LETTER Communicated by John Guiver

A Grouped Ranking Model for Item Preference Parameter

Hideitsu Hino
hideitsu.hino@toki.waseda.jp
School of Science and Engineering, Waseda University, Shinjuku, Tokyo 169-8555, Japan

Yu Fujimoto
yu.fujimoto@it.aoyama.ac.jp
Department of Integrated Information Technology, Aoyama Gakuin University, Sagamihara, Kanagawa 229-8558, Japan

Noboru Murata
noboru.murata@eb.waseda.ac.jp
School of Science and Engineering, Waseda University, Shinjuku, Tokyo 169-8555, Japan

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 such model is the Plackett-Luce model, which parameterizes the preference level of each item by a real value. In this letter, 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: one focusing on the permutational order of items and the other 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 were well studied by Diaconis (1988) and remain an active research area (e.g., Murphy & Martin, 2003; Meila, Phadnis, Patterson, & Bilmes, 2007;

Huang, Guestrin, & Guibas, 2007; Lebanon & Mao, 2007; Busse, Orbanz, & Buhmann, 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 θ_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 a popular analytical tool in the machine learning community (e.g., Hastie & Tibshirani, 1998; Huang, Weng, & Lin, 2006; Takenouchi & Ishii, 2008). A natural extension of the Bradley-Terry model is given by Plackett (1975). He has proposed a sequential ranking model that we refer to as the Plackett-Luce model in this letter. 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

$$P(I_{a(1)} \succ I_{a(2)} \succ \cdots \succ I_{a(N)}) = \frac{\theta_{a(1)}}{\sum_{j=1}^N \theta_{a(j)}} \frac{\theta_{a(2)}}{\sum_{j=2}^N \theta_{a(j)}} \cdots \frac{\theta_{a(N-1)}}{\theta_{a(N-1)} + \theta_{a(N)}}$$
$$= \prod_{i=1}^{N-1} \frac{\theta_{a(i)}}{\sum_{j=i}^N \theta_{a(j)}}, \tag{1.1}$$

where $a(j)$ denotes the index of the item that occupies the jth 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. The importance of analyzing these kinds of data has been growing with the prevalence of word-of-mouth data such as movies, books, and restaurants. In this letter, 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 observe only M groups of items, which are divisions of N items. For example, when seven items $I = \{I_1, \ldots, I_7\}$ are rated with $M = 3$, we get a grouped ranking observation from a user u such as $D^u = \{G_1^u, G_2^u, G_3^u\}$, $G_1^u = \{3, 5\}$, $G_2^u = \{2, 6, 7\}$, $G_3^u = \{1, 4\}$, where G_m^u is an index set of items rated at the mth ranking by user u. Those observations are available from U users, that is, we get a set of observations $\{D^u\}_{u=1}^U$. We assume that each of N items has a preference level θ_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), formal modeling of ranking data can be divided into two types: modeling of the ranking process and modeling

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

The rest of this letter 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 (Amari, 1995; Amari & Nagaoka 2000).[1] 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, Fujimoto, & Murata, 2009).

2 Grouped Ranking Model

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

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, G_2^u be that of the next preferred items, and so on.[2] All of the index sets for M rankings by user u are 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 letter, 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 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 π_m^u on G_m^u. Let $\pi_m^u(i)$ be the index of the ith item in the ordered group $\pi_m^u(G_m^u)$. For example, by the action of a permutation $\pi_m^u = \left(\begin{smallmatrix}1&2&3\\2&3&1\end{smallmatrix}\right)$ 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 from section 1, we 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 seven 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 $\pi_2^u = \left(\begin{smallmatrix}1&2&3\\2&3&1\end{smallmatrix}\right)$. We

[1] The *em* algorithm is an information-geometric formalization of the EM algorithm (Dempster, Laird, & Rubin, 1977).

[2] For simplicity, we assume that all users rate all items. We modify the proposed algorithm in section 3 to deal with unrated items. See appendix D for the parameter estimation algorithm with unrated items.

define $\Theta^u_m = \sum_{i \in G^u_m} \theta_i$, the sum of the parameters θ_i, whose indices belong to group G^u_m. We call Θ^u_m a grouped parameter of group $G^u_m \in D^u$ henceforth. Since the items in G^u_1 are already chosen and excluded, the probability that the items in G^u_2 are chosen in this order from the remaining items is given by

$$\begin{aligned} &P\big((\pi^u_2, G^u_2)\big|G^u_1\big) \\ &= \left(\frac{\theta_7}{\sum_{n=2}^3 \Theta^u_n}\right)\left(\frac{\theta_2}{\sum_{n=2}^3 \Theta^u_n - \theta_7}\right)\left(\frac{\theta_6}{\sum_{n=2}^3 \Theta^u_n - (\theta_7+\theta_2)}\right) \\ &= \prod_{i=1}^{\gamma^u_2} \frac{\theta_{\pi^u_2(i)}}{\sum_{n=2}^3 \Theta^u_n - \sum_{j<i} \theta_{\pi^u_2(j)}}, \end{aligned} \tag{2.1}$$

where $P((\pi^u_m, G^u_m)|G^u_1, \ldots, G^u_{m-1})$ denotes the conditional probability that the items in group G^u_m are chosen in the order specified by π^u_m when the items in groups $\{G^u_1, \ldots, G^u_{m-1}\}$ have been already chosen. Note that the orderings in the previous groups $G^u_1, \ldots, G^u_{m-1}$ do not influence the ordering π^u_m in the mth group.

Now let us generalize the above example. When we consider the complete observation $\{(\pi^u_1, G^u_1), \ldots, (\pi^u_M, G^u_M)\}$, the permutations π^u_m's can be regarded as latent variables. We can decompose the probability of observing this complete ranking data as

$$\begin{aligned} &P\big((\pi^u_1, G^u_1), \ldots, (\pi^u_M, G^u_M)\big) \\ &= P\big((\pi^u_1, G^u_1)\big) P\big((\pi^u_2, G^u_2)\big|G^u_1\big) \cdots P\big((\pi^u_M, G^u_M)\big|G^u_1, \ldots, G^u_{M-1}\big). \end{aligned} \tag{2.2}$$

We define the probability of observing the group G^u_m with the order π^u_m under the condition that groups $\{G^u_1, \ldots, G^u_{m-1}\}$ have been already chosen as

$$P\big((\pi^u_m, G^u_m)\big|G^u_1, \ldots, G^u_{m-1}\big) = \prod_{i=1}^{\gamma^u_m} \frac{\theta_{\pi^u_m(i)}}{\sum_{n=m}^M \Theta^u_n - \sum_{j<i} \theta_{\pi^u_m(j)}}. \tag{2.3}$$

Noting that the latent order in each group is expressed by the action of a permutation π^u_m on G^u_m, we let $\mathcal{S}(G^u_m)$ be the set of all the possible $\gamma^u_m!$ permutations of the items on the group G^u_m. Then the probability of occurrence of group G^u_m is a sum of the joint probabilities over possible latent orders

$$P\big(G^u_m\big|G^u_1, \ldots, G^u_{m-1}\big) = \sum_{\pi^u_m \in \mathcal{S}(G^u_m)} P\big((\pi^u_m, G^u_m)\big|G^u_1, \ldots, G^u_{m-1}\big), \tag{2.4}$$

which is nothing but a marginalization of permutations. We note that when $G^u_m = \emptyset$, we skip the correspondent index m in summation or product by

convention. From this probability for the mth rated group of items, the likelihood of the datum $D^u = \{G_1^u, \ldots, G_M^u\}$ is defined as

$$P(D^u) = \prod_{m=1}^{M} \sum_{\pi_m^u \in \mathcal{S}(G_m^u)} P\left((\pi_m^u, G_m^u) \middle| G_1^u, \ldots, G_{m-1}^u\right), \tag{2.5}$$

