Personalized Mobile App Recommendation: Reconciling App Functionality and User Privacy Preference

Bin Liu*
Rutgers University
binben.liu@rutgers.edu

Deguang Kong
Samsung Research America
deguang.k@samsung.com

Lei Cen
Purdue University
lcen@purdue.edu

Neil Zhenqiang Gong
UC Berkeley
neilz.gong@berkeley.edu

Hongxia Jin
Samsung Research America
hongxia@acm.org

Hui Xiong
Rutgers University
hxiong@rutgers.edu

ABSTRACT

Recent years have witnessed a rapid adoption of mobile devices and a dramatic proliferation of mobile applications (Apps for brevity). However, the large number of mobile Apps makes it difficult for users to locate relevant Apps. Therefore, recommending Apps becomes an urgent task. Traditional recommendation approaches focus on learning the *interest* of a user and the *functionality* of an item (*e.g.*, an App) from a set of user-item ratings, and they recommend an item to a user if the item's functionality well matches the user's interest. However, Apps could have privileges to access a user's sensitive resources (*e.g.*, contact, message, and location). As a result, a user chooses an App not only because of its functionality, but also because it respects the user's *privacy preference*.

To the best of our knowledge, this paper presents the *first* systematic study on incorporating both interest-functionality interactions and users' privacy preferences to perform personalized App recommendations. Specifically, we first construct a new model to capture the trade-off between functionality and user privacy preference. Then we crawled a real-world dataset (16, 344 users, 6, 157 Apps, and 263, 054 ratings) from Google Play and use it to comprehensively evaluate our model and previous methods. We find that our method consistently and substantially outperforms the state-of-the-art approaches, which implies the importance of user privacy preference on personalized App recommendations. Moreover, we explore the impact of different levels of privacy information on the performances of our method, which gives us insights on what resources are more likely to be treated as private by users and influence users' behaviors at selecting Apps.

*This work was done during an internship at Samsung Research America, San Jose, CA.

WSDM'15, February 2–6, 2015, Shanghai, China.

http://dx.doi.org/10.1145/2684822.2685322.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms

Algorithms, Experimentation

Keywords

Recommender Systems; Mobile Apps; Privacy and Security

1. INTRODUCTION

Mobile devices are becoming more and more popular in the past few years. For instance, it was reported that the smartphone market was bigger than the PC market in 2011 for the first time in history [28]. Thereafter, the smartphone market has continued to increase dramatically, *e.g.*, the smartphones shipped in the third quarter of 2013 increased 44% year-on-year [27]. One of the reasons lies in the fact that users are able to augment the mobile devices' functions via taking advantage of various feature-rich third-party *applications* (or Apps for brevity), which can be easily obtained from centralized markets such as Google Play and App Store. However, the number of Apps has recently increased dramatically, which makes it hard for a user to locate relevant Apps. For instance, as of July 2013, Google Play had over 1 million Apps with over 50 billion cumulative downloads, and the number of Apps has reached over 1.2 million in June 2014 [12]; as the beginning of June 2014, App Store had 1.2 million Apps and a cumulative of 75 billion downloads [4]. Therefore, it is urgent to develop effective *personalized* App recommendation systems.

Conventional recommender systems [1, 18, 25, 19, 8, 13] essentially aim to learn the *interest* of each user and the *functionality* of each item (*e.g.*, an App in our problem), given the list of items used or rated by each user. Then, an item is recommended to a user if the item's functionality matches the user's interest. For instance, matrix-factorization-based approaches [18, 25] model a user's interest as a latent vector and an item's functionality as another latent vector; and an item is recommended to a user if the item's functionality vector is close to the user's interest vector in the latent space. Such *interest-functionality* driven recommendation systems have been successfully used to recommend products in e-commence (*e.g.*, Amazon) [21], movies (*e.g.*, Netflix) [5], musics [3], point-of-interests [29, 22], and used for link prediction and attribute inference [10].

However, these approaches are not appropriate for App recommendations. Specifically, unlike items such as music, movies, and point-of-interests, Apps could have privileges to access the user's personal information such as locations, contacts, and messages. Moreover, users might have different *privacy preferences*, *e.g.*, user A tends to not share contacts with the App while user B tends to not allow the App to access her/his locations. Although an App's functionality may matches a user's interest well, the user could still choose to not install it or dislike it if it does not respect the user's privacy preference. Indeed, according to a recent report [7], 54% of surveyed users have decided not to install Apps that want to access their sensitive personal information and 30% of users have uninstalled at least one App after they realized that the App was collecting unexpected personal information. Therefore, whether a user selects/likes an App is a result of the trade-off between two factors:

(1) the degree of match between the user's interest and the App's functionality, which we call *functionality match*;

(2) the degree to which the App respects the user's privacy preference, which we call *privacy respect*.

However, conventional recommendation approaches do not capture this trade-off, which limits their performances on recommending Apps.

Our work: In this paper, we aim to bridge this gap via incorporating both interest-functionality interactions and users' privacy preferences. Specifically, we first construct a new latent factorization model to capture the trade-off between functionality and user privacy preference. Different users might have different definitions on *private data* and could have different privacy concerns on different operations (*e.g.*, read or write) on the private data. Thus, in our model, we consider three levels of privacy information to characterize users' privacy preferences. Moreover, our model takes the sparse user-app rating matrix and the set of privacy-sensitive privileges (*e.g.*, App's permissions) of each App at a given privacy level as an input, and it automatically learns the interest and privacy preference of each user, and the functionality of each App in the latent space, which are further used to predict users' preferences for new Apps. Then, we crawled a real-world dataset which consists of $16,344$ users, $6,157$ Apps, and $263,054$ rating observations from Google play, and we use the dataset to comprehensively evaluate our method and previous approaches. We find that our method consistently and substantially outperforms the state-of-the-art approaches. Furthermore, we explore the impact of different privacy levels on the performance of our method, and we observe that treating different operations with different privacy concerns achieves better recommendation performances.

Our key contributions are summarized as follows:

- We provide the *first* systematic study on leveraging both interest-functionality expectation and user privacy preference to provide personalized App recommendations.
- We propose a new model to capture the trade-off between functionality and user privacy preference.
- We crawled a real-world dataset from Google Play, and we use it to comprehensively evaluate our approach and state-of-the-art methods and explore the impact of privacy levels on the performance of our method. We find that our method consistently and substantially outperforms state-of-the-art approaches.

Table 1: Six dangerous permissions. They manipulate sensitive information *locations*, *contacts*, and *messages*, respectively.

