PERSONALIZED RECOMMENDATIONS IN MOBILE BUSINESS ENVIRONMENTS

by

BIN LIU

A dissertation submitted to the

Graduate School-Newark

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Management

written under the direction of

Dr. Hui Xiong

and approved by

____

____

____

____

Newark, New Jersey

October 2016

ABSTRACT OF THE DISSERTATION

Personalized Recommendations in Mobile Business Environments

By BIN LIU

Dissertation Director: Dr. Hui Xiong

Recent years have witnessed a rapid adoption of smart mobile devices and their increased pervasiveness into peoples daily life. As a result of this quick development, the demand for better mobile services is increasing with an even faster speed. Recommender systems become essential to deliver the right services to the right mobile users. For instance, Point of Interest (POI) recommendation enables us to recommend the right places to the right users based on their preferences. Also effective mobile app recommendation will help mobile users to make better utility of their smartphones.

In this dissertation, we identify several unique challenges for recommendation in mobile business environments, and then introduce how we use advanced data mining techniques to address these challenges. First, rich semantic information such tags and descriptions are associated with POIs in location based services. To this end, we proposed a *topic and location aware* POI recommender system by exploiting associated textual and context information. Second, many mobile services are location-dependent, which means a users choice can be influenced by location-dependent factors, in particular are the *user mobility* and *geographical influence*. User mobility refers to the factor that a users interest would change among different regions. Geographical influence is related to the cost of the option. Therefore, it is important to capture a users spatial choice behavior to make better recommendations. Along this

line, we have proposed a geographical probabilistic factor model framework, which strategically captures user mobility and geographical influence, to model user spatial choice behavior. Extensive experiments demonstrate the effectiveness of the proposed approach. Third, services are usually organized into hierarchy structure such as category hierarchy. We then introduce a structural user choice model (SUCM) to learn fine-grained user choice patterns by exploiting hierarchy structure. Moreover, we design an efficient learning algorithm to estimate the parameters for the SUCM model. Evaluation on an app adoption data demonstrates that our approach can better capture user choice patterns and thus improve recommendation performance. Finally, privacy becomes a big issue for mobile service adoption. In particular, mobile apps could have privileges to access a user's sensitive resources (e.g., contact, message, and location). As a result, a user chooses an app not only because of its *functionality*, but also because it respects the user's privacy preference. We present the *first* systematic study on incorporating both interest-functionality interactions and users' privacy preferences to perform personalized app recommendations. Moreover, we explore the impact of different levels of privacy information on the performances of our method, which gives us insights on what resources are more likely to be treated as private by users and influence users' behaviors at selecting apps.

ACKNOWLEDGEMENTS

I would like to thank all the people who provided me support and help during my PhD study. First, I would like to express my deep gratitude to my advisor, Prof. Hui Xiong, for his continuous support, guidance and encouragement. I thank him for generously giving me motivation, flexibility, support, opportunities; for helping me improving communication and presentation.

I also sincerely thank my other dissertation committee members: Prof. Martin Ester, Prof. Spiros Papadimitriou, and Prof. Xiaodong Lin. All of them not only provide constructive suggestions and comments on my research work and this dissertation, but also offer numerous support and help in my career development, and I am very grateful for them.

I also want to thank my colleagues and collaborators. I would like to thank group members at the Center of Data Mining and Business Analytics: Yong Ge, Zhongmou Li, Keli Xiao, Chuanren Liu, Zijun Yao, Yanjie Fu, Meng Qu, Constantine Alexander Vitt, Jingyuan Yang, Yanchi Liu, Hao Zhong, Farid Razzak, Qingxin Meng, Junming Liu, Mingfei Teng, Can Chen, Qi Liu, Hengshu Zhu, Chunyu Luo, Xinjiang Lu, Yanhong Guo, Yaoyao Zuo, Jie Liu, Huang Xu, Leilei Sun, Dowen Du, Hongting Niu, Ling Yin, Xiaolin Li, Bo Jin, Aimin Feng, Chang Tan, Xue Bai, LiYang Tang, Yuanchun Zhou, Tong Xu, Guannan Liu, for their help and valuable suggestion. Also

I would like to Dr. Haifeng Chen, Dr. Guofei Jiang, Dr. Yannis Pavlidis, Dr. Wei Shen, Dr. Hongxia Jin, Dr. Deguang Kong and Dr. Bing Hu, with whom I had great time during my internships. Special thanks go to Neil Zhenqiang Gong, Yao Wu and Min Xie, who provided me support and help, and made my research journey more enjoyable.

Finally, I would like to thank my wife, my daughter, and my family for their love, support and understanding. Without their encouragement and help, this dissertation would be impossible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Recent years have witnessed the tremendous growth in mobile devices among an increasing number of users, and the penetration of mobile devices into every component of modern life. As the new platforms for modern life, mobile ecosystems deliver billions of dollars of value to the economy and a huge amount of social capital to society. This dissertation research has been broadly motivated by the following major mobile technology trends:

- The huge and rapid explosion in mobile devices among an increasing number of users, and their penetration into every component of modern life.
- The high penetration of mobile apps among the huge number of mobile users.
- The availability of rich location-based user behavior data generated while mobile users interacting with a diversity of services using mobile devices.

By analyzing the rich data generated by the mobile devices, we can easily observe that mobile technology has revolutionized the computing industry and the society. First, it changes the way how services (*e.g.*, mobile apps) are digitized, monetized and consumed in a **centralized** marketplace (*e.g.*, Google Play) . Second, it changes the way of how services (*e.g.*, location based services) are delivered to, consumed by, and interacted by mobile users in a **decentralized** mobile environment. Third, it changes how people are connected with each other, *e.g.*, the social network is enhanced with rich location and interaction information. Fourth, the complexity of the mobile ecosystem raises new privacy and ethics issues.

Mobile ecosystems are made up of two fundamental components: **people** and **services**, which are connected by interactions in different forms, *e.g.*, app adoptions, location check-ins, and reviews. It is extremely important to analyze and understand user behavior from large-scale mobile data for technical and business innovations in mobile ecosystems.

On the other side, it is always difficult for a user to locate relevant services in mobile environments given the huge number of choices. For example, there are more than 1.4 million apps in centralized marketplaces Google Play and App Store respectively; the number of locations (a.k.a points of interest) in Foursquare reached 65 millions in 2015. Therefore, this dissertation focuses on designing effective and scalable methods to model user behavior from large-scale digital records of user activities, and apply them for effective recommendations in mobile environments.

1.1 Research Contributions

In this dissertation, we identify several unique challenges for recommendation in mobile business environments, and then introduce how we use advanced data mining techniques to address these challenges.

- **Exploit rich semantic and location information for POI recommendation**: Unlike traditional recommendation tasks, POI recommendation is personalized, location-aware, and context depended. In light of this difference, we proposes a *topic and location aware* POI recommender system by exploiting associated textual and context information. Specifically, we first exploit an aggregated latent Dirichlet allocation (LDA) model to learn the interest topics of users and to infer the interest POIs by mining textual information associated with POIs. Then, a Topic and Location-aware probabilistic matrix factorization (TL-PMF) method is proposed for POI recommendation. A unique perspective of TL-PMF is to consider both the extent to which a user interest matches the POI in terms of topic distribution and the word-of-mouth opinions of the POIs. Finally, experiments on real-world LBSNs data show that the proposed recommendation method outperforms state-of-the-art probabilistic latent factor models with a significant margin. Also, we have studied the impact of personalized interest topics and word-of-mouth opinions on POI recommendations.

- **Exploit geographical influences and user mobility for POI recommendation**: The increasing prevalence of LBSNs services, such as Foursquare, poses significant new opportunities and challenges for better recommendations of places (a.k.a POIs) such as restaurants and malls. We investigate *geographical choices* among *point-of-interests* (POIs) in location based social networks (LBSNs). T However, the decision process for a user to choose a POI is complex and can be influenced by numerous factors, such as personal preferences, geographical considerations, and user mobility behaviors. Besides personal preferences, there are two characteristics of LBSNs which distinguish POI recommendation from traditional recommendation tasks (such as movie or music recommendations). First, due to geographical constraints and the cost of traveling large distances, the probability of a user visiting a POI is inversely proportional to the geographic distance between them. Second, users may check into POIs at different regions, *e.g.*, an LBSN user may travel to different cities. Varying *user mobility* imposes huge challenges on POI recommendations, especially when a user arrives at a new city or region. We proposed a general geographical probabilistic factor model (Geo-PFM) framework which strategically takes various factors into consideration. Specifically, this framework allows to capture the geographical influences on a user's check-in behavior. Also, user mobility behaviors can be effectively leveraged in the recommendation model. Moreover, based on our Geo-PFM framework, we further developed a Poisson Geo-PFM which provides a rigorous probabilistic generative process for the entire model and is effective in modeling the skewed user check-in count data as implicit feedback for better POI recommendations. Extensive experimental results on three real-world LBSN datasets (which differ in terms of user mobility, POI geographical distribution, implicit response data skewness, and user-POI observation sparsity) demonstrated the effectiveness of the proposed recommendation methods.

- **Modeling Structured Choices for App Recommendation:** This work studied *structured user choices* among mobile apps in a centralized market, where apps are organized in a hierarchical taxonomy. Furthermore, apps with similar functionalities are competing with each other to win users. We developed a *structured user choice model* (SUCM) to learn fine-grained user choice patterns by exploiting the hierarchical taxonomy of apps as well as the competitive relationships among apps. Since apps are organized in a category tree, we model structured user choice by a unique *choice path* over the tree hierarchy, where the choice path starts from the root of the hierarchy and goes down to the app that is selected by a user. In each step of moving along the choice path, the competitions between the candidates (*i.e.*, either the same level categories/subcategories or apps in a chosen subcategory) play an important role in affecting user's choices. We captured the structured choice procedure by cascading user preferences over the choice paths through a probabilistic model. Seeing each step in traversing the choice path as a discrete choice model, we modeled the probability that a user reaches a certain node in the choice path as a *softmax* of the user's preference on the chosen node over the user's preference on all the nodes at the sample level. The softmax function was used to capture the competitions between categories/subcategories or apps in a category/subcategory. Moreover, we modeled a user's preference over one node using latent factors, which enabled us to capture the correlations between nodes.

Moreover, we designed an efficient learning algorithm to estimate the parameters of the SUCM model. The major challenge of learning the parameters rested in the softmax on the leaf nodes (apps) of the tree hierarchy. Indeed, it is not practical to learn these softmax functions for a subcategory of apps by directly applying Stochastic Gradient Descent (SGD), because the time complexity of one SGD step is linear to the number of apps under the subcategory, which might be very large. To address this challenge, we relaxed the softmax term in each subcategory into a hierarchical softmax, thus the time complexity of learning parameters was reduced to be logarithm of the number of apps under the subcategory. We collected a large-scale dataset from Google Play to evaluate our approach and compared SUCM with state-of-the-art approaches. The experimental results showed that SUCM consistently outperforms these methods with a significant margin in terms of a variety of widely used evaluation metrics for Top-N recommendation.

- **Privacy-aware app recommendation**: Recent years have witnessed a rapid adoption of mobile devices and a dramatic proliferation of mobile applications. However, the large number of mobile apps makes it difficult for users to locate relevant Apps. Therefore, recommending apps becomes an urgent task. Traditional recommendation approaches focus on learning the *interest* of a user and the *functionality* of an item from a set of user-item ratings, and they recommend an item to a user if the item's functionality well matches the user's interest. However, apps could have privileges to access a user's sensitive resources (*e.g.*, contact, message, and location). As a result, a user chooses an app not only because of its functionality, but also because it respects the user's *privacy preference*. We present the *first* systematic study on incorporating both interest-functionality interactions and users' privacy preferences to perform personalized app recommendations. Specifically, we first construct a new model to capture the trade-off between functionality and user privacy preference. Comprehensively evaluations show our method consistently and substantially outperforms the state-of-the-art approaches, which implies the importance of user privacy preference on personalized app recommendations. Moreover, we explore the impact of different levels of privacy information on the performances of our method, which gives us insights on what resources are more likely to be treated as private by users and influence users' behaviors at selecting apps.

CHAPTER 2

TOPIC AND LOCATION AWARE POINT-OF-INTEREST RECOMMENDATION

2.1 Introduction

Recent years have witnessed the increased development of location-based social networking (LBSN) services, such as Foursquare, Facebook Places and Google Latitude. LBSNs allow users to explore Places-of-Interests (POIs) for better services through sharing check-in experiences and opinions on the POIs they have checked in. For example, in Foursquare, users can (1) categorize a POI to help describe what type of places this POI is; (2) tag a POI to let people know what they can expect from it; (3) share their experiences of check-ins with others; (4) know how many people have visited a specific POI and how much time they spent there.

Indeed, the task of Point-of-Interest (POI) recommendation is to provide personalized recommendations of places of interest. It plays an important role in providing better location based services in location based social networks. Both LBSN users and POI owners are expected to have effective POI recommendations. For owners, they could have more targeted customers. Also, for users, they could identify the most relevant POIs and have better user experiences.

Unlike traditional recommendation tasks, POI recommendation is personalized, location-aware, and context depended. This can be illustrated by the following scenario. Bob lives in the New York City, usually he has a coffee in the morning at a Starbucks near his home, then has his lunch at an Italian restaurant near his office. Also, he prefers to hang out with his friends at a certain bar before he returns home. At weekends, he sometimes go to the Central Park with his family. Now, if Bob would spend the holiday in Florida, then what kind of POIs Bob would be interested in for his trip? This POI recommendation will certainly be personalized, location-aware, and context depended.

The development of POI recommender systems is much more complex than the development of traditional recommender systems. The reasons are as follows. First, for POI recommendations, the users' interest can vary dramatically at different time and locations. For instance, what POIs should we recommend to a resident in the New York City when he travels to Florida? Second, the LBSN user behaviors are intrinsically spatio-temporally correlated. The heterogeneous nature of spatio-temporal data is a big challenge for recommendation. Third, a POI is usually associated with categories and tags to describe the POI. However, unlike traditional recommendation (i.e. article recommendation (Wang & Blei, 2011)), the textual information associated with POIs is usually incomplete and ambiguous. Finally, even two POIs with similar or even the same semantic topics can be ranked differently if they are in two different regions.

In light of the above challenges, we propose a topic and location aware method for POI recommendation. The proposed method allows to effectively exploit the textual information associated with POIs to better profile users and POIs, as well as to take into account of context aware information. Then, we develop a *Topic and*

Location-aware probabilistic matrix factorization (TL-PMF) method for POI recommendation based on the learned user and POI topic distribution, and simultaneously incorporating location information. A unique perspective of this proposed method is to consider both the extent to which a user interest matches the POI in terms of topic distribution and the word-of-mouth opinions of the POI.

Finally, experimental results on real-world LBSNs data show that the proposed POI recommendation method outperforms state-of-the-art probabilistic latent factor models with a significant margin in terms of both prediction and Top-N recommendation.

2.2 Problem Formulation

Here, we consider POI recommendations in LBSNs. Intuitively, a user chooses a POI at a given time by matching her/his personal preferences with the service content of that POI. A user would have her/his own taste for the choice of POIs, and the personal preference can be represented by an interest topic distribution. However, even two POIs with similar or the same semantic terms can be rated differently if they are located differently. For example, a certain kind of outdoor recreation is very popular in warm and sunny California can be much less popular in a chilly northeastern area. Therefore, to provide better personalized recommendations of POIs, we need to consider both the extent to which a user's interest matches a POI in terms of topics as well as the word-of-mouth opinions of the POI.

Typically, there is textual and location-aware information associated with a POI as shown above in Table 2.1, which can be mined to improve location services. LBSNs such as Foursquare allow users to (1) categorize a POI; (2) tag a POI; (3) record how many different people have visited a POI and the total number of visits to this POI. As a result, the category and tag words provide semantic information about this POI. Meanwhile, the check-in numbers provide important local popularity information of that POI, which represents the word-of-mouth opinion of the POI.

From an example of POI and its associated information in Table 2.1, we can know detailed semantic and location information of this POI. The textual information, the categories and tags, provides meaningful semantics which can be presented in terms of topics. The last two numbers, the total number of people associated with and total number of visits to the POI, indicate the word-of-mouth opinion of the POI. The larger these numbers, the more popular this POI is in this area.

Formally, we are given the historical check-in records $R_{M\times N}$ of M LBSN users $U = \{u_1, u_2, ..., u_M\}$ and N POIs $C = \{c_1, c_2, ..., c_N\}$ with r_{ij} as the number of times user u_i checked in POI c_j. r_{ij} is similar to the rating score of user u_i for item c_j in general recommendation setting. Also, for each POI, we have additional profile information such as location information, regional information in terms of city and state names, textual information in terms of categorical and tag words, and the regional popularity score P_j of POI c_j in terms of how many people associated with and how many times people visited this POI. Categories and tags are words that are assigned to describe the POI. So we have a document d_{c_j} for each POI c_j.

We build the location aware recommender system by exploring the textual and context information associated with the POIs. We argue that the rating r_{ij} of user u_i for a POI c_j is determined by two factors: (1) The extent to which the POI's interest

Name	Columbia Heights Coffee
Address	3416 11th Street Northwest, Washington, DC 20010
Categories	Coffee Shop, General Entertainment, Sandwich Place
Tags	lounge chairs, tea, closes early, hipsters, coffee, outdoor, seating, sandwiches, bagles, pastries, free wifi, neighborhood
Other	Total people: 630, Total check-ins: 2,056

Table 2.1: A POI and its associated information.

matches a user's personalized interest in terms of topic, and (2) The regional level word-of-mouth opinion for a POI in terms of popularity score. We profile users and PoIs by mining the textual information through topic modeling.

We will use following mathematical notations in this chapter. $U = \{u_1, u_2, ..., u_M\}$: a set of M users. $C = \{c_1, c_2, ..., c_N\}$: a set of N POIs. $R_{M \times N}$ with r_{ij} being the number of times user u_i checked in POI c_j. d_{c_j}: the textual items, both the tags and categories, associated with POI c_j. d_{u_i}: the items associated with POIs that user u_i visited. P_{c_j}: popularity score of POI c_j derived from the "total people" and "total check-ins". $W = \{w_1, w_2, ..., w_V\}$: unique V words set of all the associated textual information.

2.3 User and POI Profiling

In this section, we profile users and POIs in terms of interest distribution by performing topic models on the associated textual information.

2.3.1 Topic Distillation

The goal of topic distillation is to learn the interest of a user in terms of topic distribution based on the textual information of the POIs the user have checked in. Also, we need to infer the topic of interest a POI can provide. Unlike previous studies on collaborative filtering which only rely on other user's ratings to infer a given user's rating on a specific item, we propose to profile user and POI through topic distillation. The Latent Dirichlet Allocation (LDA) model (Blei, Ng, & Jordan, 2003) is an popular technique to identify latent topic information from a large document collection. In LDA, each document is represented as a probability distribution over topics and each topic is represented as a probability distribution over a number of words. The model has two latent variables that can be inferred from the data: (1) the document-topic distributions Θ, and (2) the topic-word distributions Φ. Then information can be obtained about which topics users are typically interested in as well as textual representation of POIs in terms of these topics.

To distill the topics in which LBSN users are interested by applying LDA, we propose to aggregate all the documents of the POIs user u_i have checked in into a user document d_{u_i}. We combine all the terms, both the tags and categories associated with a POI, into a POI document d_{c_j}. One reason for aggregation is that the terms associated with a single POI are usually short, incomplete and ambiguous. The aggregation process can better learn a user's interest in terms of topic. Thus, the topics of d_{u_i} can represent user u_i's interest topics.

In this way, we build an aggregated LDA model as shown in Figure 2.1. Each

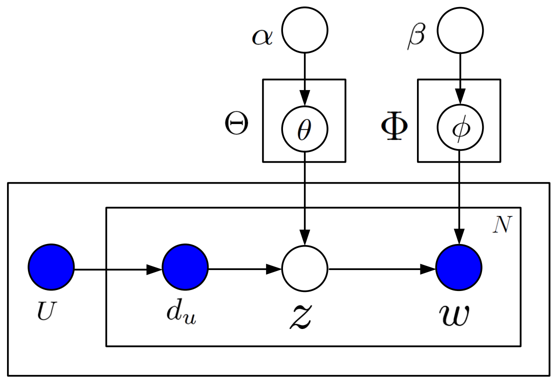


Figure 2.1: The aggregated LDA model.

document essentially corresponds to a LBSN user. As a result, the topic distribution of document d_{u_i} represents the interests of u_i. Each user u is associated with a multinomial distribution over topics, represented by θ. Each interest topic is associated with a multinomial distribution over textual terms, represented by ϕ. The generation process of the area aware user interest topic is as following:

1. For each topic $z \in \{1, ..., K\}$, draw a multinomial distribution over terms, $\phi_z \sim Dir(\beta)$.
2. For the document d_{u_i} given a user u_i
   (a) Draw a topic distribution, $\theta_{d_{u_i}} \sim Dir(\alpha)$
   (b) For each word $w_{d,n}$ in document d_{u_i}:
      i. Draw a topic $z_{d,n} \sim Mult(\theta_{d_{u_i}})$
      ii. Draw a word $w_{d,n} \sim Mult(\phi_{z_{d,n}})$

Then, we have: (1) Matrix $\Theta_{M \times K}$, where M is the number of users and K is the number of topics. θ_{ij} represents the probability that user i is interested in topic t_j. (2) Matrix $\Phi_{K \times V}$ where K is the number of topics and V is the number of unique terms in the dataset. Vector $\phi_{i\cdot}$ is the probability distribution of topic i over the V

terms.

We further infer the topic distribution π_j of POI c_j based on the learned user topic term distribution $\Phi_{K\times V}$. Therefore, we can compute the topic similarity.

2.3.2 Model Parameter Learning

For the aggregated LDA model, we have two sets of unknown parameters of interest: the user level document-topic distributions Θ, and the topic-word distributions Φ. There is also the latent variable z corresponding to the assignments of individual words to topics. We also need to infer the topic distribution π_j for each POI through the learned model as well as the POI document d_{c_j}.

