Beyond Personalization and Anonymity: Towards a Group-Based Recommender System

Shang Shang
Department of Electrical Engineering
Princeton University
Princeton, NJ 08540 U.S.A.
sshang@princeton.edu

Yuk Hui
Centre for Digital Cultures
Leuphana University
Scharnhorststr. 1 Lüneburg
yuk.hui@leuphana.de

Pan Hui
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China
panhui@cse.ust.hk

Paul Cuff
Department of Electrical Engineering
Princeton University
Princeton, NJ 08540 U.S.A.
cuff@princeton.edu

Sanjeev Kulkarni
Department of Electrical Engineering
Princeton University
Princeton, NJ 08540 U.S.A.
kulkarni@princeton.edu

ABSTRACT

Recommender systems have received considerable attention in recent years. Yet with the development of information technology and social media, the risk in revealing private data to service providers has been a growing concern to more and more users. Trade-offs between quality and privacy in recommender systems naturally arise. In this paper, we present a privacy preserving recommendation framework based on groups. The main idea is to use groups as a natural middleware to preserve users' privacy. A distributed preference exchange algorithm is proposed to ensure the anonymity of data, wherein the effective size of the anonymity set asymptotically approaches the group size with time. We construct a hybrid collaborative filtering model based on Markov random walks to provide recommendations and predictions to group members. Experimental results on the MovieLens dataset show that our proposed methods outperform the baseline methods, L+ and ItemRank, two state-of-the-art personalized recommendation algorithms, for both recommendation precision and hit rate despite the absence of personal preference information.

Keywords

Recommender system, privacy, group-based social networks

1. INTRODUCTION

With the recent development of social media, personalization and privacy preservation are often in tension with each other. Private companies such as Google and Facebook are accumulating and recording enormous amounts of personal data for the sake of personalization. Personalization provides users with conveniences, and it can have a direct impact on marketing, sales, and profit. On the other hand, privacy, which is a serious concern for many users, is the price users have to pay for the convenience of recommender systems in a world with booming information. Users normally have no choice but to trust the service provider to keep their sensitive personal profile safe. However, it is not always "safe." For example, a shopping website one has visited once might keep appearing on the advertising block for days when browsing some other web pages.

Current approaches to protect privacy in recommender systems mostly address two different privacy concerns: protecting users' privacy from curious peers or malicious users [14, 15], and protecting against unreliable service providers [1, 6, 16]. In order to make the outcome of recommendation insensitive to single input so as to protect users' private preference data from other users, privacy preserving algorithms from the differential privacy literature [8] have been modified to provide privacy guarantees. McSherry et al. [15] adapted the leading approaches from the Netflix Challenge by adding noise to data to provide differential privacy and recommendations on movies. Machanavajjhala et al. [14] studied recommendations based on a user's social network with differential privacy constraints. In addition to differential privacy [8], other notions of privacy such as k-anonymity [25], l-diversity [13], effective size of anonymity set [17], etc., have also been studied and are used to protect individual privacy. On the other side, in order to prevent a single party, e.g. the service provider, from gaining access to every user's data, cryptographic solutions based on secure multi-party computation are proposed in [1, 6], and a distributed hierarchical neighborhood formation was proposed in [3] to reduce the privacy hazard. In [6] cryptography and recommendation are computed by end-users, which is likely to suffer from the limitation of personal computation devices. Aïmeur et al.[1] introduced a semi-trusted third party to share the sensitive information with service provider, and a two-party computation is then performed

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SAC'14 March 24-28, 2014, Gyeongju, Korea.
Copyright 2014 ACM 978-1-4503-2469-4/14/03 ...$15.00.

for recommendation.

With the thriving development of group based social networks, such as Diaspora, Crabgrass, Lorea and Douban, in this paper, we try to address this issue from a social science perspective. Instead of adding noise or using cryptography, we find that it is possible to give reasonably accurate recommendations based on groups while maintaining privacy from the service provider. Differential privacy [8] essentially captures the risk to one's privacy incurred by participating in a database. It is originally defined for the randomized algorithms using noise to obscure the appearance of individuals. Our approach has some similarities with differential privacy, in the sense that the group replaces the role of the randomized mechanism, and the group size is analogous to noise level. Individual data thus are protected by grouopwise preference aggregation.

We propose a framework for using groups as a natural middleware to recommend products to users. The idea of using groups as a natural protective mechanism is inspired by the French philosopher Gilbert Simondon [23]. An intriguing and interesting aspect of Simondon's theory of systems and technical objects is the idea of adopting an "associated milieu" into the operation of the system. This associated milieu can be natural resources. For example, Simondon spoke of the Guimbal turbine, which used oil to lubricate the engine and at the same time isolate it from water to solve the problem of loss of energy and overheating; it can then also integrate a river as the cooling agent of a turbine [23]. The river here is the associated milieu for the technical system; it is part of the system rather than simply the environment. Groups for us serve a similar function as an associated milieu that contributes to the preservation of individual privacy, while still supporting the functioning of the social network. The focus of our work is to protect users from unreliable service providers, and to mitigate users' fear of potential intrusions of privacy by keeping a certain amount of anonymity. We design a simple distributed protocol to preserve users' privacy through a peer-to-peer preference exchange process. In this process, neither a third party nor cryptography is needed. The service provider only receives mixed preferences for the purpose of preference aggregation. Although data uploaded by individuals might not be k-identical as in a k-anonymity dataset [25], the origins cannot be identified by the service provider. We evaluate the privacy by the effective size of the anonymity set [17], which is a generalized concept of k-anonymity [25]. After group opinion is aggregated, we construct a recommendation graph and use a random walk based method to make recommendations. The stable distribution resulting from a random walk on the graph is interpreted as a ranking of nodes for the purpose of prediction and recommendation. Personalized recommendation is only performed locally so that no private information is revealed to the service provider. We evaluate the performance of the proposed algorithm using the MovieLens dataset, and we compare the results with recommendation algorithms designed for individual users.