Permission	Description
ACCESS_FINE_LOCATION	allow App to access precise (*e.g.*, GPS) location
ACCESS_COARSE_LOCATION	allow App to access approximate (*e.g.*, cell towers, Wi-Fi) location
READ_CONTACTS	allow App to read contacts info
WRITE_CONTACTS	allow App to write contacts info
READ_SMS	allow App to read SMS messages
WRITE_SMS	allow App to write SMS messages
SEND_SMS	allow App to send SMS messages

2. PROBLEM FORMALIZATION

We first identify that whether a user adopts an App is a result of the trade-off between the App's functionality and the user's privacy preference. Second, we introduce our defined hierarchy of user privacy concerns. Third, we formally define our *privacy-respect App recommendation* problem.

2.1 Trade-off between Functionality and Privacy

We focus on Android Apps, though our approach is also applicable to other types of Apps. Android system is a permission-based framework. A permission is related to a critical resource (*e.g.*, Internet, contact, and message) on the mobile device, and granting a permission to an App allows the App to either read or write the corresponding resource. Table 1 shows some permission examples and their corresponding descriptions. For instance, giving the permission READ_CONTACTS to an App makes it capable to read the user's contact data.

We identify that whether a user adopts an App is a result of the trade-off between the App's functionality and the user's privacy preference. To achieve the functionality desired by the user, the App might need to manipulate the user's certain type of private data through requesting the corresponding sensitive permissions. For instance, Google Map, a navigation App, requires the user's GPS location data and thus needs the ACCESS_FINE_LOCATION permission. Moreover, the App could also request other sensitive permissions intentionally [26] or unintentionally [9] for non-functionality purposes such as advertisements. For instance, Shekhar *et al.* [26] found that around 25% of Android Apps access users' location data only for advertisements; Felt *et al.* [9] found that around 30% of Android Apps request sensitive permissions that are not used by them at all. A user with low privacy concerns with the requested permissions/resources might sacrifice its privacy for the App's functionality, while a user with high privacy concerns might sacrifice the App's functionality for privacy or might transfer to another App that provides the same functionality but uses less private resources.

2.2 User Privacy Levels

Different users could have different definitions on *private resources* and could have different privacy concerns to different operations (*e.g.*, read or write) on the resources. We define three privacy levels, each of which consists of a set of resources and corresponding operations. A user's privacy

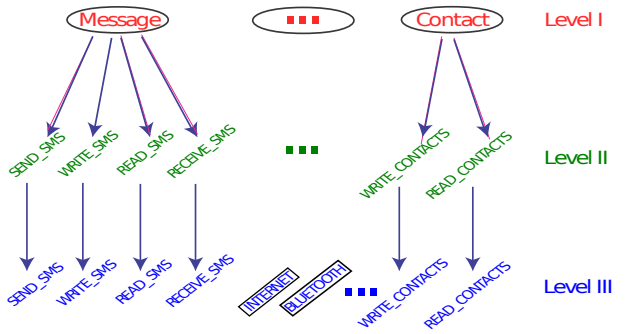


Figure 1: Illustration of the three privacy levels. Level I corresponds to *privacy-sensitive resources*; Level II corresponds to *privacy-sensitive permissions* (refer to Table 2); Level III is a superset of Level II.

preference essentially characterizes the concerns for the operations on the private resources in a given privacy level. Figure 1 illustrates the hierarchy of the three privacy levels.

- **Level I:** This level considers 10 resources (*e.g.*, contact, message, and location) as private. The 10 resources are listed in the first column of Table 2. However, this level does not distinguish the operations that can be applied to the private resources. Thus, this privacy level is represented as a binary vector of the 10 resources. If a user does not concern a certain private resource such as message, the user would accept an App to read, write, or even send messages.
- **Level II:** This level considers the same 10 resources in Level I as private. However, this level explicitly distinguishes different operations that can be applied to the resources. In this level, a user could have a low privacy concerns on reading messages but a high privacy concern on writing messages. This level of privacy can be expressed by the set of Android permissions that are related to the 10 resources. In total, there are 23 such permissions. Level II is more fine-gained than level I, and Table 2 described mappings between level I and level II.
- **Level III:** This level considers all critical resources including the 10 resources in the Level I and II and other resources (*e.g.*, Internet and bluetooth) on a mobile device as private, and it also distinguishes different operations. This level is more complete and more fine-grained than the Level II, and it can be expressed as a binary vector of all dangerous Android permissions. In total, we identified 72 such permissions, which are a superset of level II permissions.

For the same App, users with different privacy levels could behave differently at whether adopting the App or not. In our experiments, we will explore the impact of the three privacy levels on the performance of our method.

2.3 Privacy-respect App Recommendation

We use M and N to denote the number of users and the number of apps, respectively; we denote by the set of users as $U = \{u_1, u_2, \cdots, u_M\}$ and the set of Apps as $V = \{v_1, v_2, ..., v_N\}$. Let S be the set of privacy-sensitive operations or resources at a given privacy level. Depending on the privacy level, S can be the 10 private resources (Level I), the 23 sensitive Android permissions (Level II), or all dangerous Android permissions (Level III). For each App j, we have its privacy-sensitive operations/resources Π_j at a given privacy level, where $\Pi_j \subseteq S$. Π_j can be obtained by App code analysis.

Suppose we are given a sparse matrix of user-App response records (*e.g.*, ratings or likes) and the set of privacy-sensitive operations/resources of each App at a given privacy level, our goal is to recommend most relevant Apps for each user by learning both interest and privacy preference of each user and functionality of each App.

Table 2: *Privacy-sensitive resources* (Level I) vs. corresponding *privacy-sensitive permissions* (Level II).

Privacy-sensitive Resources	Privacy-sensitive Permissions
Contact	READ_CONTACTS
	WRITE_CONTACTS
Message	READ_SMS
	WRITE_SMS
	SEND_SMS
	RECEIVE_SMS
	RECEIVE_MMS
	SEND_RESPOND_VIA_MESSAGE
Location	ACCESS_FINE_LOCATION
	ACCESS_COARSE_LOCATION
Phone_state	MODIFY_PHONE_STATE
	READ_PHONE_STATE
Phone_call	CALL_PHONE
	CALL_PRIVILEGED
Calendar	READ_CALENDAR
	WRITE_CALENDAR
Call_log	READ_CALL_LOG
	WRITE_CALL_LOG
Browser_history	READ_HISTORY_BOOKMARKS
	WRITE_HISTORY_BOOKMARKS
Camera	CAMERA
Audio	RECORD_AUDIO
	MODIFY_AUDIO_SETTINGS

3. PROPOSED METHOD

This section presents our proposed user privacy-respect App recommendation model.

3.1 General Idea

We aim to quantify the trade-off between App's functionality and user privacy preference. Suppose $g_{\text{func},i,j}$ is the *functionality match score* of the interest of user i and functionality of App j and $g_{\text{privacy},i,j}$ is the *privacy respect score* of the privacy preference of user i and the privacy information used by App j.