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 A. From equations 2.3 and 2.4, we get a log likelihood of a group G_m^u as

$$l(\theta; m, u) = \log \left(\sum_{\pi_m^u \in \mathcal{S}(G_m^u)} \prod_{i=1}^{\gamma_m^u} \frac{\theta_{\pi_m^u(i)}}{\sum_{n=m}^{M} \Theta_n^u - \sum_{j<i} \theta_{\pi_m^u(j)}} \right). \tag{2.6}$$

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

$$L(\theta) = \sum_{u=1}^{U} \sum_{m=1}^{M} \log \left(\sum_{\pi_m^u \in \mathcal{S}(G_m^u)} \prod_{i=1}^{\gamma_m^u} \frac{\theta_{\pi_m^u(i)}}{\sum_{n=m}^{M} \Theta_n^u - \sum_{j<i} \theta_{\pi_m^u(j)}} \right). \tag{2.7}$$

Note that maximization of the likelihood for our model is apparently a difficult task, particularly for large numbers of items or users. The main complexity comes from latent ordering 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 section, we approximate the log-likelihood function 2.7 to exclude marginalization or permutations. The denominator in expression 2.6 reflects the normalization of parameters θ_i's in the sequential item selection. By replacing the denominator in equation 2.6 by $\sum_{n=m}^{M} \Theta_n^u$, we get a lower bound of the equation as

$$\begin{aligned} \underline{l}(\theta; m, u) &= \log \left(\sum_{\pi \in \mathcal{S}(G_m^u)} \prod_{i=1}^{\gamma_m^u} \frac{\theta_{\pi_m^u(i)}}{\sum_{n=m}^{M} \Theta_n^u} \right) \\ &= \log \left(\gamma_m^u! \prod_{i \in G_m^u} \frac{\theta_i}{\sum_{n=m}^{M} \Theta_n^u} \right) \leq l(\theta; m, u). \end{aligned} \tag{2.8}$$

Note that by this replacement, the marginalization in equation 2.7 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 section 3.2.

We next consider the upper bound of $\underline{l}(\theta; m, u)$. Using the arithmetic-geometric mean inequality

$$\left(\prod_{i \in G_m^u} \theta_i\right)^{1/\gamma_m^u} \leq \frac{1}{\gamma_m^u} \sum_{i \in G_m^u} \theta_i = \frac{1}{\gamma_m^u} \Theta_m^u, \tag{2.9}$$

we get an inequality

$$\gamma_m^u! \prod_{i \in G_m^u} \frac{\theta_i}{\sum_{n=m}^M \Theta_n^u} = \gamma_m^u! \frac{\prod_{i \in G_m^u} \theta_i}{\left(\sum_{n=m}^M \Theta_n^u\right)^{\gamma_m^u}} \leq \gamma_m^u! \left(\frac{\Theta_m^u/\gamma_m^u}{\sum_{n=m}^M \Theta_n^u}\right)^{\gamma_m^u}. \tag{2.10}$$

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

$$\begin{aligned} \tilde{l}(\theta; m, u) &= \log \left\{ \gamma_m^u! \left(\frac{\Theta_m^u/\gamma_m^u}{\sum_{n=m}^M \Theta_n^u}\right)^{\gamma_m^u} \right\} \\ &= \log \gamma_m^u! + \gamma_m^u \left(\log \Theta_m^u - \log \gamma_m^u - \log \left(\sum_{n=m}^M \Theta_n^u \right) \right) \\ &= \gamma_m^u \left\{ \log \Theta_m^u - \log \left(\sum_{n=m}^M \Theta_n^u \right) \right\} + \log \gamma_m^u! - \gamma_m^u \log \gamma_m^u. \end{aligned} \tag{2.11}$$

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 $\tilde{L}(\theta) = \sum_{u=1}^U \sum_{m=1}^M \tilde{l}(\theta; m, u)$ is smaller than the exact log likelihood $L(\theta)$ with quite high probability. Therefore, we have the following approximate expression:

$$\underline{l}(\theta; m, u) \leq \tilde{l}(\theta; m, u) \lesssim l(\theta; m, u). \tag{2.12}$$

An experiment to support this approximation is shown in section 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

$$\tilde{L}(\theta) = \sum_{u=1}^U \sum_{m=1}^M \gamma_m^u \left(\log \frac{\Theta_m^u}{\sum_{n=m}^M \Theta_n^u} \right) + const. \tag{2.13}$$

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

We can now apply any nonlinear optimizer to maximize $\tilde{L}(\theta)$ with respect to θ. However, direct optimization of $\tilde{L}(\theta)$ may cause two possible problems. The first is computational complexity. Because the approximation $\tilde{L}(\theta)$ is a nonlinear 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 superlinearly with the size of items. The second is more serious when online 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 propose an alternative approach to estimate the preference parameter.

Huang et al. (2006) 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

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

This reduces to the original case, equation 1.1, when each G_m^u contains only one item and M equals 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, equation 2.13. 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 give only rankings of those groups. It may be suitable for estimating an individual player's skill from scores of teams such as football or baseball games where each player plays many times and possibly sometimes plays on 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 $\underline{l}(\theta; m, u)$, and to get closer to the exact likelihood, we gave an upper bound of $\underline{l}(\theta; m, u)$. Intuitively speaking, expression 2.13 for $\tilde{L}(\theta)$ corresponds to the log likelihood of the probability that each user sequentially chooses the mth group γ_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 & Nagaoka, 2000) and show the effectiveness of the algorithm by simple experiments using synthetic data.

3.1 Algorithm Derivation. Our motivation for maximizing the likelihood is to find the preference parameter θ, which is consistent with all the observations as much as possible. For this purpose, we decompose the first term of equation 2.13 to U independent optimization problems with respect to group parameters Θ_m^u, which correspond to one grouped ranking observation $D^u = \{G_1^u, \ldots, G_M^u\}$:

$$\max_{\{\Theta_m^u\}} \sum_{m=1}^{M} \gamma_m^u \log \left(\frac{\Theta_m^u}{\sum_{n=m}^{M} \Theta_n^u} \right), \quad \text{subject to} \quad \Theta_m^u > 0, \quad \sum_{m=1}^{M} \Theta_m^u = 1. \tag{3.1}$$

These 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 equation 3.1 and solve

$$\begin{aligned} &\max_{\{\Theta_m^u\}} \sum_{m=1}^{M} \gamma_m^u \log \left(\frac{\Theta_m^u}{\sum_{n=m}^{M} \Theta_n^u} \right) + \sum_{m=1}^{M} \frac{1}{M} \log \Theta_m^u, \\ &\text{subject to} \quad \sum_{m=1}^{M} \Theta_m^u = 1 \end{aligned} \tag{3.2}$$

instead of equation 3.1. 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 a detailed discussion of identifiability and regularization.)