A summary of the contributions of this paper is as follows: (1) We propose a recommender system using groups as a natural protective mechanism for privacy preservation. To the best of our knowledge, this is the first work to incorporate group-based social networks in recommender systems for the purpose of protecting users' privacy. (2) A distributed peer-to-peer preference exchange protocol is designed to guarantee anonymity and privacy, which does not require a third party nor cryptography. We use a random walk model to analyze the evolution of effective size of the anonymity set with time. (3) We introduce a random walk based hybrid collaborative filtering graph model that incorporates group based social network information for recommendations. Experiments are designed on the MovieLens dataset to evaluate the performance of the proposed recommender system.

The remainder of the paper is organized as follows. We formulate the recommendation problem in Section 2. We then introduce the group-based recommender system in Section 3. The performance of the proposed framework is evaluated in Section 4, followed by conclusions in Section 5.

2. PROBLEM STATEMENT

In a typical setting, there is a list of m users $\mathcal{U} = \{u_1, u_2, ..., u_m\}$, and a list of n items $\mathcal{I} = \{i_1, i_2, ..., i_n\}$. Each user u_j has a list of items I_{u_j}, which the user has rated or from which user preferences can be inferred. The ratings can either be explicit, for example, on a 1-5 scale as in Netflix, or implicit such as purchases or clicks. This information is stored locally. In a group-based social network, the basic atoms are groups instead of individuals. $\mathcal{G} = \{g_1, g_2, ..., g_k\}$ is a list of k groups. $\mathcal{S} = \{\mathcal{G}, \mathcal{E}_s\}$ is a group-based social network, containing social network information, represented by an undirected or directed graph. $\mathcal{G}$ is a set of nodes and $\mathcal{E}_s$ is a set of edges. For all u, v, $(u, v) \in \mathcal{E}_s$ if v is an associated group of u. Let $\mathcal{T} = \{t_1, t_2, ..., t_y\}$ be a set of tagging information for the items. For example, for movies, $\mathcal{T}$ can be genre, main actor, release date, etc. $T_i \in \{0, 1\}^y$ denotes the features of item i, where y is the total number of tags. We want to make privacy preserving recommendations to users by using groups as natural middleware while no individual preference information is revealed to the central server.

3. GROUP-BASED PRIVACY PRESERVING RECOMMENDER SYSTEM

The structure of the recommender system is as follows:

- **Module 1**: Peer-to-peer preference exchange. Users exchange preference information with other group members in a distributed manner. Only the exchanged information is then uploaded to the central node, thus the individual preferences are kept private.

- **Module 2**: Intra-group preference aggregation. The central server aggregates group preferences to minimize the disagreement heuristically. The group preference will serve as an input for inter-group recommendation and prediction.

- **Module 3**: Inter-group recommendation. A recommendation graph is constructed. A random walk based algorithm is performed for recommendations.

- **Module 4**: Local recommendation personalization. The top k recommendations are returned to group members. Items that have been rated by the user are removed from the recommendation list.

In the rest of this section, we describe and analyze the system in detail.

3.1 Peer-to-peer Preference Exchange

Preference exchange is a process to mix individual preferences so that no full rating profile is collected by the recommendation service provider. Some of the benefits of our preference exchange scheme could be obtained by anonymous communications such as *The Onion Router* [18]. Users could use persistent pseudo-identities and make anonymous ratings, either directly on the central server or let a trustful third party collect this information. However, pseudo-identities still expose users to privacy risks unless the user data is further protected [6] (e.g. Netflix Prize lawsuit due to privacy concerns). Multi-party computation were introduced in [1, 6], but either requiring heavy computation by end users or a third party is introduced. Our proposed peer-to-peer preference exchange procedure lets users exchange information within the group in a distributed manner. Only the mixed preferences are sent to the central server. In a group based social network, such as Douban, group members are maintained by group masters, thus we assume that users within the group are trustful and uncorrupted. Otherwise, techniques of fake accounts and malicious users detection in social networks can be used [24, 29]. Note that the proposed P2P procedure also protects users preference information among peers, since this is beyond the scope of this work, we do not measure the privacy guarantee among users quantitatively.

In the rest of Section 3.1, we describe our peer-to-peer preference exchange scheme in detail and analytically give the privacy guarantee towards the service provider.