Modeling functionality match: Following the latent factor models in standard recommendation systems [18, 25], we model a user i's interest as a user latent vector $\mathbf{u}_i^{\text{interest}} \in \mathbb{R}^K$ and an App j's functionality as an App latent vector $\mathbf{v}_j \in \mathbb{R}^K$, where K is the number of latent dimensions of user interests and App functionalities. More specifically, each

Table 3: Mathematical Notations

Symbol	Size	Description
$\mathbf{U}$	$K \times M$	user latent factor
$\mathbf{V}$	$K \times N$	App latent factor
$\mathbf{P}$	$K \times S$	privacy information latent factor
Π_j	$\Pi_j \in S$	privacy information set for App j
y_{ij}	$\mathbb{R}$	user i's rating for App j

element $u_{ik} \in \mathbf{u}_i^{\text{interest}}$ encodes the preference of user i to "preference aspect" k, and each element $v_{ik} \in \mathbf{v}_j$ reflects the aspect affinity of App j to aspect k, where $k = 1, 2, \cdots, K$. Then the functionality match score $g_{\text{func},i,j}$ is modeled as:

$$g_{\text{func,i,j}} = f\left(\mathbf{u}_i^{\text{interest}}, \mathbf{v}_j; \Theta_1\right).$$

Modeling privacy respect: We also adopt a latent factor model to describe user privacy preference and App's private information. This latent factor model assumes that it is possible to group users by a relatively small number of privacy profiles. Specifically, we denote a user i's privacy preference as a latent factor $\mathbf{u}_i^{\text{privacy}} \in \mathbb{R}^K$. Accordingly, we model each privacy information (*i.e.*, a privacy-sensitive resource or permission) in the set of privacy information S at a given privacy level as a privacy latent factor $\mathbf{p}_s \in \mathbb{R}^K$. Note that although different number of latent dimensions can be applied to model functionality factors and privacy factors, we assume they are the same for simplicity. Therefore, we model the privacy respect score as:

$$g_{\text{privacy}} = f\left(\mathbf{u}_i^{\text{privacy}}, \sum_{s \in \Pi_j} \mathbf{p}_s; \Theta_2\right),$$

where Π_j is the set of privacy information associated with the App j.

Trade-off between functionality and privacy: We model a user i's overall preference (denoted as $g_{i,j}$) for an App j as a weighted sum of the functionality match score and the privacy respect score. Specifically, we have:

$$g_{i,j} = g_{\text{func},i,j} + \lambda g_{\text{privacy},i,j}, \tag{1}$$

where λ is used to balance App functionality and user privacy preference.

3.2 Model Specifications

Here we present a detailed model specification. Instead of separately representing user interest and user privacy preference with two latent factors, we amalgamate user interest latent vector and user privacy latent vector as one user profile latent factor $\mathbf{u}_i \in \mathbb{R}^K$, which is a K-dimension vector. This amalgamation can reduce parameters to learn and thus improve computational efficiency.

Each App j is modeled by a functionality latent factor and a privacy latent factor as $\mathbf{v}_j + \lambda \sum_{s \in \Pi_j} \mathbf{p}_s$, where Π_j is the privacy information set for App j. For example, if App j requests three permissions `ACCESS_FINE_LOCATION` (index: 2), `READ_CONTACTS` (index: 4), and `INTERNET` (index: 7), then $\Pi_j = \{2, 4, 7\}$ at the privacy level Level III. The cardinality of the set Π_j is the number of elements in Π_j, i.e., $|\Pi_j| = 3$ in our example. Privacy latent factor representation $\sum_{s \in \Pi_j} \mathbf{p}_s$ provides flexibility for Apps with different number of privacy information.

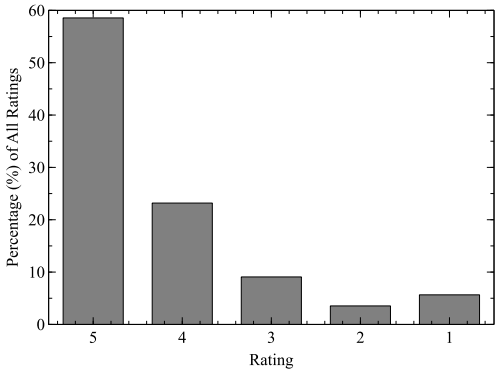


Figure 2: Rating distribution.

Then a user i's preference score for an App j can be represented as

$$x_{ij} = \mathbf{u}_i^T \left(\mathbf{v}_j + \lambda \frac{1}{|\Pi_j|} \sum_{s \in \Pi_j} \mathbf{p}_s\right) \tag{2}$$

where $\frac{1}{|\Pi_j|}$ is placed for each App to adjust the unbalanced number of privacy informations.

To model user profile and App profile, it is practical to formulate the user-App preference score x_{ij} to follow some probability distribution $\Pr(y_{ij}|x_{ij}, \Theta)$, then we can infer the latent factors $\mathbf{u}_i$, $\mathbf{v}_j$, and $\mathbf{p}_s$ through statistical inference methods. One most used probabilistic model, as used in probabilistic matrix factorization (PMF) [25], is to assume $\Pr(y_{ij}|x_{ij}, \Theta)$ as a Gaussian distribution. However, the rating distribution in an App dataset is polarized as shown in Figure 2, which indicates that Gaussian distribution would not be a good choice for our problem. Therefore, instead of using a Gaussian distribution, we adopt a Poisson distribution:

$$\begin{aligned} y_{ij} &\sim \text{Poisson}(x_{ij}) \\ \Pr(y_{ij}|x_{ij}) &= (x_{ij})^{y_{ij}} \frac{\exp\{-x_{ij}\}}{y_{ij}!}. \end{aligned} \tag{3}$$

As noticed by [13, 8, 23], Poisson distribution is a better choice for modeling discrete user-item responses. Firstly, it better captures real user-item response data. By setting non-negative constrains on latent factors, Poisson latent factor model force response variables to be in a wider range than the rating based response. As a result, it can better capture preference order. Secondly, due to the form of Poisson distribution, only the observed part of user-item matrix needs to be iterated during modeling, which provides advantage for the sparsity of user-item matrix in recommendation problems. Therefore, we model user-App preference as:

$$\Pr(y_{ij}|u_i, v_i, p_s) = (x_{ij})^{y_{ij}} \frac{\exp\{-x_{ij}\}}{y_{ij}!}, \tag{4}$$

where $x_{ij} = \mathbf{u}_i^T \left(\mathbf{v}_j + \lambda \frac{1}{|\Pi_j|} \sum_{s \in \Pi_j} \mathbf{p}_s\right)$. Further, u_{ik}, v_{ik}, and p_{sk} can be given Gamma distributions as empirical priors, *i.e.*, the user-App preferences can be modeled as a generative process:

1. For each user i, generate user latent factor:
$$u_{ik} \sim \text{Gamma}(\alpha_U, \beta_U), \tag{5}$$
2. For each App j, generate App functionality latent factor:
$$v_{jk} \sim \text{Gamma}(\alpha_V, \beta_V), \tag{6}$$

3. For each privacy information s, generate privacy latent factor:

$$p_{sk} \sim \text{Gamma}(\alpha_P, \beta_P), \tag{7}$$

4. For each user-App pair $\langle i, j \rangle$, generate Poisson response:

$$\Pr(y_{ij}|u_i, v_i, p_s) = (x_{ij})^{y_{ij}} \frac{\exp\{-x_{ij}\}}{y_{ij}!},$$

where $\Theta = \{\mathbf{U}, \mathbf{V}, \mathbf{P}\}$ are parameters to be estimated, and $\Phi = \{\alpha_U, \beta_U, \alpha_V, \beta_V, \alpha_P, \beta_P\}$ are model hyperparameters.

