Mobile App Recommendation: An Involvement-Enhanced Approach

ABSTRACT

Given the ubiquitous and critical role of mobile apps in people's lives as well as the sheer size of the mobile app market, developing effective mobile app recommendation methods that can help users locate the mobile apps they desire is critical for both mobile app users and platforms. Premised in involvement theory, we propose a novel mobile app recommendation method that integrates both users' app download behaviors and app browsing behaviors for mobile app recommendations, in contrast to existing methods that rely on download behaviors but neglect browsing behaviors. Specifically, we introduce a novel model that appropriately combines download and browsing behaviors to learn users' overall interests in and involvement with apps, we develop a new algorithm to infer the model's parameters, and we propose an innovative mobile app recommendation strategy that combines users' overall interests and their current interests to recommend apps. Finally, using data collected from one of the largest mobile app platforms in China, we demonstrate and analyze the superior performance of our method over several state-of-the-art mobile app recommendation methods.

Keywords: mobile app recommendation, data mining, machine learning, graphical model, product involvement

1. INTRODUCTION

Fostered by the growing popularity of smartphones and other mobile devices, mobile apps are playing an increasingly important role in people's lives, partly due to their ease of use, affordable cost, and wide range of functionalities that serve almost every aspect of people's lives. People now spend more time using mobile devices rather than PCs to access the Internet.[1] Moreover, 90% of people's mobile time is spent using mobile apps.[2] Indeed, mobile apps have become an indispensable part of our lives and enrich almost every aspect, from social networking and health monitoring to shopping and entertaining (Ghose and Han 2014). Given the ubiquitous and critical role of mobile apps in people's lives, it is imperative that mobile app users can easily find the apps they need. However, the huge number of mobile apps that are available to download on mobile app platforms poses a significant challenge for users trying to locate the apps they desire (Yin et al. 2013). For example, Apple's app store offers 1.5 million mobile apps, and the number of mobile apps for Android users is even larger, around 1.6 million. Therefore, it is necessary to develop effective recommendation methods that can recommend a handful of mobile apps on a mobile app platform to meet a user's needs.

Effective mobile app recommendation methods also benefit the platforms by reaping revenues and profits from apps downloaded from the platforms. The global mobile app market is enormous and continues to grow rapidly. In 2015, global mobile app revenue amounted to $69.7 billion, and it is projected to reach $188.9 billion in

[1] http://money.cnn.com/2014/02/28/technology/mobile/mobile-apps-internet/index.html

[2] http://www.phunware.com/blog/mobile-stat-snack-90-of-mobile-time-spent-in-apps/

2020.[3] Given the sheer size of the global mobile app market, effective mobile app recommendation methods can generate significant financial gains for mobile app platforms. An effective mobile app recommendation method not only increases the click-through rates of mobile apps but also improves mobile app sales (Jannach and Hegelich 2009). Indeed, prior studies have argued for and convincingly shown the positive impacts of recommendation methods in general, and mobile app recommendation methods in particular, on sales and profits (Chen et al. 2012; Fleder and Hosanagar 2009; Jannach and Hegelich 2009; Pathak et al. 2010).

Given the importance of mobile app recommendation for both mobile app users and platforms, a number of mobile app recommendation methods have been proposed. Some methods recommend apps that are most similar to a user's previously downloaded apps (Natarajan et al. 2013; Shi and Ali 2012; Yan and Chen 2011). Some others build a user-app matrix based on users' app download behaviors and propose a latent factor model to reduce the dimensionality of the matrix for high-speed mobile app recommendations (Liu et al. 2015; Rendle and Freudenthaler 2014; Takács and Tikk 2012). Recently, mobile app recommendation methods based on Latent Dirichlet Allocation model (LDA; Blei et al. 2003) have received attention. LDA-based mobile app recommendation methods discover users' interests in apps from their app download behaviors and recommend apps that best match users' interests (Lin et al. 2013; Park et al. 2016; Zhu et al. 2015). Existing methods predominantly focus on download behaviors. However, users often browse different

[3] https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/

apps for comparison before their downloads. Such browsing behaviors vividly reflect users' decision processes that lead to their downloads. Thus, intuitively, both download and browsing behaviors should be considered to achieve effective mobile app recommendations. Theoretically, involvement theory (Beatty et al. 1988; Bloch et al. 1986; Zaichkowsky 1985) suggests that different app categories can elicit different degrees of user involvement and that different degrees of involvement will result in different browsing behaviors. For example, a user who is interested in a high-involvement app category is likely to browse and compare many apps in that category before downloading an app of that category. In contrast, a user who is interested in a low-involvement app category may engage in little or even no browsing behavior before downloading an app in that category. Hence, users have interests in apps with different involvement degrees that affect and thus are reflected in their browsing behaviors. Consequently, both browsing and download behaviors are essential to be considered for learning users' interests and therefore necessary for making effective mobile app recommendations.

Premised in involvement theory, we propose a method that innovatively integrates both download and browsing behaviors for mobile app recommendations, in contrast to existing methods, which rely on download behaviors but neglect browsing behaviors. Specifically, we introduce a novel graphical model that appropriately combines download and browsing behaviors to learn users' overall interests in and involvement with apps; we develop a new algorithm to infer the model's parameters; and we propose an innovative mobile app recommendation

strategy based on the users' overall interests and their current interests, which are discovered based on their most recent browsing behaviors (i.e., browsing behaviors since their last download). The rest of this paper is organized as follows. We review existing mobile app recommendation methods and discuss the key differences between these methods and our proposed method in §2. We then describe involvement theory, which motivates and guides the development of our method, in §3. Next, we formally define the mobile app recommendation problem in §4, and we propose our method in §5. We demonstrate and analyze the effectiveness of our method using a large-scale real-world dataset in §6, and we conclude the paper with a discussion of its contributions, its business implications, and future research directions in §7.

2. RELATED WORK

Existing methods for mobile app recommendations can be broadly categorized as item-based collaborative filtering, latent factor models, or LDA-based recommendations. In the following, we review representative methods of each category and then discuss how our method differs from existing methods.

2.1 Item-based Collaborative Filtering

Item-based collaborative filtering recommends items (e.g., mobile apps) that are most similar to a user's previously selected (e.g., downloaded) items (Sarwar et al. 2001). The similarities between items are computed based on users' selections of these items (e.g., downloads of apps in the context of mobile app recommendations). Item-based collaborative filtering is easy to implement and has therefore been commonly

employed by mobile app recommender systems such as AppJoy (Yan and Chen 2011), EigenApp (Shi and Ali 2012), and iConRank (Natarajan et al. 2013) as well as e-commerce recommender systems such as Amazon's recommender system (Linden et al. 2003). However, item-based collaborative filtering is computationally expensive, due to the large number of mobile apps offered by mobile app platforms and the tremendous cost of computing similarities between all pairs of apps (Shi and Ali 2012).

2.2 Latent Factor Model

Latent factor models aim to discover the latent features of users and items through dimension reduction techniques, which reduces the high computational cost associated with item-based collaborative filtering. Representative latent factor models include matrix factorization and tensor-based factorization. Matrix factorization maps both users and items onto a low-dimensional latent feature space and then approximates a user's selection of an item (e.g., download of an app) using the dot product of the corresponding user latent factor and item latent factor. Matrix factorization techniques such as AoBPR (Rendle and Freudenthaler 2014) and RankALS (Takács and Tikk 2012) are commonly used by recommender systems. Liu et al. (2015) propose a matrix-factorization-based method for mobile app recommendations that considers the trade-off between users' functionality requirements for apps and their privacy concerns. As an extension of matrix factorization, tensor-based factorization decomposes an N-dimensional tensor into N low-dimensional matrices. Typically, a three-dimensional tensor factorization

technique that considers users, items, and contexts is used for formulating context-aware recommendations. In this vein, Karatzoglou et al. (2012) develop a tensor-based factorization method that incorporates contextual information to formulate mobile app recommendations. However, latent factor models are ineffective for making recommendations based on an extremely sparse user-item matrix (Shi and Ali 2012). This problem can be effectively addressed by LDA-based recommendation methods.

2.3 LDA-based Recommendation

LDA-based recommendation discovers users' interests in items based on their previous selections of items (e.g., downloads of apps) and recommends items that best match users' interests. In general, LDA-based recommendation models each user as a distribution over interests and represents each interest as a distribution over items. Interests are referred to as preferences, intentions, or motivations in various studies. For example, Lin et al. (2013) propose a method to discover users' preferences regarding apps, while Zhu et al. (2015) employ LDA to learn users' preferences regarding different app categories for context-aware app recommendations. Park et al. (2016) propose an LDA-based recommendation method that infers users' intentions to download apps and then recommends apps that best match the target user's intentions. Lin et al. (2014) design a semi-supervised LDA model for mobile app recommendations; their method discovers topic distributions for users and apps from users' app ratings as well as app descriptions and recommends apps to users according to their topic distributions. He and Liu (2017) combine LDA model with a mixture of

Gaussians to discover users' goals of downloading mobile apps. LDA-based recommendation has also been used in other domains. Li et al. (2011) use the LDA model to discover latent topics of news articles for personalized news recommendations. Jacobs et al. (2016) propose a LDA-based product recommendation method that models each user as a distribution over motivations and each motivation as a distribution over products. And LDA-based recommendation methods have been proposed for learning travelers' travel package preferences in different contexts (e.g., travel seasons) to make travel package recommendations (Liu et al. 2014; Tan et al. 2014).

Our literature review reveals that most existing methods recommend mobile apps based on users' download behaviors but neglect their browsing behaviors. However, according to involvement theory (Beatty et al. 1988; Bloch et al. 1986; Zaichkowsky 1985) both download and browsing behaviors are essential for learning users' interests in apps. Therefore, a mobile app recommendation method that properly integrates both download and browsing behaviors to learn users' interests can provide more accurate recommendations than a method that relies solely on download behaviors. Conceptually, the key difference between our proposed method and existing methods is that our method is premised in involvement theory and considers both download and browsing behaviors for making mobile app recommendations, while existing methods rely on download behaviors but neglect browsing behaviors. Moreover, our method differs methodologically from existing methods. To properly integrate both download and browsing behaviors for making mobile app

recommendations, we propose a novel graphical model that infers users' interests and involvement from their download and browsing behaviors, develop a new algorithm for learning the model parameters, and design an innovative mobile app recommendation strategy. The proposed graphical model, the model learning algorithm, and the recommendation strategy represent the methodological novelty of this study.

3. THEORETICAL FOUNDATION

Involvement has been an important variable explaining consumer behaviors for several decades. Overall, involvement has been studied from two different perspectives. From the psychological perspective, involvement is defined as a consumer's perceived relevance evoked by a stimulus (e.g., a product category such as jewelry) (Mitchell 1979; Zaichkowsky 1985). Commonly reported factors determining the degree of involvement include perceived risk, self-concept, pleasure, interests, etc. (Beatty et al. 1988; Laurent and Kapferer 1985; Mitchell 1979; Zaichkowsky 1985). Involvement affects and thus is reflected in consumer behaviors (Bloch et al. 1986; Laurent and Kapferer 1985). Hence, involvement has also been studied from the behavioral perspective and measured with information search efforts (Engel et al. 1993; Michaelidou and Dibb 2008). From the behavioral perspective, involvement is defined as the intensity of efforts expended in the process of information search and product acquisition (Stone 1984). Extensive search for information would indicate high involvement while few information search would show low involvement (Stone 1984).

The positive relationship between a consumer's degree of involvement and information search efforts has been widely reported in the literature (Bloch 1986; Harris 1987; Mittal 1989; Stone 1984; Zaichkowsky 1985). The degree of involvement is often dichotomized as low or high (Engel et al. 1993). A consumer who has high involvement with a product category actively searches and collects information about products in that category, deeply processes the collected information, carefully compares differences and similarities among products in that category, and finally makes a choice. In contrast, a consumer who has low involvement with a product category makes a decision with few or even no searches and comparisons. For instance, a consumer who has high involvement with a product category tends to compare many products in that category (Chaiken 1980; McColl-Kennedy and Fetter Jr 2001), asks others for advice prior to purchases (McColl-Kennedy and Fetter Jr 2001), and seeks information from external word-of-mouth sources (Gu et al. 2012). Moreover, the positive relationship becomes stronger as the hedonic value of a product category increases, because searching for information about products in this category is accompanied by an ongoing desire for the pleasure inherent in the products of this category (Chaudhuri 2000). In this study, we measure a user's involvement with a mobile app using the user's information search efforts before downloading the app. In the context of mobile apps, information search before an app download is embodied in users' browsing behaviors on a mobile app platform. By browsing and reading apps' descriptive information such as their functional description, number of downloads, language, rating, reviews, etc., a user compares

differences and similarities among alternative apps and decides which one to download. The intensity of such browsing behaviors reflects information search efforts a user expends before downloading a mobile app, and thus demonstrates the user's involvement with the app.

The degree of involvement can be explained by factors such as perceived risk, self-concept, pleasure, interests, etc. (Beatty et al. 1988; Laurent and Kapferer 1985; Mitchell 1979; Zaichkowsky 1985), which are partly driven by the characteristics of the product category, such as endurance, price, complexity, hedonic value, emotional appeal, and symbolic value (Houston and Rothschild 1978; Laurent and Kapferer 1985; Zaichkowsky 1985). For example, durable products with high price tags, such as consumer electronics, appliances and automobiles, are typical examples of high-involvement products, since purchasing a poorly chosen product in one of these categories often has costly and long-lasting consequences (Gu et al. 2012; Laurent and Kapferer 1985; Zaichkowsky 1985). In contrast, consumable, low-cost products such as groceries, music CDs, instant coffee, and bubble bath soap are representative examples of low-involvement products (Kannan et al. 2001; Zaichkowsky 1985). For these products, the cost of purchasing a poorly chosen item is low. In addition, products of high complexity (e.g., products with a large number of performance-related dimensions), are likely to cause high involvement (Houston and Rothschild 1978). Besides, products with high hedonic value and emotional appeal, such as tourism packages and champagne, are deemed to be high-involvement products (Nicolau 2013). In addition, products with high symbolic value that many consumers

use as a form of self-expression and self-imaging, such as clothing, are also regarded as high-involvement products (Bloch 1986; Zaichkowsky 1985). In short, because of their inherent characteristics, some product categories are high-involvement categories while others are low-involvement categories. In the case of mobile apps, those apps of high complexity often induce high-involvement because of high risk and uncertainty associated with their downloads (Houston and Rothschild 1978; Laurent and Kapferer 1985). In addition, game apps generally have more hedonic value and emotional appeal than utility apps; thus, the former is more likely to cause high involvement than the latter (Beatty et al. 1988).

Overall, involvement theory maintains that different product categories can elicit different degrees of involvement and that different degrees of involvement result in different information search efforts. According to involvement theory, in the context of mobile apps, a user who is interested in a high-involvement app category may browse many apps in that category to make comparisons before downloading an app of that category, while on the other hand, a user who is interested in a low-involvement app category may do little or even no browsing before downloading an app in that category. Hence, users have interests in apps with different involvement degree that affect and thus are reflected in their browsing behaviors. Consequently, to learn a user's overall interests in apps, we need to consider both browsing and download behaviors. Furthermore, a user's most recent browsing behaviors reveal the user's current interest. Thus, an effective mobile recommendation strategy is to make recommendations based on a user's overall interests and the user's current interest. In

the following sections, we propose a novel mobile app recommendation method that integrates both browsing and download behaviors to realize this recommendation strategy.

4. PROBLEM FORMULATION

We formulate the mobile app recommendation problem in this section. Let *I* denote a set of *V* apps offered by a mobile app platform and let *U* be a set of *M* app users. Each app is described by its name and the category it belongs to. Users' behaviors on a platform, such as downloading or browsing apps, are recorded in behavioral logs. Each record in the behavioral logs consists of the user ID, the app name, the behavior type (e.g., downloading or browsing), a timestamp, and the app category, indicating who downloads or browses which app, and when. For each user, we can extract her or his *download sequence* (Definition 1) as well as the *browsing sequence* associated with each download (Definition 2) from the behavioral logs.

*Definition 1 (**Download Sequence**).* A user's *download sequence* is the sequence of apps downloaded by the user, sorted in ascending time order. Formally, the download sequence for user $u_m \in U, m = 1,2, \dots, M$, is denoted as $D_m =< i_{m,1}, \dots, i_{m,n}, \dots, i_{m,N_m} >$, where N_m is the number of apps downloaded by u_m and $i_{m,n} \in I$ denotes the n^{th} app downloaded by u_m.

*Definition 2 (**Browsing Sequence**).* The *browsing sequence* associated with an app download is the sequence of apps browsed between the preceding download and the focal one. Formally, the browsing sequence associated with user u_m's n^{th} app download, $i_{m,n} \in D_m$, is denoted as $B_{m,n} = < i_{m,n,1}, \dots, i_{m,n,s}, \dots, i_{m,n,N_{m,n}} >$, where

$N_{m,n}$ is the number of apps browsed between $i_{m,n-1} \in D_m$ (the $n-1^{th}$ app downloaded by user u_m) and $i_{m,n} \in D_m$, and $i_{m,n,s}$ denotes the s^{th} app browsed by u_m between $i_{m,n-1}$ and $i_{m,n}$.