3.1.1 Pairwise Comparison Matrix

Before sending preference information to the server, group users exchange information with other group members distributedly. Users then upload the mixed information. Suppose every user has a partial ranking on $\mathcal{I}$. Each user keeps an $n \times n$ pairwise comparison matrix M locally. $M_{xy}^{(u)} = 1$ if user u considers x is better than y; $M_{xy}^{(u)} = 0$ if otherwise, including when no comparison is made between x and y or they are equally liked. When the preference information is p-rating records, i.e. users rate products by the scale of 1 to p, we can transform p-rating history into a partial rank, which naturally normalized the individual ratings. For example, user A who gives ratings 7,8,9 to items a,b,c has the same pairwise comparison matrix to user B who rates a, b, c as 1,2,3 respectively. Let $r_x^{(u)}$ denote the rating of user u on item x.

- If $r_x^{(u)} > r_y^{(u)}$, $M_{xy}^{(u)} = 1$, and $M_{yx}^{(u)} = 0$.
- If $r_x^{(u)} = r_y^{(u)}$, $M_{xy}^{(u)} = 0$, and $M_{yx}^{(u)} = 0$.

3.1.2 Pre-exchange Preparation

Although our focus is to prevent the central server from collecting individual preference, the proposed P2P preference exchange scheme also protects users preference information from other group members. Before the preference exchange starts, each user u randomly chooses p pairwise comparison pairs x, y with $M_{xy}^{(u)} = M_{yx}^{(u)} = 0$, and changes it to $M_{xy}^{(u)} = M_{yx}^{(u)} = 1$, where

$$p = \frac{1}{2}\left(\frac{1}{2}n(n-1) - \sum_{i,j} \mathbf{1}_{\{M_{ij}^{(u)} + M_{ji}^{(u)} = 1\}}\right), \qquad (1)$$

i.e. after inserting some 1s in the pairwise comparison matrix, there are an equal number of 0s and 1s among all non-diagonal entries in the matrix (diagonal entries of the matrix are always 0).

3.1.3 Preference Exchange Rules

Although in a group-based social network, a user can belong to multiple groups, in the recommender system, each user only subscribes to one group for recommendations (If assigning users to multiple groups for recommendations, trivial changes are needed, e.g. preference aggregation on the recommendation results from multiple groups). Consider a group g_i of N members. Group members form a network of N nodes, labeled 1 through N, which form a complete graph. As in some distributed systems [4], each node has a clock which ticks according to a rate 1 exponential distribution. In addition, a synchronized clock is also present at each node.

The preference exchange phase is a process to mix individual preferences so that users do not upload anyone's full rating profile but the mixed preference of the group. The only requirement for the preference exchange is sum conservation. When a user u's local Poisson clock ticks, u randomly picks another user v in the same group, and randomly picks a non-diagonal entry in the pairwise comparison matrix M_{xy} to exchange the corresponding pairwise comparison matrix entry with v.

This phase ends at synchronized time $t = T_{th}$. All nodes then check all pairwise comparisons: if $M_{xy} = M_{yx} = 1$, reset both entries to be 0, i.e. make $M_{xy} = M_{yx} = 0$. Then upload their current preference information to the central server. Because the information uploaded is a mixed preference, individual preference information is not provided and user privacy is protected.

Remark: Note that in the pre-exchange stage, changing pairwise comparison entries from 0 to 1 does not change the individual preference profile, but only to protect user's privacy from revealing to peers in the preference exchange process.

3.1.4 Anonymity Analysis

Definition 1. *Anonymity is the state of being not identifiable within a set of subjects, which is called the anonymity set [17].*

One popular measurement of anonymity is the notion of an *anonymity set*, which was introduced for the dining cryptographers problem [7]. However, a rating does not necessarily arise with equal probability from each of the group members, and so the size of the group is not necessarily a good indicator of anonymity. Instead, we adopt an information theoretic metric for anonymity proposed in [20]:

Definition 2. *Define the effective size $\mathcal{A}$ of an anonymity probability distribution as,*

$$\mathcal{A} = 2^{\sum_{u \in g_i} -p_u \log_2 p_u} \qquad (2)$$

where p_u is the probability that a rating record is from user u. Note that the exponent is the entropy of the distribution p_u.

In order to find the probability distribution of a certain rating record, we first analyze the random process of preference exchange. Because of the superposition property of the

exponential distribution, the setup is equivalent to a single global clock with a rate N exponential distribution ticking at times $\{Z_k\}_{k\geq 0}$. The communication and exchange of preferences occurs only at $\{Z_k\}_{k\geq 0}$.

Theorem 1. *The effective size of the anonymity set of any preference record $\mathcal{A}$ approaches the group size N asymptotically, i.e.*

$$\lim_{t\to\infty} \mathcal{A}(t) = N. \tag{3}$$

Proof. In this random process, there are two sources stimulating the random walk from i to j, $\forall (i,j) \in \mathcal{E}$: one is the clock of the node i, $P^1_{ij} = P^N_{ij}$; the other one is the clock of its neighbor j, $P^2_{ij} = P^N_{ji}$. Thus $P_{ij} = P^1_{ij} + P^2_{ij}$, i.e., each rating record α in a node takes a *biased random walk* on a complete graph, with marginal transition matrix $P = (P_{ij})$:

- $P_{ii} := 1 - \frac{2}{N}\frac{1}{n'}$ for $\forall i \in \mathcal{V}$,
- $P_{ij} := \frac{1}{n'}\frac{1}{N}\frac{2}{N-1}$ for $i \neq j$,

where n' is the number of entries exchanged in the pairwise comparison matrix, i.e., $n' = n(n-1)$, n is the number of items, and N is the number of members in the group.