Given the two hyperparameters α and β, the complete likelihood of the model of the M user documents as shown in Figure 2.1 is:

$$
\begin{aligned}
&p(W, Z, \Theta, \Phi|\alpha, \beta) = \\
&\prod_{m=1}^{M}\prod_{n=1}^{N_m} p(w_{m,n}|\phi_{z_{m,n}})p(z_{m,n}|\theta_m)\cdot p(\theta_m|\alpha)\cdot p(\Phi|\beta)
\end{aligned}
\tag{2.1}
$$

Note that it is computational intractable to directly estimate Θ and Φ in the likelihood of the LDA model as shown in Equation (2.1). During parameter estimation, we only need to keep track of $\Phi_{K\times V}$(word by topic) matrix, and $\Theta_{M\times K}$ (user by topic) matrix. From these matrices, we can estimate the topic-word distributions and user-topic distributions using Gibbs sampling (Griffiths & Steyvers, 2004). First we need to sample the conditional distribution of the latent variable z as follows.

$$
p(z_i = k|\mathbf{w}_i = w_i, \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{k,-i}^{(w_i)} + \beta}{n_{k,-i}^{(\cdot)} + V\beta} \cdot \frac{n_{d_i,-i}^{(k)} + \alpha}{n_{d_i,-i}^{(\cdot)} + K\alpha}
$$

where the counts $n_{\cdot,-i}^{(\cdot)}$ indicate term i is excluded from the corresponding document or topic.

With the sampling results, we can estimate ϕ and θ using $\phi_{kw} = \frac{n_k^{(w)}+\beta}{\sum_{w=1}^{V} n_k^{(w)}+V\beta}$ and $\theta_{ik} = \frac{n_i^{(k)}+\alpha}{\sum_{k=1}^{K} n_i^{(k)}+K\alpha}$ where $n_k^{(w)}$ is the frequency of word assigned for topic k and $n_i^{(k)}$ the topic observation counts for document d_{u_i} of user u_i. V is the number of the unique words and K is the number of topics. α and β are two priors and here we set symmetrical priors.

Next, we infer the topic distribution $p(\pi_j|d_{c_j}, \mathcal{M})$ of a POI with document d_{c_j} given the trained model $\mathcal{M} : \{\Theta, \Phi\}$ and hyperparameters α and β. Similar to the parameter estimation for the aggregated LDA model, we use the Gibbs sampling method to derive the topic distribution for each POI (Heinrich, 2009). The full conditional distribution of the Gibbs sampling is

$$p(z_{d_{c_j}} = k|\mathbf{w}_i = w_i, \mathbf{z}_{-i}, \mathbf{w}_{-i}, \mathcal{M}) \propto \phi_{k,w_i}(n_{d_{c_j},-i}^{(k)} + \alpha)$$

Then, the topic distribution for POI d_{c_j} is: $\pi_{jk} = \frac{n_j^{(k)}+\alpha}{\sum_{k=1}^{K} n_j^{(k)}+K\alpha}$, where $n_j^{(k)}$ is the topic observation count for POI document d_{c_j}.

2.3.3 Interest Matching Score

After deriving the interests of both users and POIs in terms of topic distribution, we can compute the extent to which a POI's interest matches a user's personalized interest by a matching score. The matching score between user u_j and POI c_j is defined as the similarity in terms of user interest topic distribution θ_i and POI topic distribution π_j. We use the symmetric Jensen-Shannon divergence between user u_i and POI c_j is:

$$D_{JS}(u_i, c_j) = \frac{1}{2}D(\theta_i \parallel M) + \frac{1}{2}D(\pi_j \parallel M)$$

where $M = \frac{1}{2}(\theta_i + \pi_j)$ and $D(\cdot \parallel \cdot)$ is the Kullback-Leibler distance. Then we define the matching score as $S(u_i, c_j) = 1 - D_{JS}(u_i, c_j)$.

2.4 A Topic and Location Aware Probabilistic Matrix Factorization (TL-PMF) Model

Since the POI recommendation is personalized, location-aware, and context depended, we introduce a *Topic and Location-aware* probabilistic matrix factorization (TL-PMF) method for POI recommendation by considering both the extent to which a user interest matches the POI in terms of topic distribution and the word-of-mouth opinions of the POI.

2.4.1 The Topic and Location-Aware POI Recommendation in LBSNs

In addition to the POI textual information and word-of-mouth opinions, we have the LBSN user's historical check-in record matrix R with r_{ij} being the number of times user u_i has checked in POI c_j. This also applies when r_{ij} is binary variable ($r_{ij} = 1$ meaning u_i interested in POI c_j and $r_{ij} = 0$ meaning not). We see r_{ij} as the rating of a user u_i for POI c_j.

For POI recommendation in LNSNs, we need to consider both (1) the extent to which the POI interest topic matches a user's personalized interest in terms of topics, and (2) the regional level word-of-mouth opinion for a POI in terms of popularity scores in a region. The rating r_{ij} of a user u_i for POI c_j is determined by user factors and POI factors. On the one hand, the rating should reflect the matching between the POI topic and the user interest topic. The rating is higher if two topic distributions

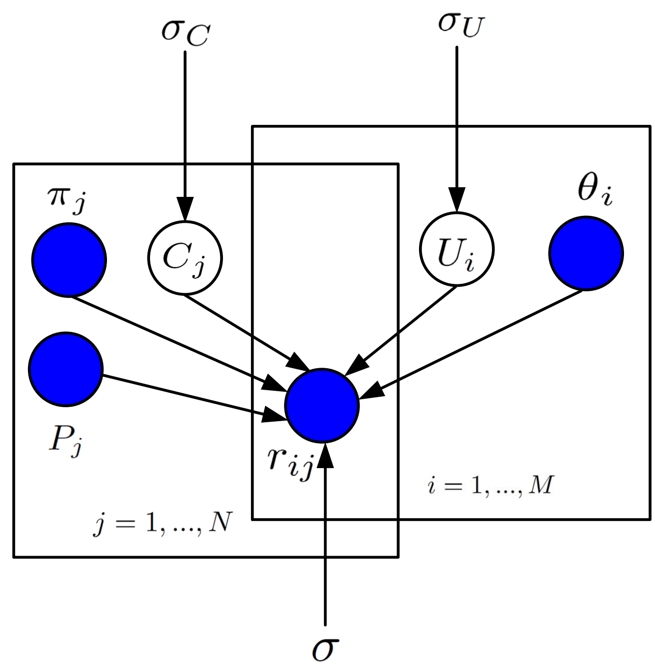


Figure 2.2: The TL-PMF model.

match better. On the other hand, the rating should reflect the word-of-mouth opinion index P_j of the local area.

We define the Topic and Location influence index of user u_i for POI c_j as

$$TL_{ij} = \gamma S(u_i, c_j) + (1 - \gamma)P_j \tag{2.2}$$

Here, $S(u_i, c_j)$ is a marching score between user u_j and POI c_j in terms of user interest topic distribution θ_i and POI topic distribution π_j. The second term P_j is a regional level popularity factor for POI c_j as a word-of-mouth opinion on the POI. γ is a factor to balance these two factors. Then TL_{ij} considers both interest topic match between user and POI, and location aware word-of-mouth opinions for a POI.

2.4.2 The TL-PMF Model

To leverage the influence interest topic and location aware word-of-mouth opinions for POI recommendation, we propose a *Topic and Location-aware* probabilistic matrix factorization (TL-PMF) model. The graphical representation of TL-PMF is shown in Figure 2.2. Let r_{ij} be the rating of user u_i for POI c_j, U_i and C_j are the user and POI latent feature space vector respectively. The distribution over the observed ratings as well as the textual information is

$$p(R|U, C, TL, \sigma^2) = \prod_{i=1}^{M} \prod_{j=1}^{N} \left[\mathcal{N}(r_{ij}|f(U_i, C_j, TL_{ij}), \sigma^2)\right]^{I_{ij}} \tag{2.3}$$

where $\mathcal{N}(\cdot|\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. I_{ij} is the indicator function. Function $f(U_i, C_j, TL_{ij})$ is to approximate the rating of user u_i for POI c_j.

Consider the influence interest topics and location aware word-of-mouth opinions on user u_i's preference for POI c_j, we define

$$f(U_i, C_j, TL_{ij}) = TL_{ij} \cdot U_i^T C_j \tag{2.4}$$

where U_i and C_j are $D-$dimensional latent factors for user u_i and POI c_j respectively, TL_{ij} is the topic and location index of user u_i for POI c_j. Here we use a weighted product of user latent factors and POI factors by incorporating topic and location index to improve PMF model. TL_{ij} is derived from the aggregated topic model and the popularity score as shown in Section 2.3.

We set zero mean Gaussian prior to user and POI latent space (Salakhutdinov & Mnih, 2008b): $p(U|\sigma_U^2) = \prod_{i=1}^{M} \mathcal{N}(U_i|0, \sigma_U^2\mathbf{I})$ and $p(C|\sigma_C^2) = \prod_{j=1}^{N} \mathcal{N}(C_j|0, \sigma_C^2\mathbf{I})$. Then, the posterior distribution of Equation (3.1) becomes

$$p(U,C|R,\sigma^2,TL,\sigma_U^2,\sigma_C^2) \propto p(R|U,C,\sigma^2,TL,\sigma_U^2,\sigma_C^2)p(U|\sigma_U^2)p(C|\sigma_C^2)$$
$$= \prod_{i=1}^{M}\prod_{j=1}^{N}\left[\mathcal{N}(r_{ij}|f(U_i,C_j,TL_{ij}),\sigma^2)\right]^{I_{ij}}$$
$$\times \prod_{i=1}^{M}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}) \times \prod_{j=1}^{N}\mathcal{N}(C_j|0,\sigma_C^2\mathbf{I})$$

We need to estimate parameters in terms of maximizing likelihood. The log posterior distribution is:

$$\mathcal{L}(U,C|R,\sigma^2,TL,\sigma_U^2,\sigma_C^2)) =$$
$$-\frac{1}{2\sigma^2}\sum_{i=1}^{M}\sum_{j=1}^{N}I_{ij}(r_{ij}-f(U_i,C_j,TL_{ij}))^2 - \frac{1}{2\sigma_U^2}\sum_{i=1}^{M}U_i^TU_i$$
$$-\frac{1}{2\sigma_C^2}\sum_{j=1}^{N}C_j^TC_j - \frac{1}{2}\left[\left(\sum_{i=1}^{M}\sum_{j=1}^{N}I_{ij}\right)\ln\sigma^2 + MD\ln\sigma_U^2 + ND\ln\sigma_C^2\right]$$

where D is the dimension of the latent factors. Maximizing the log posterior equals to minimizing the following function

$$\begin{aligned} E &= \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}I_{ij}\left(r_{ij}-TL_{ij}\cdot U_i^TC_j\right)^2 \\ &+ \frac{\lambda_U}{2}\sum_{i=1}^{M}||U_i||_F^2 + \frac{\lambda_C}{2}\sum_{j=1}^{N}||C_j||_F^2 \end{aligned} \tag{2.5}$$

where $\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_C = \sigma^2/\sigma_C^2$, and $||\cdot||_F^2$ is the Frobenius norm. Performing a gradient descent method on U and C can lead to a local minimum solution to Equation (2.5) using: $\frac{\partial E}{\partial U_i} = -\sum_{j=1}^{N} I_{ij}\left(r_{ij}-TL_{ij}\cdot U_i^TC_j\right)\cdot TL_{ij}C_j + \lambda_U U_i$ and $\frac{\partial E}{\partial C_j} = -\sum_{i=1}^{M} I_{ij}\left(r_{ij}-TL_{ij}\cdot U_i^TC_j\right)\cdot TL_{ij}U_i + \lambda_C C_j$

2.4.3 Prediction and Recommendation

After the user interest topic and parameters U, C are learned, the TL-PMF model prediction of the rating of a user for a given POI is estimated as $\mathbb{E}(r_{ij}|u_i, c_j) = TL_{ij} \cdot U_i^T C_j$ where γ can adjust the weight of matching score and the local popularity score.

Since recommendation in LBSNs is highly location sensitive, the recommendation list should be close to the user's current region and thus it advisable to recommend POIs near the user's physical location. Our TL-PMF model provides global predicted preference scores global. In real practice, we need take into consideration of location information to make reasonable personalized POI recommendations. Given a user's current location L_{u_i}, one possible way to make recommendations is to recommend N POIs corresponding to top N prediction scores within a certain range $\text{Range}_{L_{u_i}}$.

2.5 Experimental Results

In this section, we provide an empirical evaluation of the performances of the proposed model. All the experiments were performed on a large real-world LBSN dataset collected from Foursquare, one of the largest and most popular LBSN community.

2.5.1 The Experimental Data

The dataset is formulated as follows (Z. Cheng, Caverlee, Kamath, & Lee, 2011): Foursquare users usually report their check-ins of POIs via Twitter. When a LNSN user posted a Tweet, which indicates a check-in of a POI, we consider it as the user has checked in physically. Also from Foursuqre, we have detailed information of each POI with its location in terms of latitude and longitude, region, the associated categories,

Table 2.2: A check-in trace of a user.

The Wonderland Ballroom, Washington, DC, 2010-07-24, 04:25:41
Black Squirrel, Washington, DC, 2010-07-24, 16:42:28
Columbia Heights Coffee, Washington, DC, 2010-07-25, 01:19:02
The Wonderland Ballroom, Washington, DC, 2010-07-25, 02:08:44
Commonwealth Gastropub, Washington, Dc, 2010-07-25, 07:45:51
Washington National Airport, Arlington, Va, 2010-07-26, 19:20:47
.
Fornelletto, Atlantic City, NJ, 2010-08-10, 18:45:42
.
Panera Bread, Knoxville, TN, 2010-08-26, 17:10:08
.
Lou Malnati's Pizzeria, Chicago, IL, 2010-10-19, 00:26:25
.

tags, the total number of people, and the total number of check-ins. With both the LBSN's tweet check-in reports, in which latitude and longitude are available, and the LBSN check-in profiles have latitude and longitude values, we can match these two sources of information to obtain LBSN users' check-in profiles with additional information for the POIs.

Table 2.2 shows an example of the check-in trace for a user, who had reported her/his visit to different POIs at different states in USA. Figure 2.3 shows the check-

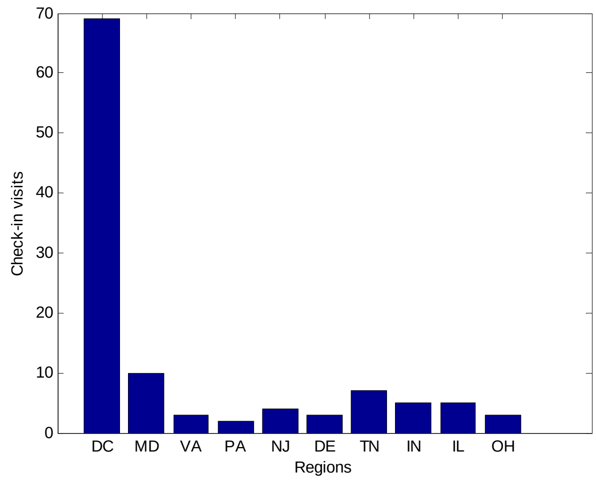


Figure 2.3: An example of check-in reports for a user

in report times for the user in different regions. A user would usually have her/his home address, which corresponds to the highest frequent report region, and may visit POIs at different regions.

Table 2.3: Data Description

user	POIs	rating	avg # rates	sparsity
35,025	49,779	1,080,824	30.85	99.94%

As a lot of users may have checked in or reported few check-ins, we exclude those users with less than 6 check-in records. As the number of words associated with each POI vary dramatically, we select the POIs with the minimum 10 tags. We finalized a dataset as shown in Table 5.4. Here, we use implicit rating, namely the number of checks-in for a POI as the rating for the POI. This is different from the rating in movie recommendation, in which the rating is usually in a range from 1 to 5. So, we need to transfer the discrete rating to a value between [0,1] by using $f(x) = (x-1)/(K-1)$ with K is the maximum rating value (Salakhutdinov & Mnih, 2008b). We can see

Table 2.4: Some selected topics (identified by aggregated LDA when $K =$ 30)

topic	terms
1	airport terminal travel airlines delta gate tsa high mile gogo gogoinflight united southwest wifi baggage continental airplane handler airways
2	san technology apple office bar diego home iphone gym shop computer ipod store video mac coffee center ipad restaurants
3	sea seattle tac bar seatac wifi coffee beer free waterfront airport food fish limo restaurant limousine square hour colman
4	train station transit new bus subway rail metro public food nj transportation line amtrak york penn city jersey express
5	bar food beer coffee wine restaurant free bbq music burgers trivia patio pool delivery italian chicken american outdoor bon
6	theater movie movies theatre food gallery photo booth photobooth popcorn mall cinema pizza douchebag cineplex shopping imax art store
7	college university frat gas food library school pizza student coffee center state boys bar building campus store station gym
8	marketing design media social web office music corporate advertising food coffee search seo agency development restaurant internet digital toronto
9	attorney law injury accident lawyer personal lawyers attorneys city atlanta firm bar beer restaurant bankruptcy office food sports oc
10	mall store food mobile accessories shopping american wireless apple cell court phone department macy coffee photobooth body women shoes

that the rating matrix is very spare with 99.94% missing ratings.

2.5.2 Evaluation Metrics

We adopt the *Root Mean Squared Error* (RMSE) to measure the prediction error. RMSE is defined as RMSE $= \sqrt{\frac{1}{N}\sum_{i,j}(\hat{r}_{ij} - r_{ij})^2}$ where r_{ij} denotes the rating of POI j by user i, $\hat{r}_{ij}$ denotes the corresponding rating predicted by the model, and N denotes the total number of the tested rating. The smaller the value of RMSE, the more precise a recommendation.

Ranking a recommendation list is often more important than the rating prediction. So we also evaluate the algorithms in terms of ranking. We present each user with N POIs sorted by their predicted rating and evaluate based on which of these POIs were actually visited by the user. However, a direct use of top N based metric like $recall@N$ and $precision@N$ would incur underlying biases as this metric depends heavily on the percentage of relevant items that each user has rated (Herlocker, Konstan, Terveen, & Riedl, 2004). In our dataset, a user has rated only a very small percentage (about 0.06%). We adopt the relative rank evaluation method as described in (Koren, 2008)

First we select the $|\mathcal{T}|$ highest rating set $\mathcal{T}$ from the test dataset. For each POI $c_j \in \mathcal{T}$ for user u_i, we add another $|\mathcal{C}|$ randomly selected POIs $\mathcal{C}$, and predict the rating for $\{c_j, \mathcal{C}\}$. Then, we sort the $|\mathcal{C}| + 1$ predicted rating scores in a descending order. In this way, we can find the relative place of these interesting POIs in the total order of the recommendation list for a given user. We can obtain a cumulative distribution of the relative ranking based on the selected rating set $\mathcal{T}$.

2.5.3 Implementation Details

We divided the data into training (80%) and testing (20%) data. We compared TL-PMF with PMF. We did not use other matrix factorization methods like SVD based methods as the benchmark because it has been shown that PMF outperforms SVD approaches (Salakhutdinov & Mnih, 2008b).

For TL-PMF, we further set different parameter γ in the topic and location index $TL_{ij} = \gamma S(u_i, c_j) + (1-\gamma)P_j$ to test how local popularity factor influence user's preference choice. When $\gamma = 1$, it means that the recommendation is made by only including user interest topic and is denoted as TL-PMF$_{\mathbf{T}}$; $\gamma = 0$ means that the rating mainly relies on local word-of-mouth popularity information and is denoted as TL-PMF$_{\mathbf{L}}$, and $0 < \gamma < 1$, denoted as TL-PMF$_{\mathbf{TL}}$, means that the recommendation is made by combining both user interest topic and local popularity opinion.

We normalize the local rating score for POI j in area to $[0, 1]$ range by the following equation. $\hat{P}_j = \frac{1}{2}\left\{\frac{\text{totalPeo}_j - 1}{\max_j\{\text{totalPeo}_j\} - 1} + \frac{\text{totalCk}_j - 1}{\max_j\{\text{totalCk}_j\} - 1}\right\}$ where $\max_j\{\text{totalPeo}_j\}$ and $\max_j\{\text{totalCk}_j\}$ are the maximum total people value and total check-in value in the area respectively.

We set $\lambda_U = 0.01$ and $\lambda_C = 0.01$ for PMF, TL-PMF$_{\mathbf{T}}$, TL-PMF$_{\mathbf{TL}}$ and TL-PMF$_{\mathbf{L}}$. We set $\alpha = 50/K$ and $\beta = 0.1$ in the aggregated LDA model.

Table 2.4 shows some of the user interest topics learned from the aggregated LDA when $K = 30$. These topics include transportation, technology, recreation, restaurant, school, company, shopping and so on.

2.5.4 Performance Comparisons

Here, we compare the performances of different approaches in terms of RMSE and Top-N metrics.

Performance comparison I: RMSE

With RMSE, we compare TL-PMF and PMF at different settings. We first set the number of topics $K = 30$ and $K = 50$ to learn user topic interest, and thus get the topic and location index TL_{ij}. We do not directly use $\mathbb{E}(r_{ij}|u_i, c_j) = g(TL_{ij} \cdot U_i^T C_j)$ for prediction but pass the results through a logistic function $g(x) = \frac{1}{1+\exp(-x)}$ to bound the prediction score to range $[0, 1]$. Then the prediction becomes: $\mathbb{E}(r_{ij}|u_i, c_j) = g(TL_{ij} \cdot U_i^T C_j)$. In each topic number case, we perform TL-PMF with different user and POI factor dimensions ($D = 10$ and $D = 30$). Also, we compare the effect of local popularity P_j in recommendation.

As shown in Table 2.5, no matter whether incorporating only topic model or both topic model and local popularity rating, TL-PMF outperforms PMF. For example, when topic number $K = 30$ and factor dimension $D = 10$, comparing to PMF, TL-PMF$_{\mathbf{T}}$ improves RMSE by 5.1%, and TL-PMF$_{\mathbf{TL}}$ with $\gamma = 0.5$ improves RMSE by 7.3%. We can see that TL-PMF$_{\mathbf{T}}$ improves recommendation performances by incorporating user's personal interest learned by topic model. TL-PMF$_{\mathbf{TL}}$ further improves recommendation by balancing both a user's personal interest and the word-of-mouth opinion.