User ID	App Name	Behavior Type	Timestamp	App Category
1	Ice Crush	Browse	2015-11-15	Game
1	Happy Crush	Browse	2015-11-15	Game
1	Candy Crush	Browse	2015-11-15	Game
1	Candy Crush	Download	2015-11-15	Game
1	Youku	Download	2015-11-20	Entertainment
1	CorePlayer	Browse	2015-11-21	Entertainment
1	Happy Crush	Download	2015-12-03	Game
1	Kids Math	Browse	2015-12-06	Education
1	Kids Reading	Browse	2015-12-06	Education

Table 1: An Example Behavioral Log

We illustrate Definitions 1 and 2 using the example behavioral log shown in Table 1. According to the log, user 1's download sequence is <*Candy Crush*, *Youku*, *Happy Crush*>. The browsing sequence associated with the download of *Candy Crush* is <*Ice Crush*, *Happy Crush*, *Candy Crush*>, while that associated with the download of *Youku* is empty because there is no browsing behaviors between the download of *Candy Crush* and the download of *Youku*. In this example, user 1's most recent browsing behaviors are the user's app browses after her or his last download (i.e., browses of *Kids Math* and *Kids Reading*).

We are now ready to define the mobile app recommendation problem studied in this paper:

Given a set of *V* apps *I* offered by a mobile app platform, a group of *M* app users *U*, the users' download sequences $\bigcup_{m=1}^{M} D_m$, the browsing sequence $B_{m,n}$ associated with each download, and the users' most recent browsing behaviors

(i.e., their browsing behaviors after their last download), predict the probability that a user $u_m \in U$ will download an app $i \in I$.

Once we have determined these probabilities, the apps with the highest probabilities of being downloaded by a user are recommended to the user. For the convenience of readers, important notation used in this paper is listed in Table 2 and a more complete list of notation is summarized in Appendix A.

Notation	Description
$i_{m,n}$	the n^{th} app downloaded by user u_m
$f_{m,n}$	browsing intensity level associated with $i_{m,n}$
$z_{m,n}$	interest associated with $i_{m,n}$
$e_{m,n}$	involvement state associated with $i_{m,n}$
$\boldsymbol{\theta_m}$	K-dimensional interest distribution for user $\boldsymbol{u_m}$
$\boldsymbol{\varphi_k}$	V-dimensional app distribution for interest k
$\boldsymbol{\lambda_k}$	E-dimensional involvement distribution for interest k
$\boldsymbol{\pi_e}$	F-dimensional browsing intensity distribution for involvement state e
$\boldsymbol{b}_m$	most recent browsing behaviors by user u_m
$c_{m,z,i,e,f}$	number of app i downloaded by user u_m due to interest z and with involvement state e and browsing intensity level f

Table 2: Important Notation

5. INVOLVEMENT-ENHANCED MOBILE APP RECOMMENDATION

In this section, we propose a novel mobile app recommendation method, involvement-enhanced mobile app recommendation (IMAR), that integrates both browsing and download behaviors for mobile app recommendations. We first describe LDA-based mobile app recommendation (LMAR), an existing method that has been widely applied for mobile app recommendations and that is closely related to our study. We then propose a graphical model for IMAR and discuss its differences from that of LMAR. Next, we develop a novel algorithm for learning the graphical model of IMAR and design a new recommendation strategy. Overall, IMAR's novel

graphical model, its model learning algorithm, and its recommendation strategy constitute the methodological contribution of this study.

5.1 LMAR

A user's download behaviors are driven by the user's interests (Liu et al. 2015). Therefore, to predict the probability of a user downloading an app, it is necessary to model the user's interests, for which LDA is a popular tool (Lin et al. 2013; Zhu et al. 2015). Specifically, LMAR (LDA-based mobile app recommendation) models each user as a probability distribution over interests and represents each interest with a probability distribution over apps (Zhu et al. 2015). Figure 1 illustrates users, interests, and apps in LMAR. As shown, *Social Networking* interest is represented with a probability vector (0.4, 0.3, 0.2, 0.05, 0.05) over five apps: *Wechat*, *Facebook*, *Twitter*, *Candy Crush*, and *Diamond Dash*. Each element of the vector indicates the probability of downloading the corresponding app—for example, the probability of downloading *Wechat* given the *Social Networking* interest is 0.4. Clearly, the *Social Networking* interest has a higher probability of driving a social networking app download (e.g., *Wechat*, *Facebook*, *Twitter*) while the *Crush Games* interest is more likely to motivate a game app download (e.g., *Candy Crush*, *Diamond Dash*). Two users, one female and one male, are defined over the interests. For example, the female user is modeled using the probability vector (0.86, 0.14) over the two interests: *Social Networking* and *Crush Games*, which indicates a high probability (0.86) of the *Social Networking* interest.

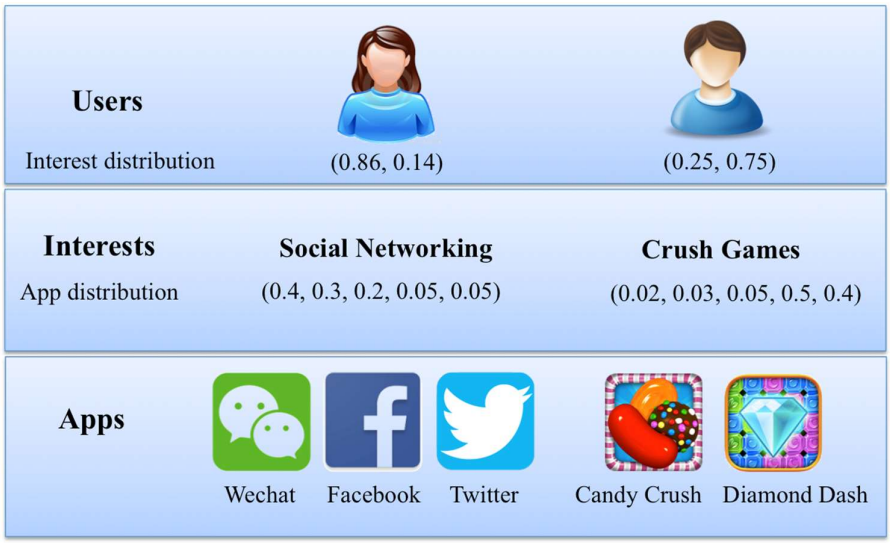

Figure 1: Users, Interests, and Apps in LMAR

Formally, a user $u_m \in U$ is modeled as a multinomial distribution over K interests with the parameter $\boldsymbol{\theta_m}$, which is a K-dimensional vector in which each element denotes the probability of an interest. The parameter $\boldsymbol{\theta_m}$ is generated using a *Dirichlet* distribution with a K-vector hyper-parameter $\boldsymbol{\alpha}$, i.e., $\boldsymbol{\theta_m} \sim Dir(\boldsymbol{\alpha})$. Each interest k, k=1, 2, …, K, is represented as a multinomial distribution over V apps with the parameter $\boldsymbol{\varphi_k}$, which is a V-dimensional vector in which each element denotes the probability of downloading the corresponding app given interest k. The parameter $\boldsymbol{\varphi_k}$ is generated using a *Dirichlet* distribution with a V-vector hyper-parameter β, i.e., $\boldsymbol{\varphi_k} \sim Dir(\beta)$. LDA is a generative probabilistic model that defines a generative process through which observable outcomes (e.g., download behaviors) are generated from unobservable variables (e.g., interests) (Blei et al. 2003). In LMAR's generative process, illustrated in Figure 2, a download behavior is generated in two steps. First, an interest $z_{m,n}$ is selected in accordance with the user's multinomial distribution over interests, i.e., $z_{m,n} \sim multi(\boldsymbol{\theta_m})$. Next, an app download $i_{m,n}$ is drawn based on

the selected interest's multinomial distribution over apps, i.e., $i_{m,n} \sim multi(\boldsymbol{\varphi}_{\boldsymbol{z_{m,n}}})$. Figure 3 shows the graphical model for LMAR. In the figure, the large outer plate denotes M users, whereas the inner plate represents a user's repeated choices of an interest ($z \in \{1,2,\dots,K\}$) and an app ($i \in I$). The shaded circles in the figure represent observed variables, while unshaded circles denote hidden variables.

1 **for** each interest k =1, 2, …, K
2 Draw $\boldsymbol{\varphi_k} \sim Dir(\boldsymbol{\beta})$
3 **for** each user u_m, m =1, 2, …, M
4 Draw $\boldsymbol{\theta_m} \sim Dir(\boldsymbol{\alpha})$
5 **for** user u_m's n^{th} app download, n = 1, 2, …, N_m
6 Draw an interest $z_{m,n} \sim multi(\boldsymbol{\theta_m})$, $z_{m,n} \in \{1, 2, \dots, K\}$
7 Draw an app $i_{m,n} \sim multi(\boldsymbol{\varphi}_{\boldsymbol{z_{m,n}}})$

Figure 2: LMAR's Generative Process

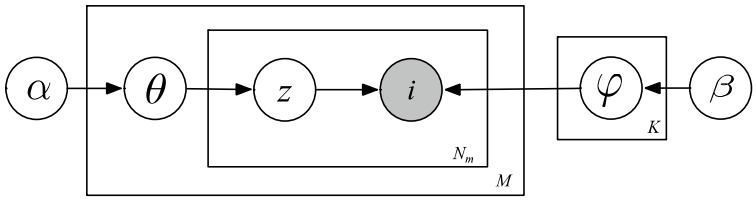

Figure 3: The Graphical Model for LMAR

Notes: $\boldsymbol{\alpha}, \beta$: hyper-parameters; $\boldsymbol{\theta}$: distribution of interests; $\boldsymbol{\varphi}$: distribution of apps; z: an interest; i: an app

The parameters $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ can be inferred from observed download sequences (Heinrich 2008). The probability that user u_m will download app i can then be computed using the following equation:

$$p(i|u_m) = \sum_{k=1}^{K} \theta_{m,k} \varphi_{k,i}, \quad (1)$$

where $\theta_{m,k} \in \boldsymbol{\theta_m}$ denotes user u_m's probability of interest k, k =1,…,K, and $\varphi_{k,i} \in \boldsymbol{\varphi_k}$ is the probability of downloading app i given interest k.

5.2 IMAR: Model

LMAR discovers users' interests based solely on their download behaviors. However,

the browsing sequences preceding downloads are also very valuable for learning users' interests because these browsing behaviors vividly reflect users' decision processes that lead to their downloads. Let us consider two scenarios of downloading the same app, with different interests and different browsing sequences associated with each download. In one scenario, a user is interested in crush games; the user browses and compares several crush games and finally picks *Candy Crush* to download. In the other scenario, a user wants to download an app to kill time and randomly selects *Candy Crush* to download without any browsing beforehand. In this case, LMAR, which only considers download behaviors, cannot differentiate between these two scenarios. However, different browsing behaviors before a download could indicate different interests. It is therefore necessary to integrate both download and browsing behaviors to discover users' interests for mobile app recommendations.

The browsing sequences that precede downloads vary in length. Specifically, some users have long browsing sequences consisting of apps that belong to the same category as the downloaded app, indicating that these users are highly involved in selecting an app to download and thus deliberately browse among alternatives for comparison. Other users have short browsing sequences or even no browsing behavior before an app download, signifying their low-involvement in the app download. To quantify this difference, we introduce the concept of browsing intensity.

*Definition 3 (**Browsing Intensity**)*. The comparison set for an app download is the set of apps included in the browsing sequence associated with the download that belong to the same category as the downloaded app. The browsing intensity

associated with an app download is the size of its comparison set.

For the example shown in Table 1, the browsing intensity associated with the download of *Candy Crush* is 3, while that for the download of *Youku* is 0. We further discretize browsing intensities into F different levels.

According to involvement theory (Beatty et al. 1988; Bloch et al. 1986; Stone 1984), a user's browsing intensity is driven by the user's involvement state. High involvement tends to motivate a user to explore and compare different apps before downloading, and thus the user is more likely to demonstrate a high browsing intensity. Low involvement, on the other hand, more likely results in a low browsing intensity. While a user's involvement state is unobservable, it is reflected by and thus can be represented using the user's browsing intensity. Therefore, an involvement state e, e=1, 2, …, E, can be modeled as a multinomial distribution over F browsing intensity levels with the parameter $\boldsymbol{\pi_e}$, which is an F-dimensional vector with each element denoting the probability of being at a browsing intensity level. Clearly, high-involvement states concentrate more at high browsing intensity levels than low-involvement states. The parameter $\boldsymbol{\pi_e}$ is generated using a *Dirichlet* distribution with an F-vector hyper-parameter $\boldsymbol{\varepsilon}$, i.e., $\boldsymbol{\pi_e} \sim Dir(\boldsymbol{\varepsilon})$.

Different apps can elicit different involvement states. For example, game apps are likely to cause higher involvement than utility apps, due to the fact that game apps have more hedonic value and emotional appeal, which are essential characteristics for eliciting high involvement (Beatty et al. 1988). Considering that an interest is a probability distribution over apps, different interests can result in different states of

involvement as well. For example, an interest concentrating on game apps is more likely to result in high-involvement states than an interest concentrating on utility apps. Therefore, an interest can also be represented as a distribution over involvement states. Formally, each interest k, k=1, 2, …, K, can be represented with a multinomial distribution over involvement states with the parameter $\boldsymbol{\lambda_k}$, which is an E-dimensional vector in which each element denotes the probability of being at the corresponding involvement state. The parameter $\boldsymbol{\lambda_k}$ is generated using a *Dirichlet* distribution with the hyper-parameter $\boldsymbol{\tau}$, i.e., $\boldsymbol{\lambda_k} \sim Dir(\boldsymbol{\tau})$.

Having introduced browsing intensities and involvement states, we are now ready to propose a graphical model for IMAR, as illustrated in Figure 4. Overall, the large outer plate in the figure denotes M users, while the inner plate represents a user's repeated choices of an interest ($z \in \{1,2,\dots,K\}$) and an app ($i \in I$) as well as the user's involvement state ($e \in \{1,2,\dots,E\}$) and browsing intensity level ($f \in \{1,2,\dots,F\}$) associated with the app download. In this figure, the shaded variables i and f are observable, while the rest are hidden. Compared to LMAR, as shown in Figure 3, the right half of Figure 4 is new. In particular, the key novelties of IMAR include: (i) the introduction of an involvement state e and a browsing intensity level f, which capture users' browsing behaviors, and (ii) the link from the interest z to the involvement state e, which connects users' download and browsing behaviors for learning their interests. As shown in Figure 4, the interest z is not only linked to the downloaded app i but also connected to the involvement state e. Thus, IMAR integrates both download and browsing behaviors to discover users' interests for

mobile app recommendations.

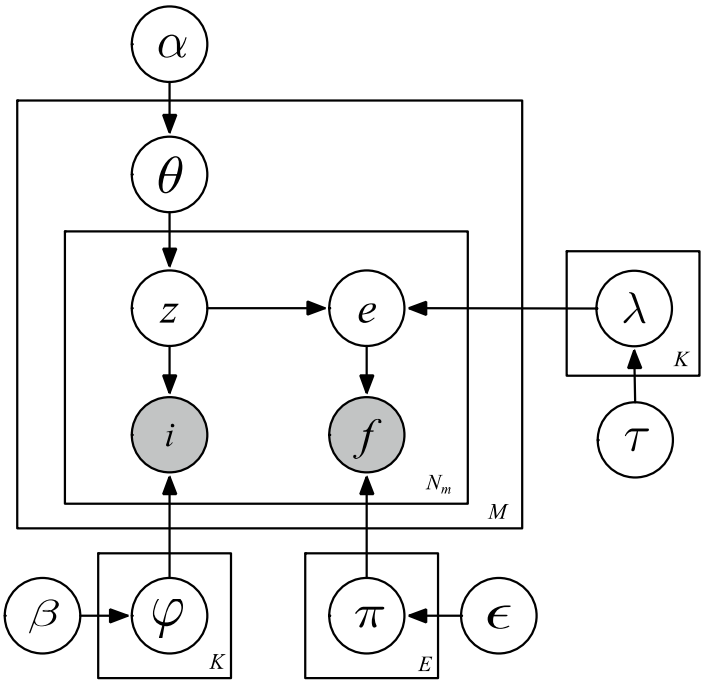

Figure 4: The Graphical Model for IMAR

Notes: $\boldsymbol{\alpha}, \beta, \boldsymbol{\tau}$: hyper-parameters; $\boldsymbol{\theta}$: distribution of interests; $\boldsymbol{\varphi}$: distribution of apps; z: an interest; i: an app; $\boldsymbol{\pi}$: distribution of browse intensities; $\boldsymbol{\lambda}$: distribution of involvement states; e: an involvement state; f: a browsing intensity level

Due to the two key novelties of IMAR, its generative process (illustrated in Figure 5) differs from that of LMAR (shown in Figure 2) in the following ways. First, in addition to being represented as a distribution over apps, the interest in IMAR is also represented with a multinomial distribution over involvement states, and the involvement states are modeled as multinomial distributions over browsing intensity levels (lines 3–5 in Figure 5). Second, the generation of a download behavior is accompanied by the generation of a browsing intensity level. While a download behavior (lines 9–10 in Figure 5) is generated the same way as in LMAR, the generation of a browsing intensity level (lines 11–12 in Figure 5) is new. Specifically,

an involvement state $e_{m,n}$ is selected from the selected interest's multinomial distribution over involvement states, i.e., $e_{m,n} \sim multi(\boldsymbol{\lambda}_{\boldsymbol{z_{m,n}}})$, and then a browsing intensity level $f_{m,n}$ is drawn from the selected involvement state's multinomial distribution over browsing intensity levels, i.e., $f_{m,n} \sim multi(\boldsymbol{\pi}_{\boldsymbol{e_{m,n}}})$.