Now our problem is finding the optimal parameter θ that is the most consistent with U sets of grouped parameters $\{\Theta_m^u\}_{m=1}^M$, $u = 1, \ldots, U$. 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 3.2 can be seen as incomplete observations of the preference parameter. Because the preference parameters θ have constraints $^\forall i,\ \theta_i > 0$ and $\sum_{i=1}^{N} \theta_i = 1$, they form a manifold

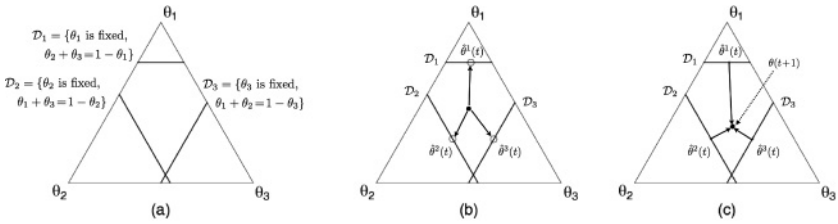

Figure 1: (a) Each observation defines submanifold $\mathcal{D}_u$ of Δ_{N-1}. (b) Find the nearest point $\hat{\theta}^u(t)$ from the previous estimate $\theta(t)$ to each submanifold (*e*-step). (c) Find the nearest point $\theta(t+1)$ from each the points $\hat{\theta}^u(t)$ of all the submanifolds (*m*-step).

known as the standard $(N-1)$-simplex denoted by Δ_{N-1}. The solution $\{\hat{\Theta}^u_m\}_{m=1}^M$ of equation 3.2 defines a submanifold of Δ_{N-1} (see Figure 1a):

$$\mathcal{D}_u = \left\{ \theta \mid \sum_{i \in G^u_m} \theta_i = \hat{\Theta}^u_m,\ m = 1, \ldots, M \right\} \subset \Delta_{N-1}. \tag{3.3}$$

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

$$L_{em}(\theta) = \sum_{u=1}^U KL(\mathcal{D}_u, \theta) = \sum_{u=1}^U \min_{\theta^u \in \mathcal{D}_u} KL(\theta^u, \theta) \tag{3.4}$$

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 tth iteration. In the *e*-step, we find a point $\hat{\theta}^u(t)$ on the submanifold $\mathcal{D}_u$ that is the nearest in terms of the KL divergence from the previous estimate $\theta(t)$ (see Figure 1b), that is,

$$\hat{\theta}^u(t) = \arg\min_{\theta \in \mathcal{D}_u} KL(\theta, \theta(t)). \tag{3.5}$$

This procedure is called the *e*-projection. We use the Lagrangian multiplier method to write the *e*-projection as

$$\hat{\theta}^u_i(t) = \frac{\theta_i(t)}{\sum_{j \in G^u_{m|i}} \theta_j(t)} \hat{\Theta}^u_{m|i}, \quad i = 1, \ldots, N,\ u = 1, \ldots, U, \tag{3.6}$$

where $G^u_{m|i}$ denotes the group to which I_i belongs and $\hat{\Theta}^u_{m|i}$ denotes the corresponding group parameter.

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

$$\theta(t+1) = \arg\min_{\theta} \sum_{u=1}^{U} KL(\hat{\theta}^u(t), \theta). \tag{3.7}$$

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

$$\theta_i(t+1) = \frac{1}{U} \sum_{u=1}^{U} \hat{\theta}^u_i(t) \ , \quad i = 1, \ldots, N. \tag{3.8}$$

Iterating these e- and m-steps until neither $\theta(t)$ nor $\hat{\theta}^u(t)$ changes, we get a (locally) optimal parameter $\theta(t)$. The derivations of the e- and m-projection formulas 3.6 and 3.8 are given in appendix C. 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 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 θ may still be computationally demanding. As an approximation approach, instead of maximizing $\tilde{L}(\theta)$, we considered the small-sized optimization problems 3.2 with respect to the grouped parameters $\{\Theta^u_m\}_{m=1}^M$ for individual users. The solutions of problems 3.2 are used to define observation submanifolds, and the most consistent preference parameter θ 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.

3.2.1 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

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:

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

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 θ.

Figure 2: The *em* algorithm for the grouped ranking model. The algorithm finds a preference parameter that 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 equations D.1 and D.2 in appendix D for the data with unrated items.

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 the 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.

3.2.2 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 $\tilde{l}(\theta; m, u)$, we used two inequalities. The first inequality comes from omitting the term $\sum_{j<i}\theta_{\pi^u_m(j)}$ in

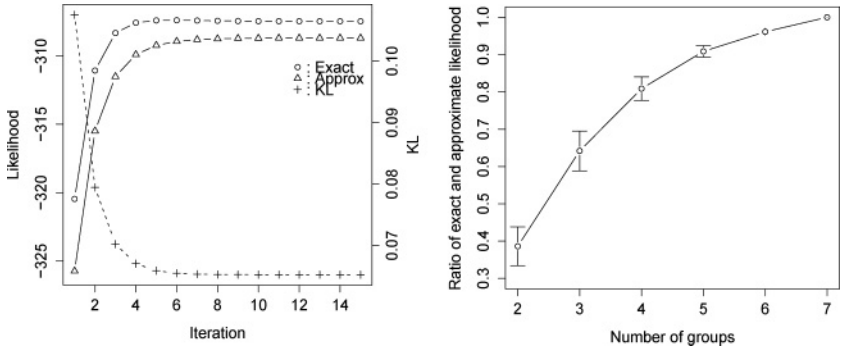

Figure 3: (Left) Average values of the exact and approximate likelihood, and the average KL divergence between true and estimated parameters. (Right) Average ratios $|L(\theta)/\tilde{L}(\theta)|$ with one standard deviation error bars for various group numbers.

the denominator of equation 2.6. When the group size γ_m^u is small enough, the number of possible permutations in the group is also small, so the effect of omitting the term $\sum_{j<i} \theta_{\pi_m^u(j)}$ is small. The second is the arithmetic-geometric mean inequality, where equality holds if and only if all $\theta_{\pi_m^u(i)}$'s in the group G_m^u are the same value. It is unlikely that all $\theta_{\pi_m^u(i)}$'s in a group are the same, except the case that γ_m^u is very small. When M is large, most 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 1. We calculated the ratio $|L(\theta)/\tilde{L}(\theta)|$ 100 times using randomly chosen parameters θ for each setting of M. Figure 3 (right) shows the average of the ratio $|L(\theta)/\tilde{L}(\theta)|$ with one-standard-deviation error bars. We can see that $L(\theta)$ is bounded below by $\tilde{L}(\theta)$, and the ratio approaches 1 as the number of groups increases. Therefore, the exact likelihood $L(\theta)$ is bounded below by the approximate likelihood $\tilde{L}(\theta)$, and we can expect maximization of $\tilde{L}(\theta)$ to obtain a good estimate of parameter θ 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 B, we analytically consider the gap between the exact and approximate likelihoods.

3.2.3 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

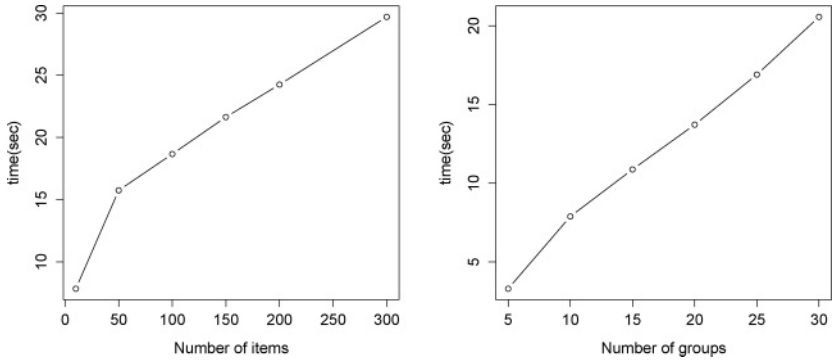

Figure 4: Average CPU time of the proposed algorithm for various item numbers (left) and group numbers (right).

time depends on the number of users U linearly, we fixed $U = 100$ and consider only N and M.

We measured the time for estimating the parameter by the proposed method; 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 input-output is excluded. The initialization part of the algorithm involves solving U optimization problems (equation 2.15). 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.

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

$$P(x) = \sum_{k=1}^{K} \omega_k P(x; \theta^k), \tag{4.1}$$

where x is a datum, ω_k is a prior probability that a datum is generated from the kth element model, and θ^k is the preference parameter of the kth model. In principle, the parameters of the mixture model, equation 4.1, $\{\omega_k, \theta^k\}_{k=1}^K$ can be estimated by the EM algorithm (Dempster et al., 1977).