3.3 Model Estimation

Let $\Pr(\mathbf{U}, \mathbf{V}, \mathbf{P}|\mathbf{Y}, \Phi)$ be the posteriori probability of generation of $\mathbf{U}, \mathbf{V}, \mathbf{P}$, given observations of $\mathbf{Y}$ and prior distribution Φ, according to the maximum a posteriori (MAP) rule, we need to maximize:

$$\begin{aligned} &\max_{\mathbf{U},\mathbf{V},\mathbf{P}} \Pr(\mathbf{U}, \mathbf{V}, \mathbf{P}|\mathbf{Y}, \Phi) \\ &\propto \max_{\mathbf{U},\mathbf{V},\mathbf{P}} \Pr(\mathbf{Y}|\mathbf{U}, \mathbf{V}, \mathbf{P}) \Pr(\mathbf{U}, \mathbf{V}, \mathbf{P}|\Phi) \end{aligned} \tag{8}$$

where $\Pr(\mathbf{u}_i, \mathbf{v}_j, \mathbf{p}_s|\alpha_u, \beta_u, \alpha_v, \beta_v, \alpha_s, \beta_s)$ are the prior distributions for $\mathbf{U}$, $\mathbf{V}$, $\mathbf{P}$ generated from Eqs.(5, 6, 7), and $\Pr(y_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j, \mathbf{p}_s)$ can be computed using Eq.(4).

Following the likelihood principle, we can determine the optimal solution for $\mathbf{U}, \mathbf{V}, \mathbf{P}$ to Eq.(8) by Maximum a Posteriori (MAP) estimation. Specifically, we have:

$$\begin{aligned} \Pr(\mathbf{Y}|\mathbf{U}, \mathbf{V}, \mathbf{P}) &= \prod_{i=1}^{M} \prod_{j=1}^{N} (x_{ij})^{y_{ij}} \exp\{-x_{ij}\} / y_{ij}! \\ \Pr(\mathbf{U}|\alpha_U, \beta_U) &= \prod_{i=1}^{M} \prod_{k=1}^{K} \frac{u_{ik}^{\alpha_U - 1} \exp(-u_{ik}/\beta_U)}{\beta_U^{\alpha_U} \Gamma(\alpha_U)} \\ \Pr(\mathbf{V}|\alpha_V, \beta_V) &= \prod_{j=1}^{N} \prod_{k=1}^{K} \frac{v_{jk}^{\alpha_V - 1} \exp(-v_{jk}/\beta_V)}{\beta_V^{\alpha_V} \Gamma(\alpha_V)} \\ \Pr(\mathbf{P}|\alpha_P, \beta_P) &= \prod_{s=1}^{S} \prod_{k=1}^{K} \frac{p_{sk}^{\alpha_P - 1} \exp(-p_{sk}/\beta_P)}{\beta_P^{\alpha_P} \Gamma(\alpha_P)}, \end{aligned} \tag{9}$$

where

$$x_{ij} = \boldsymbol{u}_i^\top \left(\boldsymbol{v}_j + \lambda \frac{1}{|\Pi_j|} \sum_{s \in \Pi_j} \boldsymbol{p}_s \right)$$

Then the log-likelihood of Eq.(8) is given by

$$\begin{aligned} \mathcal{L} &= \log \Pr(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{F}|\mathbf{Y}, \Phi) \\ &= \sum_{i=1}^{M} \sum_{k=1}^{K} \Big((\alpha_U - 1) \ln u_{ik} - u_{ik}/\beta_U \Big) \\ &+ \sum_{j=1}^{N} \sum_{k=1}^{K} \Big((\alpha_V - 1) \ln v_{jk} - v_{jk}/\beta_V \Big) \\ &+ \sum_{s=1}^{S} \sum_{k=1}^{K} \Big((\alpha_P - 1) \ln p_{sk} - p_{sk}/\beta_P \Big) \\ &+ \sum_{i=1}^{M} \sum_{j=1}^{N} (y_{ij} \ln x_{ij} - x_{ij}) + \text{const.} \end{aligned} \tag{10}$$

Thus maximization of Eq.(8) w.r.t $\mathbf{U}, \mathbf{V}, \mathbf{P}$ is equivalent to maximization of Eq.(10). To control the model complexity, we further add a penalty term, then the objective function becomes

$$\mathcal{Q} = \mathcal{L} - \frac{\eta}{2} \left(||\mathbf{U}||_{\text{F}}^2 + ||\mathbf{V}||_{\text{F}}^2 + ||\mathbf{F}||_{\text{F}}^2 \right) \tag{11}$$

Taking derivatives on $\mathcal{Q}$ with respect to u_{ik}, v_{jk}, and p_{sk}, we have

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial u_{ik}} &= \frac{\alpha_U - 1}{u_{ik}} - \frac{1}{\beta_U} - \eta \times u_{ik} \\ &\quad + \sum_{j=1}^{N} \left(\frac{y_{ij}}{x_{ij}} - 1 \right) \left(v_{jk} + \frac{\lambda}{|\Pi_j|} \sum_{s \in \Pi_j} p_{sk} \right) \\ \frac{\partial \mathcal{Q}}{\partial v_{jk}} &= \frac{\alpha_V - 1}{v_{jk}} - \frac{1}{\beta_V} - \eta \times v_{jk} + \sum_{i=1}^{M} \left(\frac{y_{ij}}{x_{ij}} - 1 \right) u_{ik} \\ \frac{\partial \mathcal{Q}}{\partial p_{sk}} &= \frac{\alpha_P - 1}{p_{sk}} - \frac{1}{\beta_P} - \eta \times p_{sk} \\ &\quad + \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\frac{y_{ij}}{x_{ij}} - 1 \right) \lambda u_{ik} \frac{\mathbb{I}(s \in \Pi_j)}{|\Pi_j|} \end{aligned} \tag{12}$$

We adopt the ascending gradient method [6] to infer the latent factors. Specifically, parameters θ are updated by the following equation:

$$\theta \leftarrow \theta + \epsilon \times \frac{\partial \mathcal{Q}}{\partial \theta}, \tag{13}$$

where θ is an element in $\{\mathbf{U}, \mathbf{V}, \mathbf{P}\}$, $\frac{\partial \mathcal{Q}}{\partial \theta}$ is the derivatives according to Equation (12), and ϵ is the learning rate. The algorithms iterate over $\{\mathbf{U}, \mathbf{V}, \mathbf{P}\}$ until one of the following termination conditions is reached (a) the value of objection function $\mathcal{Q}$ in Equation (11) keeps stable, or (b) the maximum number of iterations is reached.

