Performance Evaluation 77 (2014) 72–95

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Mathematical modeling of group product recommendation with partial information: How many ratings do we need?

Hong Xie, John C.S. Lui *

Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 7 January 2013
Received in revised form 8 January 2014
Accepted 25 March 2014
Available online 12 April 2014

Keywords:
Group recommendation systems
Partial preference information
Randomized algorithm
Performance evaluation

ABSTRACT

Product recommendation is one of the most important services in the Internet. In this paper, we consider a product recommendation system which recommends products to a *group of users*. The recommendation system only has *partial preference information* on this group of users: a user only indicates his preference to a small *subset* of products in the form of ratings. This partial preference information makes it a challenge to produce an accurate recommendation. In this work, we explore a number of fundamental questions. What is the desired number of ratings per product so to guarantee an accurate recommendation? What are some effective voting rules in summarizing ratings? How users' misbehavior such as *cheating*, in product rating may affect the recommendation accuracy? What are some efficient rating schemes? To answer these questions, we present a formal mathematical model of a group recommendation system. We formally analyze the model. Through this analysis we gain the insight to develop a randomized algorithm which is both computationally efficient and asymptotically accurate in evaluating the recommendation accuracy under a very general setting. We propose a novel and efficient *heterogeneous rating scheme* which requires equal or less rating workload, but can improve over a homogeneous rating scheme by as much as 30%. We carry out experiments on both synthetic data and real-world data from TripAdvisor. Not only we validate our model, but also we obtain a number of interesting observations, i.e., a small of misbehaving users can decrease the recommendation accuracy remarkably. For TripAdvisor, one hundred ratings per product is sufficient to guarantee a high accuracy recommendation. We believe our model and methodology are important building blocks to refine and improve applications of group recommendation systems.

1. Introduction

Nowadays, we are living in the information age with information overload. To deal with such overload, *recommender systems* [1] were introduced which suggest products (hotels, books, songs, etc.) to a user by taking into account the preference of that user. Recommender systems have drawn a lot of attention from both commercial and academic communities over the last decade. We see a number of successful commercial recommender systems like Amazon.com [2], MovieLens [3], etc. A lot of research works have been done on investigating various algorithmic and complexity issues in designing recommender systems [1,4–7]. This type of recommender systems aim to make recommendations to one user and they are also called *classic recommender systems*.

* Corresponding author.
E-mail addresses: hxie@cse.cuhk.edu.hk (H. Xie), cslui@cse.cuhk.edu.hk (J.C.S. Lui).

http://dx.doi.org/10.1016/j.peva.2014.03.004
0166-5316/© 2014 Published by Elsevier B.V.

However, when users operate in *groups, classic recommender systems* are not appropriate, because the system has to make recommendations by taking into account the preferences of all users within a group instead of one user. Examples of such contexts can be, recommending movies to a number of friends planning to watch together [8], recommending videos to an interest group on YouTube, etc. To deal with such contexts, *group recommendation systems* [9] were introduced. They aim to provide recommendations to a group of users maximizing the overall utility of that group. Recently, a number of successful commercial products of group recommendation systems have emerged [8,10–13]. In this paper, we consider group recommendation systems with *partial preference information*: there are a number of products and a number of users operates in a group, and each user only show his preference to a small *subset* of products in the form of ratings. The system applies some rating aggregation policies to summarize the ratings, and recommends a subset of products to a group users.

The *partial preference information* makes it a challenge to make an accurate recommendation. It is important for us to understand the *accuracy* and *effectiveness* of a group product recommendation system. However this is a challenging work, since a number of rating and human factors may affect the recommendation accuracy and effectiveness. Little attention has been made to this fundamental problem. In this paper, we explore a number of fundamental questions to fill in this void. *What is the desired number of ratings per product so to guarantee an accurate recommendation? What are some effective voting rules in summarizing ratings? How users' misbehavior such as cheating, in product rating may affect the recommendation accuracy? What are some efficient rating schemes?* To the best of our knowledge, this is the first paper which provides a formal model and analysis of such kind of systems. To summarize, our paper makes the following contributions:

- We propose a mathematical model to capture various factors which may influence the accuracy of a group product recommendation system under *partial preference information* settings.
- We formally analyze the model. Through this we gain the insight to develop a randomized algorithm to evaluate the recommendation accuracy under a general setting. We show that this algorithm is computationally efficient and also provides theoretical performance guarantees.
- We propose an efficient *two round heterogeneous rating scheme* which outperforms the homogeneous rating scheme by as much as 30% in recommendation accuracy with the same or less rating workload.
- We carry our experiments on both synthetic data and real-world data (rating data from TripAdvisor). We not only validate our model, but also examine various factors that may affect the recommendation accuracy. We find a number of interesting observations, for example, a small of misbehaving users can decrease the recommendation accuracy remarkably. For TripAdvisor, one hundred ratings per product is sufficient to guarantee a high recommendation accuracy.

This is the outline of our paper. In Section 2, we present the mathematical model of a group recommendation system. In Section 3, we present the formal analysis of the model. In Section 4, we present an efficient randomized algorithm with theoretical performance guarantees to evaluate the recommendation accuracy. In Section 5, we present the experimental results on synthetic data. In Section 6, we present the experimental results on a real-world dataset (from TripAdvisor). Related work is given in Sections 7 and 8 concludes.

2. Mathematical model

We consider a group product recommendation system which recommends k products from a finite set of N candidates denoted by $P_1, \dots, P_N$, to a group of M users $\mathcal{U} = \{U_1, \dots, U_M\}$, taking into account the collective preference of the whole user population with that group. Note that $1 \le k \le N$. Users show their preferences in the form of product rating. More concretely, a user only expresses ratings to a small *subset* of products on an m-level cardinal metric denoted by $\{1, \dots, m\}$. Higher rating implies higher preference. For example, a 2-level (or binary) cardinal metric could be: $\{1 = \text{dislike}, 2 = \text{like}\}$. Ratings from different users are independent. We use the notation $\mathbf{r}_i = \{r_{i,1}, \dots, r_{i,M}\}$ to denote a set of ratings for product P_i, where $r_{i,j} \in \{1, \dots, m\}$ if user U_j rates product P_i, otherwise $r_{i,j} = 0$ denotes a missing rating. Let $n_i = |\{r_{i,j} \in \{1, \dots, m\}, \forall j\}|$ denote the number of observed ratings for product P_i. We treat the observed ratings, say $r_{i,j} \in \{1, \dots, m\}, \forall i, j$, as *partial preference information*. To decide whether a product should be recommended, the systems infers the collective preference of the user group $\mathcal{U}$ via evaluating ratings. The *partial preference information* makes it challenge to infer the collective preference accurately. There are a number of interesting questions to explore, i.e., how will the number of ratings per product affect the accuracy of the overall recommendation? To guarantee an accurate recommendation, what is the minimum number ratings per product? How users' misbehavior (such as *cheating*) in product rating may affect the final recommendation? *The objective of this work is to examine how various factors can influence the recommendation accuracy.*

To make a recommendation, the system applies a voting rule $\mathcal{V}$ to summarize ratings of a product. Many voting rules are possible. A simple and widely used voting rule is the *average score rule*. Let $\gamma_i = \mathcal{V}(\mathbf{r}^i)$ denote the aggregate rating of product P_i. For the *average score rule*, we compute the aggregate rating as $\gamma_i = \sum_j r_{i,j}/n_i$. There are a number of interesting questions to explore, i.e., what are some effective voting rules? Can one voting rule be more accurate than others?

However, specifying the voting rule is not enough. Recall that the system can only recommend k products. Suppose we rank products based on their aggregate ratings. It may happen that the aggregate rating of the kth ranked product, equals to that of the $(k+1)$th ranked product. In this case, we need to specify a *tie-breaking rule* to decide which product should be recommended. Let $\mathcal{T}$ denote a tie-breaking rule. In this work, we also explore whether the recommendation accuracy is sensitive to a particular tie-breaking rule.

To answer the above questions, we present the recommendation accuracy measure, the probabilistic model in describing collective preference as well as the product quality, in the following.

2.1. Recommendation accuracy measure

We use the notation $Q_i \in (1, m)$ to represent the overall *quality* of product P_i in the view of the user group $\mathcal{U}$. The value of Q_i reflects the true collective preference of the user group $\mathcal{U}$. We formally define it as follows.

Definition 1 (*Product Quality*)**.** We define the quality of a product in the view of the user group $\mathcal{U}$ as its average rating under full preference information, i.e., all users within the group $\mathcal{U}$ express ratings to it, mathematically

$$Q_i \triangleq \sum_j r_{i,j}/M,$$

where all ratings are observed, say $r_{i,j} \in \{1, \ldots, m\}, \ \forall j$.

Remark. In the presence of full information, we could have access to the quality of a product, say the true collective preference of a group of users. But in reality, it is costly or impossible to achieve that ideal place, instead we only have access *partial preference information* on ratings. Thus the system and users do not have knowledge on product quality $Q_i, \ \forall i$ in general.

In the following of this work, we call Q_i the product quality, for simplicity. We use the term *full information* to denote the ideal scenario where for each product, all users express ratings to it.

Higher value of Q_i implies higher collective preference. In other words $Q_i > Q_j$ implies that the user group $\mathcal{U}$ prefers product P_i over P_j. Without loss of any generality, let us assume $Q_1 > Q_2 > \cdots > Q_N$. It is important to emphasize that users and the system do *not* have any a-prior knowledge of $Q_i, \ \forall i$. Let $\mathcal{R}^I(k)$ and $\mathcal{R}(k)$ denote two sets of k recommended products according to the product quality, say $Q_i, \ \forall i$, or according to the system's recommendation criteria respectively (i.e., applying a voting rule and a tie breaking rule to ratings, etc.). It is clear that $\mathcal{R}^I(k) = \{P_1, P_2, \ldots, P_k\}$, and if a group recommendation system is perfect, we should have $\mathcal{R}^I(k) = \mathcal{R}(k)$. But in general, many human factors may influence the final recommendation, hence $\mathcal{R}^I(k) \neq \mathcal{R}(k)$. To measure the accuracy of a group product recommendation system, we aim to determine how many products in $\mathcal{R}(k)$ are also in $\mathcal{R}^I(k)$. Formally, we seek to derive the following probability mass function (pmf):

$$\Pr\left[|\mathcal{R}^I(k) \cap \mathcal{R}(k)| = i\right], \quad \text{for } i = 0, 1, \ldots, k.$$

Intuitively, if $\Pr[|\mathcal{R}^I(k) \cap \mathcal{R}(k)| = k]$ occurs with a high probability, then the group product recommendation system is *accurate* and at the same time, robust against different rating and human factors.

2.2. Model for collective preference

When makes a recommendation, the system extracts the collective preferences of a group of users. We consider one most important factor that may affect the collective preference, say degree of homophyly among users. Homophyly degree measures the similarity of users. More specifically, the higher the homophyly degree, the more likely that users express similar product ratings. Under the ideal scenario of *full information*, for product P_i, we have access to ratings from all users, which form a distribution $\mathcal{P}_i = \{p_{i,1}, \ldots, p_{i,m}\}$, over the space of rating points $\{1, \ldots, m\}$. Specifically, we express this distribution as

$$p_{i,\ell} \triangleq \text{ fraction of users that assign a rating } \ell \text{ to the product } P_i.$$

Note that users and the system do not have any a-prior knowledge on $\mathcal{P}_i$. One can vary the variance of $\mathcal{P}_i$ to reflect the homophyly degree. Specifically, the higher the homophyly degree, the lower the variance of $\mathcal{P}_i$. We use the notation $h \in [0, 1]$ to represent the homophyly degree among users within the user group $\mathcal{U}$. With the usual convention, higher value of h represents higher homophyly degree. The distribution $\mathcal{P}_i$ should have the following two properties:

Property 1. *Its mean equals to Q_i. This captures that all users rate honestly.*

Property 2. *Its variance reflects the degree of homophyly h. Specifically, the higher the value of h, the lower the variance.*

In our study, we capture the above characteristics of the collective preference by discretizing normal distributions. More concretely, we obtain the probability distribution $\mathcal{P}_i$ by mapping a normal distribution $\mathcal{N}\left(Q_i, \sigma^2(h)\right)$ to a discrete distribution on $\{1, \ldots, m\}$. Note that the standard variance $\sigma(h)$ is a monotonic decreasing function of h and we will specify

it later. The probability distribution mapping process can be described by the following two steps:
Discretization: We transform the normal distribution $\mathcal{N}\left(Q_i, \sigma^2(h)\right)$ into a discrete probability distribution $\widetilde{\mathcal{N}}\left(Q_i, \sigma^2(h)\right)$ on $\{1, \dots, m\}$, whose pmf is expressed as:

$$\begin{aligned}\Pr[L = \ell] &= \Pr[\ell - 0.5 \le X \le \ell + 0.5] / \Pr[0.5 \le X \le m + 0.5] \\ &= \frac{\Phi\left((\ell + 0.5 - Q_i)/\sigma(h)\right) - \Phi\left((\ell - 0.5 - Q_i)/\sigma(h)\right)}{\Phi\left((m + 0.5 - Q_i)/\sigma(h)\right) - \Phi\left((0.5 - Q_i)/\sigma(h)\right)}, \quad \text{for } \ell = 1, \dots, m \end{aligned} \tag{1}$$

where $\Phi(x) = \int_{-\infty}^{x} \exp(-t^2/2)/\sqrt{2\pi}\,dt$ and X and L are two random variables with probability distribution $\mathcal{N}\left(Q_i, \sigma^2(h)\right)$ and $\widetilde{\mathcal{N}}\left(Q_i, \sigma^2(h)\right)$ respectively. Observe that this discrete distribution satisfies Property 2 but not Property 1. In the following step, we adjust the distribution so that it satisfies Property 1 also.
Adjustment: Adjust the distribution $\widetilde{\mathcal{N}}\left(Q_i, \sigma^2(h)\right)$ such that its mean equals to Q_i. The idea is that if its mean is smaller than Q_i, then we increase its mean by scaling up the probability:

$$\Pr[L = \ell], \quad \text{for all } \ell = \lfloor Q_i \rfloor + 1, \dots, m,$$

with the same portion. Otherwise, we decrease its mean by scaling the above probability down. Applying this idea to adjust the mean of $\widetilde{\mathcal{N}}\left(Q_i, \sigma^2(h)\right)$, we obtain the probability distribution $\mathcal{P}_i = \{p_{i,1}, \dots, p_{i,m}\}$, namely:

$$p_{i,\ell} = \begin{cases}[1 - \beta(Q_i, h)]\Pr[L = \ell], & \ell = 1, \dots, \lfloor Q_i \rfloor \\ [1 + \alpha(Q_i, h)]\Pr[L = \ell], & \ell = \lfloor Q_i \rfloor + 1, \dots, m,\end{cases} \tag{2}$$

where $\alpha(Q_i, h)$ and $\beta(Q_i, h)$ are:

$$\alpha(Q_i, h) = \frac{\sum_{\ell=1}^{\lfloor Q_i \rfloor} \Pr[L = \ell](Q_i - E[L])}{\sum_{\ell=1}^{\lfloor Q_i \rfloor} \Pr[L = \ell](E[L] - \ell)}, \tag{3}$$

$$\beta(Q_i, h) = \frac{\sum_{\ell=\lfloor Q_i \rfloor + 1}^{m} \Pr[L = \ell](Q_i - E[L])}{\sum_{\ell=1}^{\lfloor Q_i \rfloor} \Pr[L = \ell](E[L] - \ell)}, \tag{4}$$

and L is a discrete random variable with probability distribution $\widetilde{\mathcal{N}}\left(Q_i, \sigma^2(h)\right)$. Note that probability distribution $\mathcal{P}_i$ now satisfies both Properties 1 and 2.