To further investigate the effect of word-of-mouth opinions on recommendation performances, we perform another experiment by adjusting γ, which controls the

Table 2.5: A prediction comparison of TL-PMF with PMF in terms of RMSE with two different factor dimensions in two different topic number settings (Note: PMF does not involve topics.).

	D=10		D=30	
Model	30 topics	50 topics	30 topics	50 topics
PMF	0.2488		0.2470	
TL-PMF$_{\mathbf{T}}$	0.2362	0.2345	0.2388	0.2380
TL-PMF$_{\mathbf{TL}}$	0.2305	0.2301	0.2324	0.2319

weight of personal and word-of-mouth opinion factors. Table 2.6 shows that RMSE varies according to the different rating determination factor parameter γ. The change of RMSE with λ is shown in Figure 2.4. In the figure, we can see that word-of-mouth opinions not always compensate personalized interests to improve recommendation performances. When we depend too much on local popularity score, happening when γ approaches to 0, the recommendation performance starts decreasing, and even can be worse than PMF without additional information. Another problem with too much weight to local popularity score is the slow convergence of the algorithm. Note that the RMSE value 0.398 (corresponding to $\gamma = 0$) is the result after 5000 iterations. One explanation is that the personal interest is not always consistent with word-of-mouth opinions.

Performance comparison II: Top N

Since POI recommendation in LBSNs is highly location sensitive, the recommendation list should be close to the user's current region. Figure 2.3 shows an example of check-in reports for a user in different regions. A user would visit POIs at different regions. Therefore, we measure the Top N performance by considering the recommendation list within a certain range of the target user's current location.

We use the relative ranking measure as introduced in Section 2.5.2. We select the highest $|\mathcal{T}|$ rating $\mathcal{T}$ set from the test data as the probe POIs. Then, for each probe POI and the corresponding user, we randomly select $|\mathcal{C}| = 500$ POIs $\mathcal{C}$ within a certain range $\text{Range}_{L_{c_j}}$ of the probe POI location L_{c_j}, In this way, we can find the relative rank of these probe POIs in the total order of the recommendation list for a given user.

We compare the top N performances of TL-PMF$_{\mathbf{TL}}$, TL-PMF$_{\mathbf{T}}$ and PMF using the relative ranking measure. Figure 2.5 shows the cumulative distribution of the percentile relative rank for TL-PMF$_{\mathbf{TL}}$, TL-PMF$_{\mathbf{T}}$ and PMF. Note that the straight line connecting the bottom-left and top-right corners is for random prediction. As can be seen, our TL-PMF models, both the TL-PMF$_{\mathbf{T}}$ model with just topic model and the TL-PMF$_{\mathbf{TL}}$ model with topic model as well as regional level word-of-mouth opinion, outperforms PMF (dot blue) significantly. Indeed, for the case when x-axis value is equal to 0.1 or 10%, which corresponds to recommend top-50 POI recommendation: probabilistically, the numbers of POIs will match user interest are $50 \times 22.38\% \approx 11$ with PMF, $50 \times 91.33\% \approx 45$ with TL-PMF$_{\mathbf{T}}$ model, and $50 \times 96.52\% \approx 48$ with TL-PMF$_{\mathbf{TL}}$ model respectively. In the experiment, we set the location range $\text{Range}_{L_{c_j}}$ as a state level. We can potential expect to make more relevant POI recommendations by narrowing the location range value.

Table 2.6: A comparison of TL-PMF$_{\mathbf{TL}}$ with different γ values in topic and location index $TL_{ij} = \gamma S(u_i, c_j) + (1-\gamma)P_j$. Here, $K = 30$ and $D = 10$.

γ	0	0.1	0.2	0.5	0.7	1
RMSE	0.398	0.2418	0.2318	0.2305	0.2320	0.2362

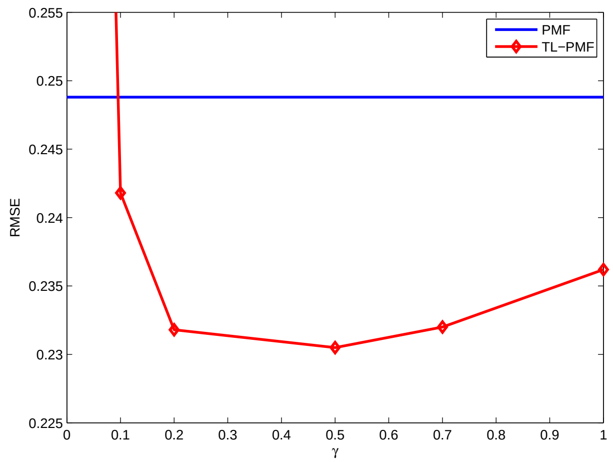


Figure 2.4: The RMSE values of TL-PMF$_{\mathbf{TL}}$ with different γ values in topic and location index $TL_{ij} = \gamma S(u_i, c_j) + (1-\gamma)P_j$ (red line) vs. PMF (blue line).

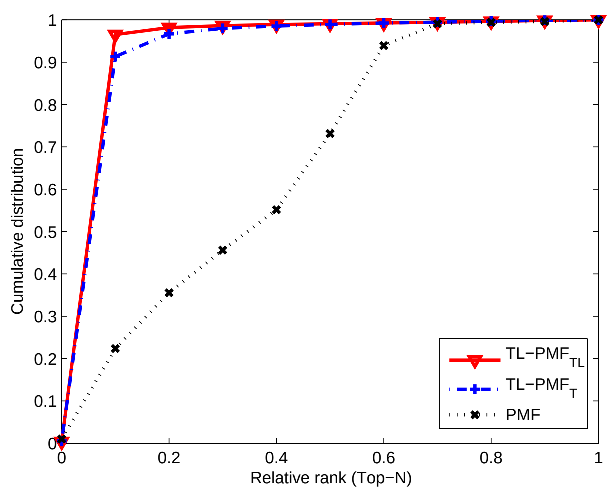


Figure 2.5: The Top N comparison of TL-PMF with PMF in terms of relative rank. X-axis stands for the relative rank in percentage of the probe POIs.

Summary

In summary, the proposed models can outperform the baseline method dramatically in terms of both RMSE and Top N metrics. We have observed that both personalized user interest topic as well as location dependent word-of-mouth opinion can be incorporated into the proposed flexible framework to improve recommendation performance.

2.5.5 Topic Analysis in LBSNs

Here, we analyze the topic characteristics of POIs across different geographical regions. We have shown that the user generated textual tags, which are aimed to better describe what type of places a POI is, help to improve POI recommendation. We are further interested in studying whether different areas would present different topics, and what is the effect of topic difference on recommendation. To this end, we select eight areas from all the POI dataset: California (CA), Arizona (AZ), Texas (TX), Florida (FL), Chicago area (IL), Washington DC (DC), Boston area (MA) and New York area (NY), and form a region level POI data set. These areas cover different regions and are representative of regional differences.

We aggregate all the POIs in an area into a region-level document and have eight region-level documents. For each region, we infer the region document-topic distribution π based on the topics we learned by setting $K = 30$. For each region pair $\{R_i, R_j\}$ within the selected regions, we can compute the correlation of the topic distribution Corr_{ij} by using $\mathrm{Corr}_{ij} = \frac{\sum_{k=1}^{K}(\pi_{ik}-\bar{\pi}_i)(\pi_{jk}-\bar{\pi}_j)}{\sqrt{\sum_{k=1}^{K}(\pi_{ik}-\bar{\pi}_i)^2}\sqrt{\sum_{k=1}^{K}(\pi_{jk}-\bar{\pi}_j)^2}}$ where $\bar{\pi}_i$ and $\bar{\pi}_j$ are the average topic probability for regions R_i and R_j respectively. Then, we have

Table 2.7: The regional level topic correlation when the topic number $K = 30$.

	AZ	TX	FL	IL	DC	MA	NY
CA	0.1163	0.0836	0.2974	0.0572	0.0302	0.0943	0.0937
AZ		-0.0637	0.0577	-0.1096	-0.0061	-0.0330	-0.0876
TX			0.1294	-0.0588	-0.0372	-0.0152	-0.0495
FL				0.0202	-0.0529	-0.0247	-0.0071
IL					-0.0200	-0.0504	-0.0598
DC						0.0159	0.0198
MA							-0.0424

the region-level topic correlation in Table 2.7 and its visualization in Figure 2.6.

In Table 2.7 and Figure 2.6, we can see that the region difference poses different topics because the correlation between the region level topics are almost near 0 or negative. In the selected regions, the California and Florida areas share the highest correlation. Florida shares high correlation with both Arizona and Texas, but Arizona and Texas do not have high correlation (the correlation between Arizona and Texas is -0.0637).

Through the topic analysis of both user interest topic and regional level topic comparison, we revealed that (1) Most POIs of LBSNs are dominated by a few topics, which are common life topics, as shown in Table 2.4; (2) Topics differ in different regions even in contiguous regions. This implies that we should take into consideration of both personalized user interests as well as the regional level word-of-mouth opinions for POI recommendations in LBSNs.

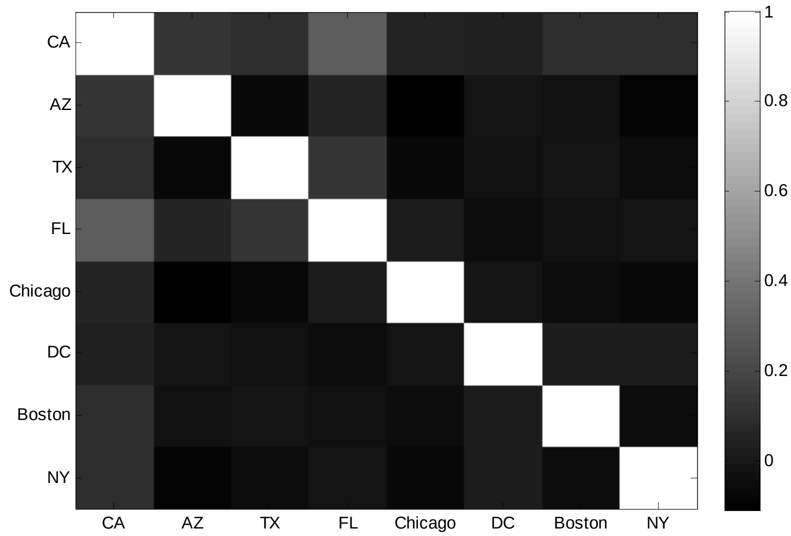


Figure 2.6: The correlation of topic distributions between different selected areas.

2.6 Related Work

Related work can be grouped into two categories: works study place of interest recommendation from the application perspective, and works study how to exploit textual information to improve recommendation from the methodology perspective.

With increasing popularity of LBSNs, applying POI recommendation to provide better location based service has caught a lot of attentions from both academia and industry. Previous studies on POI recommendations mainly relied on user trajectory data. For example, various works (Y. Zheng, Zhang, Xie, & Ma, 2009; Berjani & Strufe, 2011; V. W. Zheng, Zheng, Xie, & Yang, n.d.; Ge et al., 2010; Q. Liu, Ge, Li, Chen, & Xiong, 2011; Ge, Liu, Xiong, Tuzhilin, & Chen, 2011) applied collaborative filtering based method to recommend locations and travel packages based on user trajectory data. By considering the geographical influence due to the spatial clustering phenomenon in LBSN users, Ye et al (Ye, Yin, Lee, & Lee, n.d.) explored user preference, social influence and geographical influence for recommending POIs in LBSNs.

More recent work began to explore textual information to better understand patterns in LBSN and to improve LBSN services. For instance, (Farrahi & Gatica-Perez, 2011) applied topic models to identify daily location-driven routines by mining text from mobile phone data. (Ye, Shou, Lee, Yin, & Janowicz, 2011) presented a work on semantic annotation for LBSNs to annotate places with category tags by exploring explicit patterns of individual places and implicit relatedness among similar places. (Z. Yin, Cao, Han, Zhai, & Huang, n.d.) proposed a latent geographical topic analysis method to explore both location and associated text of locations and found this can help to discover meaningful geographical topics. Finally, (Ferrari, Rosi, Mamei, & Zambonelli, 2011) analyzed Twitter posts and performed LDA on the data to extract urban patterns, such as hotspots and crowd behaviors.

There are works to explore textual information for recommendation. A straightforward way is to combine collaborative filtering with topic models. By mining the textual information associated with each item, we could combine probabilistic matrix factorization (PMF) (Salakhutdinov & Mnih, 2008b) and topic models (Wang & Blei, 2011). The fLDA model in (Agarwal & Chen, n.d.) follows this line, but they associated the rating by regularizing both user and item factors simultaneously through user features and words associated with each item. In addition to exploring topic models for item recommendation, there are also studies which use topic models to learn social-media user interests to recommend new friends with similar interests

(Pennacchiotti & Gurumurthy, 2011).

Unlike the tasks of recommending movies and scientific papers (Wang & Blei, 2011), the problem of POI recommendation in LBSN services is location-aware, personalized, and context depended. In addition, the textual terms associated with POIs are usually incomplete and ambiguous. This study explores both associated textual and context information to address these challenges.

2.7 Summary

In this chapter, we studied the POI recommendation problem in LBSNs by exploiting textual information as well as regional word-of-mouth opinions. There are several advantages of the proposed recommendation method. First, the textual terms associated with POIs are usually incomplete and ambiguous. To meet this challenge, the proposed method exploits location dependent word-of-mouth opinions in addition to users' personalized interests learnt from the insufficient POI textual information. Second, the location-aware aggregated LDA recommendation approach allows to profile user interests by performing topic modeling of the users' historical textual information. This provides a way to match the user interests to the POI topic, and thus alleviate the cold start problem in recommendation. Third, the proposed recommendation method can strike a balance between the use of individual information and the use of location-aware word-of-mouth opinions. This helps to avoid the excessive use of personalized information, and thus reducing the possibility of overfitting. Last but not least, the proposed method is flexible and could be extended to incorporate other types of context-aware information to enhance POI recommendation.

CHAPTER 3

A GENERAL GEOGRAPHICAL PROBABILISTIC FACTOR MODEL FOR POINT OF INTEREST RECOMMENDATION

3.1 Introduction

Recent years have witnessed the increased development and popularity of location-based social network (LBSN) services, such as Foursquare, Gowalla, and Facebook Places. LBSNs allow users to share their check-ins and opinions on places they have visited, ultimately helping each other find better services. Data collected through LBSN activity can enable better recommendations of places, or Points of Interest (POIs) such as restaurants and malls. This can drastically improve the quality of location-based services in LBSNs, simultaneously benefiting not only LBSN users but also POI owners. On one hand, mobile users can identify favorite POIs and improve their user experience via good POI recommendations. On the other hand, POI owners can leverage POI recommendations for better targeted acquisition of customers. In this chapter we address exactly the problem of POI recommendation. We first identify the key challenges specific to geographical settings. Then, we propose a general framework to address these, as well as two instantiations of this framework.

Challenges. While latent factor models, such as matrix factorization (Koren, Bell, & Volinsky, 2009), probabilistic matrix factorization (PMF) (Salakhutdinov & Mnih, 2008b, 2008a), and many other variants (Koren, 2008; Agarwal & Chen, 2009; R. Bell, Koren, & Volinsky, 2007; Koren, 2010; L. Zhang, Agarwal, & Chen, 2011; Liang Xiong, 2010), have been proved effective and are widely used in diverse recommendation settings, adapting them to POI recommendations requires delicate consideration of unique characteristics of LBSNs. Indeed, there are several characteristics of LBSNs which distinguish POI recommendation from traditional recommendation tasks (such as movie or music recommendations). More specifically:

- Geographical influence. Due to geographical constraints and the cost of traveling large distances, the probability of a user visiting a POI is inversely proportional to the geographic distance between them.

- Tobler's first law of geography. The law of geography states that "Everything is related to everything else, but near things are more related than distant things" (Tobler, 1970). In other words, geographically proximate POIs are more likely to share similar characteristics.

- User mobility. Users may check into POIs at different regions; e.g., an LBSN user may travel to different cities. Varying user mobility imposes huge challenges on POI recommendations, especially when a user arrives at a new city or region.

- Implicit user feedback. In the study of POI recommendations, explicit user ratings are usually not available. The recommender system has to infer user preferences from implicit user feedback (e.g., check-in frequency).

The first three mutually related challenges due to geography imply interrelationships among items. However, traditional recommender systems usually ignore these, assuming that the items are independent and identically distributed. In fact, the decision process of a user choosing a POI is complex and can be influenced by many factors. First, geographical distance plays an important role. According to the Tobler's first law of geography and the law of demand, a user's propensity for a POI is inversely proportional to the distance between them. This is similar to the observation that the probability of purchasing an item is inversely proportional to its cost. Second, utility matters. In economics, utility is an index of preferences over sets of items and services when a user makes purchasing decisions. In other words, a user may still prefer a remote POI to a nearby one, if higher satisfaction (utility) outweighs the overhead of travel. Finally, LBSN users have varying mobility behaviors, which further impose challenges on modeling check-in decisions.

An additional fourth challenge is that user check-in counts follow a distribution with power-law form. This is different from ratings in traditional recommender systems, in which explicit ratings are available to reflect users' item preferences. In other words, in LBSNs a user can visit a POI only once and another POI hundreds of times. Since we do not have explicit user ratings for POIs, we can only make use of implicit user behavior data in the check-in records for POI recommendations.

POI Recommendation Framework. All the above challenges demand a reconsideration of the recommendation model, to achieve effective POI recommendation in LBSNs. While there are some studies on POI recommendations, they lack an integrated analysis of the joint effect of the above factors, such as user preferences, geographical influences and user mobility behaviors.

To address these challenges, we propose a framework for geographical probabilistic factor modeling (Geo-PFM) which can strategically take various factors into consideration. This framework can capture the geographical influences on a user's check-in behaviors, can effectively model the user mobility patterns, and can deal with the skewed distribution of check-in count data. Specifically, we introduce a latent region variable and use a multinomial distribution over latent regions to model user mobility behaviors over different activity regions. These latent regions reflect the activity areas for all the users through collective actions. A Gaussian distribution is used to represent a POI over a sampled region. This can reflect the first law of geography; that is, similar POIs are more related than distant POIs. Moreover, geographical influence can be effectively modeled in the latent region. Finally, implicit user feedback in the form check-in counts is taken into account.

In our earlier work (B. Liu, Fu, Yao, & Xiong, 2013), we introduced Geo-PFM by specifically instantiating a geographical Bayesian non-negative matrix factorization(Geo-BNMF), to model user preferences. As a result, this model is capable of taking personal preferences, geographical influence, and user mobility into consideration, and can effectively handle the skewed distribution of POI count data.

In this chapter, we study the Geo-PFM framework in more detail and we further develop a Poisson Geo-PFM, which is also able to capture the geographical influences on a user's check-in behavior and effectively model the user mobility patterns. In addition, the Poisson Geo-PFM provides much more flexibility and interpretability than Geo-PFM based on non-negative matrix factorization (B. Liu et al., 2013). First, the Poisson Geo-PFM provides a rigorous probabilistic generative process for the model, while the NMF-based Geo-PFM uses an approximation solution. Second, the nature of Poisson distribution is more suitable and effective for modeling the skewed user check-in count data, which provide implicit feedback, for better POI recommendations.

Finally, we provide extensive experimental results on three real-world LBSNs data, which differ in terms of user mobilities, POI geographical distributions, implicit response data skewness and user-POI observation sparsity. The experimental results show that the proposed POI recommendation method consistently outperforms state-of-the-art probabilistic latent factor models with a significant margin in terms of Top-N recommendation. Moreover, the proposed Poisson Geo-PFM outperforms Geo-BNMF (B. Liu et al., 2013) even further.

3.2 Background

Latent factors models aim to characterize user-item interactions assuming that each user and each item can be expressed as a user and item latent vector $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ respectively. Consequently, the *response* (rating, like, or implicit frequency) is modeled as $p(y_{ij}|i,j) = p\left(y_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j; \Theta\right)$. In this section we summarize two types of latent factor models: probabilistic matrix factorization methods which are widely used for recommendations when explicit user feedback (e.g., item ratings) is available, and the Poisson factor model which is more effective when user feedback is implicitly provided via heavily skewed frequency counts (as in the case of POI recommendation).

3.2.1 Probabilistic Matrix Factorization

Matrix factorization models (Koren et al., 2009) have been generalized into probabilistic matrix factorization (PMF) (Salakhutdinov & Mnih, 2008b), which is a Bayesian version. In PMF the response y_{ij} of user u_i for item v_j is assumed to follow a Gaussian distribution $y_{ij} \sim \mathcal{N}(y_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j, \sigma^2)$. When response y_{ij} is not normalized to a standard rating score, one solution is to scale the discrete response to a value between $(0, 1]$ by using $f(x) = (x-1)/(x_{max}-1)$, where x_{max} is the maximum response value for each user (Salakhutdinov & Mnih, 2008b). Furthermore, a zero-mean Gaussian prior is placed toon the user and item latent spaces

$$P(U|\sigma_u^2) = \prod_{i=1}^{M} \mathcal{N}(\boldsymbol{u}_i|0, \sigma_u^2\mathbf{I}), P(V|\sigma_v^2) = \prod_{j=1}^{N} \mathcal{N}(\boldsymbol{v}_j|0, \sigma_v^2\mathbf{I}).$$

Then the latent factors $\boldsymbol{u}$ and $\boldsymbol{v}$ can be inferred by maximize thing likelihood over the observed ratings

$$P(Y|U, V, \sigma^2) = \prod_{i=1}^{M}\prod_{j=1}^{N} \left[\mathcal{N}(y_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j, \sigma^2)\right]^{I_{ij}} \tag{3.1}$$

where I_{ij} is the indicator function. Maximizing the log-posterior over user and item latent factors with hyperparameters is equivalent to minimizing the sum-of-squared-errors objective function:

$$\begin{aligned} \mathcal{L} =& \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}\left(y_{ij} - \boldsymbol{u}_i^\top \boldsymbol{v}_j\right)^2 \\ &+ \frac{\lambda_U}{2}\sum_{i=1}^{M} ||\boldsymbol{u}_i||_F^2 + \frac{\lambda_V}{2}\sum_{j=1}^{N} ||\boldsymbol{v}_j||_F^2 \end{aligned} \tag{3.2}$$

where $\lambda_U = \sigma^2/\sigma_u^2$, $\lambda_V = \sigma^2/\sigma_v^2$, and $||\cdot||_F^2$ is the Frobenius norm. Gradient descent can be applied to infer the latent factors with partial derivatives $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ respectively,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_i} &= -\sum_{j=1}^{N} I_{ij}\left(y_{ij} - \boldsymbol{u}_i^\top \boldsymbol{v}_j\right) \cdot \boldsymbol{v}_j + \lambda_U \boldsymbol{u}_i \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_j} &= -\sum_{i=1}^{M} I_{ij}\left(y_{ij} - \boldsymbol{u}_i^\top \boldsymbol{v}_j\right) \cdot \boldsymbol{u}_i + \lambda_V \boldsymbol{v}_j.\end{aligned} \tag{3.3}$$

3.2.2 Poisson Factor Model

The Poisson distribution is a more appropriate choice for response variables y_{ij} that represent frequency counts. The Poisson probabilistic factor model (Poi-PFM) (Ma, Liu, King, & Lyu, 2011; Y. Chen, Kapralov, Pavlov, & Canny, 2009; Gopalan, Hofman, & Blei, 2013) factorizes the user-item count matrix Y as $Y \sim \text{Poisson}(UV)$. More specifically, for each user-item response y_{ij}, we assume a Poisson distribution over the mean f_{ij}: $y_{ij} \sim \text{Poisson}(f_{ij})$. The mean matrix F is factorized into two matrices $U_{M\times K}$ and $V_{N\times K}$. Each element $u_{ik} \in U$ encodes the preference of user i for "topic" k, and each element $v_{ik} \in V$ reflects the topical affinity of item j to topic k. Further, u_{ik} and v_{ik} can be assigned empirical priors following Gamma distributions. We then have the following generative process.