Hence at time t, the probability distribution $\mathbf{P}_t(i)$ of a certain rating record α starting from node i is $\mathbf{P}_t(i) = P^t \cdot e_i$ P is a symmetric stochastic matrix,

$$P = \begin{pmatrix} 1-\frac{2}{N}\frac{1}{n'} & \frac{1}{n'}\frac{1}{N}\frac{2}{N-1} & \cdots & \frac{1}{n'}\frac{1}{N}\frac{2}{N-1} \\ \frac{1}{n'}\frac{1}{N}\frac{2}{N-1} & 1-\frac{2}{N}\frac{1}{n'} & \cdots & \frac{1}{n'}\frac{1}{N}\frac{2}{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n'}\frac{1}{N}\frac{2}{N-1} & \frac{1}{n'}\frac{1}{N}\frac{2}{N-1} & \cdots & 1-\frac{2}{N}\frac{1}{n'} \end{pmatrix}, \tag{4}$$

with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. It is a basic property of eigenvalues that the sum of all eigenvalues, including multiplicities, is equal to the trace of the matrix. It is easy to check that $\lambda_1 = 1$, and $\lambda_2 = \cdots = \lambda_N = 1 - \frac{2}{n'(N-1)}$.

We can express P as $P = \sum_{i=1}^N \lambda_i \mathbf{v}_i^T \mathbf{v}_i$, where the row eigenvectors $\mathbf{v}_i$ are unitary and orthogonal. Specifically, $\mathbf{v}_1 = (\frac{1}{\sqrt{N}}, ..., \frac{1}{\sqrt{N}})$. and $P^t = \sum_{i=1}^N \lambda_i^t \mathbf{v}_i^T \mathbf{v}_i$.

Notice that $\lambda_1 \mathbf{v}_1^T \mathbf{v}_1 = \lambda_1^k \mathbf{v}_1^T \mathbf{v}_1 = \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Hence $P = \frac{1}{N}\mathbf{1}\mathbf{1}^T + \sum_{i=2}^N \lambda_i \mathbf{v}_i^T \mathbf{v}_i$. We thus have

$$P^t = \frac{1}{N}\mathbf{1}\mathbf{1}^T + \left(1 - \frac{2}{n'(N-1)}\right)^{t-1} \cdot \begin{pmatrix} 1-\frac{2}{N}\frac{1}{n'}-\frac{1}{N} & \frac{1}{n'}\frac{1}{N}\frac{2}{N-1}-\frac{1}{N} & \cdots & \frac{1}{n'}\frac{1}{N}\frac{2}{N-1}-\frac{1}{N} \\ \frac{1}{n'}\frac{1}{N}\frac{2}{N-1}-\frac{1}{N} & 1-\frac{2}{N}\frac{1}{n'}-\frac{1}{N} & \cdots & \frac{1}{n'}\frac{1}{N}\frac{2}{N-1}-\frac{1}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n'}\frac{1}{N}\frac{2}{N-1}-\frac{1}{N} & \frac{1}{n'}\frac{1}{N}\frac{2}{N-1}-\frac{1}{N} & \cdots & 1-\frac{2}{N}\frac{1}{n'}-\frac{1}{N} \end{pmatrix} \cdot \tag{5}$$

As $t \to \infty$, each rating record α shows up at each node with equal probability, i.e. $\lim_{t\to\infty} \mathbf{P}_t(i) = \frac{1}{N}\mathbf{1}$, for $\forall i \in \{1, 2, ..., N\}$. Then the effective size $\mathcal{A}$ of the anonymity distribution for α is $\mathcal{A}(t) = 2^{-\sum_{u\in g_i} p_u(t)\log_2(p_u(t))}$, where $p_u(t)$ is the u^{th} element in $\mathbf{P}_t(i)$.

Hence $\lim_{t\to\infty} \mathcal{A}(t) = N$. □

3.2 Intra-group preference aggregation

Suppose every member has a preference profile π_i (full ranking or partial ranking). In the recommender system, we focus on the top-k rank π^k, which is a partial rank consisting of the k most popular alternatives. One way to define top-k rank is that a partial rank contains k items which minimizes the disagreement with all individual user's preferences, as explicitly formulated below:

$$\underset{\pi^k}{\text{minimize}} \quad \sum_{i=1}^{|g_j|} K(\pi^k, \pi_i) \tag{6}$$

$K(\pi^k, \pi_i)$ is the *Kendall tau distance* [12], defined by the number of disagreement of pairwise comparisons between two (partial) ranks. More specifically,

$$\begin{aligned} K(\pi_1, \pi_2) &= |\{(i,j) : i < j, (\pi_1(i) < \pi_1(j) \wedge \pi_2(i) > \pi_2(j)) \\ &\vee \quad (\pi_1(i) > \pi_1(j) \wedge \pi_2(i) < \pi_2(j))\}| \end{aligned} \tag{7}$$

If k is the size of the items, i.e. $k = n$ and π^k satisfies (6), π^k is called a *Kemeny ranking* [28]. For example, suppose $\pi_1 = \{1, 2, 3\}$, $\pi_2 = \{2, 1, 3\}$, $\pi_3 = \{3, 2, 1\}$, $K(\pi_1, \pi_2) = 1$, $K(\pi_1, \pi_3) = 2$, and the Kemeny Ranking is $\pi^3 = \{1, 2, 3\}$. When the size of items is large, computation becomes expensive. Several heuristic algorithms are available, e.g. the methods proposed in [21].