Remark. In our study we treat each observed rating as a random sample from the probability distribution $\mathcal{P}_i$.

2.3. Model for product quality via preference matching

We model product quality via preference matching. Specifically, we classify products into types. For example, films can be classified into romantic films, science fiction films, etc. And users can be of many types also, for example prefer romantic films, or prefer science fiction films, etc. If a product–user pairing is of the same type, then the user is likely to express a high rating, otherwise a low rating is likely to be assigned. Collectively, if users within group $\mathcal{U}$ and a product match in the same type, then that product trend to receive high ratings, which result in high quality of that product (in the view of user group $\mathcal{U}$), otherwise, result in low quality. The product quality $Q_1, \dots, Q_N$ can be treated as N independent random samples generated from distribution $\mathcal{Q}$, and one can vary the mean or variance of $\mathcal{Q}$ to reflect the collective matching degree between users and products. Specifically, a high value of mean and a small value of variance imply that products are of high quality (in the view of user group $\mathcal{U}$) and these products have small variation in quality. On the other hand, a low value of mean and a large value of variance imply that products have low quality and have high variability.

In this study, the probability distribution $\mathcal{Q}$ is a truncated normal distribution. More concretely, $\mathcal{Q}$ is obtained by truncating the normal distribution $\mathcal{N}(q, \sigma_q^2)$ to keep those values in $(1, m)$ and scaling up the kept values by $1/\Pr[1 < X < m]$, where X is a random variable with probability distribution $\mathcal{N}(q, \sigma_q^2)$. It should be clear that after this truncation, the mean q and the variance σ_q^2 can still reflect the collective matching degree. In this study, we use the following parameters to reflect three representative types of collective matching degree:
High collective matching degree: The mean q and variance σ_q^2 are:

$$q = m, \quad \sigma_q^2 = 1. \tag{5}$$

This indicates that products tend to have high quality (or high mean), and most of the probability mass concentrates around high quality.

Medium collective matching degree: The mean q and variance σ_q^2 are:

$$q = (m+1)/2, \quad \sigma_q^2 = 1. \tag{6}$$

This reflects that products tend to have an medium quality and most of the probability mass concentrates around the medium quality.

Low collective matching degree: The mean q and variance σ_q^2 are:

$$q = 1, \quad \sigma_q^2 = 1. \tag{7}$$

This indicates that products tend to have low quality (or low mean), and most of the probability mass concentrates around low quality.

3. Theoretical analysis

We first derive the following probability mass function (pmf):

$$\Pr\left[|\mathcal{R}^I(k) \cap \mathcal{R}(k)| = i\right], \quad \text{for } i = 0, 1, \ldots, k.$$

With this pmf, we can then derive the expectation $E[|\mathcal{R}^I(k) \cap \mathcal{R}(k)|]$ and variance $\mathrm{Var}[|\mathcal{R}^I(k) \cap \mathcal{R}(k)|]$. The above probability measures can provide us with a lot of insights, e.g., if $\Pr[|\mathcal{R}^I(k) \cap \mathcal{R}(k)| = k]$ occurs with a high probability or $E[|\mathcal{R}^I(k) \cap \mathcal{R}(k)|] \approx k$, then the group product recommendation system is very accurate and robust against different human factors. To derive this pmf, we first consider the following special case. The purpose of deriving the special case is to show the general idea of derivation, and more importantly, illustrate the underlying computational complexity.

3.1. Derivation of the special case

Let us consider a group product recommendation system which has only one type of products and one type of users (e.g., all products are romantic films and all users prefer romantic films). To simplify the case, we specify the quality of each product as follows:

$$Q_i = m - i(m-1)/(N+1), \quad \text{for } i = 1, \ldots, N. \tag{8}$$

Each product will have the same number of ratings, or $n_i = n, \ \forall i$. The voting rule $\mathcal{V}$ is the *average score rule* and we use a random rule, which randomly pick some products from a tie, to tie-break any products whose aggregate scores are the same.

Recall that the score set for product P_i is $\mathbf{r}_i = \{r_{i,1}, \ldots, r_{i,M}\}$ where $r_{i,j} \in \{1, \ldots, m\}, \ \forall j$, are observed ratings, and $r_{i,j} = 0, \ \forall j$, are missing ratings. For convenience, we use the notation $\mathbf{r}_i^o = \{r_{i,1}^o, \ldots, r_{i,n}^o\}$ to represent a set of all the observed ratings for product P_i. Recall that observed ratings $r_{i,1}^o, \ldots, r_{i,n}^o$ are independent random samples from probability distribution $\mathcal{P}_i$. We can then express the probability mass function (pmf) for these observed ratings in the following lemma.

Lemma 1. *The pmf of observed ratings $r_{i,1}^o, \ldots, r_{i,n}^o$, can be expressed as*

$$\Pr[r_{i,j}^o = \ell] = \begin{cases} \left[1 - \beta\left(m - \dfrac{i(m-1)}{N+1}, h\right)\right] \Pr[L = \ell], & \forall \ell = 1, \ldots, \lfloor Q_i \rfloor \\ \left[1 + \alpha\left(m - \dfrac{i(m-1)}{N+1}, h\right)\right] \Pr[L = \ell], & \forall \ell = \lfloor Q_i \rfloor + 1, \ldots, m \\ 0, \quad \textit{otherwise} \end{cases} \tag{9}$$

where L is a discrete random variable with probability distribution $\widetilde{\mathcal{N}}\left(m - i(m-1)/(N+1), \sigma^2(h)\right)$ whose pmf is derived by Eq. (1), and $\alpha(m - i(m-1)/(N+1), h)$, $\beta(m - i(m-1)/(N+1), h)$ are derived by Eqs. (3) and (4) respectively.

The average rating for product P_i is $\gamma_i = \sum_j r_{i,j}^o/n, \ \forall i$. By applying the above lemma, we express the probability mass function (pmf) for γ_i.

Lemma 2. *The pmf for the averages rating $\gamma_i, \ \forall i$, can be expressed as*

$$\Pr\left[\gamma_i = \frac{\ell}{n}\right] = \sum_{\sum_{j=1}^n s_j = \ell} \prod_{j=1}^n \Pr[r_{i,j}^o = s_j], \quad \textit{for all } \ell = n, \ldots, nm, \tag{10}$$

and its cumulative distribution function (CDF) is

$$\Pr\left[\gamma_i \le \frac{\ell}{n}\right] = \sum_{\sum_{j=1}^n s_j \le \ell} \prod_{j=1}^n \Pr[r_{i,j}^o = s_j], \quad \textit{for all } \ell = n, \ldots, nm, \tag{11}$$

where $\Pr[r_{i,j}^o = s_j]$ is specified in Eq. (9).

Table 1
Examples of the analytical expression of $\Pr[\mathcal{R}(1) = \{P_{i_1}\}]$ $(n = 1, m = 2, N = 3, k = 1)$.

$\{P_{i_1}\}$	$\Pr[\mathcal{R}(1) = \{P_{i_1}\}]$
$\{P_1\}$	$\Pr[\gamma_1 = 2]\Pr[\gamma_2 = 1]\Pr[\gamma_3 = 1] + \Pr[\gamma_1 = 2]\Pr[\gamma_2 = 2]\Pr[\gamma_3 = 1]/2 + \Pr[\gamma_1 = 2]$ $\Pr[\gamma_3 = 2]\Pr[\gamma_2 = 1]/2 + \Pr[\gamma_1 = 1]\Pr[\gamma_3 = 1]\Pr[\gamma_2 = 1]/3\Pr[\gamma_1 = 2]\Pr[\gamma_3 = 2]\Pr[\gamma_2 = 2]/3$
$\{P_2\}$	$\Pr[\gamma_2 = 2]\Pr[\gamma_1 = 1]\Pr[\gamma_3 = 1] + \Pr[\gamma_2 = 2]\Pr[\gamma_1 = 2]\Pr[\gamma_3 = 1]/2 + \Pr[\gamma_2 = 2]$ $\Pr[\gamma_3 = 2]\Pr[\gamma_1 = 1]/2 + \Pr[\gamma_2 = 1]\Pr[\gamma_3 = 1]\Pr[\gamma_1 = 1]/3\Pr[\gamma_2 = 2]\Pr[\gamma_3 = 2]\Pr[\gamma_1 = 2]/3$
$\{P_3\}$	$\Pr[\gamma_3 = 2]\Pr[\gamma_2 = 1]\Pr[\gamma_1 = 1] + \Pr[\gamma_3 = 2]\Pr[\gamma_2 = 2]\Pr[\gamma_1 = 1]/2 + \Pr[\gamma_3 = 2]$ $\Pr[\gamma_1 = 2]\Pr[\gamma_2 = 1]/2 + \Pr[\gamma_3 = 1]\Pr[\gamma_1 = 1]\Pr[\gamma_2 = 1]/3\Pr[\gamma_3 = 2]\Pr[\gamma_1 = 2]\Pr[\gamma_2 = 2]/3$

Proof. Note that, the ratings of each product are independent random variables. The probability mass function (pmf) of each rating has been derived in Lemma 1. Since $\gamma_i = \sum_{j=1}^{n} r_{i,j}^{o}/n$, thus by enumerating all the cases satisfying the condition $\sum_{j=1}^{n} r_{i,j}^{o} = \ell$, we could obtain the pmf of γ_i, or

$$\Pr\left[\gamma_i = \frac{\ell}{n}\right] = \Pr\left[\sum_{j=1}^{n} r_{i,j}^{o} = \ell\right] = \sum_{\sum_{j=1}^{n} s_j = \ell} \prod_{j=1}^{n} \Pr[r_{i,j}^{o} = s_j].$$

Similarly, by enumerating all the cases satisfying the condition $\sum_{j=1}^{n} r_{i,j}^{o} \le \ell$, we could obtain the CDF of γ_i, or

$$\Pr\left[\gamma_i \le \frac{\ell}{n}\right] = \Pr\left[\sum_{j=1}^{n} r_{i,j}^{o} \le \ell\right] = \sum_{\sum_{j=1}^{n} s_j \le \ell} \prod_{j=1}^{n} \Pr[r_{i,j}^{o} = s_j],$$

which completes the proof. □

We now apply Lemmas 1 and 2, to derive the probability that a specific set of products is recommended as follows.

Lemma 3. *Let $\{P_{i_1}, \ldots, P_{i_k}\}$ be a set of k products. The probability that this set of products is recommended can be expressed as:*

$$\begin{aligned}\Pr[\mathcal{R}(k) &= \{P_{i_1}, \ldots, P_{i_k}\}] \\ &= \sum_{\ell=n}^{nm} \left(\prod_{i\in\mathcal{I}} \Pr\left[\gamma_i \le \frac{\ell}{n}\right] - \prod_{i\in\mathcal{I}} \Pr\left[\gamma_i \le \frac{\ell-1}{n}\right]\right) \prod_{j\in\overline{\mathcal{I}}} \Pr\left[\gamma_j \le \frac{\ell-1}{n}\right] \\ &\quad + \sum_{\mathcal{F}\subseteq\mathcal{I}, \mathcal{G}\subseteq\overline{\mathcal{I}}, \mathcal{F},\mathcal{G}\neq\emptyset} \binom{|\mathcal{F}\cup\mathcal{G}|}{|\mathcal{F}|}^{-1} \sum_{\ell=n}^{nm} \prod_{i\in\mathcal{I}\setminus\mathcal{F}} \left(1 - \Pr\left[\gamma_i \le \frac{\ell}{n}\right]\right) \\ &\quad \times \prod_{j\in\mathcal{F}\cup\mathcal{G}} \Pr\left[\gamma_j = \frac{\ell}{n}\right] \prod_{\kappa\in\overline{\mathcal{I}}\setminus\mathcal{G}} \Pr\left[\gamma_\kappa \le \frac{\ell-1}{n}\right],\end{aligned} \tag{12}$$

where $\mathcal{I} = \{i_1, \ldots, i_k\}$ is the index set of $\{P_{i_1}, \ldots, P_{i_k}\}$, and $\overline{\mathcal{I}} = \{1, \ldots, N\} \setminus \mathcal{I}$ is the complement of $\mathcal{I}$. And $\Pr[\gamma_i = \ell/n]$ is specified in Eq. (10), and $\Pr[\gamma_i \le \ell/n]$ is specified in Eq. (11).

Proof. Please refer to the Appendix in this paper for derivation. □

To illustrate, let us consider a simple example where $n = 1,\ m = 2,\ N = 3$ and $k = 1$. Table 1 shows some analytical expressions of $\Pr[\mathcal{R}(1) = \{P_{i_1}\}]$. One can observe that, the analytical expression is quite complicated.

We now apply Lemma 3 to derive the pmf for $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$ in the following theorem.

Theorem 1. *The pmf of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$ can be expressed as:*

$$\Pr\left[|\mathcal{R}^I(k) \cap \mathcal{R}(k)| = i\right] = \sum_{\substack{\mathcal{F}\subseteq\mathcal{R}^I(k),\\ |\mathcal{F}|=i}} \sum_{\substack{\mathcal{G}\subseteq\overline{\mathcal{R}^I}(k),\\ |\mathcal{G}|=k-i}} \Pr[\mathcal{R}(k) = \mathcal{F}\cup\mathcal{G}], \quad \text{for all } i = 0, 1, \ldots, k, \tag{13}$$

where $\overline{\mathcal{R}^I}(k) = \{P_1, \ldots, P_N\}/\mathcal{R}^I(k)$ is the complement of $\mathcal{R}^I(k)$ and $\Pr[\mathcal{R}(k) = \mathcal{F}\cup\mathcal{G}]$ is derived in Eq. (12).

Proof. The recommended product set $\mathcal{R}(k)$ can be divided into two disjoint subsets of which one is $\mathcal{R}(k) \cap \mathcal{R}^I(k)$ and the other one is $\mathcal{R}(k)\cap\overline{\mathcal{R}^I}(k)$. Note that we have derived the pmf that a specific set of recommended products in Eq. (12). Then by enumerating subsets of $\mathcal{R}(k)$ with cardinality i and all the subsets of $\overline{\mathcal{R}^I}(k)$ with cardinality $k-i$ we complete this proof. □

Table 2
Examples of the analytical expression of $\Pr[\mathcal{R}^I(1) \cap \mathcal{R}(1) = i]$, where $i = 0, 1$.

i	$\Pr[\mathcal{R}^I(1) \cap \mathcal{R}(1) = i]$
1	$\Pr[\gamma_1 = 2]\Pr[\gamma_2 = 1]\Pr[\gamma_3 = 1] + \Pr[\gamma_1 = 2]\Pr[\gamma_2 = 2]\Pr[\gamma_3 = 1]/2 + \Pr[\gamma_1 = 2]$ $\Pr[\gamma_3 = 2]\Pr[\gamma_2 = 1]/2 + \Pr[\gamma_1 = 1]\Pr[\gamma_3 = 1]\Pr[\gamma_2 = 1]/3\Pr[\gamma_1 = 2]\Pr[\gamma_3 = 2]\Pr[\gamma_2 = 2]/3$
0	$\Pr[\gamma_2 = 2]\Pr[\gamma_1 = 1]\Pr[\gamma_3 = 1] + \Pr[\gamma_3 = 2]\Pr[\gamma_2 = 1]\Pr[\gamma_1 = 1] + \Pr[\gamma_2 = 2]$ $\Pr[\gamma_1 = 2]\Pr[\gamma_3 = 1]/2 + \Pr[\gamma_3 = 2]\Pr[\gamma_1 = 2]\Pr[\gamma_2 = 1]/2\Pr[\gamma_2 = 2]\Pr[\gamma_3 = 2]$ $\Pr[\gamma_1 = 1] + \Pr[\gamma_2 = 1]\Pr[\gamma_3 = 1]\Pr[\gamma_1 = 1] \times 2/3\Pr[\gamma_2 = 2]\Pr[\gamma_3 = 2]\Pr[\gamma_1 = 2] \times 2/3$

To illustrate, let us consider a simple example with $n = 1$, $m = 2$, $N = 3$ and $k = 1$. Table 2 shows some analytical expressions for $\Pr[\mathcal{R}^I(1) \cap \mathcal{R}(1) = i]$, where $i = 0, 1$. One can observe that the analytical expressions are quite complicated.