[3]With 2.4 GHz dual processors and 4096 MB memory. The operating system is Mac OS X version 10.5.8.

When we consider a mixture of grouped ranking models, however, there is a difficulty mentioned in section 4.2. To avoid this difficulty, we introduce the entropy regularized K-means soft clustering algorithm (Sahbi & Boujemaa, 2005) in the parameter space of the grouped ranking model.

4.1 Entropy Regularized Soft Clustering. The soft clustering, also called 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 the degree of the datum x^u belongs to the kth 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 $\sum_{k=1}^K g_{uk} = 1,\ u = 1, \ldots, U$. 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 ξ^k of the kth 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 of them. 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 function

$$\begin{aligned} J_\lambda(\{g_{uk}\}, \{\xi^k\}) &= \sum_{u=1}^U \sum_{k=1}^K g_{uk} d_{uk} - \lambda \sum_{u=1}^U H(g_u) \\ &= \sum_{u=1}^U \sum_{k=1}^K g_{uk} d_{uk} + \lambda \sum_{u=1}^U \sum_{k=1}^K g_{uk} \log g_{uk}, \end{aligned} \tag{4.2}$$

is minimized with respect to the membership g_{uk} and the cluster representatives ξ^k, subject to the constraints $\sum_{k=1}^K g_{uk} = 1,\ u = 1, \ldots, U$. The first term of equation 4.2 represents a cost for cluster assignment to each datum x^u. If a datum x^u is far from a cluster center ξ^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 ξ^k, and the distance d_{uk} at the tth 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 is easily derived using the Lagrangian multiplier method as

$$g_{uk}(t) = \frac{\exp\left(-\frac{d_{uk}(t-1)}{\lambda}\right)}{\sum_{l=1}^K \exp\left(-\frac{d_{ul}(t-1)}{\lambda}\right)}. \tag{4.3}$$

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 ξ^k is defined as

$$d_{uk} = KL(D^u, \xi^k) = \min_{\theta \in \mathcal{D}_u} KL(\theta, \xi^k). \tag{4.4}$$

The membership update procedure is the same as equation 4.3. The cluster representative update is not as straightforward as that of membership and needs the *em* algorithm again:

e-**step:** *e*-projection from a cluster representative ξ^k to the data submanifolds $\mathcal{D}_u$:

$$\hat{\theta}^{uk} = \arg\min_{\hat{\theta}^u \in \mathcal{D}_u} KL(\hat{\theta}^u, \xi^k), \quad u = 1, \ldots, U.$$

m-**step:** *m*-projection from data submanifolds $\mathcal{D}_u$'s to the cluster representatives ξ^k:

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

We can derive more tangible formula for updates using the Lagrangian multipliers in each step. On a submanifold $\mathcal{D}_u$, the ith component of the point closest to the cluster representative ξ^k is given by

$$\hat{\theta}_i^{uk} = \frac{\xi_i^k}{\sum_{j \in G^u_{m|i}} \xi_j^k} \Theta^u_{m|i},$$

and the ith component of the cluster representative ξ^k is updated as

$$\hat{\xi}_i^k = \frac{\sum_{u=1}^{U} g_{uk} \hat{\theta}_i^{uk}}{\sum_{u=1}^{U} g_{uk}}.$$

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

4.2 Difficulty in Applying the EM Algorithm. In this section, we show that the EM algorithm, a standard approach for estimating 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:

$$P(D^u) = \sum_{k=1}^{K} \omega_k P(D^u; \theta^k). \tag{4.5}$$

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 λ.

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 $g_{uk}(t)$:

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

Cluster Representatives Update: update cluster representatives ξ^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 $\mathcal{D}_u$.

repeat: from $s = 0$ until convergence

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

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

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

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

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

output: converged parameters $\{g_{uk}\}, \{\xi^k\}$.

Figure 5: Soft clustering algorithm for the mixture of the grouped ranking model. The *e*-projection formula is modified according to equations D.1 and D.2 in appendix D for the data with unrated items.

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

$$P(D^u, k; \theta) = \omega_k P(D^u; \theta^k) = \omega_k \prod_{m=1}^M P\left(G^u_m \middle| G^u_1, \ldots, G^u_{m-1}; \theta^k\right), \qquad (4.6)$$

the Q function in the EM algorithm is written as

$$Q(\theta|\theta(t)) = \sum_{u=1}^{U}\sum_{k=1}^{K} P(k|D^u;\theta(t))\log P(D^u,k;\theta), \tag{4.7}$$

where θ 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 tth iteration of the EM algorithm.

Given a datum D^u by a user u, the probability that D^u is generated from the kth model is calculated as

$$P(k|D^u;\theta(t)) = \frac{P(D^u,k;\theta(t))}{\sum_{l=1}^K P(D^u,l;\theta(t))} = \frac{\omega_k P(D^u;\theta^k(t))}{\sum_{l=1}^K \omega_l P(D^u;\theta^l(t))}. \tag{4.8}$$

Substituting equations 4.6 and 4.8 into function 4.7, we maximize equation 4.7 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 ω_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

$$\tilde{Q}(\theta|\theta(t)) = \sum_{u=1}^{U}\sum_{k=1}^{K} \frac{\omega_k(t)\tilde{P}(D^u;\theta^k(t))}{\sum_{l=1}^K \omega_l(t)\tilde{P}(D^u;\theta^l(t))}(\log\omega_k + \log\tilde{P}(D^u;\theta^k)), \tag{4.9}$$

where

$$\tilde{P}(D^u;\theta^k) = \prod_{m=1}^{M}\gamma_m^u! \frac{\left(\frac{\Theta_n^{uk}}{\gamma_m^u}\right)^{\gamma_m^u}}{\left(\sum_{n=m}^M \Theta_n^{uk}\right)^{\gamma_m^u}}.$$

Since direct maximization of equation 4.9 with respect to θ_i^k is difficult, we again apply the *em* algorithm to this update step by dividing $\tilde{Q}(\theta|\theta(t))$ 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 does not work properly, because the inequality $Q(\theta|\theta(t)) \geq \tilde{Q}(\theta|\theta(t))$ does not hold in many cases even though $P(D^u;\theta^k) \gtrsim \tilde{P}(D^u;\theta^k)$ holds. The modified Q function 4.9 contains the term

$$\tilde{P}(k|D^u;\theta(t)) = \frac{\omega_k(t)\tilde{P}(D^u;\theta^k(t))}{\sum_{l=1}^K \omega_l(t)\tilde{P}(D^u;\theta^l(t))}, \tag{4.10}$$

and this term often becomes larger than the exact value

$$P(k|D^u;\theta(t)) = \frac{\omega_k(t)P(D^u;\theta^k(t))}{\sum_{l=1}^K \omega_l(t)P(D^u;\theta^l(t))}, \tag{4.11}$$

and maximizing $\tilde{Q}$ does not always guarantee Q to be maximized. This eliminates the approximating approach. In our preliminary experiments, the estimated probabilities 4.8 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 $\tilde{P}(D^u;\theta) \lesssim P(D^u;\theta)$. 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 equations 4.11 and 4.10 is written as

$$\begin{aligned}&\frac{P(D^u;\theta^k(t))}{\sum_{l=1}^K P(D^u;\theta^l(t))} - \frac{\tilde{P}(D^u;\theta^k(t))}{\sum_{l=1}^K \tilde{P}(D^u;\theta^l(t))} \\ &\quad= \frac{\alpha_k}{\sum_{l=1}^K \alpha_l} - \frac{\alpha_k - \varepsilon_k}{\sum_{l=1}^K(\alpha_l - \varepsilon_l)} = \frac{\sum_{l=1}^K(\alpha_l\varepsilon_k - \alpha_k\varepsilon_l)}{\left(\sum_{l=1}^K \alpha_l\right)\sum_{l=1}^K(\alpha_l - \varepsilon_l)}.\end{aligned} \tag{4.12}$$