1 **for** each interest k =1, 2, …, K
2 Draw $\boldsymbol{\varphi_k} \sim Dir(\boldsymbol{\beta})$
3 Draw $\boldsymbol{\lambda_k} \sim Dir(\boldsymbol{\tau})$
4 **for** each involvement state e =1, 2, …, E
5 Draw $\boldsymbol{\pi_e} \sim Dir(\boldsymbol{\varepsilon})$
6 **for** each user u_m, m =1, 2, …, M
7 Draw $\boldsymbol{\theta_m} \sim Dir(\boldsymbol{\alpha})$
8 **for** user u_m's n^{th} app download, where $n = 1, 2, \dots, N_m$
9 Draw an interest $z_{m,n} \sim multi(\boldsymbol{\theta_m})$
10 Draw an app $i_{m,n} \sim multi(\boldsymbol{\varphi}_{\boldsymbol{z_{m,n}}})$
11 Draw an involvement state $e_{m,n} \sim multi(\boldsymbol{\lambda}_{\boldsymbol{z_{m,n}}})$
12 Draw a browsing intensity level $f_{m,n} \sim multi(\boldsymbol{\pi}_{\boldsymbol{e_{m,n}}})$

Figure 5: IMAR's Generative Process

5.3 IMAR: Model Learning

In this subsection, we show how to estimate parameters in IMAR from observed download and browsing sequences. We first focus on learning the hidden variables $\boldsymbol{z} = (z_{m,n})$ (i.e., interests) and $\boldsymbol{e} = (e_{m,n})$ (i.e., involvement states), where $m = 1,2, \dots, M$ and $n = 1,2, \dots, N_m$. To estimate these hidden variables, the key learning problem is to infer the probability $p(\boldsymbol{z}, \boldsymbol{e} | \boldsymbol{i}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ of interests and involvement states given observed app downloads $\boldsymbol{i} = (i_{m,n})$, browsing intensity levels $\boldsymbol{f} = (f_{m,n})$ and the hyper-parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}$, and $\boldsymbol{\varepsilon}$. Given that $z_{m,n}$ takes on K possible values and $e_{m,n}$ takes on E possible values, $(\boldsymbol{z}, \boldsymbol{e})$ has $(KE)^{\sum_{m=1}^{M} N_m}$ possible values. Such a large number of possible values for $(\boldsymbol{z}, \boldsymbol{e})$ makes it infeasible to directly learn $p(\boldsymbol{z}, \boldsymbol{e} | \boldsymbol{i}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ for real-world mobile app recommendation

problems. To address this issue, we propose a method based on the collapsed Gibbs Sampling framework (Heinrich 2008).

A Gibbs sampler samples one dimension at a time, conditioned on the given values of all other dimensions (Heinrich 2008). In this study, a dimension is an interest $z_{m,n}$ or an involvement state $e_{m,n}$. Therefore, following the framework of collapsed Gibbs Sampling, we need to estimate the probabilities $p(z_{m,n}|\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{i}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ and $p(e_{m,n}|\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{i}, \boldsymbol{z}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ for m =1,2, …, M and $n = 1,2, \ldots, N_m$, where vector $\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})}$ is vector $\mathbf{z}$ with element $z_{m,n}$ excluded and vector $\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})}$ is vector $\boldsymbol{e}$ with element $e_{m,n}$ excluded. We note that the only variable in $p(z_{m,n}|\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{i}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ is $z_{m,n}$, while the values of $\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{i}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}$, and $\boldsymbol{\varepsilon}$ are given. Similarly, the only variable in $p(e_{m,n}|\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{i}, \boldsymbol{z}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ is $e_{m,n}$, while the values of $\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{i}, \boldsymbol{z}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}$, and $\boldsymbol{\varepsilon}$ are given. According to Equations (2) and (3), $p(z_{m,n}|\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{i}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ is proportional to $p(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ and $p(e_{m,n}|\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{i}, \boldsymbol{z}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ is proportional to $p(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$:

$$p(z_{m,n}|\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon}) = \frac{p(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})}{p(\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})} \propto p(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon}), \quad (2)$$

$$p(e_{m,n}|\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{z}, \boldsymbol{i}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon}) = \frac{p(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})}{p(\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{z}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})} \propto p(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon}). \quad (3)$$

We note that $p(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ in Equation (2) is different from $p(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ in Equation (3) because the only variable in the former is $z_{m,n}$, while the only variable in the latter is $e_{m,n}$.

According to the graphical model of IMAR as shown in Figure 4, $p(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ can be computed as follows:

$$p(\boldsymbol{z},\boldsymbol{e},\boldsymbol{i},\boldsymbol{f}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}) = \int\int\int\int p(\boldsymbol{z},\boldsymbol{i},\boldsymbol{e},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{\lambda},\boldsymbol{\pi}|\ \boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon})d\boldsymbol{\theta}d\boldsymbol{\varphi}d\boldsymbol{\lambda}d\boldsymbol{\pi}$$

$$= \int p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})\,d\boldsymbol{\theta}\int p(\boldsymbol{i}|\boldsymbol{\varphi},\boldsymbol{z})\ \ p(\boldsymbol{\varphi}|\boldsymbol{\beta})d\boldsymbol{\varphi}$$

$$\times \int p(\boldsymbol{e}|\boldsymbol{\lambda},\boldsymbol{z})p(\boldsymbol{\lambda}|\boldsymbol{\tau})\,d\boldsymbol{\lambda}\int p(\boldsymbol{f}|\boldsymbol{\pi},\boldsymbol{e})\,p(\boldsymbol{\pi}|\boldsymbol{\varepsilon})d\boldsymbol{\pi}. \quad (4)$$

By replacing each probabilistic term *p*(.) in Equation (4) with its corresponding density function and integrating out the parameters $\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{\lambda}$, and $\boldsymbol{\pi}$, we obtain Equations (5) and (6) (derivations of these two equations are given in Appendix B):

$$p\left(z_{m,n}\middle|\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})},\boldsymbol{e},\boldsymbol{i},\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}\right) \propto$$

$$(\alpha_{z_{m,n}} + c^{-(m,n)}_{m,z_{m,n},*,*,*}) \times \frac{\beta_{i_{m,n}} + c^{-(m,n)}_{*,z_{m,n},i_{m,n},*,*}}{\sum_{i=1}^{V}\beta_i + c^{-(m,n)}_{*,z_{m,n},i,*,*}} \times \frac{\tau_{e_{m,n}} + c^{-(m,n)}_{*,z_{m,n},*,e_{m,n},*}}{\sum_{e=1}^{E}\tau_e + c^{-(m,n)}_{*,z_{m,n},*,e,*}}, \quad (5)$$

$$p(e_{m,n}|\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})},\boldsymbol{z},\boldsymbol{i},\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}) \propto (\tau_{e_{m,n}} + c^{-(m,n)}_{*,z_{m,n},*,e_{m,n},*}) \times \frac{\varepsilon_{f_{m,n}} + c^{-(m,n)}_{*,*,*,e_{m,n},f_{m,n}}}{\sum_{f=1}^{F}\varepsilon_f + c^{-(m,n)}_{*,*,*,e_{m,n},f}}. \quad (6)$$

In Equations (5) and (6), $c_{m,z,i,e,f}$ denotes the number of app *i* downloaded by user u_m due to interest *z* and with involvement state *e* and browsing intensity level *f*. The notation * represents aggregation on an index of $c_{m,z,i,e,f}$. For example, $c_{m,z,*,*,*}$ denotes the number of app downloads by user u_m due to interest *z*, regardless of which apps, involvement states and browsing intensity levels are involved. The notation $-(m,n)$ represents the exclusion of the user u_m's n^{th} app download. For example, $c^{-(m,n)}_{m,z,*,*,*}$ denotes the number of app downloads by user u_m due to interest *z*, excluding the user's n^{th} download, regardless of which apps, involvement states and browsing intensity levels are involved. According to Equation (5), the probability distribution of interest $z_{m,n}$ depends on the download behavior $i_{m,n}$ (i.e., the second term of Equation (5)) and the involvement state $e_{m,n}$ (i.e., the third term of Equation (5)). Thus, Equation (5) shows that IMAR integrates both download and browsing

behaviors to discover users' interests.

Next, we show how the parameters $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\pi}$ are estimated in IMAR. As each of these parameters follows a multinomial distribution with a *Dirichlet* prior, we can estimate them using the *Dirichlet-Multinomial* conjugate property (Heinrich 2008). Specifically, the expected estimations of these parameters are as follows:

$$\theta_{m,k} = \frac{\alpha_k + c_{m,k,*,*,*}}{\sum_{k=1}^{K} \alpha_k + c_{m,k,*,*,*}}, \tag{7}$$

$$\varphi_{k,i} = \frac{\beta_i + c_{*,k,i,*,*}}{\sum_{i=1}^{V} \beta_i + c_{*,k,i,*,*}}, \tag{8}$$

$$\lambda_{k,e} = \frac{\tau_e + c_{*,k,*,e,*}}{\sum_{e=1}^{2} \tau_e + c_{*,k,*,e,*}}, \tag{9}$$

$$\pi_{e,f} = \frac{\varepsilon_f + c_{*,*,*,e,f}}{\sum_{f=1}^{F} \varepsilon_f + c_{*,*,*,e,f}}, \tag{10}$$

where m =1, 2, …, M, k =1, 2, …, K, e =1, 2, …, E, $f = 1, 2, \dots, F$, and $i \in I$. Derivations of Equations (7) – (10) are given in Appendix C.

Having discussed the estimation formulas for IMAR's model parameters, we now propose the algorithm for estimating these parameters. As illustrated in Figure 6, the algorithm randomly initializes the hidden variables $z_{m,n}$ and $e_{m,n}$ for $m = 1, 2, \dots, M$ and $n = 1, 2, \dots, N_m$ (line 1 in Figure 6). An iterative process, as shown in lines 3–7 in Figure 6, follows. The iterative process first updates $z_{m,n}$ and $e_{m,n}$ by sampling them according to Equations (5) and (6), for $m = 1, 2, \dots, M$ and $n = 1,2, \dots, N_m$; then, it calculates the parameters $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\pi}$ using Equations (7) – (10), based on $z_{m,n}$ and $e_{m,n}$, where $m = 1, 2, \dots, M$ and $n = 1, 2, \dots, N_m$. The iterative process terminates when perplexities $Perp(\boldsymbol{i})$ and $Perp(\boldsymbol{f})$ both converge. Perplexity measures how good an estimated model fits the observed data (Heinrich 2008). In this study, $Perp(\boldsymbol{i})$ and $Perp(\boldsymbol{f})$ measure how good the estimated

parameters fit the observed app downloads $\boldsymbol{i}$ and browsing intensity levels $\boldsymbol{f}$, respectively. Specifically, $Perp(\boldsymbol{i})$ and $Perp(\boldsymbol{f})$ are defined in Equations (11) and (12) respectively:

$$Perp(\boldsymbol{i}) = exp\left(-\frac{\sum_{m=1}^{M}\sum_{n=1}^{N_m} logP(i_{m,n})}{\sum_{m=1}^{M} N_m}\right), \tag{11}$$

$$Perp(\boldsymbol{f}) = exp\left(-\frac{\sum_{m=1}^{M}\sum_{n=1}^{N_m} logP(f_{m,n})}{\sum_{m=1}^{M} N_m}\right), \tag{12}$$

where $logP(i_{m,n})$ and $logP(f_{m,n})$ denote the log-likelihood of the observed app download $i_{m,n}$ and the browsing intensity level $f_{m,n}$, respectively. According to IMAR's graphical structure, $logP(i_{m,n})$ and $logP(f_{m,n})$ are calculated using Equations (13) and (14):

$$logP(i_{m,n}) = log(\sum_{k=1}^{K} \theta_{m,k}\varphi_{k,i_{m,n}}), \tag{13}$$

$$logP(f_{m,n}) = log(\sum_{k=1}^{K} \theta_{m,k}\sum_{e=1}^{E} \lambda_{k,e}\pi_{e,f_{m,n}}). \tag{14}$$

The running time of the model learning algorithm is linear in $\sum_{m=1}^{M} N_m$, suggesting that it can be applied to real-world mobile app recommendation problems with large numbers of users and downloads.

Input: observed download sequences $\boldsymbol{i}$, observed browsing intensities $\boldsymbol{f}$,$\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon}$, K.
Output: estimations for $\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{\theta}, \boldsymbol{\varphi}, \boldsymbol{\lambda}, \boldsymbol{\pi}$

Randomly initialize $z_{m,n}$ and $e_{m,n}$ for m =1, 2, …, M and $n = 1, 2, \ldots, N_m$
do
  for m =1, 2, …, M
    for $n = 1, 2, \ldots, N_m$
      Sample $z_{m,n}$ according to its distribution given in Equation (5)
      Sample $e_{m,n}$ according to its distribution given in Equation (6)
  Calculate $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$, $\boldsymbol{\lambda}$, $\boldsymbol{\pi}$ according to Equations (7) to (10)
until $Perp(\boldsymbol{i})$ and $Perp(\boldsymbol{f})$ both converge
Output estimations for $\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{\theta}$, $\boldsymbol{\varphi}$, $\boldsymbol{\lambda}$, $\boldsymbol{\pi}$

Figure 6: Model Learning in IMAR

5.4 IMAR: Recommendation Strategy

While $\boldsymbol{\theta_m}$, as learned in the previous subsection, represents a user's overall interests in apps, the user's most recent browsing behaviors (i.e., browsing behaviors after the user's last download) reveal her or his current interest. To make effective mobile app recommendations, both overall interests and the current interest need to be considered. In this section, we first show how to learn a user's current interest based on her or his most recent browsing behaviors, and then we propose how overall interests and the current interest can be integrated to make app recommendations.

Let $\boldsymbol{b}_m =< i_{m,1}, \dots, i_{m,j}, \dots, i_{m,J_m} >$ be user u_m's most recent browsing behaviors, where $i_{m,j} \in I$ is an app browsed by u_m and $j = 1, 2, \dots, J_m$. Using Table 1 as an example, user 1's most recent browsing behaviors is $\boldsymbol{b_1}$=< *Kids Math, Kids Reading*>. Let $\boldsymbol{b}$ denote the collection of all users' most recent browsing behaviors, i.e., $\boldsymbol{b} = \cup_m \boldsymbol{b}_m$. Our objective is to infer from $\boldsymbol{b}_m$ the corresponding recent sequence of interests $\boldsymbol{x}_m =< x_{m,1}, \dots, x_{m,j}, \dots, x_{m,J_m} >$, where $x_{m,j} \in \{1,2, \dots, K\}$ is an interest. To this end, we follow the framework of collapsed Gibbs Sampling by iteratively sampling each dimension (e.g., $x_{m,j}$) conditioned on the given values of all other dimensions. Specifically, we need to compute the probabilities $p\left(x_{m,j} \middle| \boldsymbol{x}_{-(m,j)}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{i}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right)$, where vector $\boldsymbol{x}_{-(m,j)}$ is vector $\boldsymbol{x}$ with element $x_{m,j}$ excluded and $\boldsymbol{x} = \cup_m \boldsymbol{x}_m$, for m =1, 2, …, M, and $j = 1, 2, \dots, J_m$. The only variable in $p\left(x_{m,j} \middle| \boldsymbol{x}_{-(m,j)}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{i}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right)$ is $x_{m,j}$, while the values of $\boldsymbol{x}_{-(m,j)}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{i}, \boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are given. Comparing $p\left(x_{m,j} \middle| \boldsymbol{x}_{-(m,j)}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{i}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right)$ with $p\left(z_{m,n} \middle| \boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon}\right)$ in Equation (5), we note that both $x_{m,j}$ and $z_{m,n}$

are interests. However, the overall interest $z_{m,n}$ is different from the current interest $x_{m,j}$ in that the former is conditioned on the browsing intensity $\boldsymbol{f}$ but the latter is not, because the current interest $x_{m,j}$ is learned from the most recent browsing behaviors and the browsing intensity $\boldsymbol{f}$ is defined for downloads and not for browses. Hence, we can approximate $p\left(x_{m,j}\middle|\boldsymbol{x}_{-(m,j)},\boldsymbol{b},\boldsymbol{z},\boldsymbol{i},\boldsymbol{\alpha},\boldsymbol{\beta}\right)$ with Equation (15), which is obtained by appropriately modifying Equation (5) (the details of obtaining Equation (15) are given in Appendix D):

$$p\left(x_{m,j}\middle|\boldsymbol{x}_{-(m,j)},\boldsymbol{b},\boldsymbol{z},\boldsymbol{i},\boldsymbol{\alpha},\boldsymbol{\beta}\right) \propto$$

$$\left(\alpha_{x_{m,j}} + d_{m,x_{m,j},*}^{-(m,j)}\right) \times \frac{\beta_{i_{m,j}} + c_{*,x_{m,j},i_{m,j},*,*} + d_{*,x_{m,j},i_{m,j}}^{-(m,j)}}{\sum_{i=1}^{V} \beta_i + c_{*,x_{m,j},i,*,*} + d_{*,x_{m,j},i}^{-(m,j)}}, \qquad (15)^4$$

where $d_{m,x_{m,j},*}^{-(m,j)}$ denotes the number of browses in the most recent browsing behaviors $\boldsymbol{b}_m$ due to interest $x_{m,j}$, excluding the j^{th} browse in $\boldsymbol{b}_m$, regardless of which apps are browsed; $c_{*,x_{m,j},i_{m,j},*,*}$ denotes the total number of app $i_{m,j}$ downloads due to interest $x_{m,j}$ in all download sequences; and $d_{*,x_{m,j},i_{m,j}}^{-(m,j)}$ denotes the total number of app $i_{m,j}$ browses due to interest $x_{m,j}$ in all most recent browsing behaviors, excluding the j^{th} browse in $\boldsymbol{b}_m$.