Examining Eq. (13), one can observe that $\Pr[\mathcal{R}(k) = \mathcal{F} \cup \mathcal{G}]$ is an essential part. The analytical expression of $\Pr[\mathcal{R}(k) = \mathcal{F} \cup \mathcal{G}]$ is derived in Eq. (12), which is quite complicated and cannot be reduced to a simple form. This make it difficult to gain some insight of a group product recommendation system by directly examining this analytical expression. An alternative way is to compute numerical results for Eq. (13). After we obtain the numerical results for the pmf of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$, we can then compute its expectation and variance. By analyzing these numerical results we may gain some important insights. Unfortunately, computing the numerical results for Eq. (13) is computationally expensive, which is shown in the following theorem.

Theorem 2. *The computational complexity in calculating the numerical results for Eq.* (13) *is exponential, or* $\Theta(2^N)$.

Proof. Examining Eq. (13), we can see that the calculation of $\Pr[\mathcal{R}(k) = \mathcal{F} \cup \mathcal{G}]$ is the core part on the calculation of the pmf of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$. Assume the running time of calculating $\Pr[\mathcal{R}(k) = \mathcal{F} \cup \mathcal{G}]$ is t, then by Eq. (13), the running time of calculating the pmf is

$$\Theta\left(\sum_{i=0}^{k}\binom{k}{i}\binom{N-k}{k-i}t\right) = \Theta\left(\binom{N}{k}t\right). \tag{14}$$

In the following we analyze the running time of calculating the numerical result of $\Pr[\mathcal{R}(k) = \mathcal{F} \cup \mathcal{G}]$ based on its analytical expression derived by Eq. (12). Examining Eq. (12), we can see that there are two basic computations of Eq. (12), of which the first one is $\left(\prod_{i\in\mathcal{I}} \Pr\left[\gamma_i \le \frac{\ell}{n}\right] - \prod_{i\in\mathcal{I}} \Pr\left[\gamma_i \le \frac{\ell-1}{n}\right]\right) \prod_{j\in\bar{\mathcal{I}}} \Pr\left[\gamma_j \le \frac{\ell-1}{n}\right]$. Assume the running time of calculating this basic part is t_1. The second basic computation is $\prod_{i\in\mathcal{I}\setminus\mathcal{F}}\left(1 - \Pr\left[\gamma_i \le \frac{\ell}{n}\right]\right)\prod_{j\in\mathcal{F}\cup\mathcal{G}} \Pr\left[\gamma_j = \frac{\ell}{n}\right]\prod_{\kappa\in\bar{\mathcal{I}}\setminus\mathcal{G}} \Pr\left[\gamma_\kappa \le \frac{\ell-1}{n}\right]$, let us assume the running time of calculating this basic part is t_2. The running time of computing $\Pr[\mathcal{R}(k) = \mathcal{F} \cup \mathcal{G}]$ by Eq. (12) is

$$\begin{aligned} t &= \Theta(npt_1 + (2^k - 1)(2^{N-k} - 1)npt_2) = \Theta(npt_1 + 2^{k-1}2^{N-k-1}npt_2) \\ &= \Theta(npt_1 + 2^{N-2}npt_2) = \Theta(2^N npt_2/4). \end{aligned}$$

By letting $t = \Theta(2^N npt_2/4)$ in Eq. (14), we can obtain the result stated in this theorem. □

Summary. Let us summarize our findings for the analysis of the above group product recommendation system: (1) We can analytically derive the pmf of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$, with which it is easy to derive the analytical expression for $E[|\mathcal{R}^I(k) \cap \mathcal{R}(k)|]$ and $\mathrm{Var}[|\mathcal{R}^I(k) \cap \mathcal{R}(k)|]$; (2) the analytical expression is quite complicated and it is not easy to obtain insights of the underlying recommendation system by examining this analytical expression; (3) computing the numerical results based on these analytical results is computational expensive.

3.2. Derivation for the general case

For the general case, we can derive the analytical expressions of the pmf, expectation and variance of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$ with similar methods as above. It is reasonable to expect that for the general case, there can be different types of products and that users are not homogeneous (e.g., they may have different types of preferences). The tie breaking rules will be more complicated than the random rule. As one can imagine the analytical expressions for the general case will be more complicated. This negative result makes it more difficult to gain some insights by examining the analytical expressions. Furthermore, computing the numerical results for these analytical expressions would be more computationally expensive.

We do not follow the above analysis paradigm, but rather, let us focus on finding a practical approach to solve the general case, which should be *computational inexpensive* to obtain numerical results for the pmf, expectation and variance of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$. In the following section, we present this practical approach, and show that not only we can have a computational efficient approach to compute these probability measures, but more importantly, provides theoretical performance guarantees on our operations.

Table 3
Execution steps of Algorithm 1. The settings are: $N = 3$, $k = 2$, $n_i = 2$, $m = 5$, average score rule, random pick (tie breaking rule), high collective matching degree, and $\sigma(h) = 1$.

Step	Round 1	Round 2
1	$\ell_0 = 0, \ell_1 = 0, \ell_2 = 0$	
4	$Q_1 = 4.24, Q_2 = 4.20, Q_3 = 3.99$	$Q_1 = 4.94, Q_2 = 4.89, Q_3 = 4.67$
5	$[0.070, 0.078, 0.083, 0.086, 0.683]_{Q_1}$ $[0.073, 0.081, 0.087, 0.089, 0.671]_{Q_2}$ $[0.073, 0.080, 0.085, 0.039, 0.373]_{Q_3}$	$[0.006, 0.006, 0.007, 0.008, 0.973]_{Q_1}$ $[0.010, 0.011, 0.012, 0.013, 0.954]_{Q_2}$ $[0.029, 0.033, 0.036, 0.038, 0.863]_{Q_3}$
6	$[5, 3]_{Q_1}, [5, 5]_{Q_2}, [4, 1]_{Q_3}$	$[5, 5]_{Q_1}, [5, 5]_{Q_2}, [5, 5]_{Q_3}$
8	$\mathcal{R}^I(2) = \{Q_1, Q_2\}$	$\mathcal{R}^I(2) = \{Q_1, Q_2\}$
9	$\gamma_1 = 4, \gamma_2 = 5, \gamma_3 = 2.5$ No tie breaking	$\gamma_1 = 5, \gamma_2 = 5, \gamma_3 = 5$ Tie breaking: random pick
10	$\mathcal{R}(2) = \{Q_1, Q_2\}, \lvert\mathcal{R}(2) \cap \mathcal{R}^I(2)\rvert = 2$ $\ell_2 = 0 + 1 = 1$	$\mathcal{R}(2) = \{Q_1, Q_3\}, \lvert\mathcal{R}(2) \cap \mathcal{R}^I(2)\rvert = 1$ $\ell_1 = 0 + 1 = 1$
12	$\widehat{\Pr}[I(2) = 0] = \frac{\ell_0}{2} = 0, \widehat{\Pr}[I(2) = 1] = \frac{\ell_1}{2} = 0.5, \widehat{\Pr}[I(2) = 2] = \frac{\ell_2}{2} = 0.5$	
13	$\widehat{E}[I(k)] = 0 \times 0 + 1 \times 0.5 + 2 \times 0.5 = 1.5$	
14	$\widehat{\mathrm{Var}}[I(k)] = 0^2 \times 0 + 1^2 \times 0.5 + 2^2 \times 0.5 - (1.5)^2 = 0.25$	

4. General methodology for evaluating group recommendation systems

As stated in the last section, it is computationally expensive to evaluate the accuracy of a group product recommendation system. To address this challenge, we present a general methodology which is computationally efficient and at the same time, provides *error bounds*. More concretely, we present an efficient randomized algorithm with theoretical performance guarantees to evaluate the pmf, expectation and variance of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$.

4.1. Randomized algorithm

We seek to approximate the pmf, expectation and variance of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$ with a randomized algorithm. Our idea is that we simulate the group product recommendation process with our model for K rounds. In each round we record the number of top-k products getting recommended. After all K rounds finished, we approximate the value of $\Pr[|\mathcal{R}^I(k) \cap \mathcal{R}(k)| = i]$ by the fraction of rounds, where exact i top-k products get recommended. Then we apply this approximate value of pmf to compute the approximate value for the expectation and variance of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$. More specifically, in each round we first draw N products from distribution $\mathcal{Q}$. We input them into the group recommendation system model and we generate a set of ratings for each product via simulating the model. By performing a voting rule and a tie breaking rule on these ratings, the system output k products as the final recommendation. Comparing this output with the input, we obtain the number of top-k products getting recommended, or the value of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$ for one round. Based on this idea, we outline our randomized algorithm in Algorithm 1. This randomized algorithm is actually a Monte Carlo algorithm. For this type of randomized algorithm, the law of large number guarantees that as the number of simulation rounds goes to infinity, the approximate value converges to the true pmf expectation and variance of $|\mathcal{R}^I(k) \cap \mathcal{R}(k)|$. But this is not practical, instead we will state a practical bound on the desired number of simulation rounds to guarantee an accurate approximation. For the ease of presentation, we use the notation $I(k) = |\mathcal{R}^I(k) \cap \mathcal{R}(k)|$ to denote the number of top-k products get recommended. Let $\widehat{\Pr}[I(k) = i]$, $\widehat{E}[I(k)]$ and $\widehat{\mathrm{Var}}[I(k)]$ denote the approximate value of $\Pr[I(k) = i]$, $E[I(k)]$ and $\mathrm{Var}[I(k)]$ respectively.

To illustrate, consider a simple case of recommending two products from a set of three candidates, i.e., $N = 3$, $k = 2$. We show some detail execution steps of Algorithm 1 in Table 3, where the simulation rounds is set to $K = 2$. To make this example, we set the voting rule as the *average score rule*, and set the tie breaking rule as the *random pick rule*, which randomly pick some products to recommend from a tie. In Table 3, $[0.070, 0.078, 0.083, 0.086, 0.683]_{Q_1}$ represents a rating distribution for product P_1, where 0.070 represents the probability of receiving a rating 1, while 0.683 represents the probability of receiving a rating 5. And $[5, 3]_{Q_1}$ represents a set of two ratings for product P_1.

We can state two properties for this algorithm. The first one is its running time complexity and the other one is its theoretical performance guarantees. The following theorem states its running time complexity.

Theorem 3. *The computational complexity of our algorithm is $\Theta(KN \log N)$, where K is the number of simulation rounds and N is the number of products.*

Proof. We prove this theorem by examining the complexity of each step of our Algorithm 1. The complexity of steps 10–12 and 1 are the same, say $\Theta(k)$. The complexity of steps 3–8 are $\Theta(N)$, $\Theta(N \log N)$, $\Theta(\sum_{i=1}^N n_i)$, $\Theta(\sum_{i=1}^N n_i)$, $\Theta(N \log N)$, and $\Theta(1)$ respectively. Since each user only rates a small subset of products, namely $n_i \ll N$, thus we have $\Theta(\sum_{i=1}^N n_i) = \Theta(N)$. By adding the complexity of steps 1–12 up we could obtain the theorem. □

The remaining technical issue is how to set the parameter K. Specifically, what is the desired number of simulation rounds K to guarantee good approximations for the pmf, expectation and variance? Let us proceed to answer this question by deriving the theoretical performance guarantees for Algorithm 1.

Algorithm 1 : Evaluating the pmf, expectation, and variance of $I(k)$

Input: Homophyly degree h, product quality distribution $\mathcal{Q}$.
Output: $\widehat{\Pr}[I(k)=i]$, $\widehat{E}[I(k)]$ and $\widehat{\mathrm{Var}}[I(k)]$

```
for all i = 0, ..., k, ℓ_i ← 0
for j = 1 to K do
    for i = 1 to N do
        generate the quality for product P_i with probability distribution Q, or Q_i ~ Q. Rank products such that Q_1 ≥ Q_2 ≥ ··· ≥ Q_N
        Generate probability distribution P_i based on the product quality Q_i and homophyly degree h with Eq. (2).
        Draw n_i random samples from probability distribution P_i, as the rating set r_i for product P_i.
    end for
    pick k products with the highest value of Q_i as the true top-k products set, say R^I(k).
    simulate the decision making process, i.e., applying the voting rule V and the tie breaking rule T to produce the set R(k) based on the score sets {r_1, ..., r_N}
    if the cardinality of the intersection of R^I(k) and R(k) is equal to i, then ℓ_i ← ℓ_i + 1.
end for
for all i = 0, ..., k, Pr^[I(k) = i] ← ℓ_i/K
E^[I(k)] ← Σ_{i=0}^{k} i Pr^[I(k) = i]
Var^[I(k)] ← Σ_{i=0}^{k} i^2 Pr^[I(k) = i] − (E^[I(k)])^2
```

4.2. Deriving theoretical performance guarantees

We derive the theoretical performance guarantees for Algorithm 1. More concretely, we quantify the tradeoff between the simulation rounds K and the approximation accuracy of the Algorithm 1. Through this we identify a practical bound on the desired number of simulation rounds to guarantee a high approximation accuracy.

A conventional measure to characterize the accuracy of a randomized algorithm is the ϵ-*approximation* [14]. We formally state its definition as follows.

Definition 2 (ϵ-*Approximation*)**.** Suppose $\widehat{X}$ is an approximation of X. We say that $\widehat{X}$ is an ϵ-*approximation*, if $|\widehat{X}-X|\le\epsilon X$, where $\epsilon\ge 0, X\ge 0$.

One can vary the value of ϵ in the above definition to attain different level of accuracy. Specifically, the smaller the value of ϵ, the higher the accuracy. We say Algorithm 1 produce an ϵ-*approximation* for the pmf of $I(k)$, if $\widehat{\Pr}[I(k)=i]$ gives an ϵ-*approximation* of $\Pr[I(k)=i]$ for all $i=0,1,\ldots,k$. The following theorem states that Algorithm 1 produces this approximation with sufficiently large number of simulation rounds.

Theorem 4. *Suppose the following condition holds*

$$K \ge \max_{\substack{i=0,1,\ldots,k,\\ \Pr[I(k)=i]\neq 0}} \frac{3\ln(2(k+1)/\delta)}{\Pr[I(k)=i]\epsilon^2}. \tag{15}$$

Algorithm 1 *guarantees an ϵ-approximation for the pmf of $I(k)$ with probability at least* $1-\delta$, *or* $\Pr[|\widehat{\Pr}[I(k)=i]-\Pr[I(k)=i]|\le\epsilon\Pr[I(k)=i],\ \forall i]\ge 1-\delta$.

Proof. Please refer to the Appendix in this paper for derivation. □

Roughly speaking, an ϵ-approximation for the pmf of $I(k)$ implies that $\widehat{E}[I(k)]$ and $\widehat{\mathrm{Var}}[I(k)]$ are ϵ-approximations for the expectation and variance of $I(k)$ respectively. We state this in the following theorem.