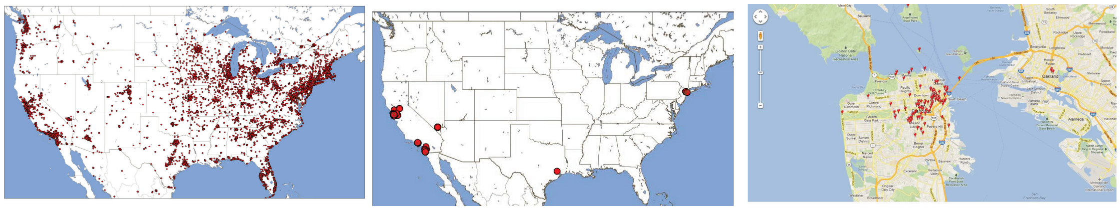
1. Generate user latent factor $u_{ik} \sim \text{Gamma}(\alpha_U, \beta_U)$.

2. Generate item latent factor $v_{jk} \sim \text{Gamma}(\alpha_V, \beta_V)$.

3. Generate $y_{ij} \sim \text{Poisson}(\boldsymbol{u}_i^\top \boldsymbol{v}_j)$.

Given user latent factor $\boldsymbol{u}_i$ and item latent factor $\boldsymbol{v}_j$, the probability of response y_{ij} is

$$P(y_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j) = \left(\boldsymbol{u}_i^\top \boldsymbol{v}_j\right)^{y_{ij}} \exp\left\{-\boldsymbol{u}_i^\top \boldsymbol{v}_j\right\} / y_{ij}!$$



(a) (b) (c)

Figure 3.1: An example of a typical user check-in pattern: (a) all the POIs; (b) the user's check-ins over different regions: San Francisco, Los Angeles, San Diego, Las Vegas, Houston, and New York City; (c) the user's check-ins in San Francisco area.

We can apply maximum a posteriori (MAP) estimation over the observed data and priors to infer the latent vectors. Specifically,

$$\begin{aligned}&P(U,V|Y,\alpha_U,\beta_U,\alpha_V,\beta_V)\\&\propto p(Y|U,V)P(U|\alpha_U,\beta_U)p(V|\alpha_V,\beta_V)\end{aligned}$$

where

$$\begin{aligned}p(Y|U,V)&=\prod_{i=1}^{M}\prod_{j=1}^{N}\left(\boldsymbol{u}_i^{\top}\boldsymbol{v}_j\right)^{y_{ij}}\exp\left\{-\boldsymbol{u}_i^{\top}\boldsymbol{v}_j\right\}/y_{ij}!\\P(U|\alpha_U,\beta_U)&=\prod_{i=1}^{M}\prod_{k=1}^{K}\frac{u_{ik}^{\alpha_U-1}\exp(-u_{ik}/\beta_U)}{\beta_U^{\alpha_U}\Gamma(\alpha_U)}\\p(V|\alpha_V,\beta_V)&=\prod_{j=1}^{N}\prod_{k=1}^{K}\frac{v_{jk}^{\alpha_V-1}\exp(-v_{jk}/\beta_V)}{\beta_V^{\alpha_V}\Gamma(\alpha_V)}\end{aligned}$$

The log of the posterior distribution over the user and item latent factors is given by

$$\begin{aligned}&\mathcal{L}(U,V,|\mathcal{D},\alpha_U,\beta_U,\alpha_V,\beta_V)\\&=\sum_{i=1}^{M}\sum_{k=1}^{K}\Big((\alpha_U-1)\ln u_{ik}-u_{ik}/\beta_U\Big)\\&+\sum_{j=1}^{N}\sum_{k=1}^{K}\Big((\alpha_V-1)\ln v_{jk}-v_{jk}/\beta_V\Big)\\&+\sum_{i=1}^{M}\sum_{j=1}^{N}(y_{ij}\ln f_{ij}-f_{ij})+\text{const.}\end{aligned}\tag{3.4}$$

Taking derivatives on $\mathcal{L}$ with respect to u_{ik} and u_{jk}, we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial u_{ik}} &= \frac{\alpha_U - 1}{u_{ik}} - \frac{1}{\beta_U} + \sum_{j=1}^{N}\left(\frac{y_{ij}}{f_{ij}} - 1\right) v_{jk} \\ \frac{\partial \mathcal{L}}{\partial v_{jk}} &= \frac{\alpha_V - 1}{v_{jk}} - \frac{1}{\beta_V} + \sum_{i=1}^{M}\left(\frac{y_{ij}}{f_{ij}} - 1\right) u_{ik}.\end{aligned} \tag{3.5}$$

Again, gradient ascent method can be applied to infer the latent factors.

3.3 Geographical Probabilistic Factor Model for POI Recommendation

In this section, we first formulate the problem of POI recommendation and then introduce a general geographical probabilistic factor analysis framework for this problem, addressing the challenges described previously.

3.3.1 Problem Definition

The problem of personalized POI recommendation is to recommend POIs to a user given user POI check-in records and other available side information. Let $U = \{u_1, u_2, ..., u_M\}$ be a set of LBSN users, where each user has a location l_i. The user location l_i is usually unknown due to user mobility. Let $V = \{v_1, v_2, ..., v_N\}$ be a set of POIs, where each POI has a location $l_j = [lon_j, lat_j]^\top$ represented by longitude and latitude. Throughout this chapter we use indices i for users and indices j for POIs, unless stated otherwise. The number of times user u_i visited POI v_j is represented by the *response variable* y_{ij}. The check-in records for a particular user are sparse (most y_{ij} values are zero), with non-zeros following a power law. In the chapter we use the terms “POI” and “item” interchangeably. Key notations are listed in Table 5.3.

Table 3.1: Mathematical Notations

Symbol	Size	Description
R	$1 \times \|R\|$	latent region set, r is a region in R
$\boldsymbol{\eta}$	$M \times \|R\|$	user level region distribution
$\boldsymbol{\mu}$	$\mathbb{R}^2$	location mean of a latent region
$\boldsymbol{\Sigma}$	$\mathbb{R}^{2\times 2}$	covariance matrix of a latent region
$\boldsymbol{U}$	$M \times K$	user latent factor
$\boldsymbol{V}$	$N \times K$	item latent factor
y_{ij}	$\mathbb{R}$	response of user i for item j
l_j	$\mathbb{R}^2$	location of item j

3.3.2 The General Idea

We aim to capture how different factors such as user preference, geographical influence and user mobility affect user POI check-in decisions. The key idea is that overall user preferences are the result of the interplay between geographical preferences and interest preferences. Our models aim to effectively capture that interplay.

Geographical preferences. To learn geographical user preferences, we need a model to encode the spatial influence and user mobility into the user check-in decision process. As shown in Figure 3.1, LBSN users are most likely to check into a number of POIs and these POIs are usually limited to certain geographical regions. This observation has two implications: first, a user's mobility always happens across a limited number regions but these regions could be different among different users; second, user check-in activities happen in a given region and the activity patterns could be different given different regions. Based on this observation, we propose to introduce a set of $|R|$ latent regions R which are inferred based on the collective actions of *all* users, reflecting activity areas for the entire population.

Although the overall distribution of POIs is irregular, we can however assume a Gaussian geographical distribution of POIs *within* each region $r \in R$. The location l_j for POI j is characterized by $l_j \sim \mathcal{N}(\mu_r, \Sigma_r)$, where μ_r and Σ_r are the mean vector and covariance matrix of the region, respectively (Z. Yin et al., n.d.; Hong, Ahmed, Gurumurthy, Smola, & Tsioutsiouliklis, 2012). Latent regions also reflect Tobler's first law of geography, which states that POIs with similar characteristics are likely to be clustered into the same geographical area. Once a region is fixed, geographical influence can be effectively modeled and applied to overall user preference profiling.

We finally model individual user mobility over the collectively inferred latent regions R by applying a multinomial distribution, $r \sim p(r|\eta_i)$, where η_i is a user-dependent distribution over latent regions for user i.

Interest preferences. Interest preferences are modeled using a latent factor model, generating a user item preference $\alpha(i,j)$ based on user latent factor variable $\boldsymbol{u}_i$ and and item latent factor variable $\boldsymbol{v}_j$.

Overall user preferences. Finally, to model a user's propensity for a POI, we assume the following factors that will affect the overall user check-in decision process: (1) the personal preference $\alpha(i,j)$ of each user i with respect to POI j; and (2) geographical influence in terms of travel distance, namely, the distance $d(i,j)$ between the user and the POI as a geographical cost. As a result, the probability of observing a user-POI pair (i,j) is directly proportional to the user interest, and monotonically decreases with the distance between them,

$$p(i,j) \propto \mathbb{F}\left(\alpha(i,j), \left[\frac{d_0}{d_0 + d(i,j)}\right]^{\tau}\right)$$

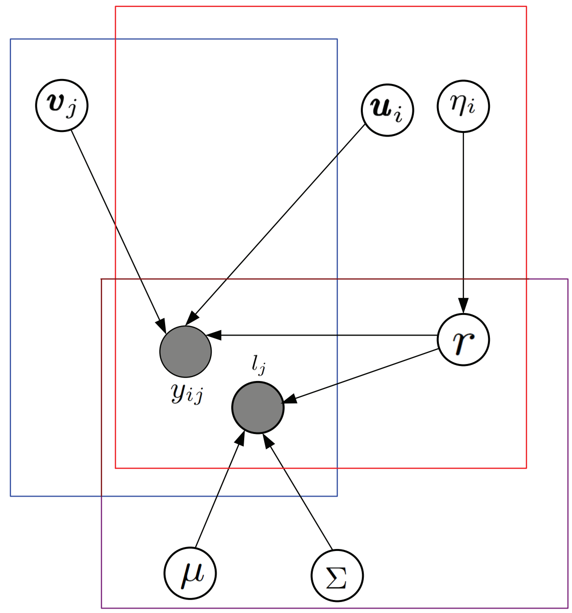


Figure 3.2: A graphical representation of the proposed geographical probabilistic factor model (Geo-PFM), where the red plate represents users, the blue plate represents POIs, and the purple plate represents latent regions. The model priors have been excluded for simplicity.

where $\mathbb{F}(\cdot)$ is a function that combines user interest preference and geographical influence. We model the distance factor in the decision making process using a parametric term $\left[\frac{d_0}{d_0+d(i,j)}\right]^\tau$ with a power-law form. This motivated by the observation that the probability of user i choosing POI j decays exponentially with respect to the distance between them.

3.3.3 Geographical Probabilistic Factor Model Framework

Based on above discussion, we proposed a geographical probabilistic factor model (Geo-PFM) to capture user mobility, and geographical influence in user profiling for POI recommendation. The complete graphical model is shown in Figure 3.2.

The corresponding generative process to draw pairs (i, j) representing user i choosing POI j can then be expressed as follows. First, a user u_i samples a region r_i from all $|R|$ regions following a multinomial distribution $r_i \sim \text{Multinomial}(\eta_i)$, on which a conjugate Dirichlet prior $\text{Dir}(\boldsymbol{\gamma})$ can be further imposed. Here η_i is a user-dependent parameter, capturing user i's mobility pattern over the latent regions. A POI is drawn from the sampled region $l_j \sim \mathcal{N}(\mu_{r_i}, \Sigma_{r_i})$. The interest preference $\alpha(i,j)$ of user i for POI j can be represented by combining latent factors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, specifically, $\alpha(i,j) = \boldsymbol{u}_i^\top \boldsymbol{v}_j$. Finally, the user-POI response y_{ij} (check-in frequency count) is assumed to follow certain distribution $y_{ij} \sim P(f_{ij})$ where f_{ij} depends on user preferences and the distance between the user and the POI. Summarizing:

1. Draw a geographical preference
   a. Draw region $r_i \sim \text{Multinomial}(\eta_i)$.
   b. Draw a POI j with location $l_j \sim \mathcal{N}(\mu_{r_i}, \Sigma_{r_i})$.
2. Draw an interest preference
   a. Draw user latent factor $\boldsymbol{u}_i \sim P(\boldsymbol{u}_i; \Psi_{\boldsymbol{u}_i})$.
   b. Draw item latent factor $\boldsymbol{v}_j \sim P(\boldsymbol{v}_j; \Psi_{\boldsymbol{v}_j})$.
   c. Draw user-item preference $\alpha(i,j) = \boldsymbol{u}_i^\top \boldsymbol{v}_j$.
3. For each user-POI pair (i,j) draw the response $y_{ij} \sim P(f_{ij})$, where

$$f_{ij} = \mathbb{F}\left(\boldsymbol{u}_i^\top \boldsymbol{v}_j, \left[\frac{d_0}{d_0 + d(i,j)}\right]^\tau\right)$$

Note that the proposed model is general and can be extended with different factor models, since we limit neither the user and item latent factor distribution, nor the user-item response distribution. $\mathbb{F}(\cdot)$ is a function of personalized preferences $\boldsymbol{u}_i^\top \boldsymbol{v}_j$ and of distance cost $\left[\frac{d_0}{d_0+d(i,j)}\right]^\tau$. User-item response $y_{ij} \sim P(f_{ij})$ can be: (i) Gaussian when explicit ratings are available, (ii) Bernoulli for binary response such as *liking*, or (iii) Poisson when count or frequency data is to be modeled.

3.3.4 Model Components

This section describes the model components of Geo-PFM in detail.

User Mobility and Geographical Influence

As discussed earlier, user mobility and geographical influence are among the most predominant factors that distinguish POI recommendation from traditional recommendation (e.g., for movies), and these two factors can interact with each other. Geographical influence has been exploited for POI recommendation due to the fact that geographical proximity could significantly affect a user's check-in decision (Ye et al., n.d.). However, check-in behavior can change as the user travels from one region to another, and little has been done to consider user mobility for POI recommendation. Capturing user mobility is important to understand user preferences in different regions, and it becomes even more important when a user travels to a new place.

To this end, as described earlier, we introduce a set of $|R|$ latent regions R, and model user mobility using multinomial distribution (Hong et al., 2012) $r \sim \text{Multinomial}(\eta_i)$, where η_i is a user-dependent distribution over latent regions for user i. The explicit location $\ell(\cdot)$ of a user is not observed. We use the region r with center μ_r to represent the user activity area and model the geographical influence as a parametric and power-law like term $\left[\frac{d_0}{d_0+d(i,j)}\right]^{\tau}$, with $d(i,j) = ||\mu_r - l_j||_2$, where μ_r approximates the current user activity area center. As a result, both user mobility and geographical influence can be effectively captured by the proposed Geo-PFM model.

Modeling Count Response

In most existing latent factor models, represented by PMF (Salakhutdinov & Mnih, 2008b), the response $P\left(y_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j;\Theta\right)$ is assumed to follow a Gaussian distribution, namely, $y_{ij} \sim \mathcal{N}(\boldsymbol{u}_i^\top \boldsymbol{v}_j, \sigma^2)$. However, a Gaussian distribution is not suitable when the response variable is implicit count data, which are heavily skewed. Therefore, it is not suitable for the POI recommendation problem, since check-in counts follow a power-law like distribution.

We need to ensure our model is suitable for count responses. By combining geographical influence with latent factors, we model user-POI response as a geographical probabilistic factor model (Geo-PFM):

$$y_{ij} \sim P(y_{ij}|f_{ij},\Theta), f_{ij} = \mathbb{F}\left(\boldsymbol{u}_i^\top \boldsymbol{v}_j, \left[\frac{d_0}{d_0 + d(i,j)}\right]^\tau\right)$$

In the above, $\mathbb{F}(\cdot)$ is a suitably chosen function that captures the joint effect of personal interest preferences $\boldsymbol{u}_i^\top \boldsymbol{v}_j$ and distance cost $\left[\frac{d_0}{d_0+d(i,j)}\right]^\tau$. Also, the response function $P(\cdot)$ suitably chosen to model count data. Potential response function distributions include Poisson (see Section 3.4).

3.4 Model Specification

This section introduces detailed model specifications of the Geo-PFM model. In particular, we introduce a Poisson Geo-PFM model, which takes into account the characteristics of count response values.

3.4.1 Poisson Geo-PFM Model

As we use count response to infer user preferences, we expect the latent vectors are constrained to be non-negative. In our earlier work (B. Liu et al., 2013), we applied

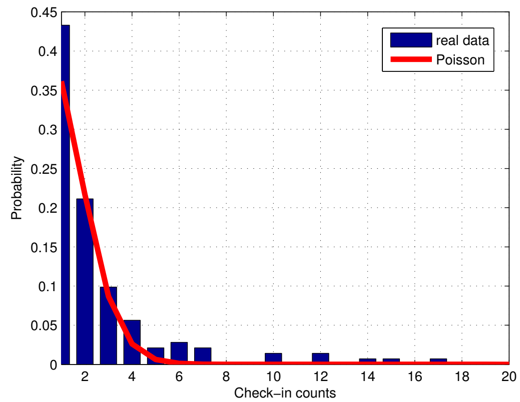


Figure 3.3: The check-in counts distribution of a randomly selected user and a Poisson approximation of this distribution (Foursquare dataset).

a rectified normal Bayesian non-negative matrix factorization model to capture the count response feature. Specifically, we assumed a rectified normal distribution on $Y \sim P(\boldsymbol{UV})$ with variance $\sigma^2\mathbf{I}$ and non-negativity constraints,

$$Y \sim \mathcal{N}^R(Y|\boldsymbol{UV}, \sigma^2\mathbf{I}), \quad \text{subject to} \;\; \boldsymbol{U} \geq 0, \boldsymbol{V} \geq 0. \tag{3.6}$$

We further placed an exponential distribution on $\boldsymbol{U}$ and $\boldsymbol{V}$, and an inverse gamma distribution on σ^2 with shape a and scale b.

However, a Poisson factor model is a better alternative. First, the Poisson distribution is a more appropriate choice for modeling skewed count data. Figure 3.3 shows a typical distribution of check-in count distribution, for a randomly selected user in the Foursquare dataset. A Poisson distribution approximates this distribution well, and can also provide a response that is non-negative. More importantly, a Poisson factor guarantees a rigorous probabilistic generative process for the model, while the rectified normal BNMF provides a probabilistic approximation. Therefore we propose a Poisson Geo-PFM model which incorporates both user interest preference and geographical influence. More specifically, for each user-item frequency y_{ij} we assume a Poisson distribution over mean f_{ij}: $y_{ij} \sim \text{Poisson}(f_{ij})$ with $f_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j \cdot \left[\frac{d_0}{d_0+d(i,j)}\right]^\tau$. Furthermore, u_{ik} and v_{ik} are given Gamma distributions as empirical priors (Ma et al., 2011; Y. Chen et al., 2009), $u_{ik} \sim \text{Gamma}(\alpha_U, \beta_U)$ and $v_{jk} \sim \text{Gamma}(\alpha_V, \beta_V)$. Then, the generative process we introduced earlier to model user-item preference becomes specifically:

1. Draw a region $r \sim \text{Multinomial}(\eta_i)$.

2. Draw a POI j with location $l_j \sim \mathcal{N}(\mu_r, \Sigma_r)$.

3. Draw user latent factor $u_{ik} \sim \text{Gamma}(\alpha_U, \beta_U)$.

4. Draw item latent factor $v_{jk} \sim \text{Gamma}(\alpha_V, \beta_V)$.

5. Draw $y_{ij} \sim \text{Poisson}(f_{ij})$ with $f_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j \cdot \left[\frac{d_0}{d_0+d(i,j)}\right]^\tau$.