Figure 7 provides the algorithm that learns recent sequences of interests $\boldsymbol{x}$ from most recent browsing behaviors $\boldsymbol{b}$. First, the algorithm randomly initializes variables $x_{m,j}$ for m =1, 2, …, M, and $j = 1, 2, \ldots, J_m$. Then it iteratively samples $x_{m,j}$

---

[4] When implementing our method for a real-world recommendation task, the second term (i.e., $\frac{\beta_{i_{m,j}} + c_{*,x_{m,j},i_{m,j},*,*} + d_{*,x_{m,j},i_{m,j}}^{-(m,j)}}{\sum_{i=1}^{V} \beta_i + c_{*,x_{m,j},i,*,*} + d_{*,x_{m,j},i}^{-(m,j)}}$) in Equation (15) can be approximated using $\varphi_{x_{m,j},i_{m,j}}$ as calculated in Equation (8) during the model learning phase. This practice is consistent with the literature (Heinrich 2008) and can greatly speed up our method's recommendation phase.

according to Equations (15), for m =1, 2, …, M, and $j = 1, 2, \dots, J_m$, until the perplexity $Perp(\boldsymbol{b})$ converges. The calculation of $Perp(\boldsymbol{b})$ is given in Appendix D. Once we have discovered a user's recent sequence of interests $\boldsymbol{x}_m =< x_{m,1}, \dots, x_{m,j}, \dots, x_{m,J_m} >$ from her or his most recent browsing behaviors. The user's current interest, denoted as $\tilde{x}$, is the last interest in $\boldsymbol{x}_m$, i.e., $\tilde{x} = x_{m,J_m}$.

Input: observed download sequences $\boldsymbol{i}$, observed most recent browsing behaviors $\boldsymbol{b}$, learned interests $\boldsymbol{z}$, K, $\boldsymbol{\alpha}, \boldsymbol{\beta}$
Output: recent sequences of interests $\boldsymbol{x}$

Randomly initialize $x_{m,j}$ for m =1, 2, …, M, and $j = 1, 2, \dots, J_m$
do
  for m =1, 2, …, M
    for $j = 1, 2, \dots, J_m$
      sample $x_{m,j}$ according to its distribution given in Equation (15)
until $Perp(\boldsymbol{b})$ converges
Output estimations for $\boldsymbol{x}$

Figure 7: Learning Recent Interest Sequences from Most Recent Browsing Behaviors

Next, we propose Equation (16) to predict user u_m's probability of downloading app i, given the user's current interest $\tilde{x}$ and overall interests $\boldsymbol{\theta_m}$:

$$p(i|\tilde{x}, \boldsymbol{\theta_m}) = \lambda_{\tilde{x},H} \times \varphi_{\tilde{x},i} + (1 - \lambda_{\tilde{x},H}) \times \sum_{k=1}^{K} \theta_{m,k} \varphi_{k,i}, \quad (16)$$

where $\lambda_{\tilde{x},H}$ denotes the probability that the current interest $\tilde{x}$ is at the high-involvement state, $(1 - \lambda_{\tilde{x},H})$ denotes the probability that the current interest $\tilde{x}$ is at the low-involvement state, $\varphi_{\tilde{x},i}$ and $\varphi_{k,i}$ are the respective probabilities of downloading app i given the interests $\tilde{x}$ and k, and $\theta_{m,k}$ is the probability of user u_m's overall interest k. The parameters $\lambda_{\tilde{x},H}$, $\varphi_{\tilde{x},i}$, $\varphi_{k,i}$ and $\theta_{m,k}$ are learned by the model learning algorithm as shown in Figure 6. In Equation (16), we follow the common practice of dichotomizing the degree of involvement as low or high (Engel et al. 1993). We identify low or high involvement state according to its distribution over

browsing intensity levels, e.g., the low-involvement state concentrates more on low browsing intensity levels than the high-involvement state.

According to the involvement literatures (Laurent and Kapferer 1985; Moe 2003), a high-involvement interest will cause the user to focus greatly on this interest; as a result, the user will repeatedly browse and compare apps of this interest and may eventually download an app of this interest. Therefore, if a user's current interest is at the high-involvement state, it is wise to formulate recommendations to the user according to her or his current interest (i.e., $\varphi_{\tilde{x},i}$). In this study, the probability that the current interest $\tilde{x}$ is at the high-involvement state is $\lambda_{\tilde{x},H}$. Hence, we use $\lambda_{\tilde{x},H} \times \varphi_{\tilde{x},i}$ (i.e., the first term of Equation (16)) to model the influence of a user's current interest on her or his download decision. On the other hand, a low-involvement interest is volatile (Hoyer and Ridgway 1984; Van Trijp et al. 1996). Hence, if a user's current interest is at the low-involvement state, the user might switch to other interests and eventually download an app not belonging to the current interest. Therefore, if a user's current interest is at the low-involvement state, it is safer to make recommendations according to her or his overall interests (i.e., $\sum_{k=1}^{K} \theta_{m,k}\varphi_{k,i}$) rather than the current interest. In this study, the probability that the current interest $\tilde{x}$ is at the low-involvement state is $(1 - \lambda_{\tilde{x},H})$. Hence, we use $(1 - \lambda_{\tilde{x},H}) \times \sum_{k=1}^{K} \theta_{m,k}\varphi_{k,i}$ (i.e., the second term of Equation (16)) to capture the impact of the user's overall interests on her or his download decision.[5]

[5]To make recommendations for users who are unseen in the training data set, namely the cold-start problem (Schein et al. 2002), we first employ the algorithm in Figure 7 to infer a user's current interest $\tilde{x}$ from the user's most recent browsing behaviors. Since the user is unseen in the training data set, we do not know the user's overall

6. EMPIRICAL EVALUATIONS

We evaluated our proposed method using data collected from Qihoo 360 Mobile Assistant, one of the largest mobile app platforms in China. In this section, we first describe the data and evaluation procedure, and then report and analyze our evaluation results.

6.1 Data and Evaluation Procedure

The evaluation dataset consists of 3.7 million behavioral log records for 113,004 users between October 1, 2015 and December 7, 2015. Each record consists of the user ID, the app name, the app category, the behavior type (i.e., downloading or browsing), and a timestamp, indicating who downloads or browses a particular app and when. Please refer to Table 1 for an example of behavioral log records. In recommendation problems, density is defined as the ratio between the average number of unique items selected per user (e.g., unique apps downloaded per user in our study) and the number of unique items available for recommendation; low density indicates the sparsity of an item-user matrix and thus the great challenge for a recommendation problem (Huang et al. 2004). The density of our data set is 0.10% (i.e., $\frac{\bar{N}}{V} \times 100\%$), which is on a par with the densities of mobile app recommendation datasets reported in the literature (Cao et al. 2017; Liu et al. 2015; Shi and Ali 2012), indicating that mobile app recommendation is in general a challenging problem. Table 3 summarizes the statistics of the evaluation dataset.

---

interests $\boldsymbol{\theta}_m$. And the recommendation is based on the user's current interest $\tilde{x}$. Thus, the probability $p(i|\tilde{x})$ of downloading app i, given the user's current interest $\tilde{x}$, is $p(i|\tilde{x}) = \varphi_{\tilde{x},i}$.

Number of Behavioral Log Records	3.7 million
Number of Users (M)	113,004
Number of Unique Mobile Apps (V)	14,057
Average Number of Unique Apps Downloaded Per User ($\bar{N}$)	14.20
Density	0.10%

Table 3: Statistics of the Evaluation Dataset

We divided the evaluation dataset into two parts: a training dataset and a test dataset. The training dataset consists of behavioral log records from October 1, 2015 to November 30, 2015 and the test dataset contains one week of behavioral log records, from December 1, 2015 to December 7, 2015. We first used the training data to train our method and benchmark methods. Next, each method (i.e., ours or a benchmark) generated a list of N app recommendations (hereafter, the recommendation list) for every app download in the test data. The performance of each method was evaluated by comparing every app download with the corresponding recommendation list generated by that method, using widely accepted performance metrics for mobile app recommendation: recall and discounted cumulative gain (Bellogin et al. 2011; Lin et al. 2014; Liu et al. 2015; Shani and Gunawardana 2011).

Recall measures the proportion of app downloads in the test data that are included in their corresponding recommendation lists (Shani and Gunawardana 2011). Let I^d denote the set of all app downloads in the test data and let $i_j^d \epsilon I^d,\ j = 1,2, \dots, |I^d|$, be an app download in I^d. For an app download i_j^d, let R_j denote the corresponding recommendation list generated by a method. We represent whether i_j^d is included in R_j using the indicator function defined in Equation (17):

$$ind(i_j^d, R_j) = \begin{cases} 1, & if\ \ i_j^d \in R_j, \\ 0, & otherwise. \end{cases} \quad (17)$$

Following Shani and Gunawardana (2011), we define recall as

$$recall = \frac{\sum_{j=1}^{|I^d|} ind(i_j^d, R_j)}{|I^d|} \quad . \tag{18}$$

Recall ranges from 0 to 1, with 0 indicating that none of the app downloads in the test data are included in their corresponding recommendation lists and 1 indicating that all app downloads are covered by their corresponding recommendation lists. The higher the value of recall, the better the performance of a recommendation method in terms of the coverage of its generated recommendation lists.[6]

Apps in a recommendation list are ranked according to the probabilities of their being downloaded, as predicted by a recommendation method. That is, the first app in a recommendation list has the highest probability of being downloaded, the second app has the second highest probability of being downloaded, and so forth. Recall measures the coverage of recommendation lists but does not gauge the ranking quality of these lists. For example, recall cannot differentiate a recommendation list that hits an app download with its first recommended app from a recommendation list that hits an app download with its last recommended app. However, in terms of ranking quality, the former is preferred over the latter. Discounted cumulative gain (DCG) measures the ranking quality of a recommendation list (Shani and Gunawardana 2011; Tan et al. 2014). Let $r_{j,l} \epsilon R_j$ be the l^{th}-ranked app in the recommendation list R_j, where $l = 1,2, \dots, N$ and N is the length of a recommendation list. Following

[6] Another performance metric for mobile app recommendation is precision. In our study, precision is defined as: $Precision = \frac{\sum_{j=1}^{|I^d|} \frac{ind(i_j^d, R_j)}{|R_j|}}{|I^d|}$. Since the size of a recommendation list $|R_j|$ is same across app downloads. We have $Precision = Recall/|R_j|$. That is, in our study, precision is proportional to recall. For example, if the size of a recommendation list is 5, $Precision = Recall/5$. Thus, for space consideration, we only report evaluation results of recall in this paper.

Shani and Gunawardana (2011), we define the ranking quality DCG_j of R_j as

$$DCG_j = \sum_{l=1}^{N} \frac{2^{ind(\, i_j^d,\, r_{j,l})} - 1}{log_2(l+1)}, \tag{19}$$

with the indicator function defined as

$$ind(\, i_j^d, r_{j,l}) = \begin{cases} 1, & if \;\; i_j^d = r_{j,l}, \\ 0, & otherwise. \end{cases} \tag{20}$$

According to Equations (19) and (20), DCG_j is 0 if the app download i_j^d is not hit by any app in R_j, and it is 1 if i_j^d is hit by the first app in R_j (i.e., $l = 1$). The value of DCG_j ranges from 0 to 1; the higher the value of DCG_j, the better the ranking quality of R_j. The ranking quality DCG of a recommendation method is the average ranking quality across all recommendation lists generated by the method:

$$DCG = \frac{1}{|I^d|} \sum_{j=1}^{|I^d|} DCG_j. \tag{21}$$

To evaluate the performance of our proposed method, we carefully chose a representative method (or methods) from each category of mobile app recommendation methods to serve as benchmarks. We selected ItemKNN (Sarwar et al. 2001), a representative item-based collaborative filtering method, as a benchmark from that category. ItemKNN is widely employed by mobile app recommender systems (Shi and Ali 2012; Yan and Chen 2011), including Qihoo 360 Mobile Assistant, from which we collected our evaluation data. From the category of latent factor models, we chose two state-of-the-art methods as benchmarks: AoBPR (Rendle and Freudenthaler 2014) and RankALS (Takács and Tikk 2012). For the category of LDA-based recommendation, we chose LMAR, as described in §4.1. LMAR is a popular mobile app recommendation method (Lin et al. 2013; Zhu et al. 2015) and the

performance difference between our method and LMAR can reveal the exact contribution of considering browsing behaviors for mobile app recommendation. As our final benchmark, we included a method that randomly recommends apps. The inclusion of this method provides a comparison baseline; we anticipated that our method and all other benchmark methods would substantially outperform the random recommendation method. We summarize each benchmark method in Table 4 and discuss the implementations and parameter settings for our method and the benchmark methods next.

Method	Category
ItemKNN	Item-based Collaborative Filtering
AoBPR	Latent Factor Models
RankALS	Latent Factor Models
LMAR	LDA-based recommendation
Random Recommendation	–

Table 4: Summary of Benchmark Methods

ItemKNN represents an app as a *M*-dimensional vector; element *m* of the vector, *m*=1, 2, …, *M*, is 1 if user u_m has downloaded the app and is 0 otherwise. The method then computes the similarity between each pair of apps according to their corresponding vectors and surrogates a user's probability of downloading a candidate app using the sum of the similarities between the candidate and its neighborhood apps, which are apps in the intersection of the Top-*Z* most similar apps to the candidate and the apps already downloaded by the user (Sarwar et al. 2001). The matrix factorization methods AoBPR and RankALS map both users and apps onto a low-dimensional latent feature space and then surrogates a user's probability of downloading an app using the dot product of their corresponding user latent factor

and item latent factor (Rendle and Freudenthaler 2014). LMAR and IMAR are implemented using the details described in §5.

To set the parameter value(s) for each method, we followed the common practice of dividing the training data into two parts: model learning and model validation (Friedman et al. 2001). Specifically, we employed the last week of the training data for model validation and the rest of the training data for model learning. We iteratively set the parameter value(s) for a method, trained it using the model learning data, and evaluated its performance (i.e., recall and DCG) using the model validation data, until the best performance was achieved. The parameter value(s) that enabled the method to achieve its best performance on the model validation data was (were) chosen for the method. Specifically, the neighborhood size Z for ItemKNN was set to 200. For AoBPR, the number of latent factors and the regularization parameter were set to 40 and 0.3, respectively. For RankALS, the number of latent factors was set to 20. The parameters for LMAR were set as: $K = 200, \alpha = 0.1, \beta = 0.01$. The parameters for our proposed method, IMAR, were set as: $K = 200, F = 5, \alpha = 0.1, \beta = 0.01, \tau = 0.1, \varepsilon = 0.1$. [7]

6.2 Recommendation Performance

Qihoo 360 Mobile Assistant recommends 5 apps to a user each time.[8] Thus, in our evaluation, we initially set the length N of the recommendation list to 5. Following the evaluation procedure described in the previous subsection, we conducted experiments

[7] We applied the widely-used equal frequency discretization method (Dougherty et al. 1995) to discretize browsing intensities into F different levels. The equal frequency discretization method divides data points into groups such that each group contains the same number of data points.

[8] The number of apps recommended by Qihoo 360 Mobile Assistant is small, i.e., 5. This small number is due to the limited size of a smartphone screen and the limited number of apps to which a user can pay attention to.

to evaluate the performance of our method and the benchmark methods. Table 5 summarizes the performance of all methods in terms of recall and DCG. As shown in the table, IMAR, our proposed method, substantially outperforms all benchmark methods in both recall and DCG. In particular, IMAR's recall is 16.10% higher than that of LMAR (the best performing benchmark method) and IMAR's DCG is 18.02% higher than that of LMAR. Moreover, IMAR surpasses ItemKNN, the mobile app recommendation method used by Qihoo 360 Mobile Assistant, by 23.02% in recall and by 25.16% in DCG. Considering that revenues from mobile apps constitute an important part of Qihoo's $1.8 billion total revenues,[9] such improvement could translate into significant financial gains for Qihoo.