Theorem 5. *Suppose* $|\widehat{\Pr}[I(k)=i]-\Pr[I(k)=i]|\le\epsilon\Pr[I(k)=i]$, *holds for all* $i=0,1,\ldots,k$, *then* $|\widehat{E}[I(k)]-E[I(k)]|\le\epsilon E[I(k)]$ *and* $|\widehat{\mathrm{Var}}[I(k)]-\mathrm{Var}[I(k)]|\le\epsilon(1+\epsilon)\mathrm{Var}[I(k)]$ *hold.*

Proof. Please refer to the Appendix in this paper for derivation. □

Remark. The value of ϵ is small, e.g., $\epsilon=0.1$ or 0.01. We then have $\epsilon(1+\epsilon)\approx\epsilon$. Thus, roughly speaking, $\widehat{\mathrm{Var}}[I(k)]$ is an ϵ-approximation of $\mathrm{Var}[I(k)]$. Combining Theorems 4 and 5, we see that Algorithm 1 guarantees ϵ-approximations for the pmf, expectation and variance of $I(k)$ with high probability, provided that the condition on the simulation rounds, i.e., Inequality (15), holds.

We show some numerical examples on the desired number of simulation rounds in Table 4, where we examine the impact of approximation error ratio ϵ, success probability $1-\delta$, and pmf of $I(k)$. From Table 4 one can see that as we decrease the value of ϵ from 0.1 to 0.01, we increase the desired simulation rounds from 13 816 to 1.38×10^8. In other words, we need to simulate remarkably more rounds so to increase the approximation accuracy. While we only need slightly more rounds to

Table 4
Impact of approximation error ratio ϵ, success probability $1-\delta$, and pmf of $I(k)$ on the desired simulation rounds ($k=4, N=50$).

$\{\Pr[I(k)=i], i=0,\ldots,4\}$	ϵ	$K(\delta=0.1)$	δ	$K(\epsilon=0.01)$
{0.1, 0.15, 0.3, 0.25, 0.2}	0.1	13 816	0.1	1 381 552
{0.1, 0.15, 0.3, 0.25, 0.2}	0.01	1 381 552	0.01	2 072 327
{0.1, 0.15, 0.3, 0.25, 0.2}	0.001	1.38×10^8	0.001	2 763 103
$\{2^{-10}, 0.15, 0.4-2^{-10}, 0.25, 0.2\}$		2.12×10^8 ($\epsilon=0.01, \delta=0.01$)		
$\{2^{-30}, 0.15, 0.4-2^{-30}, 0.25, 0.2\}$		2.23×10^{14} ($\epsilon=0.01, \delta=0.01$)		
$\{2^{-50}, 0.15, 0.4-2^{-50}, 0.25, 0.2\}$		2.33×10^{20} ($\epsilon=0.01, \delta=0.01$)		

Table 5
Numerical examples on the simulation rounds and the error bound for the pmf of $I(k)$ ($k=4, N=50, \delta=0.01$).

$\{\Pr[I(k)=i]\}$	ϵ	K	$\max\left\{\epsilon\sqrt{\Pr[I(k)=i]}, \epsilon^2\right\}$
{0.1, 0.15, 0.3, 0.25, 0.2}	0.1	2073	{0.032, 0.039, 0.055, 0.050, 0.045}
{0.1, 0.15, 0.3, 0.25, 0.2}	0.01	207 233	{0.003, 0.004, 0.005, 0.005, 0.004}
{0.1, 0.15, 0.3, 0.25, 0.2}	0.001	$2.1\cdot10^7$	$\{3,4,5,5,4\}\times10^{-4}$

increase the confidence probability $1-\delta$. In fact, as we increase it from 0.9 to 0.999, we only increase the simulation rounds from 1 381 552 to 2 763 103. When the value of $\min\{\Pr[I(k)=i]\}$, varies from 0.1 to 2^{-50}, the desired number of simulation rounds increased from 1 381 552 to 2.33×10^{20}. Namely, the desired number of simulation rounds is sensitive to the pmf of $I(k)$.

Discussion. Following the conventional randomized algorithm analysis paradigm, i.e., guarantee ϵ-approximation, we may need a very large number of simulation rounds, especially when the value of $\min\{\Pr[I(k)=i]\}$ is small. More concretely, examining the Inequality (15), one can see that the bound of K makes sense when $\Pr[I(k)=i]$ is not small for all $i=0,1,\ldots,k$. Consider the case where $\Pr[I(k)=i]\le 2^N$ for some $i\in\{0,1,\ldots,k\}$, then we have $K\ge 2^N$. For such cases, K is too large. In the following we eliminate this negative result by showing a practical bound on K, which shows a better tradeoff between simulation rounds and the approximation accuracy.

Theorem 6. *When $K\ge 3\ln(2(k+1)/\delta)/\epsilon^2$, Algorithm 1 guarantees the following $\Pr[|\widehat{\Pr}[I(k)=i]-\Pr[I(k)=i]|\le \max\left\{\epsilon\sqrt{\Pr[I(k)=i]}, \epsilon^2\right\}, \forall i]\ge 1-\delta$.*

Proof. The proof is similar to that of Theorem 4. □

This theorem relaxes the approximation accuracy from ϵ-approximation to $\max\left\{\epsilon\sqrt{\Pr[I(k)=i]}, \epsilon^2\right\}$, with the gain of reducing the desired number of simulation rounds by a factor of $\frac{1}{\min\{\Pr[I(k)=i]>0\}}$, compared with Inequality (15).

We show some numerical examples on the accuracy and the desired simulation rounds in Table 5. One can observe that with 2.1×10^7 simulation rounds, the error bound is around 0.0005, which is quite small from a practical perspective. This shows the practicability of the new bound on the number of simulation rounds.

The remaining thing is to derive the approximation error bound for the expectation and variance of $I(k)$, with the above approximation on the pmf of $I(k)$. We state them in the following theorem.

Theorem 7. *Suppose $|\widehat{\Pr}[I(k)=i]-\Pr[I(k)=i]|\le \max\{\epsilon\sqrt{\Pr[I(k)=i]}, \epsilon^2\}$, holds for all $i=0,1,\ldots,k$, then $|\widehat{E}[I(k)]-E[I(k)]|\le \epsilon k\sqrt{E[I(k)]}+\epsilon^2k^2$ and $|\widehat{\mathrm{Var}}[I(k)]-\mathrm{Var}[I(k)]|\le \epsilon(k+1)(\sqrt{k\mathrm{Var}[I(k)]+\epsilon^2k^4}+\epsilon\mathrm{Var}[I(k)]+\epsilon^3k^3)$ hold.*

Proof. Please refer to the Appendix in this paper for derivation. □

We show some numerical examples for the error bounds derived in the above theorem in Table 6, where we use the notation $\varepsilon_{\exp}=\epsilon k\sqrt{E[I(k)]}+\epsilon^2k^2$ and the notation $\varepsilon_{\mathrm{var}}=\epsilon(k+1)\left(\sqrt{k\mathrm{Var}[I(k)]+\epsilon^2k^4}+\epsilon\mathrm{Var}[I(k)]+\epsilon^3k^3\right)$ to denote the error bound on the expectation and variance of $I(k)$ respectively. One can observe that as we vary the value of ϵ from 0.1 to 0.001, the desired number of simulation rounds increased from 2073 to 2.1×10^7. It is interesting to see that the error bound on the expectation $\varepsilon_{\exp}$ dropped from 0.767 to 0.006, or the error ratio $\frac{\varepsilon_{\exp}}{E[I(k)]}$ dropped from 0.333 to 0.003. Similarly, the error bound on the variance $\varepsilon_{\mathrm{var}}$ dropped from 1.510 to 0.012, or the error ratio $\frac{\varepsilon_{\mathrm{var}}}{\mathrm{Var}[I(k)]}$ dropped from 1.042 to 0.008. This show the practicability of the new bounds on the expectation and variance of $I(k)$.

Discussion. The bounds derived in the above theorem are practical enough to deal with larger scale group recommendation system. More concretely, observe that $\epsilon k\sqrt{E[I(k)]}+\epsilon^2k^2\approx \epsilon k\sqrt{E[I(k)]}$, and $\epsilon(k+1)(\sqrt{k\mathrm{Var}[I(k)]+\epsilon^2k^4}+\epsilon\mathrm{Var}[I(k)]+\epsilon^3k^3)\approx \epsilon k^{1.5}\sqrt{\mathrm{Var}[I(k)]}$. This shows the scalability of these bounds, as we increase the recommending product set size or the candidate product set size.

Table 6
Numerical examples on the simulation rounds and the error bound for the expectation and variance of $I(k)$ ($k = 4, N = 50, \delta = 0.01$, $\{\Pr[I(k) = i]\} = \{0.1, 0.15, 0.3, 0.25, 0.2\}$).

ϵ	K	ε_{exp}	$\frac{\varepsilon_{\text{exp}}}{E[I(k)]}$	ε_{var}	$\frac{\varepsilon_{\text{var}}}{\text{Var}[I(k)]}$
0.1	2073	0.767	0.333	1.510	1.042
0.01	207 233	0.062	0.027	0.124	0.082
0.001	2.1×10^7	0.006	0.003	0.012	0.008

Table 7
Expectation and variance of I_1, I_5, and I_{30} when vary rating workload (medium collective matching degree).

	$n = 3$	$n = 4$	$n = 6$	$n = 8$	$n = 10$
$E[I_1]$	0.9832	0.9931	0.9986	0.9996	0.9999
$E[I_5]$	4.6854	4.8284	4.9352	4.9723	4.9869
$E[I_{30}]$	19.8270	20.960	22.400	23.328	23.980
$\text{Var}[I_1]$	0.0165	0.0067	0.0014	0.0004	0.0001
$\text{Var}[I_5]$	0.2942	0.1701	0.0654	0.0280	0.0132
$\text{Var}[I_{30}]$	4.1210	3.7710	3.2934	2.9580	2.7121

5. Experiments on synthetic data

In this section we use synthetic data to examine the accuracy of group product recommendation systems under various settings. We consider a group product recommendation system which recommends $k = 30$ products from $N = 200$ candidates. In consistent with realistic group recommendation systems, we set $m = 5$, or the rating metric is $\{1, \ldots, 5\}$. We apply Algorithm 1 to evaluate this system setting the simulation rounds $K = 10^8$. Let us start our evaluation from a simple case, then we extend it step by step and evaluate the impact of various factors on the overall recommendation accuracy.

5.1. Probability distribution, expectation and variance

We consider a homogeneous group recommendation system, where the user group $\mathcal{U}$ only contains one type of users and each product is rated by the *same number* of users, or $n_i = n, \ \forall i$. We set the collective matching degree to be medium. Furthermore, we set the function $\sigma(h)$, within Eq. (2) to be $\sigma(h) = 1$. The voting rule $\mathcal{V}$ is the *average score rule* and the tie breaking rule is the *least variance rule* which breaks a tie by selecting one(s) with the least variance. For further ties, products are randomly selected. For the ease of presentation, we state the following two definitions.

Definition 3. The rating workload of a group recommendation system is the number of all ratings, or $W = \sum_{i=1}^{N} n_i$.

Definition 4. Let I_i be a random variable indicating $I_i = |\mathcal{R}^I(i) \cap \mathcal{R}(k)|$. In other words, if $i = k$, I_k reflects the event that the *top k* products are recommended by the group recommendation system.

The numerical results for the pmf of $|\mathcal{R}^I(30) \cap \mathcal{R}(30)|$ are shown in Fig. 1. One can observe that the probability of less than fifteen or more than twenty seven top 30 products get recommended is around zero. This indicates that the recommendation accuracy has small variation. When we increase the number of ratings per product, or n, the probability mass function curve shifts towards the right. In other words, the higher the rating workload, the higher the recommendation accuracy.

We show the numerical results for the expectation and variance of I_1, I_5, I_{10} and I_{30} in Table 7. One can observe that when each product is rated by three users, approximately 19.8 products from the top 30 products will get recommended. It is interesting to note that the chance of recommending the most preferred product is invariant of the rating workload, since $E[I_1] = 0.98$ when $n = 3$ and it improves to 0.9999 when $n = 10$. This statement also holds for the top five products. As we increase n, we decrease the variance. In other words, the higher the rating workload, the lower the variation in the recommendation. This indicates that the group recommendation system is more accurate and robust.

Lessons learned: If users and products are matched with a medium degree, we have a pretty accurate group recommendation system. The higher the rating work load the higher the recommendation accuracy. There are number of interesting questions to explore further, i.e., are these results dependent on the distribution of product quality, or the collective matching degree? Are these results sensitive to any voting rule? Let us continue to explore.

5.2. Impact of product quality distribution

We explore the impact of *product quality* distribution (or *collective matching degree* between users and products) on the recommendation accuracy. More concretely, we consider three representative types of *collective matching degree*: high,

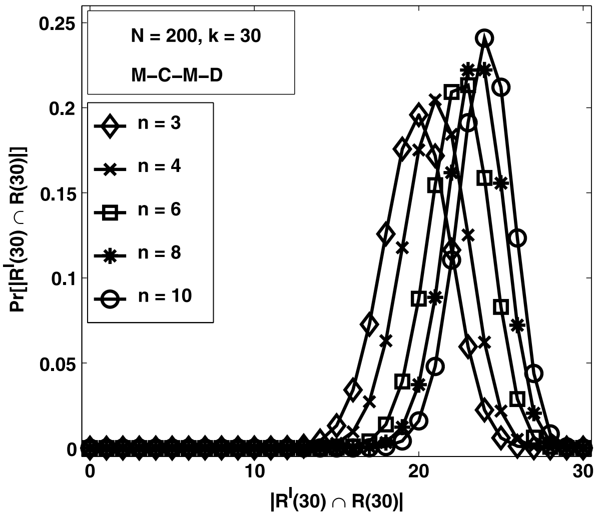


Fig. 1. The pmf of $|\mathcal{R}^I(30) \cap \mathcal{R}(30)|$ when vary the rating workload.

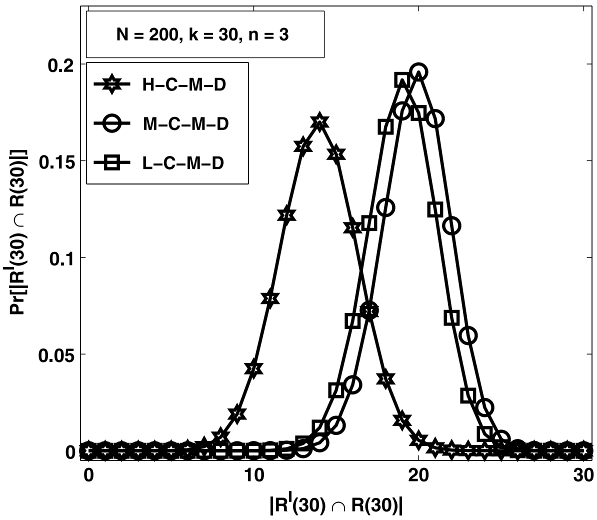


Fig. 2. The pmf of $|\mathcal{R}^I(30) \cap \mathcal{R}(30)|$ when vary the product quality distribution.

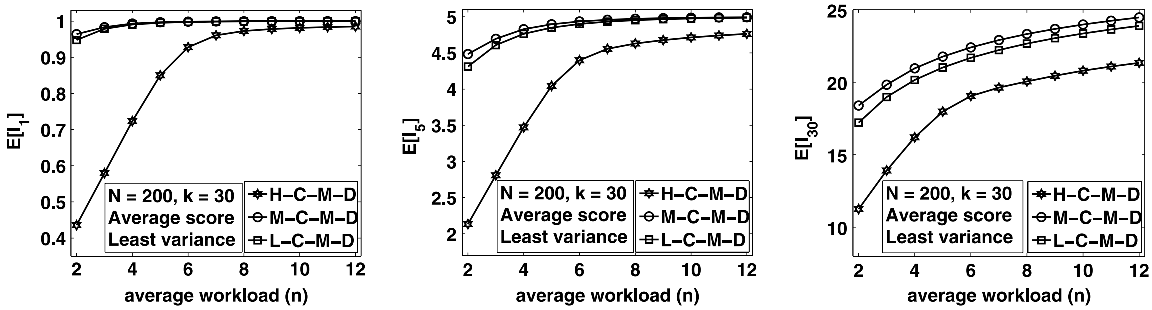
medium, and low specified by Eqs. (5)–(7) respectively. We evaluate their impact on the recommendation accuracy with the system specified in Section 5.1, setting $n = 3$. We use the following notations to present our results: (a) ***H-C-M-D:*** high *collective matching degree* specified in Eq. (5), (b) ***M-C-M-D:*** medium *collective matching degree* specified in Eq. (6), and (c) ***L-C-M-D:*** low *collective matching degree* specified in Eq. (7).