3.4.2 Parameter Estimation

Let $\Psi = \{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ denote all parameters, and let $\Omega = \{\alpha_U, \beta_U, \alpha_V, \beta_V, \gamma\}$ be the hyperparamters. We are given the observed data collection $\mathcal{D} = \{y_{ij}, l_j\}^{I_{ij}}$ where y_{ij} is the user check-in count and l_j is the location of v_j; and I_{ij} is the indicator function with $I_{ij} = 1$ when user u_i visited POI v_j, and $I_{ij} = 0$ otherwise. Then we aim to maximize the posterior probability given the observed data:

$$
\begin{aligned}
P(\Psi;\mathcal{D},\Omega) &\propto \prod_{\mathcal{D}} P\left(y_{ij},l_j|\Psi,\Omega\right) P(\Psi|\Omega) \\
&\propto \prod_{\mathcal{D}} P\left(y_{ij},l_j,\Psi|\Omega\right) P(U|\alpha,\beta)P(V|\alpha,\beta)P(\eta|\gamma) \\
&\propto \prod_{i=1}^{M}\left\{\prod_{j=1}^{N_i}|\mathbf{\Sigma}_r|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(l_j-\boldsymbol{\mu}_r)^T\mathbf{\Sigma}_r^{-1}(l_j-\boldsymbol{\mu}_r)\right)\right. \\
&\left.\frac{{f_{ij}}^{y_{ij}}\exp(-f_{ij})}{y_{ij}!}\right\}\times\eta_{i1}^{c_{i1}}\cdots\eta_{iR}^{c_{iR}}\times\prod_{i=1}^{M}\prod_{k=1}^{K}u_{ik}^{\alpha-1}\exp(\frac{-u_{ik}}{\beta}) \\
&\times\prod_{j=1}^{N}\prod_{k=1}^{K}v_{jk}^{\alpha-1}\exp(\frac{-u_{jk}}{\beta})\times\prod_{i=1}^{M}\prod_{r=1}^{R}\eta_{ir}^{\gamma_i-1}
\end{aligned}
$$

To estimate the parameters Ψ, we use a mixing Expectation Maximization (EM) and sampling algorithm to learn all the parameters (Andrieu, De Freitas, Doucet, & Jordan, 2003; Hong et al., 2012). We regions r as a latent variable and introduce the hidden variable $P(r|l_j,\Psi)$ (Z. Yin et al., n.d.; Hong et al., 2012), which is the probability of $l_j \in r$, given POI location l_j and Ψ. The algorithm iteratively updates the parameters by mutual enhancement between Geo-clustering and Geo-PFM. The Geo-clustering updates the latent regions based on both location and check-in behaviors; and Geo-PFM learns the graphical preference factors.

E-step

In the **E-step**, we iteratively draw latent region assignments for all POIs. For each POI, a latent region r is first drawn from the following distribution,

$$
r \sim P\left(r|\left\{y_{\cdot j},l_j\right\},R^{(t)},\Psi^{(t)}\right)\times P(r|\boldsymbol{\eta}^{(t)}) \tag{3.7}
$$

where

$$
P\left(\left\{y_{\cdot j},l_j\right\}|r,\Psi^{(t)}\right) = P\left(l_j|r,\Psi^{(t)}\right)\times P\left(y_{\cdot j}|r,\Psi^{(t)}\right)
$$

$$P\left(l_j|r,\Psi^{(t)}\right)=\mathcal{N}\left(l_j|\mu_r^{(t)},\Sigma_r^{(t)}\right)$$

$$P\left(y_{\cdot j}|r,\Psi^{(t)}\right)=\prod_{i=1}^{M}\text{Poission}(f_{ij}|\boldsymbol{U}^{(t)},\boldsymbol{V}^{(t)}).$$

$P(r|\boldsymbol{\eta}^{(t)})$ updates region assignment in terms of user mobility, $P(l_j|r,\Psi^{(t)})$ is the location PDF function for multivariate normal distribution with mean vector and variance matrix obtained in last iteration, and $P(y_{\cdot j}|r,\Psi^{(t)})$ updates region assignment through collective actions.

M-step

In the **M-step**, we maximize the log likelihood of the model with respect to model parameters by fixing all regions obtained in the E-step. Since we sample the regions in the E-step, we can update $\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r, \boldsymbol{\eta}$ directly from the samples,

$$\begin{aligned}\boldsymbol{\mu}_r &= \frac{1}{\#(j,r)}\sum_{j=1}^{\mathcal{D}}\mathbb{I}(r_j=r)l_j \\ \boldsymbol{\Sigma}_r &= \frac{1}{\#(j,r)-1}\sum_{j=1}^{\mathcal{D}}\left((l_j-\mu_r)(l_j-\mu_r)^\top\right)\end{aligned} \tag{3.8}$$

where $\#(j,r)$ is the number of POIs assigned to region r. Through imposing a conjugate Dirichlet prior Dir$(\boldsymbol{\gamma})$, we update $\boldsymbol{\eta}^{(t+1)}$ by

$$\eta_{ir}^{(t+1)}=\frac{C_{ir}^{(t+1)}+\gamma}{C_{i\cdot}^{(t+1)}+R\gamma} \tag{3.9}$$

where C_{ir} is the number of POIs being assigned to region r for user i, and $C_{i\cdot}$ is the number of all POIs and all regions for user i.

After updating region $R^{(t+1)}$, we update $\Psi^{(t+1)}$ by maximizing the posterior with respect to latent factors $\boldsymbol{u}$ and $\boldsymbol{v}$. We use a gradient ascent method to find $\Psi^{(t+1)}$ that maximizes the posterior. Note that we already update R as $R^{(t+1)}$, and we here need to maximize the posterior with respect to latent factor variables $\boldsymbol{u}$ and $\boldsymbol{v}$. More specifically, we maximize the following objective function:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{U},\boldsymbol{V}|R^{(t+1)}) \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K}\Big((\alpha_U - 1)\ln u_{ik} - u_{ik}/\beta_U\Big) \\
&+ \sum_{j=1}^{N}\sum_{k=1}^{K}\Big((\alpha_V - 1)\ln v_{jk} - v_{jk}/\beta_V\Big) \\
&+ \sum_{i=1}^{M}\sum_{j=1}^{N}(y_{ij}\ln f_{ij} - f_{ij}) + \text{const.}
\end{aligned} \tag{3.10}$$

where $f_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j \cdot \left[\frac{d_0}{d_0+d(i,j)}\right]^\tau$.

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

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial u_{ik}} &= \frac{\alpha_U - 1}{u_{ik}} - \frac{1}{\beta_U} + \sum_{j=1}^{N}\left(\frac{y_{ij}}{f_{ij}} - 1\right) v_{jk}\left[\frac{d_0}{d_0+d(i,j)}\right]^\tau \\
\frac{\partial \mathcal{L}}{\partial v_{jk}} &= \frac{\alpha_V - 1}{v_{jk}} - \frac{1}{\beta_V} + \sum_{i=1}^{M}\left(\frac{y_{ij}}{f_{ij}} - 1\right) u_{ik}\left[\frac{d_0}{d_0+d(i,j)}\right]^\tau.
\end{aligned} \tag{3.11}$$

We use stochastic gradient ascent to update u_{ik} and u_{ik}. Stochastic gradient ascent (descent) have been widely used for many machine learning tasks (Bottou, 2010a). The main process involves randomly scanning all training instances and iteratively updating parameters,

$$u_{ik} \leftarrow u_{ik} + \epsilon \times \frac{\partial \mathcal{L}}{\partial u_{ik}}, \quad v_{jk} \leftarrow v_{jk} + \epsilon \times \frac{\partial \mathcal{L}}{\partial v_{jk}} \tag{3.12}$$

where ϵ is a learning rate.

Remark. The region R is updated in each E-step. The latent factor model parameters are updated based on the new regions. We summarize the parameter estimation procedure for Geo-PFM in Algorithm 1.

```
Initialize region partition R^(0) by k-means (k = R)
for t ← 1 to N_iteration do
    Update region R^(t) according to Equ. (3.7)
    Update region mean μ_r^(t) and covariance Σ_r^(t) according to Equ.(3.8)
    Update user region preference distribution η_ir^(t) according to Equ. (3.9)
    Update u_ik^(t), v_jk^(t) by stochastic ascent
    repeat
        ε^nIter := ε ν/(ν+nIter−1) // annealing learn rate
        for each random {i, j} pair do
            for k ← 1 to K do
                u_ik ← u_ik + ε^nIter × ∂L/∂u_ik
                v_jk ← v_jk + ε^nIter × ∂L/∂v_jk
            end
        end
    until convergence or reach max_iter
end
```

Algorithm 1: Geo-PFM Estimation

3.4.3 Recommendation

After parameters Ψ are learned, the Geo-PFM model predicts the check-in counts of a user for a given POI as $\mathbb{E}(y_{ij}|u_i, v_j) = \boldsymbol{u}_i^\top \boldsymbol{v}_j \times \left[\frac{d_0}{d_0+d(i,j)}\right]^\tau$. We make recommendations based on the predicted check-ins as well as the user mobility. One way to combine the predicted value and user mobility is $\hat{y}_{ij} = \mathbb{E}(y_{ij}|u_i, v_j) \times \eta_{ir}$ with $j \in r$, the larger the predicted value, the more likely the user will choose this POI.

3.5 Experimental Results

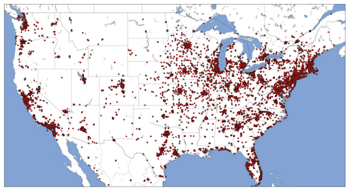
In this section we empirically evaluate the performance of our proposed methods. All experiments were performed on three real-world LBSN datasets, collected from Foursquare (one of the most popular LBSN communities), Gowalla, and Brightkite.

3.5.1 Datasets

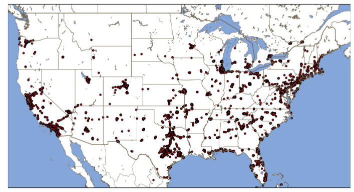
Foursquare dataset. The Foursquare dataset is formulated as follows (Z. Cheng, Caverlee, Lee, & Sui, 2011; Z. Cheng, Caverlee, Kamath, & Lee, 2011): Foursquare users usually report their check-ins at POIs via Twitter. When an LBSN user posts a Tweet check-in at a POI, we consider it as evidence that the user has physically checked into the POI. The dataset includes POIs across the Unites States (except Hawaii and Alaska), and the geographical distribution of all POIs is shown in Figure 3.4a. According to the Twitter reports from Foursquare users, we finalized a dataset of $12,422$ users for $46,194$ POIs with $738,445$ check-in observations. The user POI check-in count matrix has a sparsity of 99.87%; it is very sparse. Each user checked into 59.44 POIs on average, only a very small fraction of all the POIs. The number of check-ins for a POI ranges from 1 to 786. This range is very wide as shown in Figure 3.5, in which the user check-in count of a randomly chosen user is plotted.

Gowalla dataset. Besides the Foursquare dataset, we also evaluate the proposed models on Gowalla (Cho, Myers, & Leskovec, 2011). In this dataset, we remove those POIs with less than 10 users, and remove users with less than 30 user-POI pairs. We finalize a dataset of $7,070$ users for $30,755$ POIs with $520,950$ check-in observations. The user POI check-in count matrix has a sparsity of 99.76%, with each user checked into 73.68 POIs on average. The number of check-ins for a POI ranges from 1 to 286, and the geographical distribution of all Gowalla POIs is shown in Figure 3.4b.

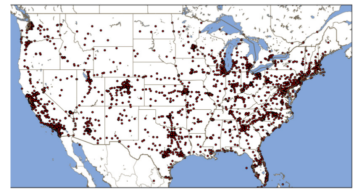
Brightkite dataset. Finally, we evaluate the proposed models on Brightkite (Cho et al., 2011). We finalize a dataset of $2,192$ users and $9,865$ POIs with $72,543$ check-



(a) Foursquare.



(b) Gowalla.



(c) Brightkite.

Figure 3.4: POI geographical distribution for the three different datasets.

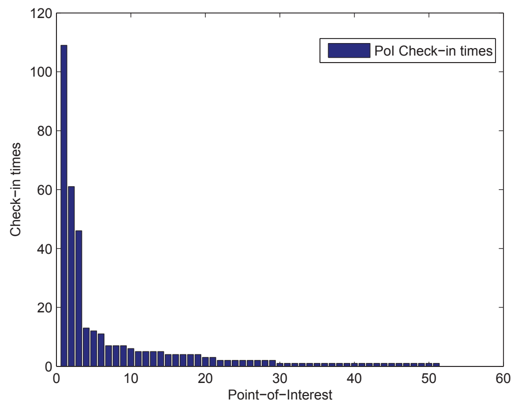


Figure 3.5: An example of wide range user check-in counts for a randomly chosen user (Foursquare).

in observations. The user POI check-in count matrix has a sparsity of 99.66%, with each user checked into 33.09 POIs on average. The number of check-ins for a POI ranges from 1 up to more than one thousand, and the geographical distribution of all Brightkite POIs is shown in Figure 3.4c. We summarize the data statistics for all datasets in Table 5.4.

Table 3.2: Data Description

	# users	# POIs	# records	sparsity	avg POIs
Foursquare	12,422	46,194	738,445	99.87%	59.44
Gowalla	7,070	30,755	520,950	99.76%	73.68
Brightkite	2,192	9,865	72,543	99.66%	33.09

3.5.2 Evaluation Metrics

Since there is no explicit rating for validation, we evaluate the models in terms of ranking. We present each user with N POIs sorted by the predicted values and evaluate based on which of these POIs were actually visited by the user.

Precision and Recall. Given a top-N recommendation list $S_{N,\text{rec}}$ sorted in descending order of the prediction values, precision and recall are defined as

$$\begin{aligned} \text{Precision@}N &= \frac{|S_{N,\text{rec}} \bigcap S_{\text{visited}}|}{N} \\ \text{Recall@}N &= \frac{|S_{N,\text{rec}} \bigcap S_{\text{visited}}|}{|S_{\text{visited}}|} \end{aligned} \tag{3.13}$$

where S_{visited} are the POIs a user has visited in the test data. The precision and recall for the entire recommender system are computed by averaging all the precision and recall values of all the users, respectively.

F-measure. F-measure combines precision and recall, and is the harmonic mean of precision and recall. Here we use the F_β measure with $\beta = 0.5$,

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \times \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{3.14}$$

The F_β measure with $\beta < 1$ indicates more emphasis on precision than recall.

3.5.3 The Method for Comparison

We experimentally compare our proposed Poisson Geo-PFM[1] model with state-of-the-art latent factor models. Specifically, we compare our proposed Poisson Geo-PFM model with following algorithms:

- Probabilistic Matrix Factorization (PMF) (Salakhutdinov & Mnih, 2008b): PMF

[1] We will refer to Poisson Geo-PFM as Geo-PFM in this Section unless stated otherwise.

is a recommendation method widely used for different recommendation tasks, and the details of PFM are summarized in Section 3.2.1.

- Bayesian Non-negative Factorization (BNMF) (Schmidt, Winther, & Hansen, 2009): This is the base model which our earlier work (B. Liu et al., 2013) adopted.

- Poisson Factor Model (PoiFM) (Ma et al., 2011): Poisson factor model provides an alternative for count data recommendation as Poisson is effective in modeling count data (more details in Section 3.2.2).

- Fused Poisson factor model (Fu-PoiFM): this method fuses the geographical influence into factor models by considering the multi-region of user check-in behaviors and the inverse distance in an *ad hoc* way (C. Cheng, Yang, King, & Lyu, 2012). Since Poison factor model also exploits the count check-in characteristics, we fuse the geographical influence into PoiFM and denote it as Fu-PoiFM.

- Geo-BNMF. This is the model we used in our earlier work (B. Liu et al., 2013).

In particular, we are interested in investigating the following questions:

- How the proposed Geo-PFM improves the non-geographical baseline models (PMF, BNMF, PoiFM) as well as the fused model (Fu-PoiFM).

- How the Poisson based model Geo-PFM improves its counterpart based on non-negative factorization, Geo-BNMF.

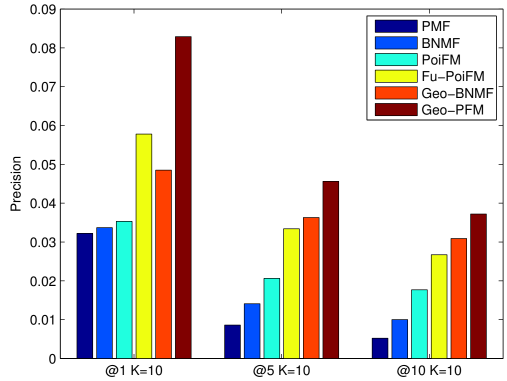
We randomly divided the data into 80% for training and 20% for testing. We set $\lambda_U = 0.005$ and $\lambda_V = 0.005$ for PMF. For Poisson factor based models used in this experiment, we set $\alpha_U = 5,\ \alpha_V = 20$ and $\beta_U = \beta_V = 0.2$. We set $1/\mathbb{R}$ for user region multinomial prior γ. We set $\tau = 1$ and $d_0 = 0.2$ for the distance model $\left[\frac{d_0}{d_0+d(i,j)}\right]^{\tau}$. For Geo-PFM and Fu-PoisonFM, we first cluster all the POIs into $|R|$ regions. This is the initialization of the Geo-PFM model. We set the number of regions $|R| = 49$, which is the number of regions partitioned according to all the states in USA (except Hawaii and Alaska). All the latent factor models are implemented with stochastic gradient ascent/descent optimization method with an annealing procedure to discount learning rate ϵ at iteration nIter with $\epsilon^{\text{nIter}} := \epsilon\frac{\nu}{\nu+\text{nIter}-1}$ by setting $\nu = 10$.

3.5.4 Performance Comparison

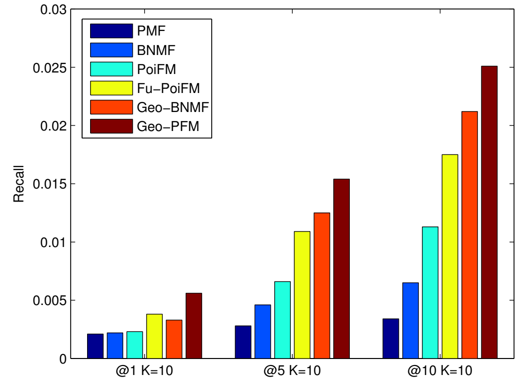
In this subsection, we present the performance comparison on recommendation accuracy between our model and the baseline methods[2] . We compare the results using both the Foursquare and the Gowalla dataset by setting latent dimensions to $K = 10$ and $K = 20$.

Foursquare Dataset. Figure 3.6 shows the precision and recall@N ($N = 1, 5, 10$) all the methods achieve on the Foursquare dataset, and Table 3.3 shows the F_β measure ($\beta = 0.5$). From the results, it is clear that the proposed Geo-PMF consistently outperforms all the baseline methods, including the non-geographical baseline models (PMF, BNMF, PoiFM) as well as the fused model (Fu-PoiFM). Specifically, nonnegative based Poison factor model (PoiFM) and BNMF outperform PMF. Furthermore, PoiFM outperforms its counterpart BNMF by making Poisson assumptions. The

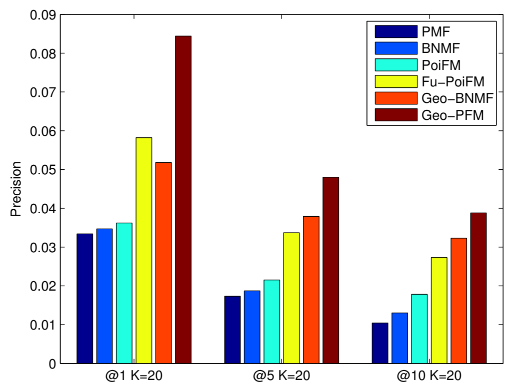
[2] In the experiments of this chapter, we rank the top-N recommendation globally, which is different from the regional way we used in (B. Liu et al., 2013). Also we further tune some parameters. Therefore, the absolute experimental values in this chapter may somewhat differ from those in (B. Liu et al., 2013)



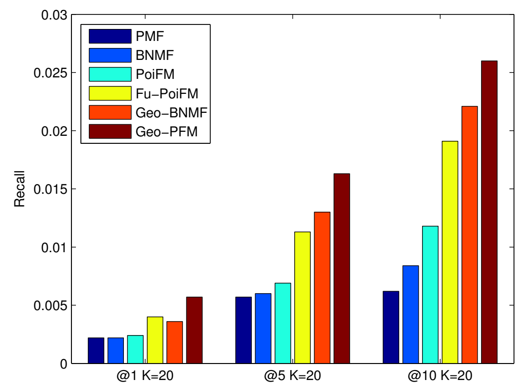
(a) Precision, K=10.



(b) Recall, K=10.



(c) Precision, K=20.

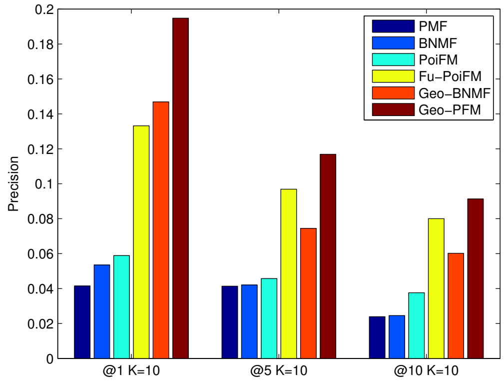


(d) Recall, K=20.

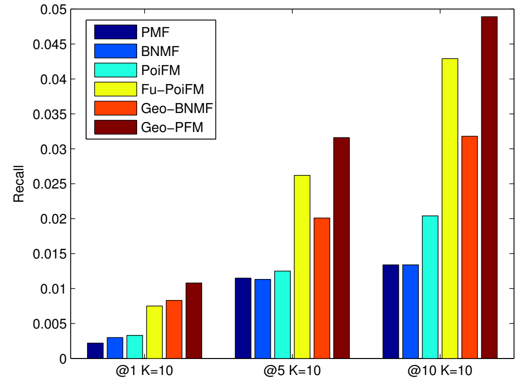
Figure 3.6: Precision and Recall with two different latent dimensions K (Foursquare dataset). Note that we focus on two comparisons: (1) How the proposed Geo-PFM improves the non-geographical baseline models (PMF, BNMF, PoiFM) as well as the fused model (Fu-PoiFM); (2) How the Poisson based model Geo-PFM improves its non-negative factorization based counterpart Geo-BNMF.

fused method, Fu-PoiFM, improves PoiFM due to the fusion of geographical influence and multi-center user activity pattern into the latent factor model. Our proposed Geo-PFM further improves Fu-PoiFM significantly. From Table 3.3, we can observe an average of 0.0089 improvement in terms of F_β measure for Geo-PFM over Fu-PoiFM.