The gap, equation 4.12, tends to be positive when α_k is relatively small and negative when α_k is large. Since $0 \le P(k|D^u;\theta) \le 1$, $\tilde{P}(k|D^u;\theta)$ is likely to be overestimated for a large $P(k|D^u;\theta)$ and 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: a tool for data analysis and visualization and 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). 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 $\theta^k = (\theta_1^k, \ldots, \theta_N^k),\ k = 1, \ldots, K$. Accordingly, each item can be characterized by K ideal users' preference and can be seen as a point in K-dimensional space as

$$(\theta_i^1, \ldots, \theta_i^K) \in \mathbb{R}^K. \tag{5.1}$$

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

$$\boldsymbol{u} = (P(1|D^u), \ldots, P(K|D^u)) \tag{5.2}$$

can be regarded as a directional vector for a user u, where the kth dimension of $\boldsymbol{u}$ represents the probability that the user u belongs to the kth "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, equations 5.1 and 5.2, 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

$$I_i \mapsto \theta_i^u, \quad \theta_i^u = \sum_{k=1}^{K} \theta_i^k P(k|D^u), \tag{5.3}$$

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 section. 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 close to each other. There is some research on the preference visualization. In Zenebe and Norcio (2007), demographic information such as gender and age is attached to each user, and items (movies in the literature) are vaguely divided into clusters of genres, such as "action," "horror," and "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 the 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 defined by equation 5.3, which are represented as half lines in $\mathbb{R}^K$.[4] 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 kinds of data. For such data, it seems to be natural to represent items as points and users as lines.

5.1.1 Experiments of the Item-User Visualization. We will demonstrate the visualization method explained above with the MovieLens rating data set (Riedl & Konstan, 2000). This data set consists of 100,000 ratings ranging from 1 to 5 for 1682 movies by 943 users. For our demonstration purpose, we used a only 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 (see equation 4.5) 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 five 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 (θ_i^1, θ_i^2). Using this visualization scheme, we can see that, for example, the movies *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 of users into the two-dimensional space. Each user is expressed by a line in this space defined by $\boldsymbol{u} = (P(1|D^u), P(2|D^u))$ as a directional vector, and we can interpret this line as representing a user's evaluation standard. From this figure, we can infer that users 4 and 6 share similar tastes, while user 1 enjoys quite different movies. 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 equation 5.3 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, 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 that are likely of interest to the user automatically. For this

[4] Note that it is not a dual space, which is defined as a vector space consisting of all linear functions.

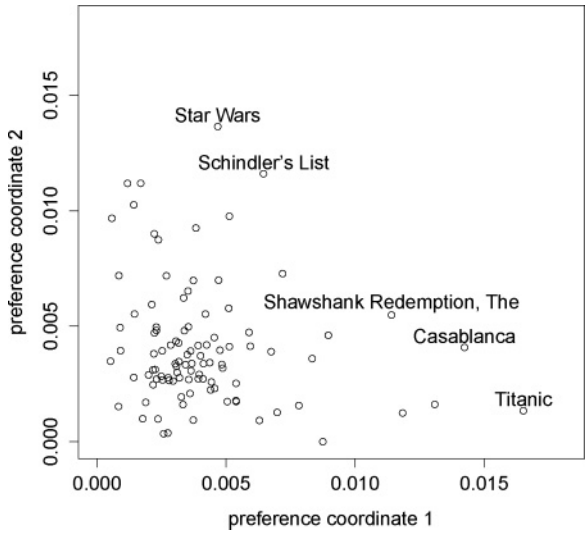

Figure 6: An example of item mapping. Items are mapped into 2D space. Movies with high parameter value are mapped with their titles.

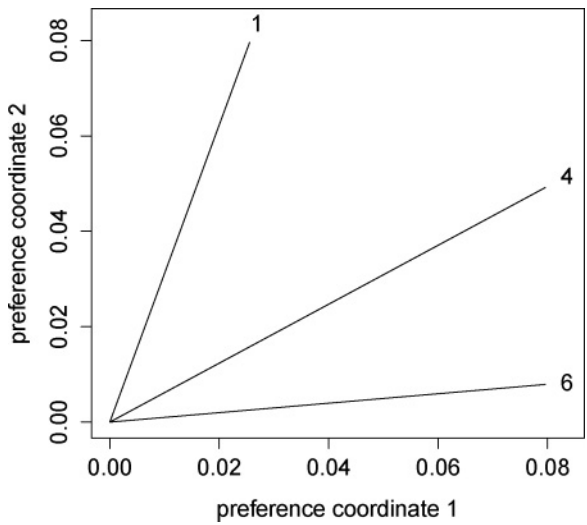

Figure 7: An example of user mapping. Three users are mapped into 2D space as lines defined by equation 5.2.

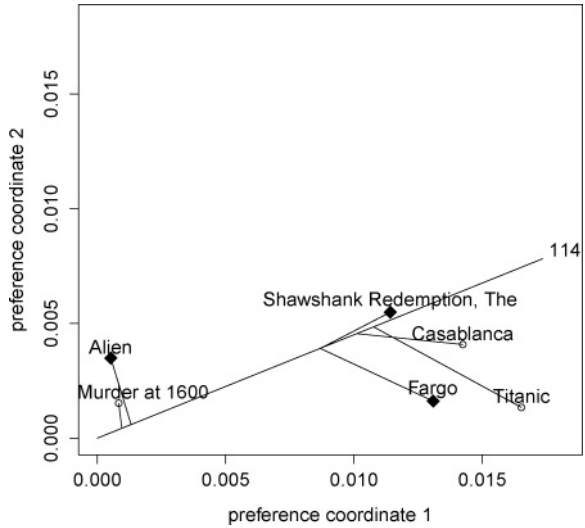

Figure 8: Example of item-user mapping for 2D space. Movies already rated by user 114 are depicted as circle, and movies not rated by this user are depicted as diamonds. The parameter values of the items for user 114 are given by the projection of the item to the user axis defined by equation 5.3.

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 they give to the same items. (See Adomavicius & Tuzhilin, 2005, for a comprehensive survey.) There is a huge amount of research on collaborative filtering systems, and each technique has advantages and drawbacks. In addition, each technique must be finely tuned for each specific occasion and data sets to be used. Here we do not attempt to make a thorough comparison and show only that our proposal works comparably to 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

$$r_{u,i} = \bar{r}_u + \frac{1}{C} \sum_{v \in ne(i)} \mathrm{sim}(u, v)(r_{v,i} - \bar{r}_v), \tag{5.4}$$

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 $C = \sum_{v \in ne(i)} |\mathrm{sim}(u, v)|$. This prediction formula highly depends on the similarity measure between users. One of the most

famous collaborative filtering systems is GroupLens (Resnick et al., 1994), which used Pearson's correlation as the similarity measure,

$$\mathrm{sim}_{Pe}(u, v) = \frac{\sum_{i \in S_{u,v}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in S_{u,v}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in S_{u,v}} (r_{v,i} - \bar{r}_v)^2}}, \tag{5.5}$$

where $S_{u,v}$ is a subset of items that 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:

$$\mathrm{sim}_{cos}(u, v) = \frac{\sum_{i \in S_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in S_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in S_{u,v}} r_{v,i}^2}}. \tag{5.6}$$

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 $\theta^u = (\theta_1^u, \ldots, \theta_N^u)$, where the parameters

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

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 the Fisher kernel (Jaakkola & Haussler, 1999) provides a principled similarity measure that takes into account an underlying probability distribution. We use the Fisher score $\boldsymbol{s}_u = \partial_\theta \log \tilde{P}(D^u; \theta^u)$ 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

$$\mathrm{sim}_{GR}(u, v) = \frac{\boldsymbol{s}_u^T \boldsymbol{s}_v}{||\boldsymbol{s}_u|| \cdot ||\boldsymbol{s}_v||}. \tag{5.7}$$

5.2.1 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, McNee, Konstan, & Lausen, 2005). The GroupLens research team kindly provided the five-level movie rating data divided for 80% training subset (that is, 80,000 rating data) and 20% test subset (20,000 rating data) for five-fold cross-validation. We implemented the similarity-based collaborative filtering system with Pearson's similarity (see equation 5.5), cosine similarity (see equation 5.6),

[5] Also known as the normalized Fisher kernel (Shawe-Taylor & Cristianini, 2004).