We show the numerical results for the pmf of $|\mathcal{R}^I(30) \cap \mathcal{R}(30)|$ in Fig. 2. From Fig. 2 we could see that as the *collective matching degree* varies in the order of high, low and medium, the corresponding mass probability distribution curves move towards right. In other words, the recommendation accuracy corresponding to the low *collective matching degree* is the highest followed by medium, and high *collective matching degree*. The numerical results for the expectation and variance of I_1, I_5, I_{10} and I_{30} are shown in Table 8. We can observe that when the *collective matching degree* is medium, or low, around 20 top 30 products will get recommended. It is interesting to note that the chance of recommending the most preferred product is invariant to these two *collective matching degree* types, since the corresponding values of $E[I_1]$ are all around 0.98. This statement also holds for recommending top five products. But when the *collective matching degree* is high, the recommendation accuracy is *remarkably lower* than that the other two types. In fact, only around 13.2 top 30 products get recommended. Even the most preferred product will get missed with a high probability, say around 0.46. This statement also holds for top five products. The variance corresponding to the high *collective matching degree* is the highest among those three, which reflects that the recommendation is the lowest and with a large variation.

Lessons learned: When users rate honestly, the above simple group recommendation system is quite accurate except when the *collective matching degree* between users and products is high. In that case, one may explore other means to improve the

Table 8
Expectation and variance of I_1, I_5, and I_{30} when $n = 3$ and users and products are matched with high, medium, and low *collective matching degree*.

	H-C-M-D	*M-C-M-D*	*L-C-M-D*
$E[I_1]$	0.5401	0.9832	0.9788
$E[I_5]$	2.6250	4.6950	4.6082
$E[I_{30}]$	13.2258	19.8270	18.9798
$\mathrm{Var}[I_1]$	0.2437	0.0165	0.0208
$\mathrm{Var}[I_5]$	1.2236	0.2942	0.3698
$\mathrm{Var}[I_{30}]$	5.4420	4.1210	4.2888



(a) #of top 1 product get in. (b) #of top 5 products get in. (c) #of top 30 products get in.

Fig. 3. Impact of rating workload on the recommendation accuracy.

accuracy. Again, there are number of interesting questions to explore, i.e., what is the desired rating workload to guarantee a high recommendation accuracy?

5.3. *Impact of rating workload*

We explore the impact of rating workload on the recommendation accuracy. Here we consider a homogeneous group recommendation system specified in Section 5.1. We consider three representation types of product quality distribution as specified in Eqs. (5)–(7). For each type, we vary the average rating workload, i.e. W/N, from 2 to 12.

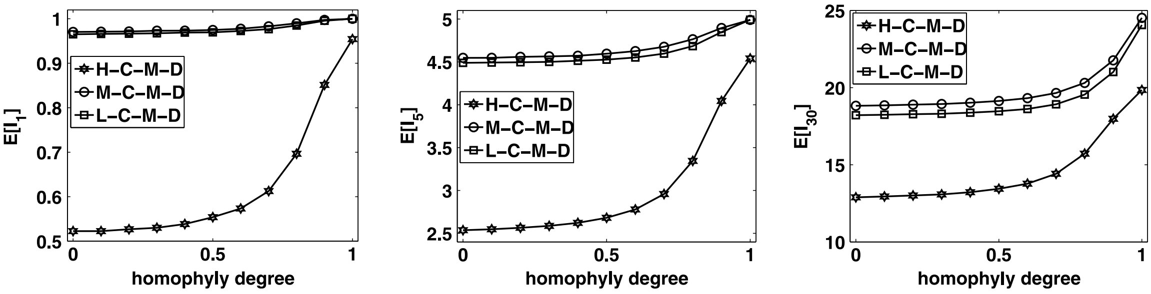
We select one probability measure, expectation of recommendation, to study. The numerical results of $E[I_1]$, $E[I_5]$ and $E[I_{30}]$ are shown in Fig. 3. One can observe that when we increase the average rating workload, the expectation increased. This reflects the improvement on the recommendation accuracy. As the *collective matching degree* varies in the order of high, low, and medium, the expectation curves move towards up. In other words, the recommendation accuracy corresponding to the medium *collective matching degree* is the highest followed by low, and high *collective matching degree*. It is interesting to observe that the chance of recommending the most preferred product is invariant of the rating workload, except the *collective matching degree* is high. This statement also holds for the top five products. When the *collective matching degree* is high, the accuracy of the group recommendation system is remarkably lower than the other two types. Especially when the rating workload is low, say $n = 3$, with less than fifteen top 30 products get recommended. The most preferred product only has a probability of less than 0.6 to be recommended when $n = 3$. Same can be said for the top five products. In closing, for high *collective matching degree*, we may have to increase the rating workload to at least $n \geq 7$ such that we have a strong guarantee that the most preferred product will be recommended.

Lessons learned: If users rate honestly, using the above simple group recommendation system achieves relatively high accuracy except when the *collective matching degree* is high. In that case, we have to increase the average rating workload to $n \geq 7$ to improve the system. Again, there are number of interesting questions to explore, i.e., is increasing the workload the only way to improve the group recommendation system? Can we improve it by using different voting rule or tie breaking rule?

5.4. *Impact of homophyly degree h*

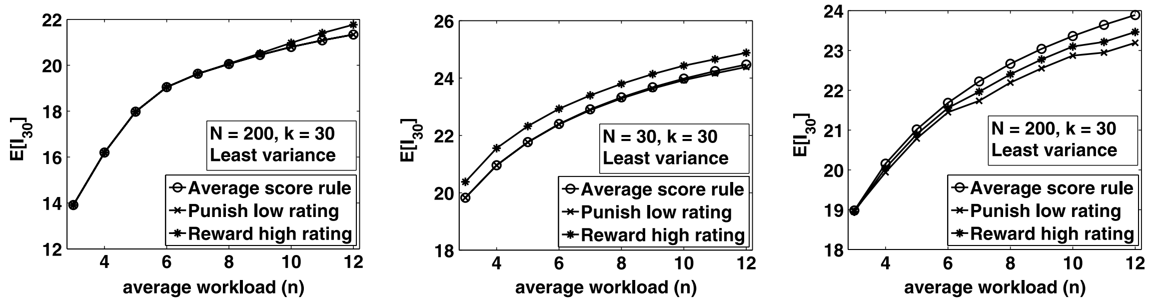
We examine the impact of homophyly degree on the recommendation accuracy. To achieve we specify the function $\sigma(h)$ within Eq. (2) as $\sigma(h) = 0.5 + 5(1 - h)$. Then we can vary the value of h to reflect different level of homophyly degree. We consider the homogeneous group recommendation system specified in Section 5.1 setting the average rating workload $n = 5$.

We vary the homophyly degree from 0 to 1. We choose expectation $E[I_{30}]$ as our performance measure. The numerical results of $E[I_{30}]$ are shown in Fig. 4. We can observe that when we increase the homophyly degree, the expectation $E[I_{30}]$



(a) #of top 1 product get in. (b) #of top 5 products get in. (c) #of top 30 products get in.

Fig. 4. Impact of homophyly degree on recommendation accuracy.



(a) High *matching degree*. (b) Medium *matching degree*. (c) Low *matching degree*.

Fig. 5. Impact of voting rule on the recommendation accuracy.

increased. In other words, the higher the homophyly degree, the higher the recommendation accuracy. When the homophyly degree is low, say less than 0.5, increasing the homophyly degree only increases the expectation $E[I_{30}]$ slightly. When the homophyly degree is above 0.5, a small increasing in the homophyly degree h increase the expectation $E[I_{30}]$ remarkably. It is interesting to note that the recommendation of the most preferred product is invariant of the homophyly degree, unless the matching degree between users and products is high. Under that matching degree, the recommendation accuracy is remarkably lower than the other two types of matching degree. Specifically, less than fifteen top 30 will be recommended, and the chance of recommending the most preferred product is less than 0.6, when the homophyly degree is less than 0.5.

Lessons learned: Increasing homophyly degree can increase the recommendation accuracy. Especially when the matching degree between users and products is high, where the improvement on the recommendation accuracy is remarkable.

5.5. Impact of voting rules

We explore the impact of voting rules on the recommendation accuracy of group recommendation systems. Specifically, we evaluate the following typical voting rules:

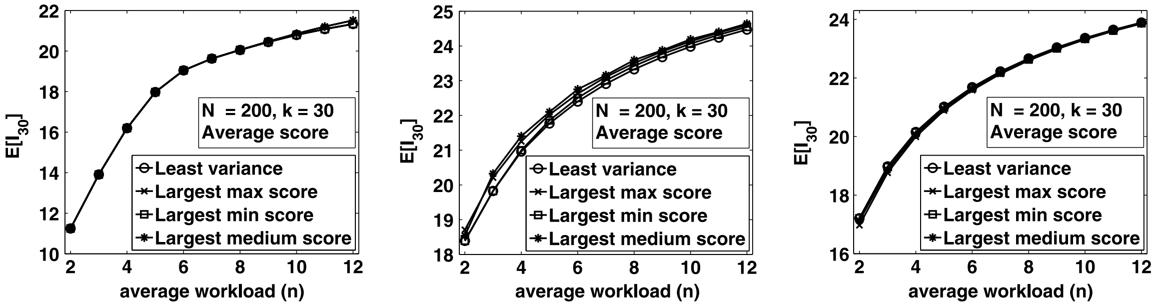
Average score rule: specified by $\gamma_i = \sum_j r_{i,j}/n_i$.

Reward high rating rule: reward high ratings. Specifically, a high rating, say 5, brings an extra reward of increasing its aggregate rating by η, or $\gamma_i = \sum_j r_{i,j}/n_i + \eta|\{r \mid r = 5, r \in \mathbf{r}_i\}|$.

Punish low rating rule: punish low ratings. Specifically, a low score, or 1, brings an extra punishment of decreasing its aggregate rating by η, or $\gamma_i = \sum_j r_{i,j}/n_i - \eta|\{r \mid r = 5, r \in \mathbf{r}_i\}|$.
We set $\eta = 0.3$ throughout this paper.

We choose the expectation $E[I_{30}]$ as our performance measure. We evaluate the recommendation accuracy under these three voting rules on the group recommendation system specified in Section 5.1. The numerical results of $E[I_{30}]$ are shown in Fig. 5. Fig. 5(a) shows that when users and products are highly matched, the expectation curves overlapped together. In other words, these three voting rules have nearly the same accuracy. While Fig. 5(b) shows that when the collective matching degree is medium, the *average score rule* and *punish low rating rule* have the same degree of accuracy, and the *reward high rating rule* has slightly higher accuracy than them. Finally Fig. 5(c), shows that when the collective matching degree is low, all three voting rules have nearly the same accuracy.

Lessons learned: These three voting rules have comparable accuracy, no rule can outperform others remarkably. The *punish low rating rule* has slightly lower recommendation accuracy than the other two rules. Thus we should avoid choosing the *punish low rating rule* to aggregate users' ratings.



(a) High *matching degree*. (b) Medium *matching degree*. (c) Low *matching degree*.

Fig. 6. Impact of tie breaking rule on recommendation accuracy.

5.6. *Impact of tie breaking rules*

We explore the impact of tie breaking rules on the recommendation accuracy. In particular, we examine the following four typical tie breaking rules:

Least variance ($\mathcal{T}_{\text{var}}$): select one with least variance.

Largest max score ($\mathcal{T}_{\text{maxs}}$): select one with the largest max rating.

Largest min score ($\mathcal{T}_{\text{mins}}$): select one with the largest min rating.

Largest medium score ($\mathcal{T}_{\text{meds}}$): select one with the largest medium rating.

For further ties, we perform random selection. We set the voting rule as the *average score rule* and we choose expectation $E[I_{30}]$ as our performance measure. We consider the group recommendation system specified in Section 5.1. The numerical results of $E[I_{30}]$ are shown in Fig. 6. We could see that the expectation curves corresponding to these four tie breaking rules overlapped together. In other words, these four rules have nearly the same accuracy. Therefore, the recommendation accuracy is invariant of tie breaking rules.

Lessons learned: These four tie breaking rules have nearly the same accuracy. This implies that, the improvement on the recommendation accuracy by tie breaking rules is limited.

5.7. *Impact of misbehavior*

We explore the impact of misbehavior on the recommendation accuracy, and we examine the robustness of voting rules against misbehavior as well. Formally, we consider the following typical cases of misbehavior:

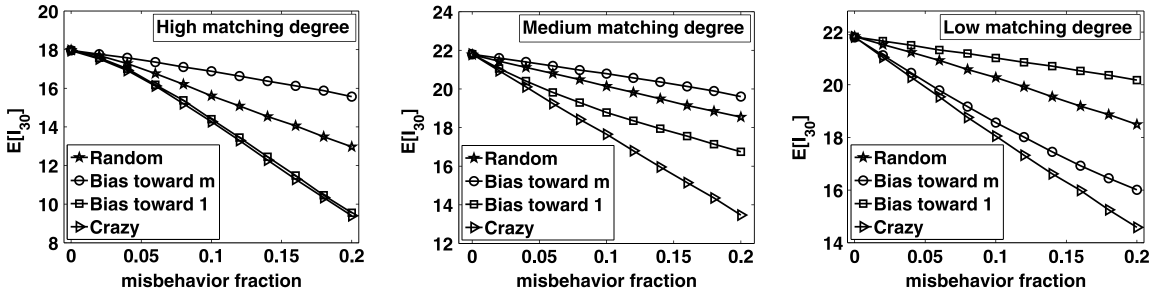
Random misbehavior: A random misbehavior indicates that a user assigns a random rating to a product. This is one typical misbehavior, because sometimes a user may not want to disclose his true preference.

Bias misbehavior: A biased misbehavior indicates that a user is intentionally biased towards one particular rating. We consider the following two typical cases of biased misbehavior: (1) *Bias towards* 1 which reflects a user assign the lowest rating to products. (2) *Bias towards m* which reflects a user assign the highest rating to products.

Crazy misbehavior. A crazy misbehavior indicates that a user shows the reverse preference to a product aiming to disrupt the recommendation. Specifically, a user gives the highest rating 5 to a product if he does not like it, or his true preference rating is lower or equal to three, otherwise assigns the lowest rating 1.