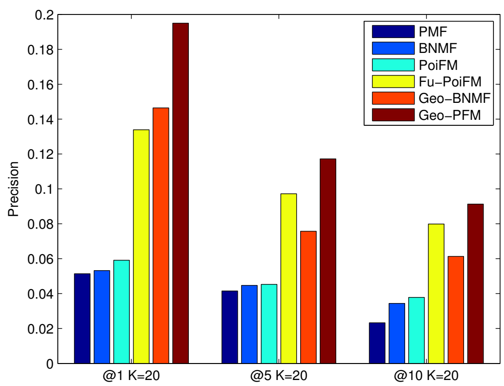
Meanwhile, from Figure 3.6 we can see that the Poisson-based model Geo-PFM



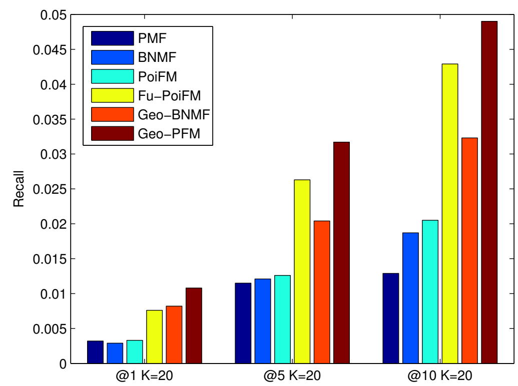
(a) Precision, K=10.



(b) Recall, K=10.



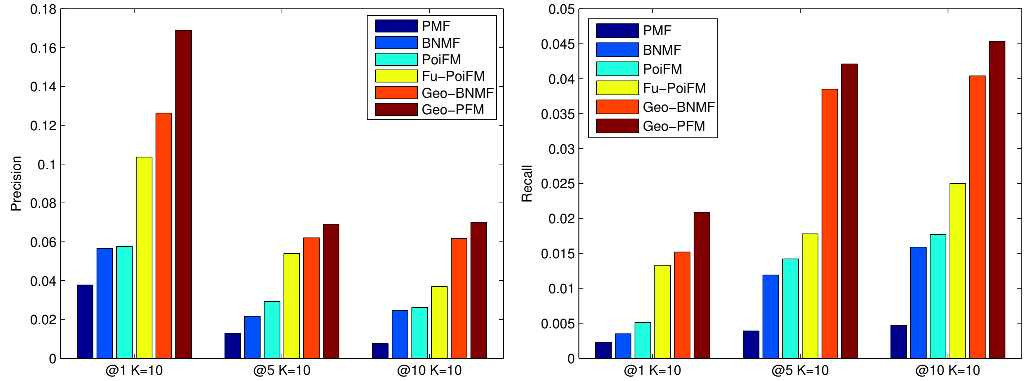
(c) Precision, K=20.



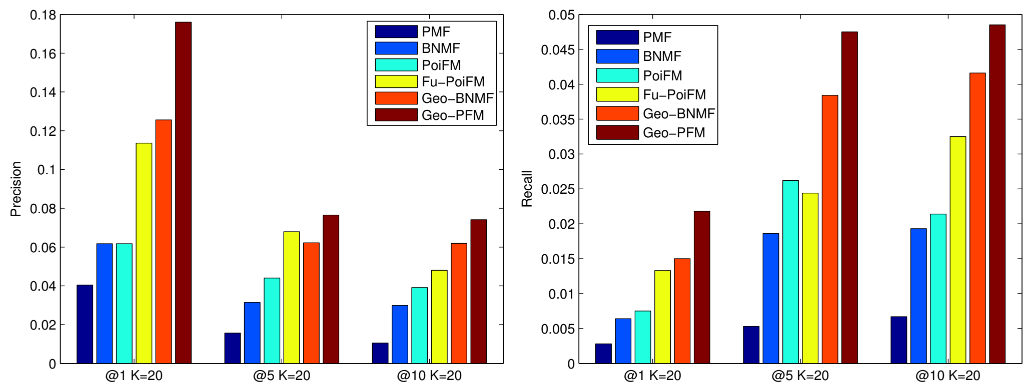
(d) Recall, K=20.

Figure 3.7: Precision and Recall with two different latent dimensions K (Gowalla dataset).

improves its counterpart based on non-negative factorization, Geo-BNMF, with an average of 0.0069 improvement in terms of F_β measure. This improvement can be ascribed to the following reasons. First, the Poisson-based latent factor is more appropriate for modeling count data. As shown, the improved performance of PoiFM over BNMF from Figure 3.6, PoiFM can improve BNMF with an average of 0.0032 improvement in terms of F_β. Second, the Poisson Geo-PFM provides a more rigorous probabilistic generative process for the model, while the non-negative matrix factor-



(a) **Precision, K=10.** (b) **Recall, K=10.**



(c) **Precision, K=20.** (d) **Recall, K=20.**

Figure 3.8: Precision and Recall with two different latent dimensions K (Brightkite dataset).

ization based Geo-PFM applied an approximation solution. As shown in the model estimation in Section 3.4.2, we need a rigorous probability for model inference. While the Poisson based model provides an exact probability representation, the Geo-BNMF applies a rectified normal distribution.

Gowalla Dataset. Figure 3.7 shows the precision and recall@N ($N = 1, 5, 10$) of all the methods evaluated on the Gowalla dataset, and the corresponding F_β measure values are shown in Table 3.4. We can clearly observe that the proposed Geo-PFM

Table 3.3: F_β measure ($\beta = 0.5$) with two different latent dimensions K (Foursquare dataset).

K	@N	PMF	BNMF	PoiFM	Fu-PoiFM	Geo-BNMF	Geo-PFM
10	@1	0.0083	0.0087	0.0091	0.0150	0.0130	**0.0220**
	@5	0.0061	0.0100	0.0145	0.0236	0.0263	**0.0328**
	@10	0.0047	0.0090	0.0159	0.0242	0.0339	**0.0339**
20	@1	0.0087	0.0088	0.0095	0.0157	0.0141	**0.0224**
	@5	0.0123	0.0131	0.0151	0.0241	0.0274	**0.0346**
	@10	0.0092	0.0117	0.0162	0.0251	0.0296	**0.0353**

performs consistently better over all the baseline methods. From Table 3.4, we can observe an average of 0.0121 improvement in terms of F_β measure for Geo-PFM over Fu-PoiFM. We further observe that the Poisson-based Geo-PFM improves Geo-BNMF by an average of 0.0207 increase in terms of F_β measure.

Table 3.4: F_β measure ($\beta = 0.5$) with two different latent dimensions K (Gowalla dataset).

K	@N	PMF	BNMF	PoiFM	Fu-PoiFM	Geo-BNMF	Geo-PFM
10	@1	0.0091	0.0123	0.0135	0.0306	0.0338	**0.0442**
	@5	0.0272	0.0272	0.0298	0.0629	0.0483	**0.0759**
	@10	0.0207	0.0221	0.0322	0.0682	0.0551	**0.0778**
20	@1	0.0128	0.0119	0.0135	0.0310	0.0335	**0.0442**
	@5	0.0273	0.0290	0.0298	0.0632	0.0491	**0.0761**
	@10	0.0201	0.0295	0.0323	0.0681	0.0520	**0.0779**

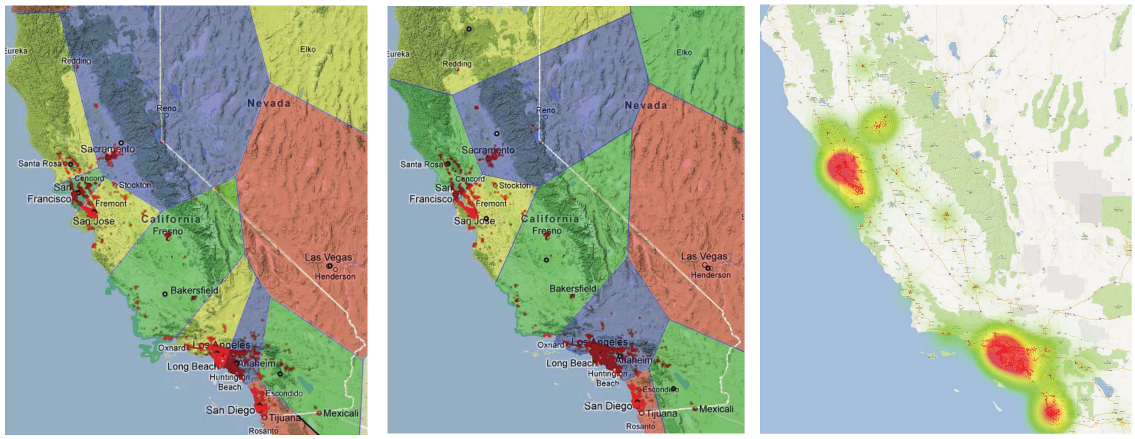
Brightkite Dataset. Figure 3.8 shows the precision and recall@N ($N = 1, 5, 10$) of all the methods evaluated on the Gowalla dataset, and the corresponding F_β measure values are shown in Table 3.4. We can still observe consistent improvements of the proposed Geo-PFM over all the baseline methods. From Table 3.5, we can observe an average of 0.0246 improvement in terms of F_β measure for Geo-PFM over Fu-PoiFM. Again, we further observe that Poisson-based Geo-PFM improves Geo-BNMF with an average of 0.0129 increase in terms of F_β measure.

Table 3.5: F_β measure ($\beta = 0.5$) with two different latent dimensions K (Brightkite dataset).

K	@N	PMF	BNMF	PoiFM	Fu-PoiFM	Geo-BNMF	Geo-PFM
10	@1	0.0092	0.0140	0.0188	0.0439	0.0513	**0.0699**
	@5	0.0088	0.0186	0.0241	0.0383	0.0553	**0.0612**
	@10	0.0067	0.0221	0.0238	0.0337	0.0558	**0.0632**
20	@1	0.0110	0.0226	0.0252	0.0453	0.0508	**0.0729**
	@5	0.0112	0.0276	0.0387	0.0501	0.0553	**0.0682**
	@10	0.0094	0.0269	0.0336	0.0438	0.0564	**0.0670**

Comparisons Across Different Datasets. First, we observed consistent improvements of the proposed Geo-PFM over all the baseline methods, though the three dataset differ in terms of user-POI observation sparsity, response skewness, and POI geographical distributions (see Figure 3.4). Second, the Poisson-based Geo-PFM improves its counterpart based on non-negative factorization, Geo-BNMF. Third, user-POI observation sparsity, response skewness and POI geographical distributions could affect the algorithm performances. For example, the results on the Gowalla dataset and the Brightkite dataset are better than those on the Foursquare dataset. The Gowalla dataset is much denser than the Foursquare dataset. Note that Gowalla dataset has a sparsity of 99.76%, and an average of 59.44 user-POI observations; while the Foursquare dataset has a sparsity of 99.87%, an average of 73.68 user-POI observations. Although Brightkite dataset has fewer user-POI observations, on average, than Foursquare dataset, its sparsity is the lowest among the three datasets. Further, the Gowalla dataset is less skewed than the Foursquare dataset. These two factors could allow the latent factor models, both PMF and PoiFM, to achieve better performances. Also, the Gowalla dataset is more geographically centralized than the Foursquare dataset. As a result, the performances of Geo-PFM would be more obvious compared to Fu-PoiFM when applied to more geographically distributed circumstances.

Latent Region Analysis. In addition to improving recommendation performance, our proposed model also provides a unique perspective on POI marketing segmentation, in the form of the learned regions. We take a representative area, California, as an example to analyze the regions learned by the Geo-PFM model. Figure 3.9 visualizes the latent regions (Figure 3.9b) learned from our model in versus its initialization by K-means (Figure 3.9a) in California. Though we have no ground truth about an optimal POI region segmentation, we can infer the user activity regions in California through the collective check-in behaviors of users who have visited California and view the region clusters formulated by collective check-ins as ground truth (see Figure 3.9c). Through analyzing the collaborative check-in frequency by those users, as shown in Figure 3.9c, we can see two clear clusters in northern California among other scattered POIs, one cluster in the Los Angeles area, one in San Diego, and some scattered POIs between southern and northern California. K-means only depends on POI distances to cluster POIs for region segmentation. As shown in Figure 3.9a, K-means segments northern California into four different regions, and segments



(a) K-means. (b) Latent region. (c) Ground truth.

Figure 3.9: Voronoi visualization of POI segmentation in California area (Foursquare): (b) latent regions learned by Geo-PFM, (a) initiation by K-means, and (c) true user collaborative activity clusters. Deeper color (red) indicates more check-ins for a POI, as contrary to light color (green). Best view in color.

Los Angeles into two regions. However, by considering the user check-in behaviors and geographical factors, our model identified a more meaningful region partition as shown in Figure 3.9b, which is more coherent to real user activity as shown in Figure 3.9c. Geo-PFM initiated by K-means leads to better POI segmentation. We can see that Geo-PFM models not only improve recommendation performance, but also provide an interesting perspective on POI marketing segmentation in the form of the learned regions.

Summary. Geographical influence and user mobility are two of the most important characteristics for LBSNs, and play an important role in POI recommendation. The fused method (Fu-PoiFM) which exploits an *ad hoc* two-step process to fuse the geographical influence and multi-center user activity pattern into user preferences can improve pure latent factor model (PoiFM). However, an integrated analysis of multiple factors for POI recommendations lead to further improvements. The proposed Geo-PFM model not only considers the geographical information of POIs and user mobility patterns for recommendation, but also updates the latent regions by considering these sources of information. The learned regions reflect the collaborative user activity pattern. As a result, we can observe obvious improvements over all the baseline algorithms. Also, as shown in the performance of Poisson factor model compared to its Gaussian counterpart PFM, we observe improvements by Poison factor model, as Poisson distribution is more suitable for modeling count data. Further evidence of this is the fact that the Poisson based model Geo-PFM improves its non-negative factorization based counterpart Geo-BNMF in all the evaluation datasets, though Geo-BNMF imposes a non-negativity constrain.

3.6 Related Work

Recommender systems can be developed based on explicit user feedback. In other words, users rate items and the user-item preference relationship can be modeled on the basis of the user ratings. Latent factor models, such as as matrix factorization (Koren et al., 2009), probabilistic matrix factorization (PMF) (Salakhutdinov & Mnih, 2008b), its non-parametric version (Salakhutdinov & Mnih, 2008a), and other other variants (Koren, 2008; Agarwal & Chen, 2009; R. Bell et al., 2007; Koren, 2010; L. Zhang et al., 2011; Liang Xiong, 2010), have become popular and widely used in recommendation. Most of the latent factors along this line of work assume that the response follows a Gaussian distribution over the product of user and item latent factors. The Gaussian-based latent factor models can achieve good prediction performance when explicit ratings are available. In contrast, recommender systems can also be developed based on implicit user feedback (Y. Hu, Koren, & Volinsky, 2008a), such as the search and click behaviors on a web site (Ma et al., 2011), advertisement targeting (Y. Chen et al., 2009), and the check-in behaviors in LBSNs, as we discussed in this chapter. In this case, the recommender system has to infer user preferences from implicit user feedback. Here, latent factor models which are suitable for implicit user feedback are preferred. One option is to set non-negative constraints on latent factors to force the response variable into a wider range than the rating-based response. As a result, methods based on non-negative matrix factorization are widely used (Lee & Seung, 2000; S. Zhang, Wang, Ford, & Makedon, 2006; Gu, Zhou, & Ding, 2010; C. Liu, Yang, Fan, He, & Wang, 2010). However, the Poisson distribution is suitable for modeling count data. As a result, Poisson factor models are widely used for count based feedback recommendation settings (Canny, 2004; Ma et al., 2011; Y. Chen et al., 2009; Gopalan et al., 2013).

Some previous studies on POI recommendation, or more precisely location recommendation, mainly relied on user trajectory data to infer user preferences. For example, previous works (Y. Zheng et al., 2009; V. W. Zheng et al., n.d.; V. W. Zheng, Cao, Zheng, Xie, & Yang, 2010; Ge et al., 2010; Y. Zheng & Xie, 2011) applied collaborative filtering (CF) methods to recommend locations and taxi pick-up locations based on user trajectory data. However, POI recommendation provide exact POIs a user would be interested rather than a “location”. Due to the development and popularity of location-based social networks, more recent works, such as (Ye et al., n.d.; Ye, Yin, & Lee, 2010), began to explore user preferences, social influence, and geographical influence for POI recommendations. However, these used a simple CF algorithm to fuse this information, and thus lack a comprehensive way to model how all this information collectively influence user POI check-in decision. The work in (B. Liu & Xiong, 2013) tried to explore side information to improve POI recommendations, but it does not explore user mobility information and does not take the skewed data characteristics of implicit user check-in counts into the consideration. Kurashima *et.al* (Kurashima, Iwata, Hoshide, Takaya, & Fujimura, 2013) extended the latent Dirichlet allocation (LDA) model to include geographical influence to profile user location preference, but it did not consider user mobility and the user activity areas modeled in this chapter are constrained only to areas that a user has traveled to.

More recently, Cheng et al. (C. Cheng et al., 2012) considered the geographical influence, the multi-center of user check-in patterns, the skewed user check-in frequency and social networks for POI recommendation. However, this work applied an *ad hoc* two-step method to fuse the geographical influence into user preferences, and did not really consider the user mobility and lacked an integrated consideration of factors that can influence POI recommendation. Moreover, the greedy clustering method applied to derive the personalized multi-centers could easily lead to overfitting problems in that it focuses on the regions a user has visited. Instead, our work is an integrated analysis of geographical influences, user mobility, and skewed data for POI recommendation. Hu and Ester (B. Hu & Ester, 2013) proposed a spatial topic model by considering the spatial and textual aspects of posts published by mobile users, and predict future user locations as POI recommendation. This is the work most closely related to ours in terms of the way to account for geographical influence and user mobility. However, their work is more similar to a location prediction problem than a POI recommendation task. Moreover, the Poisson model used in this chapter could be equivalent conditioned on the per-user sums and where the item weights are constrained to sum to one (Gopalan et al., 2013; Zhou, Hannah, Dunson, & Carin, 2012; Zhou & Carin, n.d.). However, our proposed Geo-PFM is more flexible and can be extended to different latent factor settings.

In addition, our work has a connection with recent works on mobility modeling (Hong et al., 2012; Cho et al., 2011). However, their tasks were different. Work (Hong et al., 2012) used a similar multinomial assumption over different regions to model geographical topics in Twitter stream, and the work in (Cho et al., 2011) investigated human mobility for social network analysis. Also, people have used Gaussian distribution to model region over locations (Sizov, 2010; Z. Yin et al., n.d.; Hong et al., 2012).

As described above, while there are some studies on POI recommendation, they lacks an integrated analysis of the joint effects of multiple factors that influence the decision process of a user choosing a POI. These factors include user interest preferences, geographical influences, user mobility pattern, and the skewed implicit user check-in count data. The proposed method strategically takes all these factors into consideration and presents a flexible probabilistic generative model for POI recommendations.

3.7 Summary

In this chapter, we presented an integrated analysis of the joint effect of multiple factors which influence the decision process of a user choosing a POI and proposed a general framework to learn geographical preferences for POI recommendation in LBSNs. The proposed geographical probabilistic factor analysis framework strategically takes all these factors, which influence the user check-in decision process, into consideration. There are several advantages of the proposed recommendation method. First, the model captures the geographical influence on a user's check-in behavior by taking into consideration the geographical factors in LBSNs, such as the Tobler's first law of geography. Second, the methods effectively modeled the user mobility patterns, which are important for location-based services. Third, the proposed approach extended the latent factors from explicit rating recommendation to implicit feedback recommendation settings by considering the skewed count data characteristic of LBSN check-in behaviors. Last but not least, the proposed model is flexible and could be extended to incorporate different latent factor models, which are suitable for both explicit and implicit feedback recommendation settings. Finally, extensive experimental results on real-world LBSNs data validated the performance of the proposed method.

Limitations and Discussion. Geographical influence and user mobility are among the most important characteristics in LBSNs and could greatly affect POI recommendation. The proposed Geo-PFM model captures these two factors by introducing latent regions, which represent the collective user activity areas. This method coarsely captures the geographical influence and user mobility. However, the geographical influence and user mobility can be subtle (Cho et al., 2011; Scellato, Noulas, & Mascolo, 2011). A possible future direction is to combine both the macroscopic and microscopic effects of geographical influence and user mobility.

CHAPTER 4
HIERARCHY AWARE RECOMMENDATION FOR MOBILE APPS

4.1 Introduction

Recent years have witnessed the tremendous growth in mobile devices among an increasing number of users, and the penetration of mobile devices into every component of modern life. Indeed, the smartphone market surpassed the PC market in 2011 for the first time in history [1] . Thereafter, the smartphone market has continued to increase dramatically, *e.g.*, the smartphones shipped in the third quarter of 2013 increased 44% year-on-year [2] . One of the reasons lies in the fact that users are able to augment the functions of mobile devices by taking advantage of various feature-rich third-party *applications* (or apps for brevity), which can be easily obtained from centralized markets such as Google Play and App Store. However, a huge number of mobile apps has imposed the challenge of finding the right apps to meet the user needs. For instance, as of July 2013, Google Play had over 1 million apps with over 50 billion cumulative downloads, and the number of apps had reached over 1.4 million in January 2015 [3] ; as of February 2015, App Store had over 1.4 million apps and a cumulative of over 100 billion apps downloaded [4] . As a result, there is a critical

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

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

[3] Google Play Statistics, Retrieved January 2015, `http://en.wikipedia.org/wiki/Google_Play`

[4] App Store Statistics, Retrieved January 2015, `http://en.wikipedia.org/wiki/App_Store_(iOS)`

demand for effective personalized app recommendations.

However, for the development of personalized app recommender systems, there are opportunities and challenges posed by two unique characteristics of mobile apps. First, application stores have organized apps in a hierarchical taxonomy. For instance, Google Play groups the apps into 27 categories, such as *social*, *games*, and *sports* according to their functionalities. These categories can be further divided into subcategories, *e.g* apps in the category of *games* are further divided into subcategories such as *action*, *arcade*, and *puzzle*. For the apps in the same category or subcategory, they have similar functionalities. Then, how a user navigates through the hierarchy to locate relevant apps represents a fine-grained interest preference of the user. Thus, the first challenge is how to leverage this hierarchical taxonomy of apps to better profile user interests and enhance app recommendations. Second, apps with similar functionalities are competing with each other. For instance, when a user has already adopted Google Maps as his/her navigation tool, the user might not be interested in other navigation tools such as Apple Maps. While there are a variety of existing approaches for mobile app recommendations, these approaches do not have a focus on dealing with these opportunities and challenges.