3.3 Inter-group Recommendation

Intra-group preference aggregation described above gathers existing preference information from group members. However, it is desirable to recommend new items that have similar features but that have not yet been rated by group members. Thanks to the "homophily principle"[11], a group preference can serve as a natural middleware to help make recommendation decisions while protecting the privacy of users, with the absence of individual preference records.

An intuitive approach for recommendation is collaborative filtering (CF) [2, 3, 27]. It uses the known preferences of users to make recommendations or predictions to a target user. Weighted sum is typically used to make predictions. However, traditional collaborative filtering methods are challenged by problems such as *cold start* and *data sparsity*. In the case of a group based recommender system, these problems are inevitable, since groups in a social network already form natural clusters. Hence, there may not be many co-rated items between different groups for the Pearson Correlation computation.

In order to overcome the disadvantages of collaborative filtering, we propose a random walk based inter-group recommender system, which is an extension of our previous work in [22]. Our model incorporates content information of items and social information of groups together as group preference information. We create a recommendation graph, as shown in Fig. 1, consisting of items, groups, and item genres as nodes. Similar to PageRank [5], the stable distribution resulting from a random walk on the recommendation graph is interpreted as a ranking of the nodes for the purpose of recommendation and prediction. We describe how to construct this recommendation graph and represent the flow on the graph in the rest of this section.

3.3.1 Graph settings

Let $G = \{\mathcal{V}, \mathcal{E}\}$ be a graph model for a recommender system, where $\mathcal{V} := \mathcal{G} \cup \mathcal{I} \cup \mathcal{T}$. The nodes of the graph consist of groups, items and item information. For $v_i, v_j \in \mathcal{V}$, $(v_i, v_j) \in \mathcal{E}$ if and only if there is an edge from v_i to

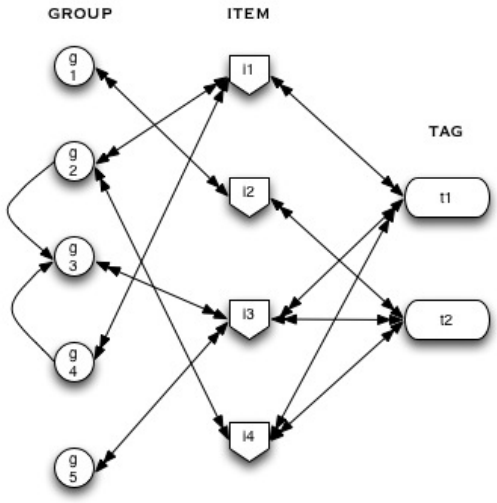


Figure 1: Example of a recommendation graph for inter-group recommendations.

v_j, which is determined as given below. The weights are specified in the next subsection.

- For $g \in \mathcal{G}, i \in \mathcal{I}$, $(g,i) \in \mathcal{E}$ and $(i,g) \in \mathcal{E}$ if and only if $i \in \pi^k(g)$. i.e., an item i and a group u are connected with weights w_{gi} and w_{ig} if i is in g's top-k list.
- For $i \in \mathcal{I}, t \in \mathcal{T}$, $(i,t) \in \mathcal{E}$ and $(t,i) \in \mathcal{E}$ if and only if $T_i^{(t)} \neq 0$. i.e., an item i and tag t are connected with weights w_{it} and w_{ti} if i is tagged by t.
- For $g_1, g_2 \in \mathcal{G}$, $(g_1, g_2) \in \mathcal{E}$ with weight $w_{g_1 g_2}$ if and only if g_1, g_2 are associated groups, i.e. $(g_1, g_2) \in \mathcal{E}_s$, as mentioned in Section 2.

3.3.2 Edge weight assignment

The main part of our rank graph is the collaborative filtering graph, which includes the group nodes, item nodes, and the edges between them. One way to assign weights on the collaborative filtering graph is by setting

$$w_{gi} = w_{ig} = \frac{k + 1 - \pi_g^k(i)}{k} w_{\max}, \tag{8}$$

where $\pi_g^k(i)$ is the rank of item i in the top-k item rank list of group g, and $w_{\max}$ is the max weight assigned on the graph. Let $\pi_g^k(i) = k+1$ if $i \notin \pi_g^k$. Note that a larger edge weight indicates greater chance that the random walk passes through that edge. An item i with better rank in $\pi_g^k(i)$ results in larger weights on edges involving i.

For the extended graph, i.e. nodes and edges containing item content, group social network information, etc., we simply assign an edge weight of 1 if an edge is present.

3.3.3 Rank Score Computation

For the recommendation graph $G = \{\mathcal{V}, \mathcal{E}\}$. Let $v = |\mathcal{V}|$ denote the number of nodes on the graph. θ is a $1 \times v$ *customized probability vector*.

$$\theta = e_g, \tag{9}$$

where $e_1, e_2, ..., e_v$ are the standard basis of row vectors. β is a *damping factor*. With probability $1 - \beta$, the random walk is teleported back to node g. The rank score s satisfies the following equation:

$$s = \beta s W + (1 - \beta)\theta, \tag{10}$$

where W is the weighted transition matrix with $W_{ij} = P_{ji}$.