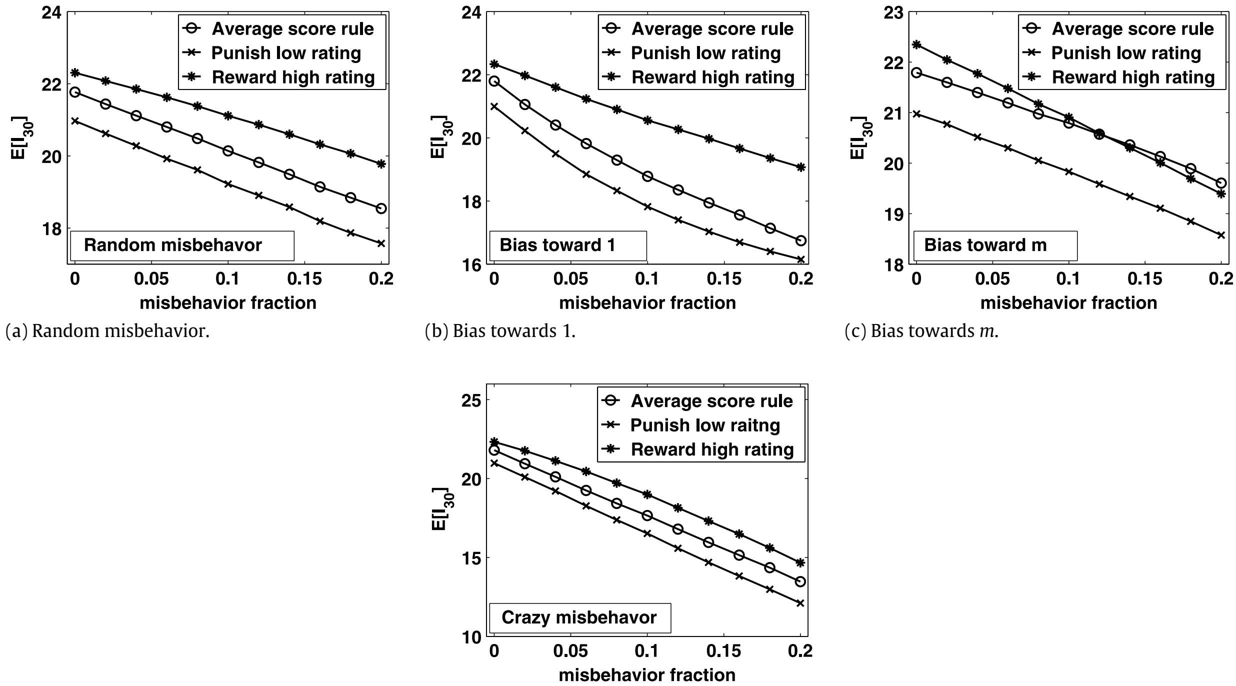
Let us explore the impact of misbehavior on the recommendation accuracy first. We vary the fraction of misbehaving users from 0 to 0.2 and we use expectation $E[I_{30}]$ as our performance measure. We consider the group recommendation system specified in Section 5.1 setting $n = 5$. The numerical results of $E[I_{30}]$ are shown in Fig. 7. One can observe that increasing the fraction of misbehaving users, decreases the expectation $E[I_{30}]$. In other words, the higher the fraction of misbehaving users, the lower the recommendation accuracy. It is interesting to note that the recommendation accuracy decreases in a nearly linear rate, where the crazy misbehavior has the largest decreasing rate. Namely, crazy misbehavior is the most harmful misbehavior. Even a small fraction crazy misbehaving users can decrease the recommendation accuracy remarkably. In fact when the fraction of crazy misbehaving users is around 0.1, only around fourteen top 30 products get recommended (high matching degree), and around eighteen get recommended (medium and low matching degree). This show that the system is sensitive to crazy misbehavior. The system is insensitive to random misbehavior, since it decreases the recommendation accuracy slightly.

We explore the robustness of voting rules against misbehavior. In particular, we consider three typical voting rules: the *average score rule*, the *reward high rating rule* and the *punish low rating rule* specified in Section 5.5. We perform our evaluation on the above experimental settings. We choose one type of matching degree, say medium matching degree, to study here. The numerical results of $E[I_{30}]$ are shown in Fig. 8. One can observe that when we increase the fraction of misbehaving users, the expectation decreased. The *punish low rating rule* is the least robust rule, since the expectation curve corresponding it lies



(a) High matching degree.
(b) Medium matching degree.
(c) Low matching degree.

Fig. 7. Impact of misbehavior on recommendation accuracy.



(a) Random misbehavior.
(b) Bias towards 1.
(c) Bias towards *m*.
(d) Crazy misbehavior.

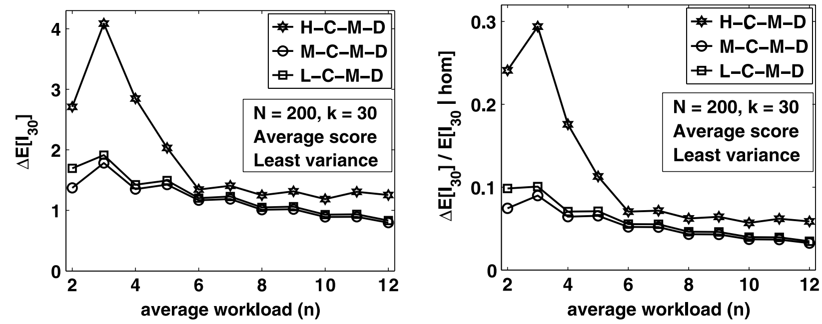
Fig. 8. Robustness of voting rules against misbehavior (medium collective matching degree).

in the bottom. Among these three voting rules, the *reward high rating rule* is the most robust rule, unless misbehaving users bias towards *m*. When misbehaving users bias towards *m*, the *average score rule* and *reward high rating rule* have comparable recommendation accuracy, since the expectation curves corresponding to them overlapped together.

Lessons learned: Crazy misbehavior can significantly decrease the recommendation accuracy. A small fraction of this kind of misbehaving users can decrease the accuracy of a group recommendation system dramatically. The group recommendation system suffers severely from this kind of misbehavior, say 10% of misbehaving users will disrupt the accuracy of a group recommendation system. Among these three voting rules, the *punish low rating rule* is the least robust rule, the *reward high rating rule* is the most robust rule, except unless misbehaving users bias towards *m*, where it has comparable robustness with the *average score rule*.

5.8. Improving the recommendation accuracy

We propose a rating scheme to improve the recommendation accuracy. In real world group product recommendation systems applications, ratings are *sparse*, say each user only rates a *small* subset of products. Here we fix the rating workload budget to be $W = N * n$, say on average, each product is rated by n users. Under this setting, we examine the accuracy of group product recommendation systems under different rating strategies. Homogeneous rating scheme is a widely adopted in many group recommendation systems applications, e.g., peer review systems, where each product is rated by the same number of users, or n. One obvious advantage of this scheme is in fairness. Here we propose a *heterogeneous rating scheme*



(a) Improvement. (b) Improvement ratio.

Fig. 9. Improvement of the heterogeneous rating scheme over the homogeneous rating scheme.

which can increase the efficiency of the homogeneous rating scheme. In other words, with the same rating workload W, it has a much higher recommendation accuracy. Our heterogeneous rating scheme works in two rounds:

Round 1: Eliminate half of the products using only half of the rating workload. Specifically, each product will receive $\lfloor n/2 \rfloor$ ratings in the first round. With these ratings, apply a voting rule and a tie breaking rule to eliminate $N/2$ products. Survived products will enter round 2.

Round 2: Select k products to recommend from the survived $N/2$ products. Each survived product will receive $2\lceil n/2 \rceil$ more ratings. After the rating process finished, combine the ratings in round 1 and round 2. Then, apply a voting rule and a tie breaking rule to select the k products to recommend.

Remark. The rating workload of the heterogeneous rating scheme is $\lfloor n/2 \rfloor \cdot N + 2\lceil n/2 \rceil \cdot N/2 = N \cdot (\lfloor n/2 \rfloor + \lceil n/2 \rceil) = Nn$.

Definition 5. Let $E[I_i \mid \text{hom}]$ and $E[I_i \mid \text{hetero}]$ represent the expectation of I_i when the homogeneous or the heterogeneous rating scheme is applied respectively, where I_i is defined in Definition 4.

Definition 6. We define the improvement of the heterogeneous rating scheme over the homogeneous rating scheme in recommendation accuracy as $\Delta E[I_i] = E[I_i \mid \text{hetero}] - E[I_i \mid \text{hom}]$, and the improvement ratio as $\Delta E[I_i]/E[I_i \mid \text{hom}]$.

We evaluate these two rating schemes on the group recommendation system specified in Section 5.1. We show the numerical results for the improvement $\Delta E[I_{30}]$ and the improvement ratio $\Delta E[I_{30}]/E[I_{30} \mid \text{hom}]$ in Fig. 9. From Fig. 9, we see an improvement of the heterogeneous rating scheme over the homogeneous rating scheme. Specifically, when the rating workload is small (rating is very sparse), say $n = 3$, with heterogeneous rating scheme at least one more top 30 products get recommended. The improvement is much higher when the collective matching degree is high, where four more top 30 products get recommended. When the average rating workload increased to six or more, the improvement becomes stabilized. The improvement is the highest when the rating workload is three, where at most four more top 30 products get recommended, and the improvement ratio for this case is around 30%.

Lessons learned: Our heterogeneous rating scheme uses equal or less resource (e.g., rating workload) than the homogeneous rating scheme and at the same time, achieve higher accuracy.

6. Experiments on real-world data

We present the experimental results on a real-world rating dataset from TripAdvisor. Specifically, we validate our model via showing that the recommendation accuracy evaluated from synthetic ratings (generated by our model) is nearly the same with that evaluated from TripAdvisor rating dataset. And we examine the desired number of ratings for TripAdvisor to guarantee an accurate recommendation as well.

Dataset. TripAdvisor is one of the largest travelling website which recommends hotels, restaurants, etc., to travellers (or website users). Users express their preferences to hotels, restaurants, etc., in the form of ratings. TripAdvisor adopts a 5-level cardinal rating metric, i.e., $\{1, \ldots, 5\}$. Higher rating implies higher preference. We use the version of TripAdvisor dataset released by authors in [15]. This dataset consists of ratings and reviews for 1850 hotels, where the maximum number ratings for a hotel is 2686 and the minimum number of ratings for a hotel is 13. Ratings are with time stamps.

We select a subset of the TripAdvisor rating dataset as our testing dataset. The selection criteria is that a hotel gets in if it has a sufficient number of ratings to guarantee an accurate estimation on its quality. Specifically, we treat the whole user population in TripAdvisor as a group and we aim to extract the quality of hotels in the view of this user group. We estimate the quality of a hotel via its average rating, say $\widehat{Q_i} = \sum_j r_{i,j}/n_i$, where $\widehat{Q_i}$ denotes the estimating value of Q_i. The larger the number of the observed ratings, the higher the estimation accuracy. To guarantee a high estimation accuracy, we select those hotels with at least 300 ratings. In total, 147 hotels in the dataset satisfy this selection criteria. We map the ID of them to $1, \ldots, 147$ and use $P_1, \ldots, P_{147}$ to denote them and their inferred quality is denoted by $\widehat{Q}_1, \ldots, \widehat{Q}_{147}$. In the following we treat $\widehat{Q_i}$ as the ground truth value of Q_i.

We present an algorithm to validate our model with the above testing dataset. We seek to compare the recommendation accuracy under our model and under the real-world rating setting. We input the above selected 147 hotels $P_1, \ldots, P_{147}$ into our model, say $N = 147$. We set $k = 30$, say the system recommends 30 hotels to the whole user group. We treat $\widehat{Q}_i$ as ground truth value of Q_i and we synthetic ratings for each hotel using our model setting homophyly degree $\sigma(h) = 1$. Let I_k denote the number of top-k hotels that get recommended. We choose expectation of I_k to measure the recommendation accuracy. To compare the recommendation accuracy, we vary the number of ratings per hotel n (each hotel receive the same number of ratings) from 5 to 300. For each n, we synthetic n ratings for each hotel, so to evaluate the recommendation accuracy under our model. And, we pick n latest ratings from the dataset for each hotel, so to evaluate the recommendation accuracy under the real-world rating setting. Based on the above idea, we outline the model validation algorithm in Algorithm 2, where $\tau_i = \{\tau_{i,1}, \ldots, \tau_{i,n_i}\}$ denote a set of sorted ratings (based ratings' times tamps) for hotel P_i, n_i is the number of ratings of P_i, and $\tau_{i,1}$ is the oldest rating. Note that, similar with Algorithm 1, Algorithm 2 is also a randomized algorithm, since evaluate the exact value of $E[I_k]$ is computationally expensive. We can derive similar performance guarantees for Algorithm 2 as that derived for Algorithm 1.

Algorithm 2 : Model validation

Input: Product quality: $\widehat{Q}_1, \ldots, \widehat{Q}_{147}$, sorted rating dataset: $\tau_1, \ldots, \tau_{147}$,
number of ratings per hotel: n, voting rule: $\mathcal{V}$, tie breaking rule: $\mathcal{T}$
data indicator: *data_ind* \\ 0 - synthetic ratings, 1 - real-world ratings
Output: $E[I_k]$
produce the group truth top-k products set $\mathcal{R}^I(k)$ via selecting k products with the highest value of $\widehat{Q}_i$.
for all $i = 0, \ldots, k$, $\ell_i \leftarrow 0$
for $j = 1$ to K **do**
  for $i = 1$ to N **do**
    if *data_ind* $== 0$ **then**
      Generate n ratings, $r_{i,1}, \ldots, r_{i,n}$, for product P_i using our model,
      $\mathbf{r}_i \leftarrow \{r_{i,1}, \ldots, r_{i,n}\}$
    else
      Pick n last ratings for the product P_i from TripAdvisor dataset as the observed ratings, say $\mathbf{r}_i \leftarrow \{\tau_{i,n_i-n+1}, \ldots, \tau_{i,n_i}\}$
    end if
  end for
  simulate the decision making process, i.e., applying the voting rule $\mathcal{V}$ and the tie breaking rule $\mathcal{T}$ to produce the set $\mathcal{R}(k)$ based on the score sets $\{\mathbf{r}_1, \ldots, \mathbf{r}_N\}$
  $i \leftarrow |\mathcal{R}^I(k) \cap \mathcal{R}^I(k)|$, $\ell_i \leftarrow \ell_i + 1$.
end for
$E[I_k] \leftarrow \sum_{i=0}^{k} i\ell_i / K$

We run Algorithm 2 to validate our model and we examine the desired number of ratings for TripAdvisor as well. In Algorithm 2, we set the voting rule to be the *average score rule* and the tie breaking rule to be the *least variance rule*. Running Algorithm 2 we obtain the numerical results for the recommendation accuracy, say $E[I_{30}]$, under our model and under the real-world setting. We show them in Fig. 10. One can observe that the expectation curves corresponding the our model and the real-world setting overlapped together. In other the recommendation accuracy under our model is nearly the same with that under the real-world setting. This implies that our model is very accurate in capturing various important factors of a group recommendation system. An interesting observation is that, for TripAdvisor, around one hundred ratings per product can ensure a high accuracy recommendation.

Lessons learned. Our model is accurate in capturing various important factors of a group recommendation system. Around one hundred ratings per hotel can guarantee a high accuracy recommendation for TripAdvisor.

7. Related work

The research on *recommender systems* [1] has received a lot of attention, since the seminal work on collaborative filtering [16–18]. In general, recommender systems suggest products to a user by taking into account the preference of that user. Researchers investigated various algorithmic and complexity issues [1,4–7]. From the industry side, we see a number of successful applications of recommender systems, e.g., www.Amazon.com [2], MovieLens [3], etc. A comprehensive survey can be found in [19].

Group recommendation systems [9] were introduced to deal with the contexts where users operate in *groups*. Previous works mainly focus on algorithmic and complexity issues of group product recommendation systems [20–24]. We can also see a number of commercial products of group product recommendation systems [8,10–13]. But little attention has been paid to examine the *accuracy* and *effectiveness* of a group product recommendation. The *partial preference information* makes it challenge to examine the *accuracy* and *effectiveness*, and the goal of this paper is to fill in this void. To the best of our knowledge, this is the first work which provides a formal model and analysis of such kind of systems.