Table 1: Accuracy Evaluation Result of Collaborative Filtering.

	Pearson	Cosine	Fisher Cosine
MovieLens	$\mathbf{0.712} \pm 0.0069$	0.720 ± 0.0055	0.721 ± 0.0056
BookCrossing	1.283 ± 0.0145	1.254 ± 0.0151	$\mathbf{1.253} \pm 0.0133$

and cosine of the Fisher score similarity (see equation 5.7). For each method, 80% of the data is used to calculate the similarity between users, and 20% is used to evaluate the rate prediction accuracy. The BookCrossing data set contains 1,149,780 ratings (433,681 explicit ratings on a 1–10 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 they are 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 eight items and items that are rated by at least 12 users. The resulting data set contains 48,484 ratings by 663 users for 1191 items. We divide this down-sized data set for five-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

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

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 existing collaborative filtering methods in rating prediction accuracy.

6 Conclusion

In this letter, we proposed a probability model for grouped ranking observations parameterized by an item-preference-level parameter. To the best of our knowledge, this is the first proposal of an item ranking model that

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 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 twofold. 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 important, advantage is applicability to online 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 have to solve only a small maximization problem, equation 3.2, with respect to the new datum, and we can obtain a revised estimate θ by iterating equations 3.6 and 3.8 of the algorithm with $U := U + 1$. For real-world applications such as collaborative filtering systems, we need to handle vast numbers of data that 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 nonparametric distance-based partial ranking model from the viewpoint of group theory and derived an efficient estimation method. It is interesting to investigate under what circumstances parametric models explain real data better than nonparametric models do, 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 is other research that considers 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, Hino, & Murata, 2009).

The accuracy of the recommendation based on Fisher scores of our proposed model is comparable to the conventional methods. We believe that research on recommender systems is shifting in 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, Herbrich, and Graepel (2009) proposed a framework to predict ratings based on Bayesian inference; 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 future work.

Appendix A: Details of the Proposed Probability Model

In this appendix, we provide details on the 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 unrated items. Almost all ranking models must address this unrated items problem, and we will show how to treat such data in appendix D. The second incompleteness is that rankings are given in the form of grouped ranking, the principal subject of this letter.

The grouped ranking observation D^u is composed of groups of items $\{G_m^u\}_{m=1}^M$. The number of items in each group G_m^u 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 & Mao, 2007).

Definition. *A composition of N into M is a sequence $\gamma^u = (\gamma_1^u, \ldots, \gamma_M^u)$ of positive integers whose sum is N.*

A composition $\gamma^u = (\gamma_1^u, \ldots, \gamma_M^u)$ corresponds to a grouped ranking with γ_1^u items in the first group G_1^u, γ_2^u items in the second group G_2^u, 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 $O^u = (I_{u(1)} \succ I_{u(2)} \succ \cdots \succ I_{u(N)})$, where $u(i)$ denotes the index of the ith ranked item.
2. Divide N items into M groups without changing item ordering.[6] We denote the divided ranking data according to the user's composition γ^u as

$$(O^u, \gamma^u) = (\underbrace{(I_{u(1)} \succ \cdots \succ I_{u(\gamma_1^u)})}_{\gamma_1^u}, \ldots, \underbrace{(I_{u(\sum_{m=1}^{M-1}\gamma_m^u+1)} \succ \cdots \succ I_{u(N)})}_{\gamma_M^u}), \tag{A.1}$$

and identify the pair (O^u, γ^u) to a grouped ranking data $D^u = \{G_1^u, \ldots, G_M^u\}$.

[6] This step is assumed to be independent of the first ranking step.

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 reports only ratings without ordering in the same rated items. The data O^u are 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 $\{D^u = \{G_m^u\}_{m=1}^M\}_{u=1}^U$, 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.

The proposed ranking model is certainly a probability model. We note that enumeration of all possible configurations in groups $\{G_m^u\}_{m=1}^M$ for all possible compositions γ^u for N items is equivalent to enumeration of all possible orderings of N items in the Plackett-Luce model, that is,

$$\sum_{\gamma^u} \sum_{\{G_m^u|\gamma^u\}_{m=1}^M} P(\{G_m^u\}_{m=1}^M) = \sum_{\gamma^u} \sum_{\{G_m^u|\gamma^u\}_{m=1}^M} P((O^u, \gamma^u)) \tag{A.2}$$

$$= \sum_u P(I_{u(1)} \succ I_{u(2)} \succ \cdots \succ I_{u(N)}) = 1, \tag{A.3}$$

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

Appendix B: Further Analysis of Approximate Likelihood

In this appendix, we 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 $\sum_{\pi_m^u \in \mathcal{S}(G_m^u)}$ by fixing π_m^u to a permutation π_m^{u*} that maximizes $l(\theta; m, u)$, and denote the log likelihood with the permutation π_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

$$\begin{aligned} l(\theta; m, u) - \tilde{l}(\theta; m, u) &\leq l^*(\theta; m, u) - \tilde{l}(\theta; m, u) \\ &= \log \gamma_m^u! + \sum_{i \in G_m^u} \log \theta_i - \sum_{i=1}^{\gamma_m^u} \log \left(\sum_{n=m}^{M} \Theta_n^u - \sum_{j<i} \theta_{\pi_m^{u*}(j)} \right) \\ &\quad - \left[\gamma_m^u \left\{ \log \Theta_m^u - \log \left(\sum_{n=m}^{M} \Theta_n^u \right) \right\} + \log \gamma_m^u! - \gamma_m^u \log \gamma_m^u \right] \end{aligned}$$

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

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

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

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 γ_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; 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 C: 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 $\mathcal{D}_u = \{ \theta \mid \sum_{i \in G_m^u} \theta_i = \hat{\Theta}_m^u \}$ as

$$\hat{\theta}^u(t) = \arg\min_{\theta \in \mathcal{D}_u} KL(\theta, \theta(t)) , \quad u = 1, \ldots, U, \tag{C.1}$$

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

$$\min_\theta \sum_{i \in G_m^u} \theta_i \log \frac{\theta_i}{\theta_i(t)} , \qquad \text{subject to} \qquad \sum_{i \in G_m^u} \theta_i = \hat{\Theta}_m^u, \ \theta_i > 0. \tag{C.2}$$

We can solve optimization problem C.2 by introducing the Lagrange multiplier λ and the Lagrangian $F(\theta, \lambda)$ as

$$F(\theta, \lambda) = \sum_{i \in G_m^u} \theta_i \log \frac{\theta_i}{\theta_i(t)} + \lambda \left(\hat{\Theta}_m^u - \sum_{i \in G_m^u} \theta_i \right). \tag{C.3}$$

Differentiating this Lagrangian with respect to θ_i, $i \in G_m^u$ and equating to 0, we get

$$\frac{\partial F}{\partial \theta_i} = \log \frac{\theta_i}{\theta_i(t)} + 1 - \lambda = 0. \tag{C.4}$$

By the constraint in equation C.2, we get

$$\lambda = \log \left(\frac{\hat{\Theta}_m^u}{\sum_{i \in G_m^u} \theta_i(t)} \right) + 1,$$

and the *e*-projection is proved to be

$$\hat{\theta}_i^u(t) = \frac{\theta_i(t)}{\sum_{j \in G_{m|i}^u} \theta_j(t)} \hat{\Theta}_{m|i}^u, \quad i = 1, \ldots, N, \ u = 1, \ldots, U, \tag{C.5}$$

where $G_{m|i}^u$ denotes the group to which I_i belongs and $\hat{\Theta}_{m|i}^u$ denotes the corresponding group parameter.