The evaluation results suggest that (i) the app recommendations produced by our method are more likely to meet users' app download requirements than those generated by any of the benchmark methods (i.e., our method has higher recall); and (ii) app recommendations that meet users' download requirements are ranked higher in the recommendation lists generated by our method than those generated by any of the benchmark methods (i.e., our method has higher DCG). This second advantage of our method is particularly useful for mobile app recommendation. Considering the limited size of a smartphone screen, placing an app recommendation that meets a user's requirements at a higher and more prominent position in a recommendation list helps the user find the recommendation more easily. It is also worth noting that the

[9] https://www.last10k.com/sec-filings/qihu

random recommendation method performs the worst among all investigated methods. The extremely low recall and DCG scores obtained by the random recommendation method, which are far below those of our proposed method and the other benchmark methods, point to the challenging nature of the mobile app recommendation problem and highlight the necessity of developing intelligent recommendation methods for mobile app recommendations.[10]

Method	Recall	Improvement by IMAR	DCG	Improvement by IMAR
IMAR (Our Method)	0.0620		0.0393	
LMAR	0.0534	16.10%	0.0333	18.02%
ItemKNN	0.0504	23.02%	0.0314	25.16%
AoBPR	0.0447	38.70%	0.0282	39.36%
RankALS	0.0441	40.59%	0.0271	45.02%
Random Recommendation	0.000366	16839.89%	0.000217	18010.60%

Table 5: Recommendation Performance of IMAR and Benchmark Methods (N=5)

To ensure the robustness of our evaluation, we performed additional experiments by varying the length of the recommendation lists. Tables 6–8 summarize evaluation results for N = 3, 10, and 15, respectively. As these tables show, our method consistently and substantially outperforms every benchmark method in both recall and DCG, across values of N. In particular, our method's performance improvement over LMAR, the best performing benchmark method, ranges from 11.48% to 20.40% in recall and from 14.37% to 21.40% in DCG as N decreases from 15 to 3. Moreover, the performance improvement by our method over any benchmark method increases as the length of the recommendation lists decreases. Considering that only a small

[10] We also develop a variant of our proposed method, namely IMAR-Gaussian. The only difference between IMAR-Gaussian and IMAR is that IMAR-Gaussian models involvement state as a Gaussian distribution over browsing intensities whereas IMAR models involvement state as a multinomial distribution over browsing intensity levels. In Appendix E, we show and explain the outperformance of our method over IMAR-Gaussian.

number of app recommendations can be displayed on a smartphone screen, our method is therefore particularly useful for mobile app recommendations.

Method	Recall	Improvement by IMAR	DCG	Improvement by IMAR
IMAR (Our Method)	0.0419		0.0312	
LMAR	0.0348	20.40%	0.0257	21.40%
ItemKNN	0.0333	25.83%	0.0244	27.87%
AoBPR	0.0282	48.58%	0.0209	49.28%
RankALS	0.0279	50.18%	0.0205	52.20%
Random Recommendation	0.000228	18277.19%	0.000161	19278.88%

Table 6: Recommendation Performance of IMAR and Benchmark Methods (N=3)

Method	Recall	Improvement by IMAR	DCG	Improvement by IMAR
IMAR (Our Method)	0.103		0.0526	
LMAR	0.0904	13.94%	0.0452	16.37%
ItemKNN	0.0862	19.49%	0.0429	22.61%
AoBPR	0.0722	42.66%	0.0371	41.78%
RankALS	0.0714	44.26%	0.0365	44.11%
Random Recommendation	0.000704	14530.68%	0.000325	16084.62%

Table 7: Recommendation Performance of IMAR and Benchmark Methods (N=10)

Method	Recall	Improvement by IMAR	DCG	Improvement by IMAR
IMAR (Our Method)	0.136		0.0613	
LMAR	0.122	11.48%	0.0536	14.37%
ItemKNN	0.115	18.26%	0.0505	21.39%
AoBPR	0.0976	39.34%	0.0438	39.95%
RankALS	0.0952	42.86%	0.0427	43.56%
Random Recommendation	0.00106	12730.19%	0.000420	14495.24%

Table 8: Recommendation Performance of IMAR and Benchmark Methods (N=15)

6.3 Analysis

The evaluation results reported in the previous subsection demonstrate the superiority of our method over the benchmark methods. In this subsection, we analyze why our method outperforms the benchmark methods.

6.3.1 Why Our Method Outperforms the Benchmark Methods

Conceptually, the key difference between our method and the benchmark methods is the consideration of browsing behaviors by our method. Methodologically, our method features two key novelties: (i) learning users' interests by appropriately integrating their browsing and download behaviors (i.e., learning novelty, as detailed in §5.2 and §5.3), and (ii) utilizing users' most recent browsing behaviors for making app recommendations (i.e., recommendation novelty, as detailed in §5.4). Thus, to explain why our method outperforms the benchmark methods, we conducted experiments to show the contribution of each novelty to the superior performance of our method. In these experiments, we dropped the recommendation novelty from our method and replaced it with the existing recommendation strategy, i.e., recommendation based on Equation (1). We denoted the resulting method without the recommendation novelty as IMAR−. We further dropped the learning novelty. The resulting method without both novelties becomes LMAR, as it learns users' interests solely based on their download behaviors and makes recommendations based on Equation (1). As summarized in Table 9, the performance advantage of IMAR over IMAR− represents the contribution of recommendation novelty to the superiority of our method while the performance advantage of IMAR− over LMAR captures the contribution of learning novelty.

Method	Learning Novelty	Recommendation Novelty
IMAR	Yes	Yes
IMAR−	Yes	No
LMAR	No	No

Table 9: Methodological Comparison: IMAR, IMAR−, and LMAR

Tables 10 and 11 report the performance of IMAR, IMAR−, and LMAR. As summarized in these tables, IMAR outperforms IMAR− by 6.25% to 12.63% in recall and by 8.11% to 13.04% in DCG as *N* decreases from 15 to 3. Similarly, the performance improvement by IMAR− over LMAR increases from 4.92% to 6.90% in recall and from 5.78% to 7.39% in DCG as *N* decreases from 15 to 3. These evaluation results reveal that each novelty contributes to the superior performance of our method; The recommendation novelty seems to contribute more than the learning novelty, suggesting the necessity of incorporating most recent browsing behaviors for mobile app recommendation.

Method	**Recall (*N* = 3)**	**Recall (*N* = 5)**	**Recall (*N* = 10)**	**Recall (*N* = 15)**
IMAR (Our Method)	0.0419	0.0620	0.103	0.136
IMAR−	0.0372	0.0565	0.0953	0.128
LMAR	0.0348	0.0534	0.0904	0.122
IMAR over IMAR−	12.63%	9.73%	8.08%	6.25%
IMAR− over LMAR	6.90%	5.81%	5.42%	4.92%

Table 10: Recommendation Performance of IMAR, IMAR− and LMAR: Recall

Method	**DCG (*N* = 3)**	**DCG (*N* = 5)**	**DCG (*N* = 10)**	**DCG (*N* = 15)**
IMAR (Our Method)	0.0312	0.0393	0.0526	0.0613
IMAR−	0.0276	0.0355	0.0481	0.0567
LMAR	0.0257	0.0333	0.0452	0.0536
IMAR over IMAR−	13.04%	10.70%	9.36%	8.11%
IMAR− over LMAR	7.39%	6.61%	6.42%	5.78%

Table 11: Recommendation Performance of IMAR, IMAR− and LMAR: DCG

6.3.2 The Value of Properly Integrating Browsing and Download Behaviors

Having shown that considering browsing behaviors is key to the superior performance of our method, we further demonstrate that properly integrating browsing and download behaviors is also critical to its performance. Premised in involvement

theory, we proposed a novel graphical model (shown in Figure 4) that seamlessly integrates browsing and download behaviors into our method. To demonstrate the importance of properly integrating browsing and download behaviors, we simply modified LMAR by replacing the variable *i* (i.e., app downloads) in its graphical model shown in Figure 3 with a new variable that represents both app downloads and app browses. As a result, the modified LMAR, namely LMAR+B, uses both download and browsing behaviors for mobile app recommendation.

Tables 12 and 13 compare the recommendation performances of IMAR, LMAR, and LMAR+B. It is important to note that both IMAR and LMAR+B consider browsing and download behaviors, while LMAR relies on download behaviors only. As expected, IMAR substantially outperforms LMAR. However, LMAR+B performs even worse than LMAR, even though LMAR+B considers both browsing and download behaviors. The underperformance of LMAR+B is partly due to the essential difference between download and browsing behaviors. Apps downloaded by a user generally indicate the user's likes of these apps, whereas apps browsed by a user are often a mixture of likes and dislikes. LMAR+B simply combines browsing and download behaviors and does not differentiate between the two kinds of behaviors. As a consequence, LMAR+B cannot accurately discover users' interests and thus makes poor recommendations. IMAR, on the other hand, premised in involvement theory, differentiates download behaviors from browsing behaviors to make mobile app recommendations. In IMAR, a user's download behaviors reveal the user's interests in apps, whereas a user's browsing behaviors reflect the user's decision process that

lead to her or his app downloads. Thus, IMAR can accurately discover users' interests based on their download and browsing behaviors and makes effective recommendations. Appendix F provides sample interests discovered by IMAR. In short, the evaluation results in Tables 12 and 13 suggest that the proper integration of browsing and download behaviors (as in IMAR) can improve the performance of mobile app recommendations. On the other hand, an inappropriate combination of browsing and download behaviors (as in LMAR+B) can even degrade the performance of mobile app recommendations.

Method	**Recall (*N* = 3)**	**Recall (*N* = 5)**	**Recall (*N* = 10)**	**Recall (*N* = 15)**
IMAR (Our Method)	0.0419	0.0620	0.103	0.136
LMAR	0.0348	0.0534	0.0904	0.122
LMAR+B	0.0303	0.0449	0.0747	0.100
IMAR over LMAR	20.40%	16.10%	13.94%	11.48%
LMAR over LMAR+B	14.85%	18.93%	21.02%	22.00%

Table 12: Recommendation Performance of IMAR, LMAR, and LMAR+B: Recall

Method	**DCG (*N* = 3)**	**DCG (*N* = 5)**	**DCG (*N* = 10)**	**DCG (*N* = 15)**
IMAR (Our Method)	0.0312	0.0393	0.0526	0.0613
LMAR	0.0257	0.0333	0.0452	0.0536
LMAR+B	0.0227	0.0287	0.0385	0.0451
IMAR over LMAR	21.40%	18.02%	16.37%	14.37%
LMAR over LMAR+B	13.22%	16.03%	17.40%	18.85%

Table 13: Recommendation Performance of IMAR, LMAR, and LMAR+B: DCG

Altogether, the superior performance of our method can be attributed to its proper integration of users' download and browsing behaviors for discovering their interests based on involvement theory as well as its utilization of users' most recent browsing behaviors for recommending apps.

7. CONCLUSION

Given the ubiquitous and critical role of mobile apps in people's lives as well as the sheer size of the mobile app market, developing effective mobile app recommendation methods that can help users easily locate the mobile apps they desire is critical for both mobile app users and platforms. Toward this end, we propose a novel method that integrates both download and browsing behaviors for making mobile app recommendations. Using data collected from one of the largest mobile app platforms in China, we demonstrate and analyze the superior performance of our method over several state-of-the-art mobile app recommendation methods.

Our study makes several research contributions and implications. First, rooted in involvement theory, we propose a novel method that considers both download and browsing behaviors for making mobile app recommendations, which contrasts with existing methods that rely on download behaviors but neglect browsing behaviors. Thus, our study contributes to the extant information systems literature by adding a novel method to the growing list of data analytics methods that address critical business and societal problems (e.g., Abbasi et al. 2010; Fang et al. 2013; Fang and Hu 2018). Second, we develop a novel graphical model for inferring users' interests and involvement states from their download and browsing behaviors, design a new algorithm for learning the model parameters, and propose an innovative mobile app recommendation strategy. The proposed graphical model, model learning algorithm, and recommendation strategy represent the methodological contribution of our study. Third, through extensive experiments with a real-word dataset, we demonstrate the

superior performance of our method over several prevalent benchmark methods. Our experimental results also shed light on the essential role of browsing behaviors for effective mobile app recommendations. Finally, our study demonstrates the critical role of theory in design science research. In particular, our method is premised in involvement theory (Beatty et al. 1988; Bloch et al. 1986; Zaichkowsky 1985), which serves as the kernel theory that motivates and guides the development of our method (Gregor and Hevner 2013). Therefore, our study suggests that informing and guiding IT artifact design using theories is a viable way to rigorous design science research (Gregor and Hevner 2013).

Our study also offers several implications for business. First, mobile app platforms can use our method to enhance the performance of mobile app recommendations, which could in turn increase their revenues and profits as well as improve users' satisfaction with their platforms. As we show in the empirical study, our method outperforms state-of-the-art methods from representative existing research and salient industry practices by 16.10% to 40.59% in terms of recall and by 18.02% to 45.02% in terms of DCG. In light of the $69.7 billion annual mobile app revenue, which is expected to grow,[11] the successful application of our method could generate significant financial gains for mobile app platforms. Furthermore, compared to state-of-the-art mobile app recommendation methods, the more accurate recommendations provided by our method make it more likely that users will easily locate the apps they desire and thus improve their experience and satisfaction with

[11] https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/

mobile app platforms. Second, our method is ready to be deployed for real-world large-scale mobile app recommendations. Taking Qihoo 360 Mobile Assistant as an example, one of the largest mobile app platform in China. When a user clicks on an app on this platform, the user will be directed to the page describing the details of the app; five other mobile apps, namely "related recommendations", will also be displayed on this page. Our proposed method can be used to generate related recommendations. Considering the huge size of real-world datasets that are used to learn our model parameters, our method's model learning phase can be deployed offline. Since the model parameters are relatively stable within a given time period, they can be updated periodically (e.g., every week). Our method's recommendation phase produces app recommendations based on learned model parameters and users' most recent browsing data, which have a much smaller magnitude than a model learning dataset. Thus, the recommendation phase can be deployed online to generate real-time mobile app recommendations. Third, the applicability of our method is not limited to mobile app recommendations; it can also be used to produce recommendations in other domains. Take product recommendation systems, which are commonly deployed on ecommerce websites, as an example. In this example, products correspond to mobile apps and product purchases correspond to app downloads. Consumers often browse alternative products for comparison before a purchase, which generates browsing behaviors. Moreover, involvement theory plays a crucial role in explaining and predicting consumers' product purchase behaviors (Laurent and Kapferer 1985). Therefore, our method can be easily extended to the

product recommendation domain.

There are several areas that merit future research attention. Our study models an interest as a distribution over involvement states. A promising future research direction would be to differentiate interest-involvement distributions among users and model a personalized interest-involvement distribution for each user. The challenge for this extension is that the number of parameters for involvement increases by M times, where M is the number of users. Learning such a huge number of model parameters poses a great computational challenge and demands novel and efficient learning algorithms. Another direction worthy of future research would be to incorporate addition factors such as familiarity and rational/emotional involvement into our proposed IMAR model. In addition to involvement, familiarity also plays an important role in driving users' browsing behaviors. For example, users tend to browse less for a familiar app. Besides high or low involvement studied in the paper, another dimension of involvement worthy of consideration is rational or emotional involvement, which refers to the degree to which reason or emotion influences an app download decision. In addition, while we follow a widely accepted practice in evaluating our method using archival data, it would also be worthwhile to conduct laboratory or field experiments to evaluate the method. In a laboratory or field experiment, we can observe in real time how users react to recommended apps and which recommended apps they actually download. By combining laboratory or field experiments and evaluations with archival data, we can produce more comprehensive evaluations of our method. Finally, additional evaluations of our proposed method in

other domains can help generate more empirical evidence regarding its effectiveness and practical value. For example, future work could evaluate the performance of our method for product recommendations.

References

Abbasi, A., Zhang, Z., Zimbra, D., Chen, H., and Nunamaker Jr., J. F. 2010. "Detecting Fake Websites: The Contribution of Statistical Learning Theory," *MIS Quarterly* (34:3), pp. 435-461.

Beatty, S. E., Homer, P., and Kahle, L. R. 1988. "The Involvement—Commitment Model: Theory and Implications," *Journal of Business Research* (16:2), pp. 149-167.

Bellogin, A., Castells, P., and Cantador, I. 2011. "Precision-Oriented Evaluation of Recommender Systems: An Algorithmic Comparison," *Proceedings of the Fifth ACM Conference on Recommender Systems*, pp. 333-336.

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. "Latent Dirichlet Allocation," *Journal of Machine Learning Research* (3), pp. 993-1022.

Bloch, P. H. 1986. "The Product Enthusiast: Implications for Marketing Strategy," *Journal of Consumer Marketing* (3:3), pp. 51-62.

Bloch, P. H., Sherrell, D. L., and Ridgway, N. M. 1986. "Consumer Search: An Extended Framework," *Journal of Consumer Research* (13:1), pp. 119-126.

Cao, D., Nie, L., He, X., Wei, X., Shen, J., Wu, S., and Chua, T.-S. 2017. "Version-Sensitive Mobile App Recommendation," *Information Sciences* (381), pp. 161-175.

Chaiken, S. 1980. "Heuristic Versus Systematic Information Processing and the Use of Source Versus Message Cues in Persuasion," *Journal of Personality and Social Psychology* (39:5), pp. 752-766.

Chaudhuri, A. 2000. "A Macro Analysis of the Relationship of Product Involvement and Information Search: The Role of Risk," *Journal of Marketing Theory and Practice* (8:1), pp. 1-15.

Chen, H., Chiang, R.H. and Storey, V.C. 2012. "Business Intelligence and Analytics: From Big Data to Big Impact," *MIS Quarterly* (36:4), pp.1165-1188.

Dougherty, J., Kohavi, R., and Sahami, M. 1995. "Supervised and Unsupervised Discretization of Continuous Features," *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 194-202.

Engel, J. F., Blackwell, R. D., and Miniard, P. W. 1993. "Consumer Behavior, 8th edition," *Fort Worth, TX: Dryden*.

Fang, X., and Hu, P. J. 2018. "Top Persuader Prediction for Social Networks," MIS Quarterly (42:1), pp. 63-82.

Fang, X., Hu, P.J., Li, Z., and Tsai, W. 2013. "Predicting Adoption Probabilities in Social Networks,"Information Systems Research (24:1), pp.128–145.