So we have,

$$s = s\big(\beta W + (1-\beta)\theta \mathbf{1}^T\big) := sM \tag{11}$$

Hence the rank score is the *principal eigenvector* of M, which can be computed by iterations fast and easily via Algorithm 1.

$s_j^{(0)} \leftarrow \frac{1}{v}$ for all j;
$t = 1$;
while $|s^{(t)} - s^{(t-1)}| < \epsilon$ **do**
  for $j = 1$ *to* v **do**
    $s_j^{(t)} = \sum_{i=1}^{v} \beta W_{ij} s_i^{(t-1)} + (1-\beta)\theta_j$;
  end
  $t \leftarrow t + 1$;
end

Algorithm 1: Iterative computation of rank score

The rank score s can be interpreted as the importance of other nodes to the target group g. It is easy to see that we can increase the rank score by shortening the distance, adding more paths, or increasing the weight on the path to g. These are desirable properties in a recommender system. For example, even if item i is not directly connected with g, but it is in a category to which many of g's top-k items belong, then i is very likely to have a high rank score. Or if group g and g' have many overlapping top-k items, g' will have high rank, so we can use g''s top-k list to make recommendations and predictions for g.

3.3.4 Recommendations

Direct Method: Solving Equation (10) iteratively, we obtain a rank score for all nodes of the recommendation graph G. Since the rank score represents the importance to the target group, we can then separate and sort them according to the categories, i.e. groups $\mathcal{G}$, items $\mathcal{I}$, tags $\mathcal{T}$, etc. Sorted items form a recommendation list to the target group g, and we can compute the recommendation for every group.

User-based Prediction: For items above the group popularity threshold, we simply take the average rating of group members as the rating prediction. For other items, we can use rank score as an influence measure to make predictions, which is similar to memory-based collaborative filtering, using *Pearson Correlation* [19] as a similarity measure between users and items. Given the rank score of the group set $\mathcal{G}$, we take the weighted sum of the groups' ratings on item i as a prediction for the target group g, as shown below:

$$\hat{r}_{gi}^{user} = \frac{\sum_{x \in G_i} s_x (\bar{r}_{xi} - \bar{r}_x)}{\sum_{x \in G_i} s_x} + \bar{r}_x. \tag{12}$$

G_i is the set of groups for which item i is above the popularity threshold. s_x is the target group's personalized rank score of group x.

Item-based Prediction: As above, in order to perform an item-based recommendation, we can use the rank score of item set $\mathcal{I}$ as weight to predict the rating of the item i

Table 1: Average percentile results obtained by 5-fold cross-validation for recommendation.

Methods	Percentile
L+	0.1157
ItemRank	0.1150
Personal Recommendation	**0.0790**
Group by Gender	0.1110
Group by Age	0.1066
Group by Occupation	**0.1060**
Random 2 Groups	0.1172
Random 5 Groups	0.1149
Random 21 Groups	0.1104

for the target group g, if the popularity of the item is below the threshold. Specifically,

$$\hat{r}_{gi}^{item} = \frac{\sum_{j \in I_g} s_j r_{gj}}{\sum_{j \in I_g} s_j}. \quad (13)$$

In Equation (13), we use u's rating on similar items to predict the rating on i. s_j is the target group's personalized rank score of item j.

After a recommendation is made, results are returned to individual users. Items that have been rated by the user, which are stored locally, are then removed from the recommendation list.

4. EXPERIMENTS AND EVALUATION

4.1 Dataset

In order to evaluate the performance of the proposed algorithm, we run experiments on the MovieLens dataset, which is a widely used benchmark for recommender systems. The MovieLens dataset consists of 1,682 movies and 943 users. Movies are labeled by 19 genres. User profile information such as age, gender, and occupation is also available. In order to evaluate the group-based recommender system, we take user profile categories provided in the dataset as groups. In the experiments, we group users in three different ways, namely, gender, age, and occupation. Detailed group category distribution is as follows:

- **Gender**: male (71.16%) and female (28.84%).
- **Age**: below 21, 21 to 30, 31 to 40, 41 to 50, above 50, indexed from 1 to 5, respectively, as shown in Fig.2a.
- **Occupation**: administrator, artist, doctor, educator, engineer, entertainment, executive, healthcare, homemaker, lawyer, librarian, marketing, none, other, programmer, retired, salesman, scientist, student, technician and writer, indexed from 1 to 21, respectively, as shown in Fig.2b.

Remarks: To the best knowledge of the authors, there are no datasets with both group-based social network and user rating information available. We thus choose MovieLens datasets and take the user personal information to categorize users. It may not be the best way to group users, but the purpose of our experiments is to show that the recommendation provided by aggregated preference performs reasonably well compared with personalized recommendation, which is a good compromise between privacy and accuracy.

4.2 Experimental Methodology and Results

We evaluate our results with two popular evaluation metrics for top-k recommendations: percentile and TOPK.

Percentile: The individual percentile score is simply the average position (in percentage) that an item in the test set occupies in the recommendation list. For example, if four items are ranked 1st, 9th, 10th and 20th in a recommendation list consisting of 100 items, with individual percentile scores of 0.01, 0.09, 0.10 and 0.20. The average percentile of the system is 0.1. A lower percentile indicates a better prediction.