In the *m*-projection, we find a point $\theta(t+1)$ in the manifold Δ_{N-1} that 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,

$$\begin{aligned}
\theta(t+1) &= \arg\min_\theta \frac{1}{U} \sum_{u=1}^{U} KL(\hat{\theta}^u(t), \theta) \\
&= \arg\min_\theta \frac{1}{U} \sum_{u=1}^{U} \sum_{i=1}^{N} \left(\hat{\theta}_i^u(t) \log \hat{\theta}_i^u(t) - \hat{\theta}_i^u(t) \log \theta_i \right) \\
&= \arg\max_\theta \frac{1}{U} \sum_{u=1}^{U} \sum_{i=1}^{N} \hat{\theta}_i^u(t) \log \theta_i \qquad \text{(C.6)} \\
&= \arg\max_\theta \sum_{i=1}^{N} p_i \log \theta_i, \qquad \text{(C.7)}
\end{aligned}$$

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 μ and the Lagrangian $H(\theta, \mu)$ as

$$H(\theta, \mu) = \sum_{i=1}^{N} p_i \log \theta_i + \mu \left(1 - \sum_{i=1}^{N} \theta_i \right). \tag{C.8}$$

Differentiating this Lagrangian with respect to θ_i, $i = 1, \ldots, N$, and equating to 0, we get

$$\frac{\partial H}{\partial \theta_i} = \frac{p_i}{\theta_i} - \mu = 0. \tag{C.9}$$

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

$$\theta_i(t+1) = p_i = \frac{1}{U} \sum_{u=1}^{U} \hat{\theta}_i^u(t). \tag{C.10}$$

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, equation 3.4, like

$$L_{em}^{\mathrm{reg}}(\theta) = \sum_{u=1}^{U} \min_{\theta^u \in \mathcal{D}_u} KL(\theta^u, \theta) + \epsilon KL(\theta_{\mathrm{unif}}, \theta), \tag{C.11}$$

where $\theta_{\mathrm{unif}} = (\frac{1}{N}, \ldots, \frac{1}{N})$. Then the m-step (see equation 3.8) of the algorithm is modified as

$$\theta_i(t+1) = \frac{1}{U+\epsilon} \left(\sum_{u=1}^{U} \hat{\theta}_i^u(t) + \epsilon \frac{1}{N} \right), \quad i = 1, \ldots, N. \tag{C.12}$$

We adopted regularized version C.12 of the algorithm in experiments with synthetic data. The optimal value for ϵ may be different depending on data sets, and it can be estimated by techniques such as cross-validation. We omit exploration of 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 D: Treatment of Unrated Items

In this appendix, 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 algorithm 1 does not change, and we just solve the M variable optimization problem, equation 3.2. However, the *e*-step of the *em* algorithm must be modified when there are unrated items because we have to define projections 3.5 and 3.6, taking account of unrated items. Omitting the user index u here for

notational simplicity, we modify the *e*-projection in algorithm 1 as

$$\hat{\theta}_i(t) = \frac{\theta_i(t)}{\sum_{j \in G_{m|i}} \theta_i(t)} \hat{\Theta}_{m|i} \times \frac{e^{-KL(\hat{\Theta}, \Theta(t))}}{e^{-KL(\hat{\Theta}, \Theta(t))} + \Theta_*(t)}, \quad i \in G_m,\ m = 1, \ldots, M \tag{D.1}$$

for rated items, and the *e*-projection for unrated items is defined as

$$\hat{\theta}_i(t) = \theta_i(t) \times \frac{1}{e^{-KL(\hat{\Theta}, \Theta(t))} + \Theta_*(t)},\ i \notin {}^{\forall}G_m, \tag{D.2}$$

where we let the sum of all the preference parameters correspond to unrated items by the user as

$$\Theta_*(t) = \sum_{i \notin {}^{\forall}G_m} \theta_i(t), \tag{D.3}$$

and $KL(\hat{\Theta}, \Theta(t)) = \sum_{l=1}^{M} \hat{\Theta}_l \log \frac{\hat{\Theta}_l}{\Theta_l(t)}$. When there are no unrated items, $\Theta_*(t) = 0$ and these formulas are reduced to the *e*-projection in algorithm 1. These modified update formulas are derived as follows. To derive the *e*-step of algorithm 1, we used the solutions $\{\hat{\Theta}_m\}_{m=1}^{M}$ of the optimization problem 3.2 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

$$\mathcal{D} = \left\{ \theta \in \Delta_{N-1} \,\middle|\, \frac{\sum_{i \in G_m} \theta_i}{\sum_{i \in G_M} \theta_i} = c_m,\ m = 1, \ldots, M-1 \right\}. \tag{D.4}$$

Then, the *e*-projection from the previous estimate $\theta(t)$ to a submanifold $\mathcal{D}$ is given by the solution of the following optimization problem:

$$\min_{\theta} \sum_{i=1}^{N} \theta_i \log \frac{\theta_i}{\theta_i(t)}, \tag{D.5}$$

$$\text{subject to } \frac{\sum_{j \in G_m} \theta_j}{\sum_{j \in G_M} \theta_j} = c_m,\ m = 1, \ldots, M-1, \tag{D.6}$$

$$\sum_{i=1}^{N} \theta_i = 1. \tag{D.7}$$

For this problem, the Lagrangian is written as

$$F\left(\theta, \{\lambda_m\}_{m=1}^{M-1}, \mu\right) = \sum_{i=1}^{N} \theta_i \log \frac{\theta_i}{\theta_i(t)} + \sum_{m=1}^{M-1} \lambda_m \left(\frac{\sum_{j \in G_m} \theta_j}{\sum_{j \in G_M} \theta_j} - c_m \right) + \mu \left(\sum_{i=1}^{N} \theta_i - 1 \right)$$

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 θ_i, we equate the derivative of $F(\theta, \{\lambda_m\}_{m=1}^{M-1}, \mu)$ to 0 and get the following equations:

$$\theta_i = \begin{cases} \theta_i(t)\bar{\mu}^{-1} e^{-\lambda_m/\Theta_M}, & i \in G_m,\ m = 1, \ldots, M-1, \\ \theta_i(t)\bar{\mu}^{-1} \exp\left(\sum_{m=1}^{M-1} \frac{\lambda_m}{\Theta_M} \frac{\Theta_m}{\Theta_M} \right), & i \in G_M, \\ \theta_i(t)\bar{\mu}^{-1}, & i \notin {}^{\forall} G_m, \end{cases} \tag{D.8}$$

where we define $\bar{\mu} = e^{1+\mu}$. Then, summing up θ_i's in each group leads us to

$$\Theta_m = \Theta_m(t)\bar{\mu}^{-1} e^{-\lambda_m/\Theta_M}, \quad m = 1, \ldots, M-1, \tag{D.9}$$

$$\Theta_M = \Theta_M(t)\bar{\mu}^{-1} \exp\left(\sum_{m=1}^{M-1} \frac{\lambda_m}{\Theta_M} \frac{\Theta_m}{\Theta_M} \right), \tag{D.10}$$

$$\Theta_* = \Theta_*(t)\bar{\mu}^{-1}. \tag{D.11}$$

Solving these simultaneous equations with respect to λ_m/Θ_M, $m = 1, \ldots, M-1$, we get

$$\frac{\lambda_m}{\Theta_M} = \sum_{l=1}^{M} \left(\log \frac{\Theta_m(t)}{\Theta_l(t)} - \log \frac{c_m}{c_l} \right) c_l \hat{\Theta}_M \tag{D.12}$$

$$= \sum_{l=1}^{M} \hat{\Theta}_l \log \frac{\hat{\Theta}_l}{\Theta_l(t)} + \log \frac{\Theta_m(t)}{\hat{\Theta}_m} \tag{D.13}$$

$$= KL(\hat{\Theta}, \Theta(t)) + \log \frac{\Theta_m(t)}{\hat{\Theta}_m}, \tag{D.14}$$

