then, in the same manner as the \( e \)-projection, we introduce the Lagrange multiplier \( \mu \) and the Lagrangian \( H(\theta, \mu) \) as

\[
H(\theta, \mu) = \sum_{i=1}^{N} p_i \log \theta_i + \mu \left( 1 - \sum_{i=1}^{N} \theta_i \right) .
\]

Differentiating this Lagrangian with respect to \( \theta_i, i=1, \ldots, N \), and equating to 0, we get

\[
\frac{\partial H}{\partial \theta_i} = p_i \theta_i - \mu = 0.
\]

Using the constraint \( \sum_{i=1}^{N} \theta_i = 1 \) and the fact \( \sum_{i=1}^{N} p_i = \frac{1}{U} \sum_{u=1}^{U} \sum_{i=1}^{N} \hat{\theta}_i^u = 1 \), we get \( \mu = 1 \) and the \( m \)-projection is proved to be

\[
\theta_i(t+1) = p_i = \frac{1}{U} \sum_{u=1}^{U} \hat{\theta}_i^u(t).
\]

It is possible that the \( em \) algorithm overfits a given data set. One of the well known methods to avoid overfitting is adding a regularization term to the objective function (18) like

\[
L_{\text{reg}}(\theta) = \sum_{u=1}^{U} \min_{\theta^u \in D_u} KL(\theta^u, \theta) + \epsilon KL(\theta_{\text{unif}}, \theta),
\]

where \( \theta_{\text{unif}} = \left( \frac{1}{N}, \ldots, \frac{1}{N} \right) \). Then, the \( m \)-step (22) of the algorithm is modified as

\[
\theta_i(t+1) = \frac{1}{U + \epsilon} \left( \sum_{u=1}^{U} \hat{\theta}_i^u(t) + \epsilon \frac{1}{N} \right), \quad i = 1, \ldots, N.
\]

We adopted regularized version (56) of the algorithm in experiments with synthetic data. The optimal value for \( \epsilon \) may be different depending on data sets, and it can be estimated by techniques such as cross validation. We omit to explore the optimal constant factor here, and set \( \epsilon = U/2 \) for experiments with synthetic data. For large scale real-world data, instead of adding a regularization term, we avoided the overfitting by an early stopping strategy.

**Appendix 4: Treatment of unrated items**

In this appendix section, we explain how we handle the unrated data in observations. In reality, users almost never rate all the objects and the way to handle the scenario where users leave some items unrated is very important.

With unrated items in observations, the initialization step of the Algorithm 1 does not change, and we just solve the \( M \) variable optimization problem (16). However, the \( e \)-step of the \( em \) algorithm must be modified when there are unrated items because we have to define projections (19) and (20) taking account of unrated items. Omitting the user index \( u \) here for the sake of notational simplicity, the \( e \)-projection in the Algorithm 1 is modified as

\[
\hat{\theta}_i(t) = \frac{\theta_i(t) \Theta_{m|i} \times e^{-KL(\hat{\theta}_i(t) + \Theta_m(t))}}{e^{-KL(\hat{\theta}_i(t)) + \Theta_m(t)}}, \quad i \in G_m, \quad m = 1, \ldots, M
\]

for rated items, and the \( e \)-projection for unrated items is defined as

\[
\hat{\theta}_i(t) = \theta_i(t) \times \frac{1}{e^{-KL(\hat{\theta}_i(t)) + \Theta_\emptyset(t)}}, \quad i \notin \forall G_m,
\]