TOPK: Given a recommendation test, we consider any item in the top-k recommendations that matches any item in the test set as a "hit", as in [26].

$$TOPK(k) = \frac{\#\text{hits of top-k}}{T}, \quad (14)$$

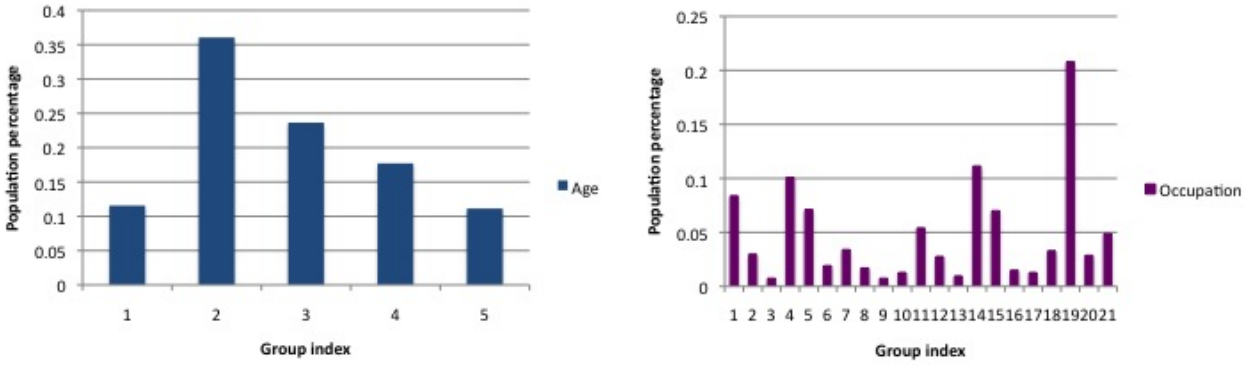
where T is the size of test set. A higher TOPK value indicates a better prediction. Note that TOPK is proportional to precision, for fixed data and k. In other words, when comparing different recommender systems on fixed data and fixed k, the one with the larger TOPK value also has the larger precision.

In this experiment, all items in the test set T are rated 5 (highest rating) by users, thus we can consider them as relevant items for recommendation. The recommendation list has a length of 900 items. The top-500 movies in the aggregated group preference list are used to construct the recommendation graph. Note that the popularity threshold of the recommender system can be decided by users, since different groups may have a different requirement for popularity. In our experiment, we set the popularity threshold at 0.01. We compare the proposed method with two state-of-art personalized recommender systems: L+ [9] and ItemRank [10]. L+ suggested a dissimilarity measure between nodes of a graph, the expected commute time between two nodes, which the authors applied to recommendation [9]. Specifically, they constructed a non-directed bipartite graph where users and movies form the nodes. A link is placed between a user and movie if the user watched that movie. Movies are then ranked in ascending order according to the average commute time to the target node. ItemRank built the recommendation graph by only using movies as nodes. In [10], two nodes are connected if at least one user rated both nodes. The weight of the edge is set as the number of users who rated both of the nodes. A random-walk based algorithm is then used to rank items according to the target user's preference record. In order to see how much information is lost by grouping users, we also compare the proposed privacy-preserving recommendation algorithm with a recommendation graph of similar structure, but with all the individual rating information, where nodes of the recommendation graph are formed by users, items, user social profile information (gender, age and occupation). The weight of an edge between users and items is given by

$$w_{ui} = w_{iu} = \exp\left(\frac{r_{ui} - \bar{r}_u}{\sqrt{\sum_{i \in I_u}(r_{ui} - \bar{r}_u)^2}}\right), \quad (15)$$

$$\bar{r}_u := \frac{\sum_{i \in I_u} r_{ui}}{|I_u|}. \quad (16)$$

where I_u denotes the set of items which user u has rated.



(a) Percentage of the population of 5 different age categories.(b) Percentage of the population of 21 different occupation categories.

Figure 2: The group distribution of MovieLens datasets.

Table 2: Average TOPK results obtained by 5-fold cross-validation for recommendation.

Methods	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30	Top-35	Top-40	Top-45	Top-50
L+	**0.1569**	0.2335	0.2776	0.3166	0.3519	0.3774	0.4118	0.4350	0.4591	0.4814
ItemRank	0.1690	0.2330	0.2851	0.3352	0.3788	0.4076	0.4364	0.4582	0.4837	0.5037
Personal Recommendation	**0.2187**	**0.3027**	**0.3477**	**0.4162**	**0.4596**	**0.4912**	**0.5139**	**0.5459**	**0.5710**	**0.5910**
Group by Gender	0.1038	0.1660	0.2435	0.3130	0.3658	0.4077	0.4417	0.4702	0.4894	0.5095
Group by Age	0.1260	0.2275	0.2861	0.3334	0.3771	0.4110	0.4422	0.4689	0.4920	0.5140
Group by Occupation	0.1490	**0.2399**	**0.3048**	**0.3479**	**0.3856**	**0.4210**	**0.4485**	**0.4729**	**0.4958**	**0.5184**

Note that a larger edge weight indicates more chance that the random walk passes through that edge. If user u's rating on item i r_{ui} is lower than the average rating $\bar{r}_u$, w_{ui} and w_{iu} are less than 1; otherwise are greater than 1. The assignment of weights do not depend on the variance of the user's ratings.