Fleder, D., and Hosanagar, K. 2009. "Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity," *Management Science* (55:5), pp. 697-

712.
Friedman, J., Hastie, T., and Tibshirani, R. 2001. *The Elements of Statistical Learning*. Springer series in statistics Springer, Berlin.
Ghose, A. and Han, S.P. 2014. "Estimating Demand for Mobile Applications in the New Economy," *Management Science* (60:6), pp.1470-1488.
Gregor, S. and Hevner, A. 2013. "Positioning and Presenting Design Science Research for Maximum Impact," MIS Quarterly (37:2), pp. 337–355.
Gu, B., Park, J., and Konana, P. 2012. "The Impact of External Word-of-Mouth Sources on Retailer Sales of High-Involvement Products," *Information Systems Research* (23:1), pp. 182-196.
Harris, G. 1987. "The Implications of Low-Involvement Theory for Advertising Effectiveness," *International Journal of Advertising*, (6:3), pp. 207-221.
He, J., and Liu, H. 2017. "Mining Exploratory Behavior to Improve Mobile App Recommendations," *ACM Transactions on Information Systems* (35:4), article 32.
Heinrich, G. 2008. "Parameter Estimation for Text Analysis," *University of Leipzig, Technical Report*.
Houston, M. J., and Rothschild, M. L. 1978. "Conceptual and Methodological Perspectives on Involvement, Educators Proceedings," Research Frontiers in Marketing Dialogues & Directions, pp.184-187.
Hoyer, W. D., and Ridgway, N. M. 1984. "Variety Seeking as an Explanation for Exploratory Purchase Behavior: A Theoretical Model," *NA-Advances in Consumer Research* (11), pp. 114-119.
Huang, Z., Chen, H., and Zeng, D. 2004. "Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering," *ACM Transactions on Information Systems* (22:1), pp. 116-142.
Jacobs, B. J., Donkers, B., and Fok, D. 2016. "Model-Based Purchase Predictions for Large Assortments," *Marketing Science* (35:3), pp. 389-404.
Jannach, D., and Hegelich, K. 2009. "A Case Study on the Effectiveness of Recommendations in the Mobile Internet," *Proceedings of the Third ACM Conference on Recommender Systems*: ACM, pp. 205-208.
Kannan, P., Chang, A.-M., and Whinston, A. B. 2001. "Wireless Commerce: Marketing Issues and Possibilities," *Proceedings of the 34th Annual Hawaii International Conference on System Sciences,* paper 6.
Karatzoglou, A., Baltrunas, L., Church, K., and Hmer, M. 2012. "Climbing the App Wall: Enabling Mobile App Discovery through Context-Aware Recommendations," *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 2527-2530.
Laurent, G., and Kapferer, J.-N. 1985. "Measuring Consumer Involvement Profiles," *Journal of Marketing Research*, pp. 41-53.
Li, L., Wang, D., Li, T., Knox, D. and Padmanabhan, B. 2011. July. "Scene: a scalable two-stage personalized news recommendation system," *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 125-134.
Lin, J., Sugiyama, K., Kan, M. Y., and Chua, T. S. 2013. "Addressing Cold-Start in App

Recommendation: Latent User Models Constructed from Twitter Followers," *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 283-292.

Lin, J., Sugiyama, K., Kan, M. Y., and Chua, T. S. 2014. "New and Improved: Modeling Versions to Improve App Recommendation," *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 647-656.

Linden, G., Smith, B., and York, J. 2003. "Amazon. Com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing* (7:1), pp. 76-80.

Liu, B., Kong, D., Cen, L., Gong, N. Z., Jin, H., and Xiong, H. 2015. "Personalized Mobile App Recommendation: Reconciling App Functionality and User Privacy Preference," *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 315-324.

Liu, Q., Chen, E., Xiong, H., Ge, Y., Li, Z., and Wu, X. 2014. "A Cocktail Approach for Travel Package Recommendation," *IEEE Transactions on Knowledge and Data Engineering* (26:2), pp. 278-293.

McColl-Kennedy, J. R., and Fetter Jr, R. E. 2001. "An Empirical Examination of the Involvement to External Search Relationship in Services Marketing," *Journal of Services Marketing* (15:2), pp. 82-98.

Michaelidou, N., and Dibb, S. 2008. "Consumer Involvement: A New Perspective," *Marketing Review*, 8(1), pp. 83-99.

Mitchell, A. A. 1979. "Involvement: A Potentially Important Mediator of Consumer Behavior," *Advances in Consumer Research* (6), pp. 191-196.

Mittal, B. 1989. "Must Consumer Involvement Always Imply More Information Search?" *NA-Advances in Consumer Research* (16), pp. 167-172.

Moe, W. W. 2003. "Buying, Searching, or Browsing: Differentiating between Online Shoppers Using in-Store Navigational Clickstream," *Journal of Consumer Psychology* (13:1), pp. 29-39.

Natarajan, N., Shin, D., and Dhillon, I. S. 2013. "Which App Will You Use Next?: Collaborative Filtering with Interactional Context," *Proceedings of the Seventh ACM Conference on Recommender Systems*, pp. 201-208.

Nicolau, J. L. 2013. "Direct Versus Indirect Channels: Differentiated Loss Aversion in a High-Involvement, Non-Frequently Purchased Hedonic Product," *European Journal of Marketing* (47:1), pp. 260-278.

Park, D. H., Fang, Y., Liu, M., and Zhai, C. 2016. "Mobile App Retrieval for Social Media Users Via Inference of Implicit Intent in Social Media Text," *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 959-968.

Pathak, B., Garfinkel, R., Gopal, R. D., Venkatesan, R., and Yin, F. 2010. "Empirical Analysis of the Impact of Recommender Systems on Sales," *Journal of Management Information Systems* (27:2), pp. 159-188.

Rendle, S., and Freudenthaler, C. 2014. "Improving Pairwise Learning for Item Recommendation from Implicit Feedback," *Proceedings of the Seventh ACM International Conference on Web Search and Data Mining*, pp. 273-282.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2001. "Item-Based Collaborative Filtering Recommendation Algorithms," *Proceedings of the Tenth International Conference on World Wide Web*, pp. 285-295.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. 2002. "Methods and Metrics for Cold-Start Recommendations," *International ACM SIGIR Conference on Research and Development in Information Retrieval* (39), pp. 253-260.

Shani, G., and Gunawardana, A. 2011. "Evaluating Recommendation Systems," in *Recommender Systems Handbook*. Springer, pp. 257-297.

Shi, K., and Ali, K. 2012. "Getjar Mobile Application Recommendations with Very Sparse Datasets," *Proceedings of of the ACM International Conference on Knowledge Discovery and Data Mining*, pp. 204-212.

Stone, R. N. 1984. "The Marketing Characteristics of Involvement," *Advances in Consumer Research,* (11:4), pp. 210-215.

Takács, G., and Tikk, D. 2012. "Alternating Least Squares for Personalized Ranking," *Proceedings of the Sixth ACM Conference on Recommender Systems*, pp. 83-90.

Tan, C., Liu, Q., Chen, E., Xiong, H., and Wu, X. 2014. "Object-Oriented Travel Package Recommendation," *ACM Transactions on Intelligent Systems and Technology* (5:3), paper 43.

Van Trijp, H. C., Hoyer, W. D., and Inman, J. J. 1996. "Why Switch? Product Category: Level Explanations for True Variety-Seeking Behavior," *Journal of Marketing Research*, pp. 281-292.

Yan, B., and Chen, G. 2011. "Appjoy: Personalized Mobile Application Discovery," *Proceedings of the Ninth International Conference on Mobile Systems, Applications, and Services*: ACM, pp. 113-126.

Yin, P., Luo, P., Lee, W. C., and Wang, M. 2013. "App Recommendation: A Contest between Satisfaction and Temptation," *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 395-404.

Zaichkowsky, J. L. 1985. "Measuring the Involvement Construct," *Journal of Consumer Research* (12:3), pp. 341-352.

Zhu, H., Chen, E., Xiong, H., Yu, K., Cao, H., and Tian, J. 2015. "Mining Mobile User Preferences for Personalized Context-Aware Recommendation," *ACM Transactions on Intelligent Systems and Technology* (5:4), paper 58.

Appendix A: Notation

I = set of mobile apps

V = the number of mobile apps

U = set of app users

M = the number of app users

K = the number of interests

E = the number of involvement states

F = the number of browsing intensity levels

$i_{m,n}$= the n^{th} app downloaded by user u_m

$f_{m,n}$= browsing intensity level associated with $i_{m,n}$

$z_{m,n}$= interest associated with $i_{m,n}$

$e_{m,n}$= involvement state associated with $i_{m,n}$

$\boldsymbol{\theta_m}$= K-dimensional interest distribution for user u_m

$\theta_{m,k} \in \boldsymbol{\theta_m}$= user u_m's probability of interest k, k =1,…,K

$\varphi_{k,i} \in \boldsymbol{\varphi_k}$ = probability of downloading app i given interest k

$\boldsymbol{\varphi_k}$=V-dimensional app distribution for interest k

$\boldsymbol{\lambda_k}$=E-dimensional involvement distribution for interest k

$\boldsymbol{\pi_e}$=F-dimensional browsing intensity distribution for involvement state e

$\boldsymbol{b}_m$= most recent browsing behaviors by user u_m

$c_{m,z,i,e,f}$ = number of app i downloaded by user u_m due to interest z and with involvement state e and browsing intensity level f

Appendix B: Derivations of Equations (5) and (6)

B1: Derivation of Equation (5)

We repeat Equation (2):

$$p\left(z_{m,n}\middle|\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})},\boldsymbol{i},\boldsymbol{e},\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}\right)=\frac{p(\boldsymbol{z},\boldsymbol{i},\boldsymbol{e},\boldsymbol{f}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon})}{p\left(\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})},\boldsymbol{i},\boldsymbol{e},\boldsymbol{f}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}\right)}\propto p(\boldsymbol{z},\boldsymbol{i},\boldsymbol{e},\boldsymbol{f}\ \mid\ \boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon})$$

(B1)

We also repeat Equation (4):

$$p(\boldsymbol{z},\boldsymbol{i},\boldsymbol{e},\boldsymbol{f}\ \mid\ \boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon})=\int\int\int\int p(\boldsymbol{z},\boldsymbol{i},\boldsymbol{e},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{\lambda},\boldsymbol{\pi}|\ \boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon})d\boldsymbol{\theta}d\boldsymbol{\varphi}d\boldsymbol{\lambda}d\boldsymbol{\pi}$$

$$=\int\int\int\int p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{i}|\boldsymbol{\varphi},\boldsymbol{z})\,p(\boldsymbol{e}|\boldsymbol{\lambda},\boldsymbol{z})p(\boldsymbol{f}|\boldsymbol{\pi},\boldsymbol{e})p(\boldsymbol{\theta}|\boldsymbol{\alpha})p(\boldsymbol{\varphi}|\boldsymbol{\beta})p(\boldsymbol{\lambda}|\boldsymbol{\tau})p(\boldsymbol{\pi}|\boldsymbol{\varepsilon})d\boldsymbol{\theta}d\boldsymbol{\varphi}d\boldsymbol{\lambda}d\boldsymbol{\pi}$$

$$=\int p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})\,d\boldsymbol{\theta}\int p(\boldsymbol{i}|\boldsymbol{\varphi},\boldsymbol{z})\ \ p(\boldsymbol{\varphi}|\boldsymbol{\beta})d\boldsymbol{\varphi}$$

$$\times\int p(\boldsymbol{e}|\boldsymbol{\lambda},\boldsymbol{z})p(\boldsymbol{\lambda}|\boldsymbol{\tau})\,d\boldsymbol{\lambda}\int p(\boldsymbol{f}|\boldsymbol{\pi},\boldsymbol{e})\,p(\boldsymbol{\pi}|\boldsymbol{\varepsilon})d\boldsymbol{\pi} \qquad \text{(B2)}$$

By integrating Equations (B1) and (B2) and dropping the term $\int p(\boldsymbol{f}|\boldsymbol{\pi},\boldsymbol{e})\,p(\boldsymbol{\pi}|\boldsymbol{\varepsilon})d\boldsymbol{\pi}$ that does not contain variable $z_{m,n}$, we have,

$$p\left(z_{m,n}\middle|\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})},\boldsymbol{i},\boldsymbol{e},\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}\right)$$

$$\propto\int p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})\,d\boldsymbol{\theta}\int p(\boldsymbol{i}|\boldsymbol{\varphi},\boldsymbol{z})\ \ p(\boldsymbol{\varphi}|\boldsymbol{\beta})d\boldsymbol{\varphi}\int p(\boldsymbol{e}|\boldsymbol{\lambda},\boldsymbol{z})p(\boldsymbol{\lambda}|\boldsymbol{\tau}). \qquad \text{(B3)}$$

$$\int p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})\,d\boldsymbol{\theta}\int p(\boldsymbol{i}|\boldsymbol{\varphi},\boldsymbol{z})\ \ p(\boldsymbol{\varphi}|\boldsymbol{\beta})d\boldsymbol{\varphi}\int p(\boldsymbol{e}|\boldsymbol{\lambda},\boldsymbol{z})p(\boldsymbol{\lambda}|\boldsymbol{\tau})$$

$$=\int\prod_{m=1}^{M}p(\boldsymbol{\theta}_{\boldsymbol{m}}|\boldsymbol{\alpha})\prod_{m=1}^{M}\prod_{n=1}^{N_m}p\left(z_{m,n}\middle|\boldsymbol{\theta}_{\boldsymbol{m}}\right)d\,\boldsymbol{\theta}$$

$$\times\int\prod_{k=1}^{K}p(\boldsymbol{\varphi}_{\boldsymbol{k}}|\boldsymbol{\beta})\prod_{m=1}^{M}\prod_{n=1}^{N_m}p\left(i_{m,n}\middle|\boldsymbol{\varphi}_{z_{m,n}}\right)d\boldsymbol{\varphi}$$

$$\times\int\prod_{k=1}^{K}p(\boldsymbol{\lambda}_{\boldsymbol{k}}|\boldsymbol{\tau})\prod_{m=1}^{M}\prod_{n=1}^{N_m}p(e_{m,n}|\boldsymbol{\lambda}_{z_{m,n}})\,d\boldsymbol{\lambda}$$

$$=\int\prod_{m=1}^{M}\frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_{m,k}^{\alpha_k-1}\prod_{m=1}^{M}\prod_{n=1}^{N_m}\theta_{m,z_{m,n}}\,d\boldsymbol{\theta}$$

$$\times$$

$$\int\prod_{k=1}^{K}\frac{\Gamma\left(\sum_{i=1}^{V}\beta_i\right)}{\prod_{i=1}^{V}\Gamma(\beta_i)}\prod_{i=1}^{V}\varphi_{k,i}^{\beta_i-1}\prod_{m=1}^{M}\prod_{n=1}^{N_m}\varphi_{z_{m,n},i_{m,n}}\,d\boldsymbol{\varphi}$$

$$\times\int\prod_{k=1}^{K}\frac{\Gamma\left(\sum_{e=1}^{E}\tau_e\right)}{\prod_{e=1}^{E}\Gamma(\tau_e)}\prod_{e=1}^{E}\lambda_{k,e}^{\tau_e-1}\prod_{m=1}^{M}\prod_{n=1}^{N_m}\lambda_{z_{m,n},e_{m,n}}\,d\boldsymbol{\lambda}$$

by replacing each probabilistic term $p(.)$ with its corresponding density function

$$=\prod_{m=1}^{M}\int\frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_{m,k}^{\alpha_k-1+c_{m,k,*,*,*}}\,d\boldsymbol{\theta}_{\boldsymbol{m}}$$

$$\times\prod_{k=1}^{K}\int\frac{\Gamma\left(\sum_{i=1}^{V}\beta_i\right)}{\prod_{i=1}^{V}\Gamma(\beta_i)}\prod_{i=1}^{V}\varphi_{k,i}^{\beta_i-1+c_{*,k,i,*,*}}\,d\boldsymbol{\varphi}_{\boldsymbol{k}}$$

by replacing the innermost products in each integral with sum

$$\times \prod_{k=1}^{K} \int \frac{\Gamma(\sum_{e=1}^{E} \tau_e)}{\prod_{e=1}^{E} \Gamma(\tau_e)} \prod_{e=1}^{E} \lambda_{k,e}^{\tau_e - 1 + c_{*,k,*,e,*}} \, d\boldsymbol{\lambda_k} \qquad \text{of counts}$$

$$= \prod_{m=1}^{M} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K} \Gamma(\alpha_k + c_{m,k,*,*,*})}{\Gamma(\sum_{k=1}^{K} \alpha_k + c_{m,k,*,*,*})} \int \frac{\Gamma(\sum_{k=1}^{K} \alpha_k + c_{m,k,*,*,*})}{\prod_{k=1}^{K} \Gamma(\alpha_k + c_{m,k,*,*,*})} \prod_{k=1}^{K} \theta_{m,k}^{\alpha_k - 1 + c_{m,k,*,*,*}} \, d\boldsymbol{\theta_m}$$

$$\times \prod_{k=1}^{K} \frac{\Gamma(\sum_{i=1}^{V} \beta_i)}{\prod_{i=1}^{V} \Gamma(\beta_i)} \frac{\prod_{i=1}^{V} \Gamma(\beta_i + c_{*,k,i,*,*})}{\Gamma(\sum_{i=1}^{V} \beta_i + c_{*,k,i,*,*})} \int \frac{\Gamma(\sum_{i=1}^{V} \beta_i + c_{*,k,i,*,*})}{\prod_{i=1}^{V} \Gamma(\beta_i + c_{*,k,i,*,*})} \prod_{i=1}^{V} \varphi_{k,i}^{\beta_i - 1 + c_{*,k,i,*,*}} \, d\boldsymbol{\varphi_k}$$

$$\times \prod_{k=1}^{K} \frac{\Gamma(\sum_{e=1}^{E} \tau_e)}{\prod_{e=1}^{E} \Gamma(\tau_e)} \frac{\prod_{e=1}^{E} \Gamma(\tau_e + c_{*,k,*,e,*})}{\Gamma(\sum_{e=1}^{E} \tau_e + c_{*,k,*,e,*})} \int \frac{\Gamma(\sum_{e=1}^{E} \tau_e + c_{*,k,*,e,*})}{\prod_{e=1}^{E} \Gamma(\tau_e + c_{*,k,*,e,*})} \prod_{e=1}^{E} \lambda_{k,e}^{\tau_e - 1 + c_{*,k,*,e,*}} \, d\boldsymbol{\lambda_k}$$

$$= \prod_{m=1}^{M} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K} \Gamma(\alpha_k + c_{m,k,*,*,*})}{\Gamma(\sum_{k=1}^{K} \alpha_k + c_{m,k,*,*,*})} \qquad \text{by using the fact that all integral terms equal to 1}$$

$$\times \prod_{k=1}^{K} \frac{\Gamma(\sum_{i=1}^{V} \beta_i)}{\prod_{i=1}^{V} \Gamma(\beta_i)} \frac{\prod_{i=1}^{V} \Gamma(\beta_i + c_{*,k,i,*,*})}{\Gamma(\sum_{i=1}^{V} \beta_i + c_{*,k,i,*,*})}$$

$$\times \prod_{k=1}^{K} \frac{\Gamma(\sum_{e=1}^{E} \tau_e)}{\prod_{e=1}^{E} \Gamma(\tau_e)} \frac{\prod_{e=1}^{E} \Gamma(\tau_e + c_{*,k,*,e,*})}{\Gamma(\sum_{e=1}^{E} \tau_e + c_{*,k,*,e,*})} \qquad \text{(B4)}$$