4. EXPERIMENTS

4.1 Experimental setup

We aim to answer the following two questions:

- **Question 1**: Whether, and to what extent, our privacy-respect App recommendation model improves upon previous recommendation approaches that do not consider user privacy preferences?
- **Question 2**: How privacy levels (as introduced in Section 2.2) influence the performance of our approach?

Towards this goal, we first crawled a user-app rating dataset from Google Play via reverse engineering the service protocol. Then, using the crawled dataset, we compare our method with state-of-the-art latent factor based recommendation models and explore the impact of the three privacy levels on the performance of our approach.

4.2 Data Collection

We collected our dataset from Google Play. On Google Play, a user's ratings about Apps he/she used are publicly available. Once we obtain the Google ID of a user, we can locate all Apps the user has rated. Therefore, we first obtain a list of Google user IDs from Gong et al. [11] and write a crawler to retrieve all rated Apps of these users. Moreover, for each retrieved App, we crawled its permissions from Google Play. Our crawls were performed during June and July of 2014.

We remove unpopular Apps and users with too few rated Apps from our collected dataset to avoid *cold start* problem. Specifically, we first remove Apps with less than 5 users and then exclude users with less than 10 Apps. After this preprocessing step, our dataset has 16,344 users, 6,157 Apps,

Table 4: Data Description

users	Apps	ratings	sparsity	avg ratings
16,344	6,157	263,054	99.74%	16.09

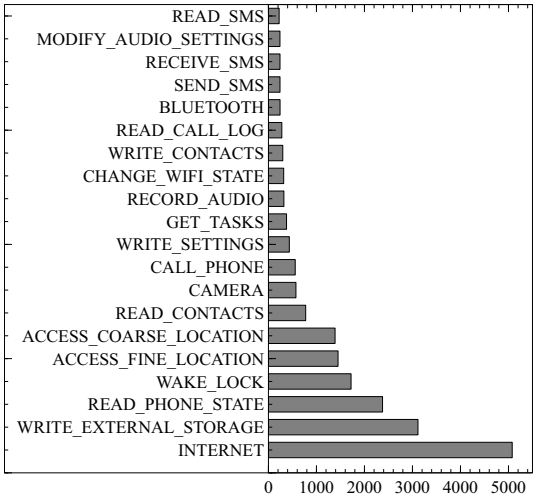


Figure 3: The top 20 most frequently used permissions and the number of Apps (out of the $6,157$ Apps in our dataset) that require those permissions.

and $263,054$ rating observations. The user App rating matrix has a sparsity as high as 99.74%. Each user rated 16.09 Apps on average, which is a very small fraction of all the Apps. Table 4 shows some basic statistics of our preprocessed dataset.

Figure 3 shows the top 20 most frequently used permissions and the number of Apps that require those permissions. While permissions such as `INTERNET` might not lead to privacy concerns for most users, some permissions (*e.g.*, `READ_CONTACTS`, `ACCESS_FINE_LOCATION`, and `CALL_PHONE`) could raise serious privacy concerns depending on how they are used by the Apps.

4.3 Compared Approaches

We compare our proposed model with the following latent factor based recommendation models:

- Singular Value Decomposition (SVD) [18]: SVD is a low dimension decomposition based recommendation method.
- Probabilistic Matrix Factorization (PMF) [25]: PMF extends SVD to a probabilistic framework.
- Non-negative Matrix Factorization (NMF) [19]: Similar to SVD, but NMF requires the latent vectors to be non-negative.
- Poisson Factor Model (Poi-FM) [13, 8]: Poisson factor model provides an alternative for latent factor model for different applications, and previous work [13] shows that Poisson factor model outperforms Gaussian based PMF.

Besides comparing to previous methods that do not consider user privacy preferences, we also investigate how different privacy levels impact the performance of our method. Specifically, we compare the following three variants of our method:

- **Privacy_Res.**: Privacy-respect App recommendation with Level I as the privacy level. Recall that this level considers the 10 resources listed in Table 2 as private data and do not distinguish different operations. Thus, each App's permissions are transformed to a 10-dimensional binary vector, which represents the private resources used by the App.
- **Sensitive_Perm.**: Privacy-respect App recommendation with Level II as the privacy level. Level II considers the 23 dangerous permissions that are related to the 10 sensitive resources. Therefore, we have a 23-dimensional binary vector to represent the private resources used by an App.
- **All_Danger_Perm.**: Privacy-respect App recommendation with Level III as the privacy level. Level III considers all critical resources (corresponding to 72 dangerour Android permissions) on a mobile device as private. Therefore, we have a 72-dimensional binary vector to represent the private resources used by an App.

Training and testing: We sample 80% of rated Apps of each user uniformly at random as training data, and we use the remaining rated Apps for testing. All the latent factor models are implemented with stochastic gradient ascent/descent optimization method with an annealing procedure to discount learning rate ϵ at iteration nIter with $\epsilon^{\text{nIter}} = \epsilon \frac{\nu}{\nu + \text{nIter} - 1}$ by setting $\nu = 50$. For SVD, PMF, and NMF, learning rates are set as $1e^{-3}$; learning rates for Poisson based methods are empirically set as $1e^{-4}$. For Poisson based latent factor models including both baseline Poi-FM and our proposed model, we set $\alpha_U = \alpha_V = \alpha_P = 20$ and $\beta_U = \beta_V = \beta_P = 0.5$; penalty weight η is set as $1e^{-5}$; functionality-privacy trade-off weight λ is empirically set as 1.

4.4 Evaluation Metrics

In App recommendation, we present to the user a list of recommendations, thus we evaluate the models in terms of ranking. Specifically, we present each user with N Apps that have the highest predicted values but are not rated by the user in the training phase, and we evaluate different approaches based on which of these Apps were actually adopted by the user in the test phase.