Instead, in this chapter, we provide a systematic study to address these challenges. Specifically, we first develop a *structural user choice model* (SUCM) to learn fine-grained user preferences by exploiting the hierarchical taxonomy of apps as well as the competitive relationships among apps.[5] Since apps are organized as a hi-

[5] Note that the model and algorithms developed in this chapter can also be applied to other domains where items are organized in a hierarchical way.

erarchical taxonomy, we model the user choice as two phases. In the first phase, a user decides which type of apps to choose and then moves to the appropriate app category/subcategory. In the second phase, the user chooses apps in the selected category/subcategory. Such structural user choice is modeled by a unique *choice path* over the tree hierarchy, where the choice path starts from the root of the hierarchy and goes down to the app that is selected by a user. In each step of moving along the choice path, the competitions between the candidates (*i.e.*, either the same level categories/subcategories or apps in a chosen category/subcategory) play an important role in affecting user's choices. We capture the structural choice procedure by cascading user preferences over the choice paths through a probabilistic model. Specifically, in our probabilistic model, motivated by the widely used discrete choice models in economics (Luce, 1959; McFadden, 1973; Manski, 1977), we model the probability that a user reaches a certain node in the choice path as a *softmax* of the user's preference on the chosen node over the user's preference on all the nodes at the sample level. The softmax function is used to capture the competitions between categories/subcategories or apps in a category/subcategory. Moreover, we model a user's preference over one node using latent factors, which enables us to capture the correlations between nodes.

Moreover, we design an efficient learning algorithm to estimate the parameters of the SUCM model. The major challenge of learning the parameters lies in the softmax on the leaf nodes (apps) of the tree hierarchy. Indeed, it is not practical to learn these softmax functions for a subcategory of apps by directly applying Stochastic Gradient Descent (SGD), because the time complexity of one SGD step is linear to the number

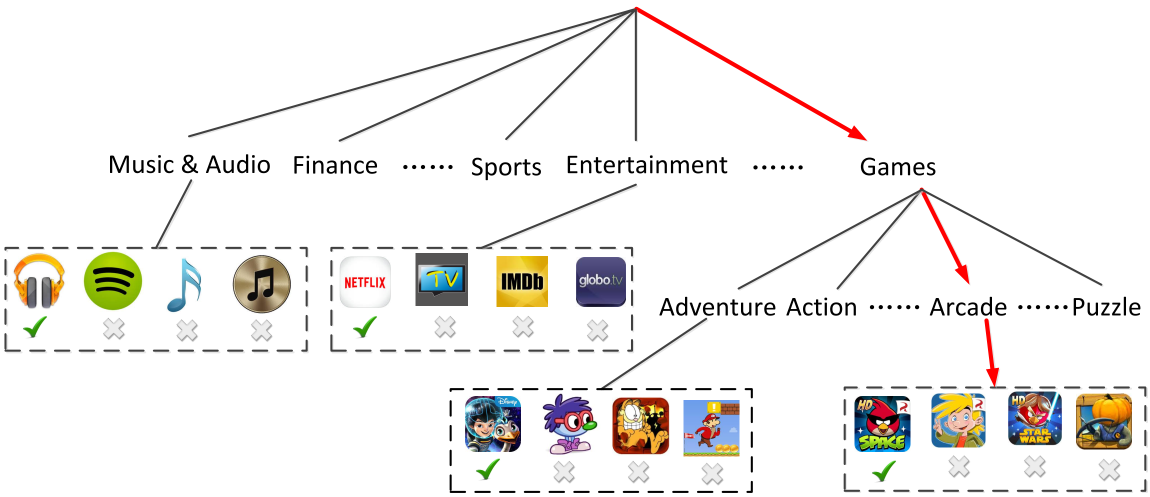


Figure 4.1: An illustrative example of structural user choice for app adoption in Google Play. First, apps are organized into a Category Tree. Second, as illustrated by the highlighted and arrowed path, a user makes an app adoption by traversing a *choice path* from the root of the tree to the chosen app.

of apps under the subcategory, which might be very large. To address this challenge, we relax the softmax term in each subcategory into a hierarchical softmax, thus the time complexity of learning parameters is reduced to be logarithm of the number of apps under the subcategory.

Finally, we collected a large-scale dataset from Google Play to evaluate our approach and compare SUCM with state-of-the-art approaches. The experimental results show that SUCM consistently outperforms these methods with a significant margin in terms of a variety of widely used evaluation metrics for Top-N recommendation.

4.2 Problem Definition

We first introduce three key concepts and then formally define our app recommendation problem.

Definition 1 (Category Tree) *A Category Tree (denoted as* Γ*) is a data structure to organize apps according to their properties (e.g functionalities). Figure 4.1 shows an example Category Tree adopted by Google Play. In a Category Tree, internal nodes represent categories or subcategories, leaf nodes represent apps, and the children of an internal node represent the subcategories or the apps that belong to the category/subcategory represented by the node. We use* z *to denote an internal node in* Γ*, and we denote by* $level(z)$*,* $c(z)$*,* $\pi(z)$*, and* $s(z)$ *the level, the children, the parents, and the siblings of* z*, respectively. Moreover, we use* z_M *to denote an internal node whose children are leaf nodes and use* i *to represent an app.*

We note that an app might belong to multiple categories due to the rich functionalities provided by it, which makes the category hierarchy not a tree. However, we found that mobile markets such as Google Play do not place an app into multiple categories based on the dataset we collected from Google Play, and thus we do not consider this scenario.

Definition 2 (Choice Path) *A choice path is a sequence of nodes that a user traverses through the Category Tree* Γ*, starting from the root and ending at a leaf node which corresponds to the app selected by the user. For instance, if a user adopts an app* i*, the choice path can be represented as* $\mathrm{path}_i = z_0 \to z_1 \to \cdots \to z_M \to i$*. Note that, given the Category Tree* Γ*, the choice path* path_i *for app* i *is unique.*

Definition 3 (Competing Apps) *For an app* i*, we denote by* $\mathcal{A}(i)$ *the set of apps that have competing properties (e.g functionalities) and compete with* i *to attract users. In this chapter, we treat the siblings of an app* i *under the same category/subcategory in the Category Tree as the competing apps.*

We note that users might have multiple ways to adopt apps, *e.g*suggestions from friends, recommendations from Google Play store, etc. However, we assume that no matter in which way a user is aware of an app, the decision is made on the functionality of the app and its competitors with similar functionalities, thus following the choice path we discuss above.

It also should be noted that we do not assume a user only adopts one app in a subcategory. The category/subcategory in the Category Tree provided by mobile markets such as Google Play is not fine-grained enough so some siblings of the app i might provide slightly different functionalities with i. For example, Facebook, LinkedIn and Twitter all belong to *Social* category. We model the process of one user adopting an app using a structural choice model. If a user selects multiple apps under a same category, the joint probability of selecting them together would be optimized (see details in Section 4.3).

Given the above three concepts, we can formally define our app recommendation problem as follows: suppose we are given a set of users denoted as $\mathcal{U} = \{1, 2, ..., U\}$, a set of apps denoted as $\mathcal{I} = \{1, 2, ..., I\}$, the apps are organized into a predefined Category Tree Γ, each app i has a set of competing apps $\mathcal{A}(i)$, a set of adoption records $\{(u, i)\}$ indicating which users have adopted which apps, then our goal is to recommend each user a list of apps that match his/her interest preference. In the rest of the chapter, we use u to index users, and i and j to index apps. Moreover, we use the two terms *app* and *item* interchangeably. Table 4.1 shows some important notations used in this chapter.

Table 4.1: Mathematical Notations

Symbol	Description
u	user index for user set $\mathcal{U} = \{1, 2, ..., U\}$
i, j	app index for app set $\mathcal{I} = \{1, 2, ..., I\}$
Γ	predefined Category Tree
z	internal node in Category Tree Γ, in particular, z_M denotes a node whose children are leaf nodes
path_i	choice path in Γ: $z_0 \to z_1 \to \cdots \to z_M \to i$
$\pi(z), s(z), c(z)$	parent, sibling, and children of internal node z in the Category Tree Γ
$\mathbf{p}_u, \mathbf{q}_i, \mathbf{q}_z$	latent factor vector for user u, app i, and internal node z in Γ
b_i, b_z	bias term for app i and internal node z
y_{ui}	affinity score of user u for app i
y_{uz}	affinity score of user u for internal node z
$\mathcal{D} = \{(u, i)\}$	observed user-app adoption instances
$\mathcal{D}_u$	adopted apps by user u

4.3 Structural User Choice Model

In this section, we present our *structural user choice model* (SUCM) to learn fine-grained user interest preference via leveraging the Category Tree and competitions between apps for app recommendation.

4.3.1 Model Structural User Choice

As shown in Figure 4.1 , given a Category Tree Γ, there exists one unique *choice path* from the root node to app i, namely,

$$\text{path}_i = \underbrace{z_0 \to z_1 \to \cdots \to z_M}_{\substack{\text{Phase I:} \\ \text{locate a subcategory}}} \underbrace{\longrightarrow i}_{\substack{\text{Phase II:} \\ \text{choose an app}}}.$$

We see the structural user choice consists of two adoption phases. In the first phase, a user decides what types of apps to choose and moves to the appropriate category or subcategory in the Category Tree, namely, traverses $z_0 \rightarrow z_1 \rightarrow \cdots \rightarrow z_M$. In the second phase, the user makes app adoption decisions by choosing app i among all competing apps under the located subcategory z_M. For example, if a user wants to select the app *Angry Birds* under the subcategory *Arcade*, he would first consider the *Games* category and then further locates himself at the *Arcade* subcategory before he finally chooses app *Angry Birds*.

We model the process of a user u traversing path $z_0 \rightarrow z_1 \rightarrow \cdots \rightarrow z_M \rightarrow i$ as a sequence of decisions made for the multiple competing choices at each choice step. Specifically, in each step among this decision-making sequence:

- for choosing category or subcategory, user u chooses one child node z from all the children $c(\pi(z))$ of z's parent node $\pi(z)$;
- for choosing app, user u chooses app i from all the children of i's parent node, namely z_M.

Each decision making step can be seen as a discrete choice model, whose theoretical foundation is the neoclassical economic theory on preferences and utility built on a set of axiomatic assumptions (Luce, 1959; McFadden, 1973; Manski, 1977). The discrete choice model implies that a user u is endowed a *utility value* $f(u, z)$ to each alternative z in a choice set $\mathcal{A}(z)$. In our recommendation task, the utility value $f(u, z)$ can be the affinity score, which captures user preferences, between user u and choice z. Following the random utility model (Manski, 1977), we model the utility as a random variable

$$\nu_{uz} = f(u, z) + \varepsilon_{uz}, \tag{4.1}$$

where $f(u, z)$ is the deterministic part of the utility reflecting user preference, and ε_{uz} is the stochastic part capturing the impact of all unobserved factors that affect the user's choice. By assuming the stochastic part ε_{uz} be an independently and identically distributed log Weibull (type I extreme value) distribution, we can obtain the the multinomial choice model (McFadden, 1973). Specifically, in a multinomial choice model, the probability of a user u choosing z from a choice set $\mathcal{A}(z)$takes the form of

$$\Pr(\text{user } u \text{ choose } z|\mathcal{A}(z)) = \frac{\exp(f(u, z))}{\sum_{z' \in \mathcal{A}(z)} \exp(f(u, z'))}, \tag{4.2}$$

where $f(u, z)$ is a user preference depended utility function. This choice model also holds for user u choosing app i from app choice set $\mathcal{A}(i)$. Note that the choice model $\frac{\exp(f(u,z))}{\sum_{z' \in \mathcal{A}(z)} \exp(f(u,z'))}$ turns out to be a *softmax* function of utility value $f(u, z)$. In the following, we elaborate how we model each phase.

Phase I: Model category/subcategory preference. Following the latent factor models that are widely used in conventional recommender systems (Salakhutdinov & Mnih, 2008b; Koren, 2008), we use a latent factor vector $\mathbf{p}_u \in \mathbb{R}^K$ to represent a user's latent interest, where K is the dimension of the latent factor vector. Intuitively, $\mathbf{p}_u$ captures the interest of the user u. To capture the hierarchical structural user choice, we associate an internal node z in the Category Tree with a latent factor vector $\mathbf{q}_z$, which represents the properties ($e.g$ functionalities) of z in the latent space. Moreover, we define the affinity score between a user u and an internal node z as

$$y_{uz} = b_z + \mathbf{p}_u^\top \mathbf{q}_z, \tag{4.3}$$

where b_z is a bias term for the node z. The category/subcategory node affinity score represents the preference of a user over the category or the subcategory of apps (*e.g* Games).

We model the process of a user locating a subcategory as a sequence of decisions made for the multiple competing choices, starting from the root node and moving along the Category Tree towards the internal node corresponding to the subcategory. Specifically, in each step among this decision-making sequence, user u chooses one child node z from all the children of z's parent node $\pi(z)$. Following the choice model as shown in Equation (4.2), we assume the utility as the affinity score between user u and internal node z, *i.e*

$$f(u, z) = y_{uz} = b_z + \mathbf{p}_u^\top \mathbf{q}_z.$$

Then we model the probability of user u choosing the child z from all the children $c(\pi(z))$ of z's parent node $\pi(z)$ as a *softmax* function of the affinity scores between the user u and the internal nodes $c(\pi(z))$. Formally, we have:

$$\Pr(z|u, \pi(z)) = \frac{\exp(y_{uz})}{\sum_{z' \in c(\pi(z))} \exp(y_{uz'})} \tag{4.4}$$

The *softmax* function is used to model the competitions between the nodes in $c(\pi(z))$. As a result, the probability of user u traverses $z_0 \to z_1 \to \cdots \to z_M$ to reach the subcategory z_M is cascaded as

$$\begin{aligned}
\Pr(z_0 \to z_1 \to \cdots \to z_M|u) &= \prod_{m=1}^{M} \Pr(z_m|u, z_{m-1}) \\
&= \prod_{m=1}^{M} \frac{\exp(y_{uz})}{\sum_{z' \in c(z_{m-1})} \exp(y_{uz'})} \\
&= \prod_{m=1}^{M} \frac{\exp(b_z + \mathbf{p}_u^\top \mathbf{q}_z)}{\sum_{z' \in c(z_{m-1})} \exp(b_{z'} + \mathbf{p}_u^\top \mathbf{q}_{z'})}.
\end{aligned} \tag{4.5}$$

Phase II: Model app adoption. After a user locates at a specific subcategory node z_M whose children are all apps, the user makes an app adoption decision by choosing an app i among all competing choices $c(z_M)$. We use a latent factor vector $\mathbf{q}_i \in \mathbb{R}^K$ to represent the latent factor of app i. Intuitively, $\mathbf{q}_i$ encodes the properties ($e.g$ functionalities) of app i. Moreover, we define the affinity score between user u and app i as

$$y_{ui} = b_i + \mathbf{p}_u^\top \mathbf{q}_i, \tag{4.6}$$

where b_i is a bias term for app i. Again, following the choice model as shown in Equation (4.2), we assume the utility as the affinity score between user u and app i, $i.e$

$$f(u, i) = y_{ui} = b_i + \mathbf{p}_u^\top \mathbf{q}_i.$$

Then we model the probability of user u selecting app i over its competing alternatives under the subcategory node z_M using a softmax function as follows:

$$\begin{aligned}
\Pr(i|u, z_M) &= \frac{\exp(y_{ui})}{\sum_{j \in c(z_M)} \exp(y_{uj})} \\
&= \frac{\exp(b_i + \mathbf{p}_u^\top \mathbf{q}_i)}{\sum_{j \in c(z_M)} \exp(b_j + \mathbf{p}_u^\top \mathbf{q}_j)},
\end{aligned} \tag{4.7}$$

where z_M is the parent node of app i and $c(z_M)$ includes all competing apps of app i and i itself. The softmax function is used to model the competitions between apps.

Model the overall structural choice probability. Note that there exists one unique *choice path* from the root node to app i, namely,

$$\text{path}_i = z_0 \to z_1 \to \cdots \to z_M \to i.$$

Then, the probability of user u choosing app i is the joint probability of u selecting each node in the choice path path_i, *i.e.*, we have:

$$\begin{aligned}
\Pr(i|u) &= \Pr(i|u, z_M) \times \Pr(z_0 \to z_1 \to \cdots \to z_M|u) \\
&= \Pr(i|u, z_M) \prod_{m=1}^{M} \Pr(z_m|u, z_{m-1}) \\
&= \frac{\exp(b_i + \mathbf{p}_u^\top \mathbf{q}_i)}{\sum_{j \in c(z_M)} \exp(b_j + \mathbf{p}_u^\top \mathbf{q}_j)} \prod_{m=1}^{M} \frac{\exp(b_{z_m} + \mathbf{p}_u^\top \mathbf{q}_{z_m})}{\sum_{z' \in c(z_{m-1})} \exp(b_{z'} + \mathbf{p}_u^\top \mathbf{q}_{z'})},
\end{aligned} \tag{4.8}$$

where the first term $\Pr(i|u, z_M)$ is user u's adoption probability of app i under subcategory node z_M and the second term $\prod_{m=1}^{M} \Pr(z_m|u, z_{m-1})$ captures the structural choice by cascading user preferences over the Category Tree Γ.

4.3.2 Model Structural App Dependences

Intuitively, nodes that are closer in the Category Tree Γ could have more similar properties. For instance, apps under the subcategory *action* are more similar to those under the subcategory *arcade* than those under the category *weather* because both *action* and *arcade* belong to the *games* category. Thus, we associate each internal node z with a latent variable $\mathbf{q}_z$ to represent the category/subcategory level properties, and we model the latent variable $\mathbf{q}_z$ as a function of the latent variable of z's parent node $\mathbf{q}_{\pi(z)}$ to capture the hierarchical structural dependences between the nodes in the

Category Tree. Formally, we have:

$$\mathbf{q}_z \sim \begin{cases} \mathcal{N}(0, \sigma^2 \mathbf{I}) & \text{if } z \text{ is the root node} \\ \mathcal{N}(\mathbf{q}_{\pi(z)}, \sigma^2 \mathbf{I}) & \text{otherwise} \end{cases} \tag{4.9}$$

where $\mathcal{N}(u, \sigma^2)$ is a normal distribution with mean u and standard deviation σ.

4.3.3 Discussion

Note that our model does not only capture the competitions between apps under the same categories, but also incorporates the correlations between apps via the latent factors.

- **Competition**. We use a softmax function to model the probability of selecting a child node (a subcategory or an app) under a category node. If user u selects a child node z from all the competing nodes $\mathcal{A}(z)$, the value of y_{uz} should be larger than all other $y_{uz'}$ where $z' \in \mathcal{A}(z)$ and $z \neq z'$. This model characteristic can address the cases when multiple apps in same categories are adopted.

- **Correlation**. The latent factor model is able to model the correlations between apps and categories. For example, if two categories are always liked by the same users, the latent factors of them will be close to each other in the latent space. As a result, if we know a user likes either of the two categories, the value of his/her preferences on the other one will also be large.

4.4 Parameter Estimation

Let $\Theta = \{\mathbf{p}_u, \mathbf{q}_i, \mathbf{q}_z, b_i, b_z\}_{u \in \mathcal{U}, i \in \mathcal{I}, z \in \Gamma}$ denote all parameters to be estimated. Given the observed user-app adoption records $\mathcal{D} = \{(u, i, \text{path}_i)\}$ and the category tree Γ, we have the posterior probability distribution of the parameters as follows:

$$\Pr(\Theta|\mathcal{D},\Gamma) \propto \prod_{u=1}^{U}\prod_{i\in\mathcal{D}_u}\Pr(i|u,z_M)\prod_{m=1}^{M}\Pr(z_m|u,z_{m-1})\prod_{\substack{m=1\\ \forall z\in\Gamma}}^{M}\Pr(\mathbf{q}_{z_m}|\mathbf{q}_{z_{m-1}},\sigma^2\mathbf{I}) \tag{4.10}$$

where the first term captures the structural user choices and the second term represents the hierarchical structural dependences of the nodes in the category tree. We estimate all the parameters via maximizing the log likelihood of the posterior,

$$\begin{aligned}\arg\max_{\Theta}\Bigg\{&\sum_{u=1}^{U}\sum_{i\in\mathcal{D}_u}\ln\Pr(i|u,z_M)+\sum_{u=1}^{U}\sum_{i\in\mathcal{D}_u}\sum_{m=1}^{M}\ln\Pr(z_m|u,z_{m-1})\\ &+\sum_{\substack{m=1\\ \forall z\in\Gamma}}^{M}\ln\Pr(\mathbf{q}_{z_m}|\mathbf{q}_{z_{m-1}},\sigma^2\mathbf{I})\Bigg\}.\end{aligned} \tag{4.11}$$

Note that the widely used regularizations for latent factor vectors (Salakhutdinov & Mnih, 2008b; Koren et al., 2009) can be applied here, but we exclude the regularization priors for presentation simplicity.