By dropping constant terms that do not contain variable $z_{m,n}$ in Equation (B4), we have,

$$p\left(z_{m,n} \middle| \boldsymbol{z}_{-(m,n)}, \boldsymbol{i}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon}\right)$$

$$\propto \frac{\prod_{k=1}^{K} \Gamma(\alpha_k + c_{m,k,*,*,*})}{\Gamma(\sum_{k=1}^{K} \alpha_k + c_{m,k,*,*,*})} \times \prod_{k=1}^{K} \frac{\Gamma\left(\beta_{i_{m,n}} + c_{*,k,i_{m,n},*,*}\right)}{\Gamma(\sum_{i=1}^{V} \beta_i + c_{*,k,i,*,*})} \times \prod_{k=1}^{K} \frac{\Gamma(\tau_{e_{m,n}} + c_{*,k,*,e_{m,n},*})}{\Gamma(\sum_{e=1}^{E} \tau_e + c_{*,k,*,e,*})} \qquad \text{(B5)}$$

Using the fact that $\Gamma(x+1) = x\Gamma(x)$, the right hand side of Equation (B5) becomes,

$$\frac{\prod_{k=1}^{K} \Gamma(\alpha_k + c_{m,k,*,*,*})}{\Gamma(\sum_{k=1}^{K} \alpha_k + c_{m,k,*,*,*})} \times \prod_{k=1}^{K} \frac{\Gamma\left(\beta_{i_{m,n}} + c_{*,k,i_{m,n},*,*}\right)}{\Gamma(\sum_{i=1}^{V} \beta_i + c_{*,k,i,*,*})} \times \prod_{k=1}^{K} \frac{\Gamma(\tau_{e_{m,n}} + c_{*,k,*,e_{m,n},*})}{\Gamma(\sum_{e=1}^{E} \tau_e + c_{*,k,*,e,*})}$$

$$= \frac{\prod_{k \neq z_{m,n}} \Gamma(\alpha_k + c_{m,k,*,*,*}^{-(m,n)})}{\Gamma(1 + \sum_{k=1}^{K} \alpha_k + c_{m,k,*,*,*}^{-(m,n)})} \times \Gamma(\alpha_{z_{m,n}} + c_{m,z_{m,n},*,*,*}^{-(m,n)}) \times (\alpha_{z_{m,n}} + c_{m,z_{m,n},*,*,*}^{-(m,n)})$$

$$\times \prod_{k \neq z_{m,n}} \frac{\Gamma(\beta_{i_{m,n}} + c_{*,k,i_{m,n},*,*}^{-(m,n)})}{\Gamma(\sum_{i=1}^{V} \beta_i + c_{*,k,i,*,*}^{-(m,n)})} \times \frac{\Gamma(\beta_{i_{m,n}} + c_{*,z_{m,n},i_{m,n},*,*}^{-(m,n)})}{\Gamma(\sum_{i=1}^{V} \beta_i + c_{*,z_{m,n},i,*,*}^{-(m,n)})} \times \frac{\beta_{i_{m,n}} + c_{*,z_{m,n},i_{m,n},*,*}^{-(m,n)}}{\sum_{i=1}^{V} \beta_i + c_{*,z_{m,n},i,*,*}^{-(m,n)}}$$

$$\times \prod_{k \neq z_{m,n}} \frac{\Gamma(\tau_{e_{m,n}} + c_{*,k,*,e_{m,n},*}^{-(m,n)})}{\Gamma(\sum_{e=1}^{E} \tau_e + c_{*,k,*,e,*}^{-(m,n)})} \times \frac{\Gamma(\tau_{e_{m,n}} + c_{*,z_{m,n},*,e_{m,n},*}^{-(m,n)})}{\Gamma(\sum_{e=1}^{E} \tau_e + c_{*,z_{m,n},*,e,*}^{-(m,n)})} \times \frac{\tau_{e_{m,n}} + c_{*,z_{m,n},*,e_{m,n},*}^{-(m,n)}}{\sum_{e=1}^{E} \tau_e + c_{*,z_{m,n},*,e,*}^{-(m,n)}}$$

$$= \frac{\prod_{k=1}^{K} \Gamma(\alpha_k + c_{m,k,*,*,*}^{-(m,n)})}{\Gamma(1 + \sum_{k=1}^{K} \alpha_k + c_{m,k,*,*,*}^{-(m,n)})} \times (\alpha_{z_{m,n}} + c_{m,z_{m,n},*,*,*}^{-(m,n)}) \qquad \text{by refolding the residual } \Gamma\text{-function terms back into their general product}$$

$$\times \prod_{k=1}^{K} \frac{\Gamma(\beta_{i_{m,n}} + c_{*,k,i_{m,n},*,*}^{-(m,n)})}{\Gamma(\sum_{i=1}^{V} \beta_i + c_{*,k,i,*,*}^{-(m,n)})} \times \frac{\beta_{i_{m,n}} + c_{*,z_{m,n},i_{m,n},*,*}^{-(m,n)}}{\sum_{i=1}^{V} \beta_i + c_{*,z_{m,n},i,*,*}^{-(m,n)}}$$

(B6)

$$\times \prod_{k=1}^{K} \frac{\Gamma(\tau_{e_{m,n}} + c_{*,k,*,e_{m,n},*}^{-(m,n)})}{\Gamma(\sum_{e=1}^{E} \tau_e + c_{*,k,*,e,*}^{-(m,n)})} \times \frac{\tau_{e_{m,n}} + c_{*,z_{m,n},*,e_{m,n},*}^{-(m,n)}}{\sum_{e=1}^{E} \tau_e + c_{*,z_{m,n},*,e,*}^{-(m,n)}}$$

By dropping constant terms that do not contain variable $z_{m,n}$ in Equation (B6), we obtain Equation (5).

B2: Derivation of Equation (6)

Equation (6) can be derived in a way similar to that of Equation (5). First, we repeat Equation (3):

$$p\left(e_{m,n}\middle|\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})},\boldsymbol{i},\boldsymbol{z},\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}\right)=\frac{p(\boldsymbol{e},\boldsymbol{i},\boldsymbol{z},\boldsymbol{f}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon})}{p\left(\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})},\boldsymbol{i},\boldsymbol{z},\boldsymbol{f}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}\right)}\propto p(\boldsymbol{e},\boldsymbol{i},\boldsymbol{z},\boldsymbol{f}\ \mid\ \boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon})$$

(B7)

We also repeat Equation (4):

$$p(\boldsymbol{e},\boldsymbol{i},\boldsymbol{z},\boldsymbol{f}\ \mid\ \boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon})=\int\int\int\int p(\boldsymbol{z},\boldsymbol{i},\boldsymbol{e},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{\lambda},\boldsymbol{\pi}|\ \boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon})d\boldsymbol{\theta}d\boldsymbol{\varphi}d\boldsymbol{\lambda}d\boldsymbol{\pi}$$

$$=\int p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})\,d\boldsymbol{\theta}\int p(\boldsymbol{i}|\boldsymbol{\varphi},\boldsymbol{z})\ \ p(\boldsymbol{\varphi}|\boldsymbol{\beta})d\boldsymbol{\varphi}$$

$$\times\int p(\boldsymbol{e}|\boldsymbol{\lambda},\boldsymbol{z})p(\boldsymbol{\lambda}|\boldsymbol{\tau})\,d\boldsymbol{\lambda}\int p(\boldsymbol{f}|\boldsymbol{\pi},\boldsymbol{e})\,p(\boldsymbol{\pi}|\boldsymbol{\varepsilon})d\boldsymbol{\pi} \quad \text{(B8)}$$

By integrating Equations (B7) and (B8) and dropping the term $\int p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})\,d\boldsymbol{\theta}\int p(\boldsymbol{i}|\boldsymbol{\varphi},\boldsymbol{z})\,p(\boldsymbol{\varphi}|\boldsymbol{\beta})d\boldsymbol{\varphi}$ that does not contain the variable $e_{m,n}$, we have,

$$p\left(e_{m,n}\middle|\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})},\boldsymbol{i},\boldsymbol{z},\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}\right)\propto\int p(\boldsymbol{e}|\boldsymbol{\lambda},\boldsymbol{z})p(\boldsymbol{\lambda}|\boldsymbol{\tau})\,d\boldsymbol{\lambda}\int p(\boldsymbol{f}|\boldsymbol{\pi},\boldsymbol{e})\,p(\boldsymbol{\pi}|\boldsymbol{\varepsilon})d\boldsymbol{\pi}$$

$$\int p(\boldsymbol{e}|\boldsymbol{\lambda},\boldsymbol{z})p(\boldsymbol{\lambda}|\boldsymbol{\tau})\,d\boldsymbol{\lambda}\int p(\boldsymbol{f}|\boldsymbol{\pi},\boldsymbol{e})\,p(\boldsymbol{\pi}|\boldsymbol{\varepsilon})d\boldsymbol{\pi}$$

$$=\int\prod_{k=1}^{K}p(\boldsymbol{\lambda_k}|\boldsymbol{\tau})\prod_{m=1}^{M}\prod_{n=1}^{N_m}p\left(e_{m,n}\middle|\boldsymbol{\lambda}_{z_{m,n}}\right)d\boldsymbol{\lambda}$$

$$\times\int\prod_{e=1}^{E}p(\boldsymbol{\pi}_e|\boldsymbol{\varepsilon})\prod_{m=1}^{M}\prod_{n=1}^{N_m}p(f_{m,n}|\boldsymbol{\pi}_{e_{m,n}})d\boldsymbol{\pi}$$

$$=\int\prod_{k=1}^{K}\frac{\Gamma\left(\sum_{e=1}^{E}\tau_e\right)}{\prod_{e=1}^{E}\Gamma(\tau_e)}\prod_{e=1}^{E}\lambda_{k,e}^{\tau_e-1}\prod_{m=1}^{M}\prod_{n=1}^{N_m}\lambda_{z_{m,n},e_{m,n}}\,d\boldsymbol{\lambda}$$

by replacing each probabilistic term $p(.)$ with its corresponding density function

$$\times\int\prod_{e=1}^{E}\frac{\Gamma\left(\sum_{f=1}^{F}\varepsilon_f\right)}{\prod_{f=1}^{F}\Gamma(\varepsilon_f)}\prod_{f=1}^{F}\pi_{e,f}^{\varepsilon_f-1}\prod_{m=1}^{M}\prod_{n=1}^{N_m}\pi_{e_{m,n},f_{m,n}}\,d\boldsymbol{\pi}$$

$$=\prod_{k=1}^{K}\int\frac{\Gamma\left(\sum_{e=1}^{E}\tau_e\right)}{\prod_{e=1}^{E}\Gamma(\tau_e)}\prod_{e=1}^{E}\lambda_{k,e}^{\tau_e-1+c_{*,k,*,e,*}}\,d\boldsymbol{\lambda_k}$$

by replacing the innermost products in each integral with sum of counts

$$\times\prod_{e=1}^{E}\int\frac{\Gamma\left(\sum_{f=1}^{F}\varepsilon_f\right)}{\prod_{f=1}^{F}\Gamma(\varepsilon_f)}\prod_{f=1}^{F}\pi_{e,f}^{\varepsilon_f-1+c_{*,*,*,e,f}}\,d\boldsymbol{\pi_e}$$

$$=\prod_{k=1}^{K}\frac{\Gamma\left(\sum_{e=1}^{E}\tau_e\right)}{\prod_{e=1}^{E}\Gamma(\tau_e)}\frac{\prod_{e=1}^{E}\Gamma(\tau_e+c_{*,k,*,e,*})}{\Gamma\left(\sum_{e=1}^{E}\tau_e+c_{*,k,*,e,*}\right)}\int\frac{\Gamma\left(\sum_{e=1}^{E}\tau_e+c_{*,k,*,e,*}\right)}{\prod_{e=1}^{E}\Gamma(\tau_e+c_{*,k,*,e,*})}\prod_{e=1}^{E}\lambda_{k,e}^{\tau_e-1+c_{*,k,*,e,*}}\,d\boldsymbol{\lambda_k}$$

$$\times\prod_{e=1}^{E}\frac{\Gamma\left(\sum_{f=1}^{F}\varepsilon_f\right)}{\prod_{f=1}^{F}\Gamma(\varepsilon_f)}\frac{\prod_{f=1}^{F}\Gamma(\varepsilon_f+c_{*,*,*,e,f})}{\Gamma\left(\sum_{f=1}^{F}\varepsilon_f+c_{*,*,*,e,f}\right)}\int\frac{\Gamma\left(\sum_{f=1}^{F}\varepsilon_f+c_{*,*,*,e,f}\right)}{\prod_{f=1}^{F}\Gamma(\varepsilon_f+c_{*,*,*,e,f})}\prod_{f=1}^{F}\pi_{e,f}^{\varepsilon_f-1+c_{*,*,*,e,f}}\,d\boldsymbol{\pi_e}$$

$$=\prod_{k=1}^{K}\frac{\Gamma\left(\sum_{e=1}^{E}\tau_e\right)}{\prod_{e=1}^{E}\Gamma(\tau_e)}\frac{\prod_{e=1}^{E}\Gamma(\tau_e+c_{*,k,*,e,*})}{\Gamma\left(\sum_{e=1}^{E}\tau_e+c_{*,k,*,e,*}\right)}$$

by using the fact that all integral terms equal to 1

$$\times\prod_{e=1}^{E}\frac{\Gamma\left(\sum_{f=1}^{F}\varepsilon_f\right)}{\prod_{f=1}^{F}\Gamma(\varepsilon_f)}\frac{\prod_{f=1}^{F}\Gamma(\varepsilon_f+c_{*,*,*,e,f})}{\Gamma\left(\sum_{f=1}^{F}\varepsilon_f+c_{*,*,*,e,f}\right)} \quad \text{(B9)}$$

By dropping constant terms that do not contain the variable $e_{m,n}$ in Equation (B9), we have,

$$p\left(e_{m,n}\middle|\boldsymbol{e}_{-(\boldsymbol{m},\boldsymbol{n})},\boldsymbol{i},\boldsymbol{z},\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\varepsilon}\right) \propto \frac{\prod_{e=1}^{E}\Gamma(\tau_e+c_{*,z_{m,n},*,e,*})}{\Gamma(\sum_{e=1}^{E}\tau_e+c_{*,z_{m,n},*,e,*})}\prod_{e=1}^{E}\frac{\Gamma\left(\varepsilon_{f_{m,n}}+c_{*,*,*,e,f_{m,n}}\right)}{\Gamma\left(\sum_{f=1}^{F}\varepsilon_f+c_{*,*,*,e,f}\right)} \quad \text{(B10)}$$