Precision and Recall: Given a top-N recommendation list $C_{N,\text{rec}}$, precision and recall are defined as

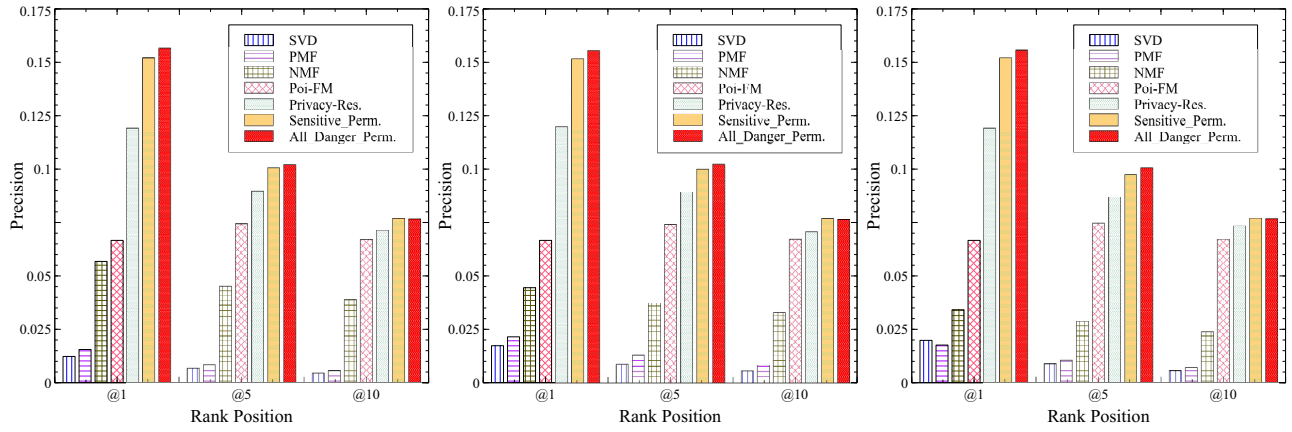
$$\begin{aligned} \text{Precision@}N &= \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}|}{N} \\ \text{Recall@}N &= \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}|}{|C_{\text{adopted}}|}, \end{aligned} \tag{14}$$

where C_{adopted} are the Apps that a user has adopted in the test data. The precision and recall for the entire recommender system are computed by averaging the precision and recall over all the users, respectively.

F-measure: F-measure balances between precision and recall. We consider the F_β metric, which is defined as

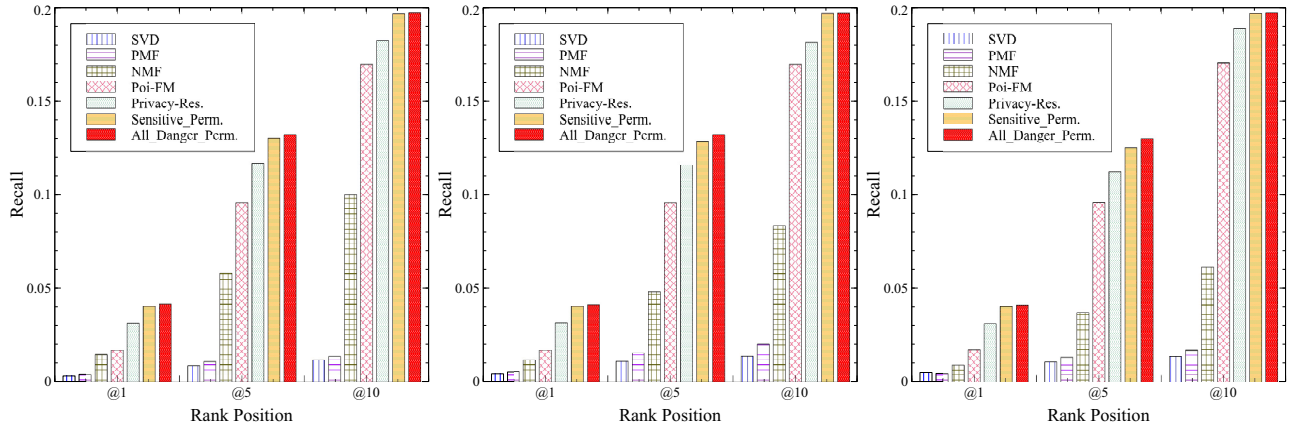
$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \times \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}. \tag{15}$$

The F_β metric with $\beta < 1$ indicates more emphasis on precision than recall. In our experiments, we use F_β metric with $\beta = 0.5$.



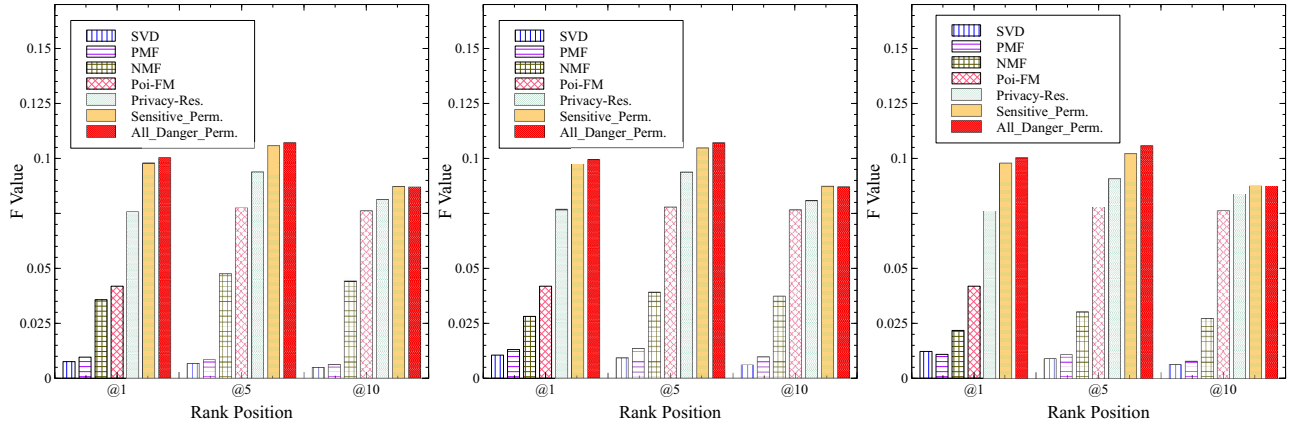
(a) $K = 20$ **(b)** $K = 30$ **(c)** $K = 50$

Figure 4: Precision @N with different latent dimensions K.



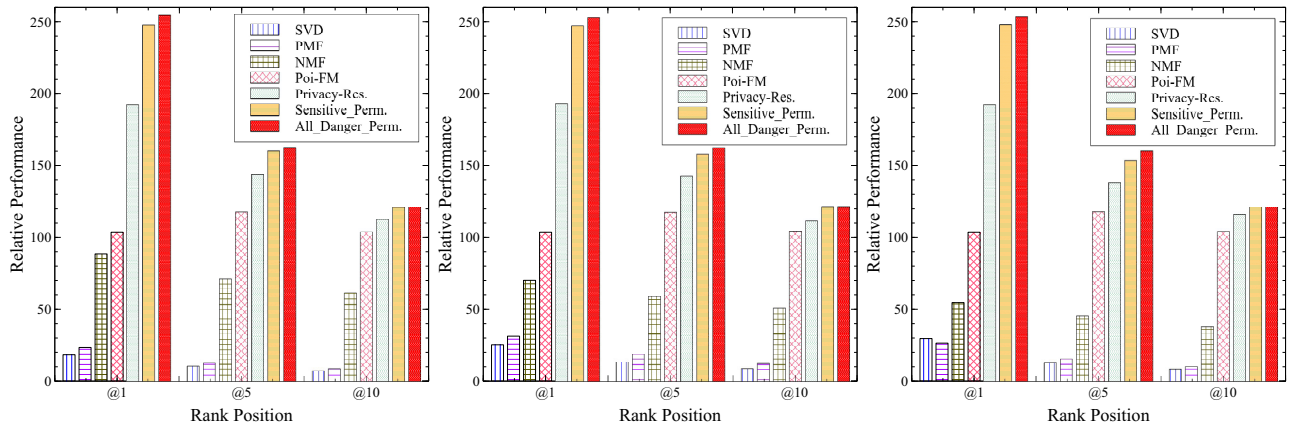
(a) $K = 20$ **(b)** $K = 30$ **(c)** $K = 50$

Figure 5: Recall @N with different latent dimensions K.