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

We next consider $\bar{\mu}$. Using the fact that $\sum_{m=1}^{M-1} \Theta_m + \Theta_M + \Theta_* = 1$, from equations D.9 to D.11, we get

$$\bar{\mu} = \sum_{m=1}^{M-1} \Theta_m(t) e^{-\frac{\lambda_m}{\Theta_M}} + \Theta_M(t) \exp\left(\sum_{l=1}^{M-1} \frac{\lambda_l}{\Theta_M} c_l \right) + \Theta_*(t) \tag{D.15}$$

$$= \sum_{m=1}^{M} \Theta_m(t) e^{-\frac{\lambda_m}{\Theta_M}} + \Theta_*(t), \tag{D.16}$$

where we define $\lambda_M = -\sum_{m=1}^{M-1} c_m \lambda_m$. We note that equation D.14 still holds for $m = M$ with this λ_M. Substituting equation D.14 into D.16,

$$\bar{\mu} = \sum_{m=1}^{M} \hat{\Theta}_m e^{-KL(\hat{\Theta}, \Theta(t))} + \Theta_*(t) = e^{-KL(\hat{\Theta}, \Theta(t))} + \Theta_*(t). \tag{D.17}$$

Finally, using equations D.14 and D.16, we get the *e*-projection formula with unrated items as equations D.1 and D.2. Since the *m*-step depends on the *e*-projections and not on observations directly, the *m*-step is conducted all the same as in algorithm 1.

Acknowledgments

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

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749.

Amari, S. (1995). Information geometry of the EM and *em* algorithms for neural networks. *Neural Networks*, *8*(9), 1379–1408.

Amari, S., & Nagaoka, K. (2000). *Methods of information geometry*. New York: Oxford University Press.

Bradley, R. A., & Terry, M. (1952). The rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, *39*, 324–345.

Busse, L. M., Orbanz, P., & Buhmann, J. M. (2007). Cluster analysis of heterogeneous rank data. In *Proceedings of the 24th International Conference on Machine learning* (pp. 113–120). San Francisco: Morgan Kaufmann.

Croon, M. A., & Luijkx, R. (1993). *Latent structure models for ranking data*. Berlin: Springer.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, *39*(1), 1–38.

Diaconis, P. (1988). *Group representations in probability and statistics*. Beachwood, OH: Institute of Mathematical Statistics.

Fligner, M. A., & Verducci, J. S. (1986). Distance based ranking models. *Journal of the Royal Statistical Society, Series B*, *48*(3), 359–369.

Fujimoto, Y., Hino, H., & Murata, N. (2009). *Item-user preference mapping with mixture models: Data visualization for item preference.* Paper presented at the International Conference on Knowledge Discovery and Information Retrieval, October 6–8, Madeira, Portugal.

Guiver, J., & Snelson, E. (2009). Bayesian inference for Plackett-Luce ranking models. In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 377–384). San Francisco: Morgan Kaufmann.

Hastie, T., & Tibshirani, R. (1998). Classification by pairwise coupling. *Annals of Statistics*, *26*(2), 451–471.

Hino, H., Fujimoto, Y., & Murata, N. (2009). Item preference parameters from grouped ranking observations. In *Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 875–882). Berlin: Springer.

Hofmann, T. (2000). Learning the similarity of documents: An information-geometric approach to document retrieval and categorization. In S. A. Solla, S. A., Leen, T. K., & Muller, K. R. (Eds.), *Advances in neural information processing systems, 12* (pp. 914–920). Cambridge, MA: MIT Press.

Huang, J., Guestrin, C., & Guibas, L. (2007). Efficient inference for distributions on permutations. In J. C. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems, 20* (pp. 697–704). Cambridge, MA: MIT Press.

Huang, T., Weng, R. C., & Lin, C. (2006). Generalized Bradley-Terry models and multi-class probability estimates. *Journal of Machine Learning Research*, *7,* 85–115.

Hunter, D. R. (2004). MM algorithms for generalized Bradley-Terry models. *Annals of Statistics*, *32*(1), 384–406.

Jaakkola, T. S., & Haussler, D. (1999). Exploiting generative models in discriminative classifiers. In M. S. Kearns, S. A., Solla, & D. A. Cohn, (Eds.), *Advances in neural information processing systems* (pp. 487–493). Cambridge, MA: MIT Press.

Kamishima, T., & Akaho, S. (2006). Efficient clustering for orders. In *Proceedings of the 2nd International Workshop on Mining Complex Data* (pp. 274–278). Berlin: Springer.

Lebanon, G., & Mao, Y. (2007). Non-parametric modeling of partially ranked data. In J. C. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems, 20* (pp. 857–864). Cambridge, MA: MIT Press.

Mallows, C. L. (1957). Non-null ranking models. I. *Biometrika*, *44*(1/2), 114–130.

Marden, J. I. (1995). *Analyzing and modeling rank data.* London: Chapman & Hall.

Mei, G., & Shelton, C. R. (2006). Visualization of collaborative data. In *Proceedings of the 22nd International Conference on Uncertainty in Artificial Intelligence* (pp. 341–348). San Francisco: Morgan Kaufmann.

Meila, M., Phadnis, K., Patterson, A., & Bilmes, J. (2007). Consensus ranking under the exponential model. In *Proceedings of 22nd Conference on Uncertainty in Artificial Intelligence* (pp. 285–294). San Francisco: Morgan Kaufmann.

Murphy, T. B., & Martin, D. (2003). Mixtures of distance-based models for ranking data. *Computational Statistics and Data Analysis*, *41*(3–4), 645–655.

Plackett, R. L. (1975). The analysis of permutations. *Applied Statistics*, *24*(2), 193–202.

R Development Core Team. (2009). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., & Riedl, J. (1994). GroupLens: An Open architecture for collaborative filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work* (pp. 175–186). San Francisco: Morgan Kaufmann.

Riedl, J., & Konstan, J. (2000). *MovieLens dataset*. Available online at http://www.grouplens.org/.

Sahbi, H., & Boujemaa, N. (2005). Fuzzy clustering: Consistency of entropy regularization. *Advances in Soft Computing*, *2*, 95–107.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge: Cambridge University Press.

Stern, D. H., Herbrich, R., & Graepel, T. (2009). Matchbox: Large scale online Bayesian recommendations. In *Proceedings of the 18th International World Wide Web Conference* (pp. 111–120). Norwell, MA: Kluwer.

Takenouchi, T., & Ishii, S. (2008). Ternary Bradley-Terry model-based decoding for multi-class classification. In *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing* (pp. 121–126). Piscataway, NJ: IEEE Press.

Zenebe, A., & Norcio, A. F. (2007). Visualization of item features, customer preference and associated uncertainty using fuzzy sets. In *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society* (pp. 7–12). Piscataway, NJ: IEEE Press.

Ziegler, C., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on the World Wide Web* (pp. 22–32). Norwell, MA: Kluwer.

Received November 19, 2009; accepted February 17, 2010.