Using the fact that $\Gamma(x+1)=x\Gamma(x)$, the right-hand side of Equation (B10) becomes,

$$\frac{\prod_{e=1}^{E}\Gamma(\tau_e+c_{*,z_{m,n},*,e,*})}{\Gamma(\sum_{e=1}^{E}\tau_e+c_{*,z_{m,n},*,e,*})}\times\prod_{e=1}^{E}\frac{\Gamma\left(\varepsilon_{f_{m,n}}+c_{*,*,*,e,f_{m,n}}\right)}{\Gamma\left(\sum_{f=1}^{F}\varepsilon_f+c_{*,*,*,e,f}\right)}$$

$$=\frac{\prod_{e\neq e_{m,n}}\Gamma(\tau_e+c^{-(m,n)}_{*,z_{m,n},*,e,*})}{\Gamma(1+\sum_{e=1}^{E}\tau_e+c^{-(m,n)}_{*,z_{m,n},*,e,*})}\times\Gamma(\tau_{e_{m,n}}+c^{-(m,n)}_{*,z_{m,n},*,e_{m,n},*})\times(\tau_{e_{m,n}}+c^{-(m,n)}_{*,z_{m,n},*,e_{m,n},*})$$

$$\times\prod_{e\neq e_{m,n}}\frac{\Gamma\left(\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e,f_{m,n}}\right)}{\Gamma\left(\sum_{f=1}^{F}\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e,f}\right)}\times\frac{\Gamma\left(\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e_{m,n},f_{m,n}}\right)}{\Gamma\left(\sum_{f=1}^{F}\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e_{m,n},f}\right)}\times\frac{\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e_{m,n},f_{m,n}}}{\sum_{f=1}^{F}\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e_{m,n},f}}$$

$$=\frac{\prod_{e=1}^{E}\Gamma(\tau_e+c^{-(m,n)}_{*,z_{m,n},*,e,*})}{\Gamma(1+\sum_{e=1}^{E}\tau_e+c^{-(m,n)}_{*,z_{m,n},*,e,*})}\times(\tau_{e_{m,n}}+c^{-(m,n)}_{*,z_{m,n},*,e_{m,n},*})$$

$$\times\prod_{e=1}^{E}\frac{\Gamma\left(\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e,f_{m,n}}\right)}{\Gamma\left(\sum_{f=1}^{F}\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e,f}\right)}\times\frac{\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e_{m,n},f_{m,n}}}{\sum_{f=1}^{F}\varepsilon_{f_{m,n}}+c^{-(m,n)}_{*,*,*,e_{m,n},f}} \quad \text{(B11)}$$

by refolding the residual Γ-function terms back into their general products

By dropping constant terms that do not contain variable $e_{m,n}$ in Equation (B11), we obtain Equation (6)

Appendix C: Derivations of Equations (7) to (10)

Let $p(\boldsymbol{\theta}_m|\boldsymbol{i},\boldsymbol{f},\boldsymbol{z},\boldsymbol{e},\boldsymbol{\alpha})$ be the posterior distribution of $\boldsymbol{\theta}_m$, $m = 1, \dots, M$, given observed app downloads $\boldsymbol{i}$, browsing intensity levels $\boldsymbol{f}$, learned hidden variables $\boldsymbol{z}$ and $\boldsymbol{e}$, and hyper-parameter $\boldsymbol{\alpha}$. We have,

$$p(\boldsymbol{\theta}_m|\boldsymbol{i},\boldsymbol{f},\boldsymbol{z},\boldsymbol{e},\boldsymbol{\alpha}) = \frac{p(\boldsymbol{\theta}_m|\boldsymbol{\alpha})\prod_{m=1}^{N_m} p(z_{m,n}|\boldsymbol{\theta}_m)}{\int p(\boldsymbol{\theta}_m|\boldsymbol{\alpha})\prod_{m=1}^{N_m} p(z_{m,n}|\boldsymbol{\theta}_m) d\boldsymbol{\theta}_m}$$

$$= \frac{\frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_{m,k}^{\alpha_k-1}\prod_{n=1}^{N_m}\theta_{m,z_{m,n}}}{\int\frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_{m,k}^{\alpha_k-1}\prod_{n=1}^{N_m}\theta_{m,z_{m,n}} d\theta_m}$$

by replacing each probabilistic term $p(.)$ with its corresponding density function

$$= \frac{\frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)\Gamma(\sum_{k=1}^{K}\alpha_k+c_{m,k,*,*,*})}{\prod_{k=1}^{K}\Gamma(\alpha_k)\prod_{k=1}^{K}\Gamma(\alpha_k+c_{m,k,*,*,*})}\prod_{k=1}^{K}\theta_{m,k}^{\alpha_k+c_{m,k,*,*,*}-1}}{\frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\int\frac{\Gamma(\sum_{k=1}^{K}\alpha_k+c_{m,k,*,*,*})}{\prod_{k=1}^{K}\Gamma(\alpha_k+c_{m,k,*,*,*})}\prod_{k=1}^{K}\theta_{m,k}^{\alpha_k+c_{m,k,*,*,*}-1} d\theta_m}$$

by replacing the innermost product with sum of counts and multiplying denominator and nominator by a same term

$$= \frac{\Gamma(\sum_{k=1}^{K}\alpha_k+c_{m,k,*,*,*})}{\prod_{k=1}^{K}\Gamma(\alpha_k+c_{m,k,*,*,*})}\prod_{k=1}^{K}\theta_{m,k}^{\alpha_k+c_{m,k,*,*,*}-1}$$

by using the fact that the integral term in the denominator equal to 1

According to the equation above, given $\boldsymbol{i},\boldsymbol{f},\boldsymbol{z},\boldsymbol{e}$, and $\boldsymbol{\alpha}$, $\boldsymbol{\theta}_m$ follows a *Dirichlet* distribution with K-vector hyper-parameter $\boldsymbol{\alpha_k} + \boldsymbol{c_{m,k,*,*,*}} = (\alpha_1 + c_{m,1,*,*,*}, \dots, \alpha_K + c_{m,K,*,*,*})$. Given a K-dimensional variable $\boldsymbol{X} = (X_1, X_2, \dots, X_K)$, which follows a *Dirichlet* distribution with K-vector hyper-parameter $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_K)$, we know the fact that $E(X_i) = \frac{\alpha_i}{\sum_i \alpha_i}$. Applying this fact to $\boldsymbol{\theta}_m$, we obtain Equation (7).

Equations (8) to (10) can be obtained in a way similar to that of Equation (7).

Appendix D: Derivations of Equation (15) and $Perp(\boldsymbol{b})$

D1: Details of Obtaining Equation (15)

By comparing $p(x_{m,j}|\boldsymbol{x}_{-(m,j)}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{i}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ with $p(z_{m,n}|\boldsymbol{z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{e}, \boldsymbol{i}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\varepsilon})$ in Equation (5), we notice that both $x_{m,j}$ and $z_{m,n}$ are interests. However, overall interest $z_{m,n}$ is different from current interest $x_{m,j}$ in that the former is conditioned on browsing intensity $\boldsymbol{f}$ but the latter is not. Specifically, we obtain Equation (15) by (i) dropping the third term in Equation (5), which is associated with browsing intensity $\boldsymbol{f}$; (ii) replacing parameters in the first two terms of Equation (5) with their corresponding parameters for $x_{m,j}$; (iii) replacing $c_{*,z_{m,n},i_{m,n},*,*}^{-(m,n)}$ in Equation (5) with $c_{*,x_{m,j},i_{m,j},*,*} + d_{*,x_{m,j},i_{m,j}}^{-(m,j)}$ because $x_{m,j}$ in $p(x_{m,j}|\boldsymbol{x}_{-(m,j)}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{i}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is conditioned on both download behaviors $\boldsymbol{i}$ and most recent browsing behaviors $\boldsymbol{b}$.

D2: Calculation of $Perp(\boldsymbol{b})$

We have,

$$Perp(\boldsymbol{b}) = exp(-\frac{\sum_{m=1}^{M}\sum_{j=1}^{J_m} logP(i_{m,j})}{\sum_{m=1}^{M} J_m}), \tag{D1}$$

$$\text{where } logP(i_{m,j}) = log(\sum_{k=1}^{K} \gamma_{m,k}\delta_{k,i_{m,j}}), \tag{D2}$$

$$\gamma_{m,k} = \frac{\alpha_k + d_{m,k,*}}{\sum_{k=1}^{K}\alpha_k + d_{m,k,*}}, \tag{D3}$$

$$\text{and } \delta_{k,i} = \frac{\beta_i + c_{*,k,i,*,*} + d_{*,k,i}}{\sum_{i=1}^{V}\beta_i + c_{*,k,i,*,*} + d_{*,k,i}}. \tag{D4}$$

In Equation (D1), $logP(i_{m,j})$ denotes the log-likelihood of browsing app $i_{m,j}$, which is calculated using Equation (D2). In Equation (D2), $\gamma_{m,k}$ denotes the probability of interest *k* discovered from most recent browsing behaviors $\boldsymbol{b}_m$, and $\delta_{k,i}$ is the probability of browsing app *i* given interest *k*. Equations (D3) and (D4) for calculating $\gamma_{m,k}$ and $\delta_{k,i}$ can be obtained by analogy to Equations (7) and (8), with two changes: (i) $c_{m,k,*,*,*}$ in Equation (7) changes to $d_{m,k,*}$ in Equation (D3); (ii) $c_{*,k,i,*,*}$ in Equation (8) changes to $c_{*,k,i,*,*} + d_{*,k,i}$ in Equation (D4), because $x_{m,j}$ in $p(x_{m,j}|\boldsymbol{x}_{-(m,j)}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{i}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is conditioned on both download behaviors $\boldsymbol{i}$ and most recent browsing behaviors $\boldsymbol{b}$.

Appendix E: Performance Comparison between IMAR and IMAR-Gaussian

We develop a variant of our proposed method, namely IMAR-Gaussian. The only difference between IMAR-Gaussian (Figure E1) and IMAR (Figure 4) is that IMAR-Gaussian models involvement state as a Gaussian distribution over browsing intensities g with mean μ_e and standard deviation σ_e whereas IMAR models involvement state as a multinomial distribution over browsing intensity levels.

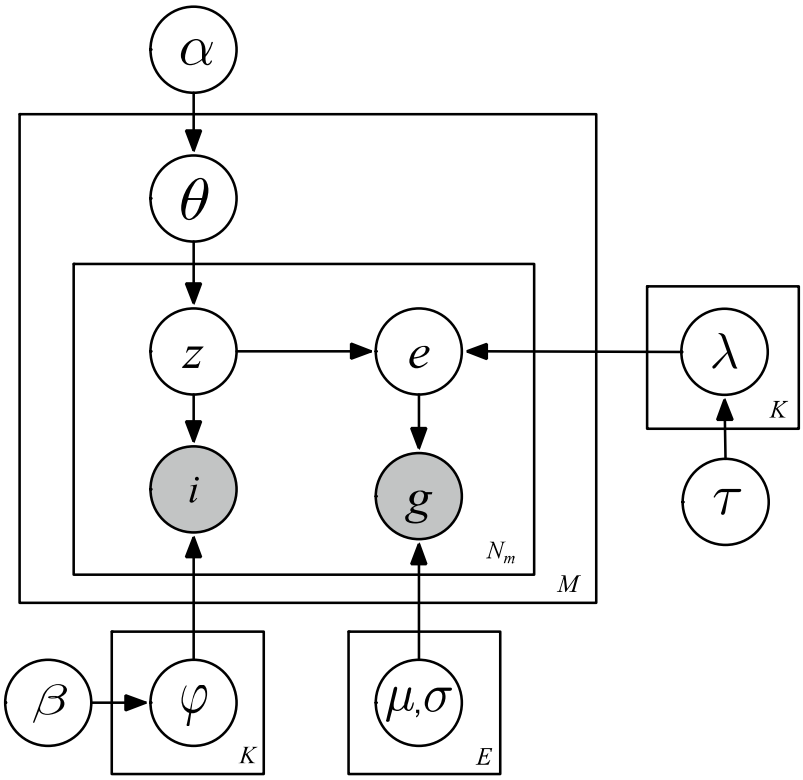

Figure E3: The Graphical Model for IMAR-Gaussian

The model parameters in IMAR-Gaussian are inferred with a combined Gibbs Sampling and EM algorithm. To evaluate its performance, we test IMAR-Gaussian on the same dataset used in this study. As shown in Tables E1 and E2, our method consistently outperforms IMAR-Gaussian in both recall and DCG, as the length N of the recommendation list increases from 3 to 15. One possible explanation of the underperformance of IMAR-Gaussian could be that multinomial distribution allows for a more flexible structure for data modeling than Gaussian distribution. For

example, the distribution of an involvement state over browsing intensities could be skewed. Gaussian distribution, a symmetric distribution, is not a good option for modeling that distribution, whereas multinomial distribution can model skewed distributions well, despite of its discrete characteristic. In addition, we would like to explain why IMAR treats involvement state as a categorical variable. In IMAR, differentiating various involvement states is sufficient for model learning and ordinal information among involvement states is not required for model learning. For example, differentiating between "high involvement" and "low involvement" is sufficient while the ordinal information that one is "higher" than the other is not necessary for model learning. Therefore, in the model learning phase, IMAR treats involvement state as a categorical variables and model it as a multinomial distribution over browsing intensity levels. In the recommendation phase, IMAR identifies low or high involvement state according to its distribution over browsing intensity levels, e.g., the low-involvement state concentrates more on low browsing intensity levels than the high-involvement state.

Method	**Recall (N=3)**	**Recall (N=5)**	**Recall (N=10)**	**Recall (N=15)**
IMAR (Our Method)	0.0419	0.0620	0.1030	0.1360
IMAR-Gaussian	0.0387	0.0587	0.0989	0.131
IMAR over IMAR-Gaussian	8.27%	5.62%	4.15%	3.82%

Table E1: Recommendation Performances of IMAR and IMAR-Gaussian: Recall

Method	**DCG (N=3)**	**DCG (N=5)**	**DCG (N=10)**	**DCG (N=15)**
IMAR (Our Method)	0.0312	0.0393	0.0526	0.0613
IMAR-Gaussian	0.0287	0.0369	0.0498	0.0583
IMAR over IMAR-Gaussian	8.71%	6.50%	5.62%	5.15%

Table E2: Recommendation Performances of IMAR and IMAR-Gaussian: DCG

Appendix F: Sample Interests Discovered by Our Method

In this appendix, we report sample interests discovered by our method. In our method, an interest is represented as a probability distribution over apps. Table F1 lists six interests discovered by our method, along with apps with top download probabilities in each interest. In this table, we manually label each interest according to the top apps in the interest. For example, the top five apps in interest "Racing Games" are *Truck Simulator City*, *Need for Speed Most Wanted*, *Crazy Taxi: Urban Surge*, *High-speed Road Race*, and *Hill Climbing Racing*, with download probabilities of 0.022, 0.021, 0.017, 0.016, and 0.015, respectively.

Our method also discovers the distribution of involvement states for each interest, shown in Table F2. For example, the probabilities that interests "Racing Games", "Mom & Kids" and "Learning English" being at the high-involvement state are 0.999, 0.851, and 0.756 respectively. The top downloaded apps in interests "Racing Games" and "Mom & Kids" are of high hedonic value and emotional appeal and thus can elicit high involvement from users (Nicolau 2013; Zaichkowsky 1985). The interest "Learning English" has a high probability at the high-involvement state because users are highly motivated to improve their English and thus carefully compare alternative apps and select the most appropriate one to download for learning English.

Comparatively, the probabilities that interests "Hot Apps", "Videos" and "Navigation Services" being at the high-involvement state are 0.00004, 0.0003, and 0.0006 respectively. These interests are more likely at the low-involvement state because (i) top downloaded apps of these interests are more utilitarian than hedonic and thus are unlikely to arouse users' involvement; (ii) top downloaded apps of these interests are similar to each other without much differences in attributes. Thus, there is no need for users to carefully compare alternatives before a download. The sample interests discussed in this Appendix show that our method can effectively discover interests and their involvement distributions.

Interest: Racing Games		**Interest: Hot Apps**	
Truck Simulator City	0.022	Wechat	0.119
Need for Speed Most Wanted	0.021	QQ	0.109
Crazy Taxi: Urban Surge	0.017	KuGou Music	0.062
High-speed Road Race	0.016	Paypal	0.061
Hill Climbing Racing	0.015	Mobile Taobao	0.051
Interest: Mom & Kids		**Interest: Videos**	
Kids Hospital	0.034	Youku	0.138
Kids Kindergarten	0.029	IQIYI Video	0.132
Kids Kitchen	0.027	Sohu Video	0.13
Kids Cleaning	0.021	Tudou Video	0.099
Kids Love Eating	0.020	Mango Video	0.094
Interest: Learning English		**Interest: Navigation Services**	
Fluent Oral English	0.059	AutoMap	0.217
Hundred Words Killer	0.057	AutoNavi	0.144
Hj Happy Words	0.043	Baidu Map	0.097
Zhimi Word Tutor	0.041	Google Map	0.072
Palm English	0.039	Tencent Map	0.048

Table F1: Sample Interests Discovered by IMAR

Interest	**Probability at the High-Involvement State**	**Probability at the Low-Involvement State**
Racing Games	0.999	0.001
Mom & Kids	0.851	0.149
Learning English	0.756	0.244
Hot Apps	0.00004	0.99996
Videos	0.0003	0.9997
Navigation Services	0.0006	0.9994

Table F2: Involvement Distributions of the Six Sample Interests

References

Nicolau, J. L. 2013. "Direct Versus Indirect Channels: Differentiated Loss Aversion in a High-Involvement, Non-Frequently Purchased Hedonic Product," *European Journal of Marketing* (47:1/2), pp. 260-278.

Zaichkowsky, J. L. 1985. "Measuring the Involvement Construct," *Journal of Consumer Research* (12:3), pp. 341-352.