(a) $K = 20$ **(b)** $K = 30$ **(c)** $K = 50$

Figure 6: F_β **@N with different latent dimensions K ($\beta = 0.5$).**



(a) $K = 20$ **(b)** $K = 30$ **(c)** $K = 50$

Figure 7: Relative performance @N with different latent dimensions K.

r-Recall and r-Precision: Similar to Yin et al. [30], we also compare the different models in terms of relative precision and recall. Let C denote the candidate Apps, the precision and recall in a top-N list of a random recommender system are $\frac{|C_{\text{adopted}}|}{|C|}$ and $\frac{N}{|C|}$, respectively. Then, the relative precision and recall [30] are defined as

$$\text{rPrecision@}N = \frac{\text{Precision@}N}{|C_{\text{adopted}}|/|C|} = \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}| \cdot |C|}{|C_{\text{adopted}}| \cdot N}$$
$$\text{rRecall@}N = \frac{\text{Recall@}N}{N/|C|} = \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}| \cdot |C|}{|C_{\text{adopted}}| \cdot N} \quad (16)$$

Note that the relative precision and recall have the same value. Therefore, we only show one of them and we name it as the relative performance, which measures the improvement upon a random recommendation method.

4.5 Comparison Results

Figure 4, Figure 5, Figure 6, and Figure 7 respectively show the precision@N, recall@N, F_β@N, and relative performance@N of all compared approaches on our dataset, where $N = 1, 5, 10$. For each approach, we explore 3 latent dimension settings, *i.e.*, $K = 20$, $K = 30$, and $K = 50$.

4.5.1 Comparing Our Method with Previous Approaches

We find that our approach consistently outperforms previous methods for different N and different K. Specifically, we observe that NMF outperforms both SVD and PMF, and that Poisson based factor model Poi-FM can further improve upon NMF with all the three considered number of latent dimensions. Moreover, our privacy-respect App recommendation methods with the three privacy levels all further improve upon Poi-FM with significant margins for all the four evaluation metrics. To better demonstrate the improvements, Table 5 shows the absolute improvements of our proposed method, with different privacy levels, as compared to the best baseline method Poi-FM when $K = 30$ and $N = 1$. We can see significant improvement margins. While precision and recall may change with different N, F-measure provides a stable comparisons as it is the harmonic mean of precision and recall. In terms of F-measure, we can observe a 3.46% improvement for **Privacy_Res.**, a 5.56% improvement for **Sensitive_Perm.**, and a 5.80% improvement for **All_Danger_perm.**.

Table 5: Absolute improvements of our proposed method with different privacy levels as compared to the best baseline method Poi-FM ($K = 30$, $N = 1$).

metric	Privacy_Res.	Sensitive_Perm.	All_Danger_perm.
F-measure	**3.46%** ↑	**5.56%** ↑	**5.80%** ↑
precision	5.31% ↑	8.49% ↑	8.86% ↑
recall	1.45% ↑	2.33% ↑	2.43% ↑
relative	89.16 ↑	143.45 ↑	149.39 ↑

Our results show that once we consider user privacy preference, performance of App recommendation gets improved no matter what privacy level is used.

4.5.2 Impact of Privacy Levels

We find that our method with Level II privacy information (*i.e.*, **Sensitive_Perm.**) can improve upon our method with Level I privacy information (*i.e.*, **Privacy_Res.**) with notable margins. This implies that users treat different operations (*e.g.*, read and write) on the 10 private resources with different privacy concerns. However, **Sensitive_Perm.** and **All_Danger_Perm.** have very close performances consistently for all settings we considered. Figure 8 shows a zoom in comparisons of the three levels. Our observations indicate that users probably do not treat resources (*e.g.*, Internet, Bluetooth) other than the 10 resources in the privacy level Level I and Level II as private resources, and thus accessing those resources would not influence whether a user likes/adopts an App.

4.5.3 Summary

We find that our privacy-respect App recommendation method significantly outperforms previous approaches that do not consider user privacy preference. Moreover, we find that users are more likely to treat the 10 resources in Level I and Level II as private resources and they treat different operations (*e.g.*, read and write) on these resources with different privacy concerns.

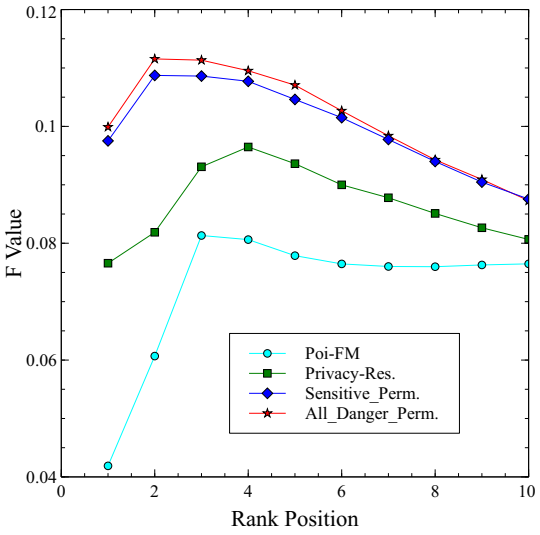


Figure 8: Zoom in comparisons between our methods with the three privacy levels (i.e., Privacy_Res., Sensitive_Perm., and All_Danger_perm.) and the best baseline method (i.e., Poi-FM) for $K = 30$.

5. RELATED WORK

Most of related works come from two research fields: personalized recommendation methodologies especially latent factor model based recommendation models, and mobile App rankings and recommendations.

Personalized Recommendation: Latent factor models have become popular and been widely used in recommendation systems. These work include matrix factorization [18], Probabilistic Matrix Factorization (PMF) [25] and its Bayesian version [24], and their other variants [17, 2, 16]. In our case and many cases alike, the recommendation system needs to infer user preferences from user feedbacks. Latent factor models, which are suitable for better capturing preference order are preferred and followed in our approach. One option in designing latent factor model is to set non-negative constrains on latent factors to force response variables to be in a wider range than the rating based response, which is normally limited to a certain range of integers. As a result, non-negative matrix factorization based methods are widely used [19, 31, 14] due to this advantage.