Experimental results of cross-validation on percentile scores of the MovieLens dataset are shown in Table 1. We create five training/testing splits. Although it does not utilize knowledge of individual's preference information, the proposed group-based privacy preserving recommendation algorithm still has a better performance than L+ and ItemRank, which are two state-of-art personalized recommendation methods. And as expected, due to the absence of personal rating information, the performance of the proposed group method is inferior to *personal recommendation*, i.e., recommendations with individual rating information. It is also worth noting that among all three different ways of grouping users, grouping by occupation outperforms the other two grouping methods, which shows the promise of group-based recommender system with finer groups. Moreover, in order to evaluate the effectiveness of groups in the dataset, we did contrast experiments on random groups, which are users divided randomly into 2, 5, 21 groups to compare with gender, age and occupation groups. Experimental results show that the natural groups outperform the random groups, as shown in Table 1.

We also perform 5-fold cross-validation experiments for TOPK values, as shown in Table 2. In real settings, a user is unlikely to browse a very long recommendation list. Thus, we only test the top-5 to top-50 TOPK values. As introduced in Section 4.2, a TOPK value of k is the probability that an item in the test set hits the top-k items recommended by the system. A higher TOPK value means a higher chance that items in the test set appear in the top-k list. Since these items all have the highest ratings, a higher TOPK value indicates better performance of the recommendation algorithm. In Table 2, *personal recommendation*, our proposed algorithm with individual preference information, trading privacy for quality, has the best performance. Otherwise, L+ has better performance on top-5 TOPK, and the recommender system based on occupation groups outperforms gender and age groups, and also has a higher TOPK value than L+ and ItemRank for top-10 to top-50 recommendations.

5. CONCLUSIONS

In this paper, we present a framework for group-based privacy preserving recommender systems. We introduce the novel idea of using groups as a natural protective mechanism to preserve individual users' private preference data from the central service provider. A distributed peer-to-peer preference exchange process is designed to provide anonymity of group members. We also introduce a hybrid recommendation model based on random walks. It incorporates item content and group social information to make recommendations for groups. Personalized recommendations are made locally to group members, so that no user preference profile is leaked to the service provider. Experimental results show that the proposed algorithm outperforms the baseline algorithms L+ and ItemRank, despite the absence of personal

preference information. By aggregating group preferences and then making recommendations, we can obtain a reasonable compromise between privacy and accuracy.

6. ACKNOWLEDGMENTS

This research was supported in part by the Center for Science of Information (CSoI), a National Science Foundation (NSF) Science and Technology Center, under grant agreement CCF-0939370, by NSF under the grant CCF-1116013, and by a research grant from Deutsche Telekom AG.

7. REFERENCES

[1] E. Aïmeur, G. Brassard, J. M. Fernandez, and F. S. M. Onana. Alambic: a privacy-preserving recommender system for electronic commerce. *International Journal of Information Security*, 7(5):307–334, 2008.

[2] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126. ACM, 2010.

[3] S. Berkovsky and J. Freyne. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 111–118. ACM, 2010.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, pages 2508–2530, 2006.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*, 1998.

[6] J. Canny. Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 238–245, New York, NY, USA, 2002. ACM.

[7] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of cryptology*, 1(1):65–75, 1988.

[8] C. Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.

[9] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *Knowledge and Data Engineering IEEE Transactions*, 19(3):355–369, March 2007.

[10] M. Gori and A. Pucci. Itemrank: a random-walk based scoring algorithm for recommender engines. *In Proceedings of the 20th international joint conference on Artifical intelligence*, 2007.

[11] J. He and W. W. Chu. *A social network-based recommender system (SNRS)*. Springer, 2010.

[12] M. Kendall. A new measure of rank correlation. *Biometrika*, pages 81–89, 1932.

[13] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[14] A. Machanavajjhala, A. Korolova, and A. D. Sarma. Personalized social recommendations - accurate or private? In *Proceedings of the VLDB Endowment*, 2011.

[15] F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

[16] A. Nandi, A. Aghasaryan, and M. Bouzid. P3: A privacy preserving personalization middleware for recommendation based services. In *4th Hot Topics in Privacy Enhancing Technologies*, 2011.

[17] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity — a proposal for terminology. In *International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability*, 2001.

[18] T. T. Project. *Tor Project: Core People*, Retrieved 17 July 2008.

[19] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 1994.

[20] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *PET'02 Proceedings of the 2nd international conference on Privacy enhancing technologies*, pages 41–53, 2002.

[21] S. Shang, Y. Hui, P. Hui, P. Cuff, and S. Kulkarni. Privacy perserving recommendation system based on groups. *http://arxiv.org/abs/1305.0540*.

[22] S. Shang, S. Kulkarni, P. Cuff, and P. Hui. A random walk based model incorporating social information for recommendations. *2012 IEEE Machine Learning for Signal Processing Workshop (MLSP)*, 2012.

[23] G. Simondon. L'invention dans les techniques. In *Cours et conférences*, 2005.

[24] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.

[25] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[26] K. H. L. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. *Proceedings of the 2008 ACM symposium on Applied computing*, 2008.

[27] S. Vucetic and Z. Obradovic. Collaborative filtering using a regression-based approach. *Knowledge and Information Systems*, 7:1–22, 2005.

[28] H. Young and A. Levenglick. A consistent extension of Condorcet's election principle. *SIAM Journal on Applied Mathematics*, 35(2):285–300, 1978.

[29] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36(4):267–278, 2006.