where \( \forall G_m \) is the set of unrated items in any group. The optimal value for \( \epsilon \) may be different depending on data sets, and it can be estimated by techniques such as cross validation. We omit to explore the optimal constant factor here, and set \( \epsilon = U/2 \) for experiments with synthetic data. For large scale real-world data, instead of adding a regularization term, we avoided the overfitting by an early stopping strategy.
where we let the sum of all the preference parameters correspond to unrated items by the user as
\[
\Theta_* (t) = \sum_{i \notin G_m} \theta_i (t),
\]
and
\[
KL (\hat{\Theta}, \Theta (t)) = \sum_{i=1}^M \hat{\Theta}_i \log \frac{\hat{\Theta}_i}{\Theta (t)_i}.
\]
When there are no unrated items, \( \Theta_* (t) = 0 \) and these formulae are reduced to the e-projection in Algorithm 1. These modified update formulae are derived as follows. To derive the e-step of the Algorithm 1, we used the solutions \( \{ \hat{\Theta}_m \}_{m=1}^M \) of the optimization problem (16) as constraints for \( \sum_{i \in G_m} \theta_i \). When there are unrated items in the grouped ranking observation, the equality constraint \( \sum_{i \in G_m} \theta_i = \hat{\Theta}_m \) is replaced by proportional constraint \( \sum_{i \in G_m} \theta_i \propto \hat{\Theta}_m \), and we use ratios of these values as constraints for \( \sum_{i \in G_m} \theta_i \). That is, letting \( \hat{\Theta}_m / \hat{\Theta}_M = c_m \), \( m = 1, \ldots, M-1 \), the submanifold defined by the grouped ranking observation with unrated items becomes
\[
D = \left\{ \theta \in \Delta_{N-1} \mid \frac{\sum_{i \in G_m} \theta_i}{\sum_{i \in G_M} \theta_j} = c_m, \; m = 1, \ldots, M-1 \right\}.
\]
Then, the e-projection from the previous estimate \( \theta (t) \) to a submanifold \( D \) is given by the solution of the following optimization problem:
\[
\min_{\theta} \quad \sum_{i=1}^N \theta_i \log \frac{\theta_i}{\Theta_i (t)},
\]
subject to
\[
\sum_{j \in G_m} \theta_j = c_m, \; m = 1, \ldots, M-1,
\]
\[
\sum_{i=1}^N \theta_i = 1.
\]
For this problem, the Lagrangian is written as
\[
F (\theta; \{ \lambda_m \}_{m=1}^{M-1}, \mu) = \sum_{i=1}^N \theta_i \log \frac{\theta_i}{\Theta_i (t)} + \sum_{m=1}^{M-1} \lambda_m \left( \frac{\sum_{j \in G_m} \theta_j}{\sum_{j \in G_M} \theta_j} - c_m \right) + \mu \left( \sum_{i=1}^N \theta_i - 1 \right)
\]
with Lagrangian multipliers \( \{ \{ \lambda_m \}_{m=1}^{M-1}, \mu \} \). Now we differentiate this Lagrangian with respect to \( \theta_i, i = 1, \ldots, N \). For each \( \theta_i \), we equate the derivative of \( F (\theta; \{ \lambda_m \}_{m=1}^{M-1}, \mu) \) to 0 and get the following equations:
\[
\theta_i = \begin{cases} 
\hat{\Theta}_i (t) \hat{\mu}^{-1} e^{-\lambda_m / \hat{\Theta}_M}, & i \in G_m, \; m = 1, \ldots, M-1, \\
\hat{\Theta}_i (t) \hat{\mu}^{-1} \exp \left( \sum_{m=1}^{M-1} \frac{\lambda_m}{\hat{\Theta}_M} \frac{\Theta_m}{\hat{\Theta}_M} \right), & i \in G_M, \\
\hat{\Theta}_i (t) \hat{\mu}^{-1}, & i \notin G_m,
\end{cases}
\]
where we define \( \hat{\mu} = e^{1+\mu} \). Then, summing up \( \theta_i \)'s in each group leads us to
\[
\Theta_m = \Theta_m (t) \hat{\mu}^{-1} e^{-\lambda_m / \hat{\Theta}_M}, \; m = 1, \ldots, M-1,
\]
\[
\Theta_M = \Theta_M (t) \hat{\mu}^{-1} \exp \left( \sum_{m=1}^{M-1} \frac{\lambda_m}{\hat{\Theta}_M} \frac{\Theta_m}{\hat{\Theta}_M} \right),
\]
\[
\Theta_* = \Theta_* (t) \hat{\mu}^{-1}.
\]
25
Solving these simultaneous equations with respect to $\lambda_m/\Theta_M$, $m = 1, \ldots, M - 1$, we get

\[
\frac{\lambda_m}{\Theta_M} = \sum_{l=1}^{M} \left( \log \frac{\Theta_m(t)}{\Theta_l(t)} - \log \frac{c_m}{c_l} \right) c_l \hat{\Theta}_M \tag{69}
\]

\[
= \sum_{l=1}^{M} \hat{\Theta}_l \log \frac{\hat{\Theta}_l(t)}{\Theta_l(t)} + \log \frac{\Theta_m(t)}{\Theta_m} \tag{70}
\]

\[
= KL(\hat{\Theta}, \Theta(t)) + \log \frac{\Theta_m(t)}{\Theta_m}, \tag{71}
\]

where we introduce $c_M = \hat{\Theta}_M = 1$.

We next consider $\bar{\mu}$. Using the fact that $\sum_{m=1}^{M-1} \Theta_m + \Theta_M + \Theta_\star = 1$, from equations (66), (67), (68), we get

\[
\bar{\mu} = \sum_{m=1}^{M-1} \Theta_m(t) e^{-\lambda_m} + \Theta_M(t) \exp \left( \sum_{l=1}^{M-1} \lambda_l \frac{c_l}{c_M} \right) + \Theta_\star(t) \tag{72}
\]

\[
= \sum_{m=1}^{M} \Theta_m(t) e^{-\lambda_m} + \Theta_\star(t), \tag{73}
\]

where we define $\lambda_M = -\sum_{m=1}^{M-1} c_m \lambda_m$. We note that equation (71) still holds for $m = M$ with this $\lambda_M$. Substituting (71) into (73), we get

\[
\bar{\mu} = \sum_{m=1}^{M} \hat{\Theta}_m e^{-KL(\hat{\Theta}, \Theta(t))} + \Theta_\star(t) = e^{-KL(\hat{\Theta}, \Theta(t))} + \Theta_\star(t). \tag{74}
\]

Finally, using (71) and (73), we get the $e$-projection formula with unrated items as (57) and (58). Since the $m$-step depends on the $e$-projections and does not depend on observations directly, the $m$-step is conducted all the same as in Algorithm 1.

**References**


26