Furthermore, most of these latent factors based studies along this line of research assume that the user response follows Gaussian distribution with expectation from the product of user and item latent factors. However, some recent work [13, 8, 23] have pointed out that Poisson distribution could be a better choice for modeling user response. Firstly, it better captures real consumption data; secondly, due to the form of Poisson distribution, only the observed part of user-item matrix need to be iterated during modeling. This is a big advantage considering the usually extreme sparsity of user-item matrix in recommendation problems, providing better scalability.

This paper follows the state-of-the-art latent factor model to propose a novel model that is more suitable for App recommendation task by introducing users' privacy preference information. Experimental evidence shows advantage of our approach against above state-of-the-art models.

Mobile App Ranking and Recommendation: There have been a few previous works on App recommendation, but these works only focused on recommending the most relevant Apps to a user without considering *user privacy preference*. Work from [15] provides a context-aware recommendation using tensor factorization by including context information such as location, moving status and time. To address the cold-start problem for App recommendation, [20] proposed to incorporate side information from Twitter. Information of followers of the App's official Twitter account is collected and utilized to model the App, providing an estimation about which users may like the App, even when the App still has no official rating yet. Yin *et al.* [30] considered a trade-off between satisfaction and temptation for App recommendation with a special focus on the case that a user would like to replace an old App with a new one. An interesting dataset is collected via an App from users, revealing users' process of choosing a new App after comparing it with those already obtained ones. And the satisfaction and temptation of an App is evaluated and used to facilitate the recommendation algorithm.

More recently, Zhu *et al.* [32] proposed a mobile App recommendation (more precisely, *ranking*) system by considering both the App's popularity and security risks. They provide an identical global ranking of Apps to every user, and thus their work is not *personalized* App recommendation.

As described above, different from all previous work on App recommendation (personalized or unpersonalized), this paper, to the best of our knowledge, is the first one that proposes to incorporate user privacy preference into mobile App recommendation. Experiments on real-life data show affirmative results for the contribution of privacy information in App recommendation.

6. CONCLUSION AND FUTURE WORK

In this paper, we present the *first* systematic study on leveraging the trade-off between a user's interest-functionality expectation and her/his privacy preference to perform personalized privacy-respect App recommendations. Specifically, we first propose a new model to capture the trade-off between functionality and user privacy preference. Our model is flexible to incorporate three levels of privacy information. Moreover, we crawled a real-world dataset from Google Play and use it evaluate our method. Our results demonstrate that our method achieves consistent and substantial performance improvement over previous approaches. This implies that it is important to consider user privacy preference on personalized App recommendations. Furthermore, we explore the impact of different levels of privacy information on the performances of our method. We find that treating different operations with different privacy concerns achieves better recommendation performances.

A few interesting future directions include measuring users' different privacy preferences in the three privacy levels in the wild and constructing a more fine-grained model to capture the trade-off between functionality expectation and privacy preference.

7. REFERENCES

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

[2] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, KDD '09, pages 19–28, 2009.

[3] N. Aizenberg, Y. Koren, and O. Somekh. Build your own music recommender by modeling internet radio streams. In *Proc. of the 21st int'l conf. on World Wide Web*, pages 1–10, 2012.

[4] App Store Statistics. `http://en.wikipedia.org/wiki/App_Store_(iOS)`.

[5] R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79, Dec. 2007.

[6] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

[7] J. L. Boyles, A. Smith, and M. Madden. Privacy and data management on mobile devices. *Pew Internet & American Life Project*, 2012.

[8] J. Canny. Gap: A factor model for discrete data. In *Proc. of the 27th ACM SIGIR Conf. on Research and Development in Information Retrieval*, SIGIR '04, pages 122–129, 2004.

[9] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner. Android permissions demystified. In *CCS*, 2011.

[10] N. Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. R. Shi, and D. Song. Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2):27, 2014.

[11] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar, and D. Song. Evolution of social-attribute networks: measurements, modeling, and implications using google+. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 131–144. ACM, 2012.

[12] Google Play Statistics. `http://en.wikipedia.org/wiki/Google_Play`.

[13] P. Gopalan, J. M. Hofman, and D. M. Blei. Scalable recommendation with poisson factorization. *CoRR*, abs/1311.1704, 2013.

[14] Q. Gu, J. Zhou, and C. H. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, pages 199–210, 2010.

[15] A. Karatzoglou, L. Baltrunas, K. Church, and M. Böhmer. Climbing the app wall: Enabling mobile app discovery through context-aware recommendations. In *Proc of the 21st ACM Int'l Conf. on Information and Knowledge Management*, CIKM '12, pages 2527–2530, 2012.

[16] D. Kong, M. Zhang, and C. H. Q. Ding. Minimal shrinkage for noisy data recovery using schatten-p norm objective. In *ECML/PKDD 2013*, pages 177–193, 2013.

[17] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, 2008.

[18] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.

[19] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

[20] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *Proc. of the 36th int'l ACM SIGIR conf. on Research and development in information retrieval*, SIGIR '13, pages 283–292, 2013.

[21] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan. 2003.

[22] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning geographical preferences for point-of-interest recommendation. In *Proc. of the 19th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, KDD '13, pages 1043–1051, 2013.

[23] B. Liu, H. Xiong, S. Papadimitriou, Y. Fu, and Z. Yao. A general geographical probabilistic factor model for point of interest recommendation. *Knowledge and Data Engineering, IEEE Transactions on*, PP(99), 2015.

[24] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. ICML '08, pages 880–887, 2008.

[25] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, volume 20, 2008.

[26] S. Shekhar, M. Dietz, and D. S. Wallach. Adsplit: Separating smartphone advertising from applications. In *Usenix Security*, 2012.

[27] Smartphone Sales in the Third Quarter of 2013. `http://www.finfacts.ie/irishfinancenews/article_1026800.shtml`.

[28] The Smartphone Market is Bigger Than the PC Market. `http://www.businessinsider.com/smartphone-bigger-than-pc-market-2011-2`.

[29] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proc of the 34th Int'l ACM SIGIR Conf on Research and Development in Information Retrieval*, SIGIR '11, pages 325–334, 2011.

[30] P. Yin, P. Luo, W.-C. Lee, and M. Wang. App recommendation: a contest between satisfaction and temptation. WSDM '13, pages 395–404, 2013.

[31] S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *SDM'06*, 2006.

[32] H. Zhu, H. Xiong, Y. Ge, and E. Chen. Mobile app recommendation with security and privacy awareness. In *Proc. of the 20th ACM Int'l Conf. on Knowledge Discovery and Data Mining*, KDD '14, 2014.