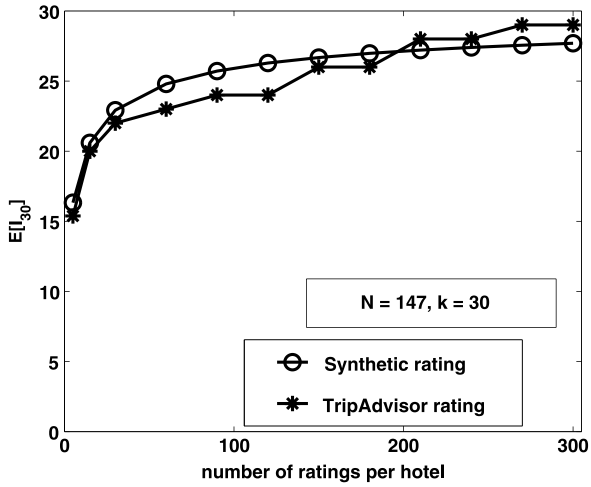


Fig. 10. Comparison of $E[I_{30}]$ under synthetic ratings and ratings from TripAdvisor.

One closely related work is one of our archival technical reports [25]. We summarize the difference between [25] and this paper as follows. First, the domain of [25] and this paper are very different. The study of [25] falls into the domain of *online quality evaluation systems*, where users act as *reviewers*, and they evaluate the quality of products in the form of ratings. For example, users evaluate the reputation index of sellers on eBay, or evaluate the quality of a page on Wikipedia. The objective is to identify the intrinsic quality of an item (or the true reputation index of a seller) relying on "wisdom of the crowd''. However, this paper falls into the domain of *recommendation systems*, where the personalized quality of a product is of essential importance. Users express their ratings in order to show their *preferences*, such as IMDB. The objective is to make personalized recommendations taking into account the preference of a user. Second, the technical details are quite different. In [25], we derived theoretical bounds on the desired number of ratings to guarantee that the aggregate rating reveals the intrinsic quality of a product. We also derived for certain conditions under which the system will be disrupted. In this paper, however, we develop a general framework and an efficient methodology to evaluate the efficiency and effectiveness of a *group recommendation system*. More precisely, we propose several probability measures to characterize the recommendation accuracy, and we derive their analytical expressions. Through this analysis, we gain some important insights on how design an efficient randomized algorithm to evaluate the recommendation accuracy. Furthermore, we formally derive theoretical performance guarantees for this algorithm. Third, the experimental studies are very different. In [25], the authors mainly evaluated the accuracy and robustness of the *majority rule* and the *average score rule* under the honest rating and the misbehavior settings. The key observation is that the *majority rule* is more accurate and robust than the *average score rule*. However, in this paper we evaluate the impact of various factors on the recommendation accuracy, such as *homophyly degree, voting rule, tie breaking rule*, etc. One interesting observation is that the recommendation accuracy under different voting rules has a very small variation. Finally, in this paper we also propose an efficient rating scheme to improve the recommendation accuracy, while [25] did not cover this technical aspect.

8. Conclusions

This paper develops a mathematical framework and efficient methodology to analyze the *accuracy* and *effectiveness* of group product recommendations. We present a mathematical model to capture various factors which may influence the accuracy of a group product recommendation system. We formally analyze the model and through this analysis, we gain the insight to develop a randomized algorithm to evaluate the recommendation accuracy. Our methodology is computationally efficient and can provide theoretical performance guarantees on various performance measures. We propose a novel and efficient *heterogeneous rating scheme* which requires equal or less rating workload, but can improve over a homogeneous rating scheme by as much as 30%. We carry out experiments on both synthetic data and real-world data (from TripAdvisor). We not only validate our model, but also find a number of important observations, for example, a small of misbehaving users can decrease the recommendation accuracy remarkably. For TripAdvisor, one hundred ratings per product is sufficient to guarantee a high accuracy recommendation. We believe that our model and methodology are important building blocks for researchers to study group recommendation systems.

Appendix

Proof of Lemma 3. Let $\mathcal{H} = \{P_{i_1}, \ldots, P_{i_k}\}$. Let $\overline{\mathcal{H}} = \{P_1, \ldots, P_N\} \backslash \mathcal{H}$ be the complement of $\mathcal{H}$. Let $\gamma_{\min}(\mathcal{H}) = \min\{\gamma_i, i \in \mathcal{I}\}$ denote the minimum average rating of product set $\mathcal{H}$. Let $\gamma_{\max}(\overline{\mathcal{H}}) = \max\{\gamma_i, i \in \overline{\mathcal{I}}\}$ denote the maximum average rating of product set $\overline{\mathcal{H}}$. The probability that the recommended product set equals to $\{P_{i_1}, \ldots, P_{i_k}\}$ can be divided into the following three parts:

$$\begin{aligned}\Pr[\mathcal{R}(k) = \{P_{i_1}, \ldots, P_{i_k}\}] = \Pr[\mathcal{R}(k) = \mathcal{H}, \gamma_{\min}(\mathcal{H}) < \gamma_{\max}(\overline{\mathcal{H}})] + \Pr[\mathcal{R}(k) = \mathcal{H}, \gamma_{\min}(\mathcal{H}) > \gamma_{\max}(\overline{\mathcal{H}})] \\ + \Pr[\mathcal{R}(k) = \mathcal{H}, \gamma_{\min}(\mathcal{H}) = \gamma_{\max}(\overline{\mathcal{H}})]. \end{aligned} \tag{A.1}$$

Let us derive these three terms individually.

According to our voting rule, the recommended product set $\mathcal{R}(k)$ equals to $\mathcal{H}$ is 0 conditioned on that $\gamma_{\min}(\mathcal{H})$ is less than $\gamma_{\max}(\overline{\mathcal{H}})$. Thus,

$$\Pr[\mathcal{R}(k) = \mathcal{H}, \gamma_{\min}(\mathcal{H}) < \gamma_{\max}(\overline{\mathcal{H}})] = 0. \tag{A.2}$$

According to our voting rule, the recommended product set $\mathcal{R}(k)$ equals to $\mathcal{H}$ is 1 conditioned on $\gamma_{\min}(\mathcal{H}) > \gamma_{\max}(\overline{\mathcal{H}})$. Thus,

$$\Pr[\mathcal{R}(k) = \mathcal{H}, \gamma_{\min}(\mathcal{H}) > \gamma_{\max}(\overline{\mathcal{H}})] = \Pr[\gamma_{\min}(\mathcal{H}) > \gamma_{\max}(\overline{\mathcal{H}})]. \tag{A.3}$$

Note that $\gamma_i = \sum_{i=1}^{n} r^o_{i,j}/n, \ \forall i$. Since the ratings $r^o_{i,j}, \ \forall i, j$, are independent random variables, thus the average ratings $\gamma_1, \ldots, \gamma_N$ are also independent random variables. Based on this fact, we derive the analytical expression of $\Pr[\gamma_{\min}(\mathcal{H}) > \gamma_{\max}(\overline{\mathcal{H}})]$ as

$$\begin{aligned}\Pr[\gamma_{\min}(\mathcal{H}) > \gamma_{\max}(\overline{\mathcal{H}})] &= \sum_{\ell=n}^{nm} \Pr[\gamma_{\min}(\mathcal{H}) = \ell/n] \Pr\left[\gamma_{\max}(\overline{\mathcal{H}}) < \ell/n\right] \\ &= \sum_{\ell=n}^{nm} \left(\prod_{i\in\mathcal{I}} \Pr\left[\gamma_i \le \frac{\ell}{n}\right] - \prod_{i\in\mathcal{I}} \Pr\left[\gamma_i \le \frac{\ell-1}{n}\right]\right) \prod_{j\in\overline{\mathcal{I}}} \Pr\left[\gamma_j \le \frac{\ell-1}{n}\right]. \end{aligned} \tag{A.4}$$

The remaining task is to derive the last term of Eq. (A.1). When $\gamma_{\min}(\mathcal{H}) = \gamma_{\max}(\overline{\mathcal{H}})$ occurs, tie breaking will be performed on the set of products with the average rating equal to $\gamma_{\min}(\mathcal{H})$. Let us provide some notations first. Let $\mathcal{F} = \{i \mid \gamma_i = \gamma_{\min}(\mathcal{H}), i \in \mathcal{I}\}$ be the index set of the products that belong to set $\mathcal{H}$ and with average ratings equal to minimum average score of $\mathcal{H}$. Let $\mathcal{G} = \{i \mid \gamma_i = \gamma_{\min}(\mathcal{H}), i \in \overline{\mathcal{I}}\}$ be the index set of the products that belong to set $\overline{\mathcal{H}}$ and with average ratings equal to the minimum average score of $\mathcal{H}$. Thus, tie breaking will perform on the products with index set $\mathcal{F} \cup \mathcal{G}$, from which only $|\mathcal{F}|$ products will be selected for recommendation. By enumerating all possible tie breaking product sets, we can divide the last term of Eq. (A.1) into the following form:

$$\Pr[\mathcal{R}(k) = \mathcal{H}, \gamma_{\min}(\mathcal{H}) = \gamma_{\max}(\overline{\mathcal{H}})] = \sum_{\mathcal{F}\subseteq\mathcal{I}, \mathcal{G}\subseteq\overline{\mathcal{I}}, \mathcal{F},\mathcal{G}\neq\emptyset} \Pr[\mathcal{F} \cup \mathcal{G}] \Pr[\mathcal{F} \mid \mathcal{F} \cup \mathcal{G}], \tag{A.5}$$

where $\Pr[\mathcal{F} \cup \mathcal{G}]$ is the probability that tie breaking is performed on products with index set $\mathcal{F} \cup \mathcal{G}$, and $\Pr[\mathcal{F}|\mathcal{F} \cup \mathcal{G}]$ is the conditional probability that products with index set $\mathcal{F}$ are selected for recommendation under the condition that tie breaking is performed on the products with index set $\mathcal{F} \cup \mathcal{G}$. Since the tie-breaking rule is the random rule, under which we just randomly pick $|\mathcal{F}|$ products, thus

$$\Pr[\mathcal{F}|\mathcal{F} \cup \mathcal{G}] = \binom{|\mathcal{F} \cup \mathcal{G}|}{|\mathcal{F}|}^{-1}. \tag{A.6}$$

Because the average ratings $\gamma_1, \ldots, \gamma_N$ are independent random variables, we can derive $\Pr[\mathcal{F} \cup \mathcal{G}]$ as:

$$\begin{aligned}\Pr[\mathcal{F} \cup \mathcal{G}] &= \Pr[\gamma_i > \gamma_{\min}(\mathcal{H}), \forall i \in \mathcal{I} \setminus \mathcal{F}] \Pr[\gamma_i = \gamma_{\min}(\mathcal{H}), \forall i \in \mathcal{F} \cup \mathcal{G}] \times \Pr[\gamma_i < \gamma_{\min}(\mathcal{H}), \forall i \in \overline{\mathcal{I}} \setminus \mathcal{G}] \\ &= \sum_{\ell=n}^{nm} \prod_{i\in\mathcal{I}\setminus\mathcal{F}} \left(1 - \Pr\left[\gamma_i \le \frac{\ell}{n}\right]\right) \prod_{j\in\mathcal{F}\cup\mathcal{G}} \Pr\left[\gamma_j = \frac{\ell}{n}\right] \prod_{\kappa\in\overline{\mathcal{I}}\setminus\mathcal{G}} \Pr\left[\gamma_\kappa \le \frac{\ell-1}{n}\right]. \end{aligned} \tag{A.7}$$

Combining Eqs. (A.5)–(A.7) we obtain

$$\begin{aligned}\Pr[\mathcal{R}(k) = \mathcal{H}, \gamma_{\min}(\mathcal{H}) = \gamma_{\max}(\overline{\mathcal{H}})] = &\sum_{\mathcal{F}\subseteq\mathcal{I}, \mathcal{G}\subseteq\overline{\mathcal{I}}, \mathcal{F},\mathcal{G}\neq\emptyset} \sum_{\ell=n}^{nm} \binom{|\mathcal{F} \cup \mathcal{G}|}{|\mathcal{F}|}^{-1} \prod_{i\in\mathcal{I}\setminus\mathcal{F}} \\ &\times (1 - \Pr[\gamma_i \le \ell/n]) \prod_{j\in\mathcal{F}\cup\mathcal{G}} \Pr[\gamma_j = \ell/n] \prod_{\kappa\in\overline{\mathcal{I}}\setminus\mathcal{G}} \Pr[\gamma_\kappa \le (\ell-1)/n]. \end{aligned} \tag{A.8}$$

Combining Eqs. (A.1)–(A.4), and (A.8), we complete the proof. □

Proof of Theorem 4. Let us state a theorem on bounding the tail probability.

Theorem 8 (*Chernoff Bound [14]*). *Let $X_1, \dots, X_n$ be n independent random variables with $X_i = 1$ with probability p and 0 otherwise. Let $X = \sum_{i=1}^{n} X_i$ and let $\mu = E[X] = np$. Then for each $\epsilon > 0$, it holds that* $\Pr[|X-\mu| \geq \epsilon\mu] \leq 2e^{-\mu \min\{\epsilon^2,\epsilon\}/3}$.

We apply Theorem 8 to show that for each given i, the following Inequality

$$|\widehat{\Pr}[I(k) = i] - \Pr[I(k) = i]| \leq \epsilon \Pr[I(k) = i] \tag{A.9}$$

holds with probability at least $1 - \delta/(k+1)$. Given i, let $\mathbf{I}_{ij}$ be an indicator random variable, where $j = 1, \dots, K$, such that

$$\mathbf{I}_{ij} = \begin{cases} 1 & \text{if in } j\text{th round, } |\mathcal{R}^I(k) \cap \mathcal{R}(k)| = i \\ 0 & \text{otherwise.} \end{cases}$$

Recall that $I(k) = |\mathcal{R}^I(k) \cap \mathcal{R}(k)|$. Consider a special case of $\Pr[I(k) = i] = 0$. The physical meaning implies that the event $I(k) = i$ never happen, or $\mathbf{I}_{ij} = 0,\ \forall j$. In other words, $\widehat{\Pr}[I(k) = i] = \sum_{j=1}^{K} \mathbf{I}_{ij}/K = 0$. Thus $\left|\widehat{\Pr}[I(k) = i] - \Pr[I(k) = i]\right| \leq \epsilon \Pr[I(k) = i]$. Consider $\Pr[I(k) = i] > 0$. Since each round runs independently, thus $\mathbf{I}_{i1}, \dots, \mathbf{I}_{iK}$ are K independent random variables with $\mathbf{I}_{ij} = 1$ with probability $\Pr[I(k) = i]$ and 0 otherwise. Let $\mathbf{I}_i = \sum_{j=1}^{K} \mathbf{I}_{ij}$. Then $E[\mathbf{I}_i] = K \Pr[I(k) = i]$. Observe that $\widehat{\Pr}[I(k) = i] = \mathbf{I}_i/K$. Then by applying Theorem 8 we have,

$$\Pr\left[|\widehat{\Pr}[I(k) = i] - \Pr[I(k) = i]| \geq \epsilon \Pr[I(k) = i]\right] = \Pr\left[|\mathbf{I}_i - K \Pr[I(k) = i]| \geq \epsilon K \Pr[I(k) = i]\right] \leq 2e^{-K \Pr[I(k)=i]\epsilon^2/3},$$

by substituting K with Inequality (15), we have $2e^{-K \Pr[I(k)=i]\epsilon^2/3} \leq 1 - \delta/(k+1)$. Then it follows that for each given i Inequality (A.9) holds with probability at least $1 - \delta/(k+1)$.

We show that Inequality (A.9) holds for all $i = 0, 1, \dots, k$ with probability at least $1 - \delta$. We apply union bound to show this, specifically

$$\begin{aligned} \Pr[\text{Inequality (A.9) holds for all } i = 0, 1, \dots, k] &= 1 - \Pr[\exists i \text{ such that Inequality (A.9) not holds}] \\ &\geq 1 - \sum_{i=0}^{k} \Pr[\text{Inequality (A.9) not holds for } i] \geq 1 - \delta, \end{aligned}$$

which completes the proof. □

Proof of Theorem 5. Let us derive the theoretical performance guarantee for the expectation of $I(k)$, or $E[I(k)]$. It is quite straightforward,

$$\begin{aligned} \left|\widehat{E}[I(k)] - E[I(k)]\right| &= \left|\sum_{i=0}^{k} i\left(\widehat{\Pr}[I(k) = i] - \Pr[I(k) = i]\right)\right| \\ &\leq \sum_{i=0}^{k} i\left|\widehat{\Pr}[I(k) = i] - \Pr[I(k) = i]\right| \leq \sum_{i=0}^{k} \epsilon i \Pr[I(k) = i] = \epsilon E[I(k)]. \end{aligned}$$

Let $\Delta E[I(k)] = \widehat{E}[I(k)] - E[I(k)]$ denote the approximation error of expectation. We show $(\Delta E[I(k)])^2 \leq \epsilon^2 \mathrm{Var}[I(k)]$. Let $\Delta p_i = \widehat{\Pr}[I(k) = i] - \Pr[I(k) = i]$ denote the approximation error of probability. Observe that the summation of all Δp_i equals to zero, or

$$\sum_{i=0}^{k} \Delta p_i = \sum_{i=0}^{k} \widehat{\Pr}[I(k) = i] - \sum_{i=0}^{k} \Pr[I(k) = i] = 1 - 1 = 0.$$

Based on this result, we derive a general upper bound of $(\Delta E[I(k)])^2$ as follows

$$\begin{aligned} (\Delta E[I(k)])^2 &= \left(\sum_{i=0}^{k} i\Delta p_i\right)^2 = \left(\sum_{i=0}^{k} i\Delta p_i - E[I(k)]\sum_{i=0}^{k} \Delta p_i\right)^2 \\ &\leq \left(\sum_{i=0}^{k} |i - E[I(k)]||\Delta p_i|\right)^2. \end{aligned} \tag{A.10}$$

Note that $|\Delta p_i| \leq \epsilon \Pr[I(k) = i]$. Substituting $|\Delta p_i|$ with this inequality we have

$$(\Delta E[I(k)])^2 \leq \left(\sum_{i=0}^{k} |i - E[I(k)]|\epsilon \Pr[I(k) = i]\right)^2,$$

by applying Cauchy's Inequality we have,

$$(\Delta E[I(k)])^2 \le \left[\sum_{i=0}^{k} \epsilon^2 \Pr[I(k)=i]\right]\left[\sum_{i=0}^{k}(i-E[I(k)])^2 \Pr[I(k)=i]\right] = \epsilon^2 \mathrm{Var}[I(k)].$$

We now apply the above result to derive the theoretical performance guarantee for the variance. With some basic probability arguments, we derive $\widehat{\mathrm{Var}}[I(k)] - \mathrm{Var}[I(k)]$ as follows:

$$\begin{aligned}
\widehat{\mathrm{Var}}[I(k)] - \mathrm{Var}[I(k)] &= \sum_{i=0}^{k} i^2 \widehat{\Pr}[I(k)=i] - \left(\widehat{E}[I(k)]\right)^2 - \sum_{i=0}^{k} i^2 \Pr[I(k)=i] + (E[I(k)])^2 \\
&= \sum_{i=0}^{k} i^2 \Delta p_i - 2E[I(k)]\Delta E[I(k)] - (\Delta E[I(k)])^2 \\
&= \sum_{i=0}^{k} (i-E[I(k)])^2 \Delta p_i - (E[I(k)])^2 \sum_{i=0}^{k} \Delta p_i - (\Delta E[I(k)])^2.
\end{aligned}$$

Observe that the summation of all Δp_i equals to zero, say $\sum_{i=0}^{k} \Delta p_i = 0$. Substituting $\sum_{i=0}^{k} \Delta p_i$ with 0 in the above equation, we derive an upper bound for $\widehat{\mathrm{Var}}[I(k)] - \mathrm{Var}[I(k)]$, namely

$$\begin{aligned}
\left|\widehat{\mathrm{Var}}[I(k)] - \mathrm{Var}[I(k)]\right| &= \left|\sum_{i=0}^{k} (i-E[I(k)])^2 \Delta p_i - (\Delta E[I(k)])^2\right| \\
&\le \sum_{i=0}^{k} (i-E[I(k)])^2 |\Delta p_i| + (\Delta E[I(k)])^2. \qquad (A.11)
\end{aligned}$$

Note that $|\Delta p_i| \le \epsilon \Pr[I(k)=i]$ and $(\Delta E[I(k)])^2 \le \epsilon^2 \mathrm{Var}[I(k)]$. By substituting Δp_i and $(\Delta E[I(k)])^2$ with these two inequalities we have

$$\begin{aligned}
\left|\widehat{\mathrm{Var}}[I(k)] - \mathrm{Var}[I(k)]\right| &\le \sum_{i=0}^{k} (i-E[I(k)])^2 \epsilon \Pr[I(k)=i] + (\Delta E[I(k)])^2 \\
&\le \epsilon \mathrm{Var}[I(k)] + \epsilon^2 \mathrm{Var}[I(k)] = \epsilon(1+\epsilon)\mathrm{Var}[I(k)],
\end{aligned}$$

which completes the proof. □

Proof of Theorem 7. Let us derive the theoretical performance guarantee for the expectation of $I(k)$, or $E[I(k)]$. Recall that $\Delta p_i = \widehat{\Pr}[I(k)=i] - \Pr[I(k)=i]$. We then have

$$|\Delta p_i| \le \max\left\{\epsilon\sqrt{\Pr[I(k)=i]}, \epsilon^2\right\} \le \epsilon\sqrt{\Pr[I(k)=i]} + \epsilon^2, \quad \forall i = 0, 1, \ldots, k.$$

Then it follow that

$$\begin{aligned}
\left|\widehat{E}[I(k)] - E[I(k)]\right| &\le \sum_{i=0}^{k} i|\Delta p_i| \le \sum_{i=0}^{k} i\left(\epsilon\sqrt{\Pr[I(k)=i]} + \epsilon^2\right) \\
&\le \sum_{i=0}^{k} \epsilon i \sqrt{\Pr[I(k)=i]} + \epsilon^2 k^2
\end{aligned}$$

by applying Cauchy's Inequality we have,

$$\begin{aligned}
\left|\widehat{E}[I(k)] - E[I(k)]\right| &\le \epsilon\sqrt{\left(\sum_{i=0}^{k} i\right)\left(\sum_{i=0}^{k} i \Pr[I(k)=i]\right)} + \epsilon^2 k^2 \\
&= \epsilon\sqrt{(k+1)kE[I(k)]/2} + \epsilon^2 k^2 \\
&\le \epsilon k \sqrt{E[I(k)]} + \epsilon^2 k^2.
\end{aligned}$$

We now derive the theoretical performance guarantee for the variance of $I(k)$, or $\mathrm{Var}[I(k)]$. For the ease of presentation, we let $\Delta p_i = \widehat{\Pr}[I(k)=i] - \Pr[I(k)=i]$ and let $\Delta E[I(k)] = \widehat{E}[I(k)] - E[I(k)]$. Inequality (A.11) gives a general upper bound for $\left|\widehat{\mathrm{Var}}[I(k)] - \mathrm{Var}[I(k)]\right|$, namely

$$\left|\widehat{\mathrm{Var}}[I(k)] - \mathrm{Var}[I(k)]\right| \le \sum_{i=0}^{k} (i-E[I(k)])^2 |\Delta p_i| + (\Delta E[I(k)])^2. \qquad (A.12)$$

Let us now proceed to individually derive these two terms of the above inequality. Observe that

$$(\Delta p_i)^2 \le \max\left\{\left(\epsilon\sqrt{\Pr[I(k)=i]}\right)^2, \epsilon^4\right\} \le \epsilon^2 \Pr[I(k)=i] + \epsilon^4.$$

With this inequality we can bound $\sum_{i=0}^{k}(i - E[I(k)])^2|\Delta p_i|$ as follows,

$$\begin{aligned}
\sum_{i=0}^{k}(i-E[I(k)])^2|\Delta p_i| &\le \left[\sum_{i=0}^{k}(i-E[I(k)])^2(\Delta p_i)^2\right]^{1/2}\left[\sum_{i=0}^{k}(i-E[I(k)])^2\right]^{1/2}\\
&\le \left[\sum_{i=0}^{k}(i-E[I(k)])^2(\epsilon^2 \Pr[I(k)=i]+\epsilon^4)\right]^{1/2}\left[\sum_{i=0}^{k} i^2\right]^{1/2}\\
&\le \left[\epsilon^2 \mathrm{Var}[I(k)] + \epsilon^4\sum_{i=0}^{k} i^2\right]^{1/2}\left[\sum_{i=0}^{k} i^2\right]^{1/2}\\
&= \left[\epsilon^2 \mathrm{Var}[I(k)] + \epsilon^4\frac{k(k+1)(2k+1)}{6}\right]^{1/2}\left[\frac{k(k+1)(2k+1)}{6}\right]^{1/2}\\
&\le \epsilon k\sqrt{k\mathrm{Var}[I(k)] + \epsilon^2 k^4},
\end{aligned} \tag{A.13}$$

where the first step is obtained by applying the Cauchy's Inequality and the last step follows that $\frac{k(k+1)(2k+1)}{6} \le k^3$. The remaining task is to bound the term $(\Delta E[I(k)])^2$. Inequality (A.10) gives a general upper bound of it, namely $(\Delta E[I(k)])^2 \le \left(\sum_{i=0}^{k}|i-E[I(k)]||\Delta p_i|\right)^2$. With this inequality, we can derive $(\Delta E[I(k)])^2$ as follows

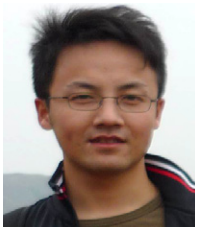
$$\begin{aligned}
(\Delta E[I(k)])^2 &\le \left(\sum_{i=0}^{k}\epsilon|i-E[I(k)]||\Delta p_i/\epsilon|\right)^2 \le \sum_{j=0}^{k}\epsilon^2\sum_{i=0}^{k}(i-E[I(k)])^2(\Delta p_i/\epsilon)^2\\
&\le \epsilon^2(k+1)\sum_{i=0}^{k}(i-E[I(k)])^2(\Pr[I(k)=i]+\epsilon^2)\\
&\le \epsilon^2(k+1)(\mathrm{Var}[I(k)]+\epsilon^2 k^3),
\end{aligned} \tag{A.14}$$

where the second step is obtained by applying Cauchy's Inequality and the last step follows the inequality $\sum_{i=0}^{k}(i - E[I(k)])^2 \le \sum_{i=0}^{k} i^2 = \frac{k(k+1)(2k+1)}{6} \le k^3$. Applying Inequality (A.13) and (A.14) to Inequality (A.12) we complete the proof. □

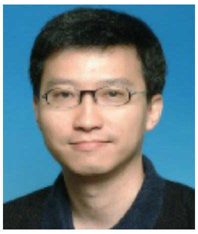
References

[1] P. Resnick, H.R. Varian, Recommender systems, Commun. ACM 40 (1997) 56–58.
[2] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Comput. 7 (2003) 76–80.
[3] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, J. Riedl, Movielens unplugged: experiences with an occasionally connected recommender system, in: Proceedings of the 8th International Conference on Intelligent User Interfaces, ACM, 2003.
[4] M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms, ACM Trans. Inf. Syst. 22 (2004) 143–177.
[5] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (2004) 5–53.
[6] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proc. of WWW, ACM, 2001.
[7] J.B. Schafer, J. Konstan, J. Riedl, Recommender systems in e-commerce, in: Proc. of ACM EC, ACM, 1999.
[8] M. O'Connor, D. Cosley, J.A. Konstan, J. Riedl, Polylens: a recommender system for groups of users, in: Proc. of ECSCW.
[9] L. Boratto, S. Carta, State-of-the-art in group recommendation and new approaches for automatic identification of groups, in: Information Retrieval and Mining in Distributed Environments, 2011, pp. 1–20.
[10] D. Goren-Bar, O. Glinansky, Fit-recommend ing TV programs to family members, Comput. Graph. 28 (2004) 149–156.
[11] E. Vildjiounaite, V. Kyllönen, T. Hannula, P. Alahuhta, Unobtrusive dynamic modelling of TV programme preferences in a finnish household, Multimedia Syst. 15 (2009) 143–157.
[12] A. Jameson, S. Baldes, T. Kleinbauer, Two methods for enhancing mutual awareness in a group recommender system, in: Proceedings of the Working Conference on Advanced Visual Interfaces, ACM, 2004.
[13] F. Lorenzi, F. dos Santos, P. Ferreira, A. Bazzan, Optimizing preferences within groups: a case study on travel recommendation, in: Advances in Artificial Intelligence-SBIA, 2008, pp. 103–112.
[14] M. Mitzenmacher, E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge Univiversity Press, 2005.
[15] H. Wang, Y. Lu, C. Zhai, Latent aspect rating analysis on review text data: a rating regression approach, in: Proc. of ACM KDD, ACM, 2010.
[16] W. Hill, L. Stead, M. Rosenstein, G. Furnas, Recommending and evaluating choices in a virtual community of use, in: Proc. of ACM CHI.
[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proc. of ACM CSCW, ACM, 1994.
[18] U. Shardanand, P. Maes, Social information filtering: algorithms for automating “word of mouth&rdquo, in: Proc. of ACM CHI, ACM Press/Addison-Wesley Publishing Co., 1995.
[19] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (2005) 734–749.

[20] A. Jameson, More than the sum of its members: challenges for group recommender systems, in: Proceedings of the Working Conference on Advanced Visual Interfaces, ACM, 2004.
[21] S. Amer-Yahia, S.B. Roy, A. Chawlat, G. Das, C. Yu, Group recommendation: semantics and efficiency, Proc. VLDB Endow. 2 (2009) 754–765.
[22] J. Masthoff, Group modeling: selecting a sequence of television items to suit a group of viewers, User Model. User-Adapt. Interact. 14 (2004) 37–85.
[23] J. McCarthy, Pocket restaurantfinder: a situated recommender system for groups, in: Proc. of ACM CHI Workshop.
[24] L.M. de Campos, J.M. Fernandez-Luna, J.F. Huete, M.A. Rueda-Morales, Group recommending: a methodological approach based on Bayesian networks, in: Proc. of the IEEE ICDE Workshop.
[25] H. Xie, John C.S. Lui, Mathematical modeling of product rating: sufficiency, misbehavior and aggregation rules, 2013. ArXiv Preprint arXiv:1305.1899.



Hong Xie received his B.Eng. degree from the School of Computer Science and Technology at the University of Science and Technology of China in 2010. He is now a Ph.D. candidate in the Department of Computer Science and Engineering at the Chinese University of Hong Kong. His advisor is Prof. John C.S. Lui. His interests include network economics, data analytics, stochastic modeling, crowdsourcing and online social networks.



John C.S. Lui was born in Hong Kong and is currently a professor in the Department of Computer Science & Engineering at the Chinese University of Hong Kong. He received his Ph.D. in Computer Science from UCLA. His current research interests are in communication networks, network/system security (e.g., cloud security, mobile security, etc.), network economics, network sciences (e.g., online social networks, information spreading, etc.), cloud computing, large scale distributed systems and performance evaluation theory. John serves in the editorial board of IEEE/ACM Transactions on Networking, IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, Journal of Performance Evaluation and International Journal of Network Security. John was the chairman of the CSE Department from 2005 to 2011. He received various departmental teaching awards and the CUHK Vice-Chancellor's Exemplary Teaching Award. He is also a corecipient of the IFIP WG 7.3 Performance 2005 and IEEE/IFIP NOMS 2006 Best Student Paper Awards. He is an elected member of the IFIP WG 7.3, Fellow of ACM, Fellow of IEEE and Croucher Senior Research Fellow. His personal interests include films and general reading.