4.4.1 Hierarchical Softmax

One challenge of directly solving the objective function as shown in Equation (4.11) rests in the updating of all the parameters over the probability distribution $\Pr(i|u, z_M)$, namely, the first term in Equation (4.11),

$$\begin{aligned}\sum_{u=1}^{U}\sum_{i\in\mathcal{D}_u}\ln\Pr(i|u,z_M) &= \sum_{u=1}^{U}\sum_{i\in\mathcal{D}_u}\ln\frac{\exp(b_i+\mathbf{p}_u^\top\mathbf{q}_i)}{\sum_{j\in c(z_M)}\exp(b_j+\mathbf{p}_u^\top\mathbf{q}_j)}\\ &= \sum_{(u,i)\in\mathcal{D}}\left\{\left(b_i+\mathbf{p}_u^\top\mathbf{q}_i\right)-\ln\left[\sum_{j\in c(z_M)}\exp\left(b_j+\mathbf{p}_u^\top\mathbf{q}_j\right)\right]\right\},\end{aligned} \tag{4.12}$$

where $c(z_M)$ represents all the apps under the subcategory z_M. The updating computation cost for all the parameters in one user-app adoption instance (u, i) is linear to the number of apps under z_M, which might be very large. To address this challenge,

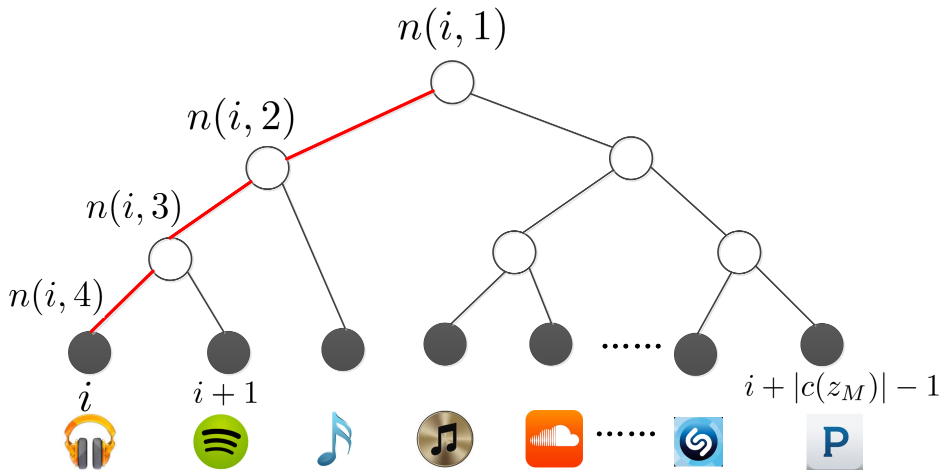


Figure 4.2: An illustrative example of binary tree for hierarchical softmax under a category/subcategory. All apps under the category/subcategory (*e.g*Music & Audio) are organized using a binary tree. The black nodes (leaf nodes) are apps, and the white nodes are internal nodes. One example path from root node to app i is highlighted as $n(i,1) \rightarrow n(i,2) \rightarrow n(i,3) \rightarrow n(i,4)$, which means the path length $L(i) = 4$.

we leverage *hierarchical softmax* to approximate $\Pr(i|u, z_M)$ efficiently. Hierarchical softmax was first introduced by Morin and Bengio (Morin & Bengio, 2005) for neural networks and recently was widely used in deep learning (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Mikolov, Chen, Corrado, & Dean, 2013). The main advantage of hierarchical softmax is that, in each training instance, instead of evaluating the parameters for all the children of z_M, we only need to evaluate parameters for $\log |c(z_M)|$ nodes.

Adapting hierarchical softmax to our model is challenging since our hierachical category tree has multiple layers and applying hierarchical softmax to different layers results in different performances. In our work, since the major computation cost comes from the large number of apps, we adapt hierarchical softmax to the apps.

Specifically, we organize the apps under a subcategory using a binary tree. As shown in Figure 4.2, we represent each app (black nodes) as a leaf node of the binary tree, and the leaf nodes are connected by internal nodes (white nodes). Let $n(i,l)$ be the l-th node on the path from the root of the binary tree to i, and let $L(i)$ be the length of this path, then $n(i,1)$ is the root and $n(i,L(i)) = i$. For each leaf node (*i.e.*, an app), there exists an unique path from the root to the node. Let $n(i,l+1) = \text{left}(n(i,l))$ indicate that $n(i,l+1)$ is the left child node of $n(i,l)$ and we define a sign function as follows:

$$\mathbb{S}\big(n(i,l+1) = \text{left}(n(i,l))\big) := \begin{cases} 1 & n(i,l+1) \text{ on left,} \\ -1 & \text{otherwise.} \end{cases} \tag{4.13}$$

Let $y_{u,n(i,l)}$ be the affinity score between user u and node $n(i,l)$, which is defined as

$$y_{u,n(i,l)} = b_{n(i,l)} + \mathbf{p}_u^\top \mathbf{q}_{n(i,l)}, \tag{4.14}$$

where $\mathbf{q}_{n(i,l)} \in \mathbb{R}^K$ is the latent factor vector and $b_{n(i,l)}$ is the bias term for node $n(i,l)$. Intuitively, at each inner node $n(i,l)$ in the hierarchical softmax binary tree, we assign the probability of moving left as

$$\Pr\big(u, n(i,l+1) = \text{left}(n(i,l))\big) = \sigma\big(b_{n(i,l)} + \mathbf{p}_u^\top \mathbf{q}_{n(i,l)}\big), \tag{4.15}$$

where $\sigma(x)$ is a sigmoid function defined as follows:

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{4.16}$$

Accordingly, the probability of moving right is

$$\begin{aligned}\Pr\big(u, n(i,l+1) \neq \mathrm{left}(n(i,l))\big) &= 1 - \sigma\big(b_{n(i,l)} + \mathbf{p}_u^\top \mathbf{q}_{n(i,l)}\big) \\ &= \sigma\Big(- \big(b_{n(i,l)} + \mathbf{p}_u^\top \mathbf{q}_{n(i,l)}\big)\Big).\end{aligned} \tag{4.17}$$

Combing Equation (4.15) and Equation (4.17), we can derive from the probability of moving from node $n(i,l)$ to node $n(i,l+1)$ as

$$\Pr\big(u, n(i,l) \rightarrow n(i,l+1)\big) = \sigma\Big(\mathbb{S}\big(n(i,l+1) = \mathrm{left}(n(i,l))\big) \cdot \big(b_{n(i,l)} + \mathbf{p}_u^\top \mathbf{q}_{n(i,l)}\big)\Big). \tag{4.18}$$

As a result, by using the path $n(i,1) \rightarrow n(i,2) \cdots \rightarrow n(i,L(i))$ in the defined hierarchical softmax binary tree, we approximate the probability $\Pr(i|u,z_M)$ as follows:

$$\begin{aligned}\Pr(i|u,z_M) &= \prod_{l=1}^{L(i)-1} \Pr(u, n(i,l) \rightarrow n(i,l+1)) \\ &= \prod_{l=1}^{L(i)-1} \sigma\Big(\mathbb{S}\big(n(i,l+1) = \mathrm{left}(n(i,l))\big) \cdot \big(b_{n(i,l)} + \mathbf{p}_u^\top \mathbf{q}_{n(i,l)}\big)\Big)\end{aligned} \tag{4.19}$$

Note that, instead of computing the affinity scores for all the apps under subcategory z_M to get the probability distribution $\Pr(i|u,z_M)$ as defined in Equation (4.12), we only need to compute $L(i)-1$ times in the order of $\log|c(z_M)|$. Also hierarchical softmax does not increase the number of parameters to be estimated. Instead of estimating parameters of $|c(z_M)|$ apps, we only need to estimate the parameters for $|c(z_M)|-1$ internal nodes.

Comments: The binary tree built for hierarchical softmax is meant for computation efficiency purpose, which is different form the category tree Γ used for structural choice modeling.

4.4.2 Parameter Learning

After building the hierarchical softmax binary tree for each most outside subcategory node z_M in the category hierarchy, the unique structural path for user u to choose app i is extended as

$$\text{path}_i = z_0 \to z_1 \cdots \to z_M \to n(i,1) \cdots \to n(i, L(i)).$$

We rewrite the log likelihood $\ell(\Theta)$ and get the following objective function

$$\begin{aligned}
\mathcal{O} =& \sum_{u=1}^{U} \sum_{i \in \mathcal{D}_u} \sum_{l=1}^{L(i)-1} \ln \Pr(u, n(i,l) \to n(i, l+1)) \\
&+ \sum_{u=1}^{U} \sum_{i \in \mathcal{D}_u} \sum_{m=1}^{M} \ln \Pr(z_m | u, z_{m-1}) + \sum_{\substack{m=1 \\ \forall z \in \Gamma}}^{M} \ln \Pr(\mathbf{q}_{z_m} | \mathbf{q}_{z_{m-1}}, \sigma^2 \mathbf{I}) \\
=& \sum_{u=1}^{U} \sum_{i \in \mathcal{D}_u} \sum_{l=1}^{L(i)-1} \ln \sigma \Big(\mathbb{S}\big(n(i, l+1) = \text{left}(n(i,l))\big) \cdot y_{u, n(i,l)} \Big) \\
&+ \sum_{u=1}^{U} \sum_{i \in \mathcal{D}_u} \sum_{m=1}^{M} \ln \frac{\exp(b_{z_m} + \mathbf{p}_u^\top \mathbf{q}_{z_m})}{\sum_{z' \in c(z_{m-1})} \exp(b_{z'} + \mathbf{p}_u^\top \mathbf{q}_{z'})} \\
&+ \sum_{\substack{m=1 \\ \forall z \in \Gamma}}^{M} \ln \mathcal{N}(\mathbf{q}_{z_m} | \mathbf{q}_{z_{m-1}}, \sigma^2 \mathbf{I})
\end{aligned}$$

Note that here we have an updated set of parameters to estimate, namely, $\Theta = \{\mathbf{p}_u, \mathbf{q}_z, \mathbf{q}_{n(i,l)}, b_z, b_{n(i,l)}\}$. Instead of estimating $\mathbf{p}_i$ and b_i for $i \in \mathcal{I}$, we estimate that of internal nodes $n(i,l)$ in the hierarchical softmax binary trees.

We use stochastic gradient ascent method to update the latent factor variables. Stochastic gradient ascent (descent) has been widely used for many machine learning tasks (Bottou, 2010b). The main process involves randomly scanning training instances and iteratively updating parameters. In each iteration, we randomly sample a user-app adoption instance $\langle u, i, \text{path}_i \rangle$, and we maximize $\mathcal{O}(\Theta)$ using the following update rule for Θ:

$$\Theta = \Theta + \epsilon \cdot \frac{\partial \mathcal{O}(\Theta)}{\partial \Theta}, \tag{4.20}$$

where ϵ is a learning rate.

Specifically, given a user-app adoption instance $\langle u, i, \text{path}_i \rangle$, the gradient with respect to $\mathbf{p}_u$ is

$$\begin{aligned} \frac{\partial \mathcal{O}}{\partial \mathbf{p}_u} &= \sum_{l=1}^{L(i)-1} \frac{\partial \ln \Pr(u, n(i,l) \rightarrow n(i,l+1))}{\partial \mathbf{p}_u} + \sum_{m=1}^{M} \frac{\partial \ln \Pr(z_m | u, z_{m-1})}{\partial \mathbf{p}_u} \\ &= \sum_{l=1}^{L(i)-1} \Big(\mathbf{1}_{l+1} - \sigma\big(y_{u,n(i,l)}\big) \Big) \cdot \mathbf{q}_{n(i,l)} \\ &\quad + \sum_{m=1}^{M} \left(\mathbf{q}_{z_m} - \frac{\sum_{z' \in c(z_{m-1})} \exp(b_{z'} + \mathbf{p}_u^\top \mathbf{q}_{z'}) \cdot \mathbf{q}_{z'}}{\sum_{z' \in c(z_{m-1})} \exp(b_{z'} + \mathbf{p}_u^\top \mathbf{q}_{z'})} \right) \end{aligned} \tag{4.21}$$

Here $\mathbf{1}_{l+1}$ is an indicator function defined as

$$\mathbf{1}_{l+1} := \begin{cases} 1 & \text{if } n(i,l+1) = \text{left}(n(i,l)), \\ 0 & \text{otherwise.} \end{cases} \tag{4.22}$$

Before moving to the internal nodes, let us define another indicator function $\mathbf{1}_{z \in \text{path}_i}$ which is defined as

$$\mathbf{1}_{z \in \text{Path}_i} := \begin{cases} 1 & \text{if } z \text{ is in } \text{path}_i, \\ 0 & \text{other siblings nodes.} \end{cases} \tag{4.23}$$

Then, for each internal node $z \in \text{path}_i$ and its siblings, we have the gradient with respect to $\mathbf{q}_z$ as

$$\begin{aligned} \frac{\partial \mathcal{O}}{\partial \mathbf{q}_z} &= \sum_{l=1}^{L(z)} \frac{\partial \ln \Pr(z | u, \pi(z))}{\partial \mathbf{q}_z} + \sum_{\substack{m=1 \\ \forall z \in \Gamma}}^{M} \frac{\partial \ln \Pr(\mathbf{q}_{z_m} | \mathbf{q}_{z_{m-1}}, \sigma^2 \mathbf{I})}{\partial \mathbf{q}_z} \\ &= \mathbf{1}_{z \in \text{path}_i} \cdot \mathbf{p}_u - \frac{\exp(b_z + \mathbf{p}_u^\top \mathbf{q}_z) \cdot \mathbf{p}_u}{\sum_{z' \in c(z_{m-1})} \exp(b_{z'} + \mathbf{p}_u^\top \mathbf{q}_{z'})} \\ &\quad - \frac{\mathbf{q}_z - \mathbf{q}_{\pi(z)}}{\sigma^2} - \frac{\sum_{z' \in c(z)} (\mathbf{q}_z - \mathbf{q}_{z'})}{\sigma^2}. \end{aligned} \tag{4.24}$$

Moreover, we have the gradient with respect to bias b_z as

$$\frac{\partial \mathcal{O}}{\partial b_z} = \mathbf{1}_{z \in \mathrm{Path}_i} - \frac{\exp(b_z + \mathbf{p}_u^\top \mathbf{q}_z)}{\sum_{z' \in c(z_{m-1})} \exp(b_{z'} + \mathbf{p}_u^\top \mathbf{q}_{z'})} \tag{4.25}$$

Finally, for each node level $l = \{1, 2, ..., L(i)-1\}$ in the hierarchical softmax binary tree, we have the gradient with respect to $\mathbf{q}_{n(i,l)}$ and $b_{n(i,l)}$ as

$$\frac{\partial \mathcal{O}}{\partial \mathbf{q}_{n(i,l)}} = \Big(\mathbf{1}_{l+1} - \sigma\big(y_{u,n(i,l)}\big)\Big) \cdot \mathbf{p}_u \tag{4.26}$$

$$\frac{\partial \mathcal{O}}{\partial b_{n(i,l)}} = \mathbf{1}_{l+1} - \sigma\big(y_{u,n(i,l)}\big), \tag{4.27}$$

where $\mathbf{1}_{l+1}$ is the indicator function defined in Equation (4.22). With gradients with respect to $\Theta = \{\mathbf{p}_u, \mathbf{q}_z, \mathbf{q}_{n(i,l)}, b_z, b_{n(i,l)}\}$ being derived, we update Θ using stochastic gradient ascent rule $\Theta = \Theta + \epsilon \cdot \frac{\partial \mathcal{O}(\Theta)}{\partial \Theta}$. We summarize the parameter estimation procedure in Algorithm 2.

4.4.3 Complexity Analysis

Note that in each iteration our structural user choice model has a linear time complexity $O\bigg(\Big(\sum_{m=1}^{M} |L_m| + \log |c(z_M)|\Big) \times |\mathcal{D}|\bigg)$, where $|\mathcal{D}|$ is the number of user-app adoption observations in the training dataset, $|L_m|$ is the number of categories or subcategories in the category hierarchy level m, and $\log |c(z_M)|$ is the logarithm of the number of apps under the most outside subcategory z_M, whose children nodes are apps. Therefore, the structural user choice model has the same complexity as the widely used latent factor models, which are usually linear to the number of observations $|\mathcal{D}|$. In most applications, value of $\Big(\sum_{m=1}^{M} |L_m| + \log |c(z_M)|\Big)$ will not be a large number. For example, in our app recommendation application with a dataset collected from Google Play, the worst case of $\Big(\sum_{m=1}^{M} |L_m| + \log |c(z_M)|\Big)$ is around

```
Input: category tree Γ, user app adoption observations D = {(u, i)}, learning
       rate ε.
Output: optimal Θ = {p_u, q_z, q_n(i,l), b_z, b_n(i,l)}
begin
    for each most outside subcategory node z_M do
        build a binary tree for hierarchical softmax
    end
    Initialize Θ
    repeat
        sample a user app adoption instance ⟨u, i, path_i⟩
        // update user latent factor
        p_u ← p_u + ε · ∂O/∂p_u (Equation (4.21))
        // update internal node latent factor
        for each internal node z ∈ path_i and its siblings do
            q_z ← q_z + ε · ∂O(Θ)/∂q_z (Equation (4.24))
            b_z ← b_z + ε · ∂O(Θ)/∂b_z (Equation (4.25))
        end
        // update hierarchical softmax binary tree node latent
           factor
        for for each node level l = {1, ..., L(i) − 1} do
            q_n(i,l) ← q_n(i,l) + ε · ∂O(Θ)/∂q_n(i,l) (Equation (4.26))
            b_n(i,l) ← b_n(i,l) + ε · ∂O(Θ)/∂b_n(i,l) (Equation (4.27))
        end
    until convergence or reach max_iter
    return Θ̂
end
```

Algorithm 2: Structural User Choice Model Estimation

70.

4.5 Experiments

This section presents an empirical evaluation of the performances of our model and previous methods. All the experiments are performed on a large-scale real-world app adoption dataset that we collected from Google Play.

4.5.1 Dataset Collection

The Google Play is a centralized marketplace where all apps are organized in a predefined category tree. Apps are organized into 27 categories, and the category *Games* is further divided into 18 subcategories. Also Google Play has both free and paid apps. Users can review (*i.e.*, rate or like) apps on Google Play. A user's review about apps he/she used are publicly available. Once we obtain the Google ID of a user, we can locate all apps the user has reviewed. Therefore, we first obtained a list of Google user IDs from the data set shared from Gong et al. (Gong et al., 2012) and wrote a crawler to collect the list of apps that had been reviewed by these users. For each retrieved app, we crawled its category and subcategory information from Google Play.

We treated a user having adopted an app if the review score, whose value is from one to five, is greater or equal to three. After excluding users who have adopted less than 40 apps to avoid cold start problem, we obtained a dataset with $52,483$ users, $26,426$ apps, and $3,286,156$ review observations. The resulting user-app adoption matrix has a sparsity as high as 99.76% and each user adopts 62.61 apps on average, which is a very small fraction of all the apps. Table 5.4 shows some basic statistics of our dataset.

Since only 11.11% of all the apps in our dataset are paid apps, we do not distinguish between paid and free apps when constructing the hierarchical category tree Γ. The $26,426$ apps are categorized into 25 categories (the categories *Live Wallpaper* and *Widgets* defined by Google Play do not appear in our dataset). Figure 4.3 shows the detailed app distributions in different categories and the subcategories of Games. We observe that game apps take the highest percentage, accounting for 26.85% of all the apps; *Arcade* (20.46%), *Puzzle* (16.50%), and *Casual* (12.35%) are among the top three subcategories in *Games*, accounting for 49.31% of all the game apps.

Table 4.2: Data Description

#users	#apps	#observations	sparsity
$52,483$	$26,426$	$3,286,156$	99.76%

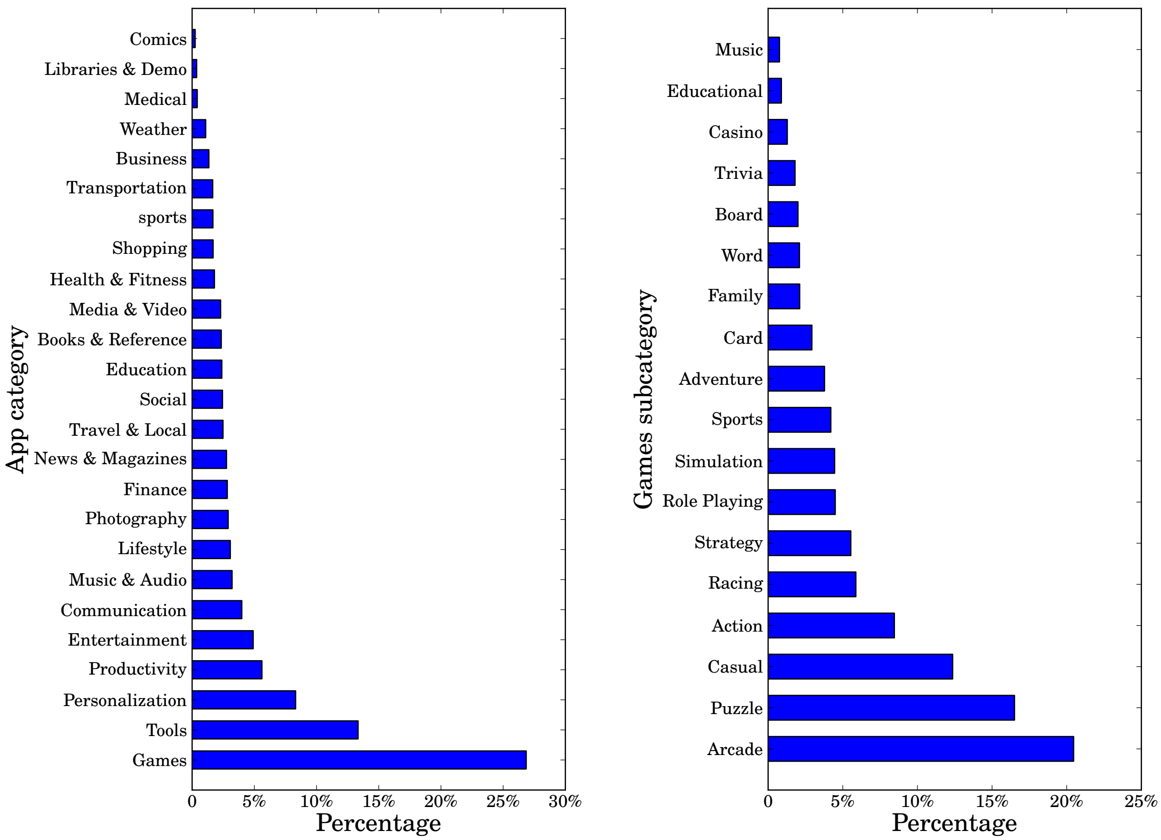
4.5.2 Compared Approaches

We compare our structural user choice model (SUCM) with the following recommendation models.

- Logistic Latent Factor Model (**LLFM**) (Agarwal & Chen, 2009). LLFM was designed to model binary response using a cross-entropy loss function. In our problem, we adapt LLFM to solve the following optimization problem:

$$\arg\min_{\mathbf{P},\mathbf{Q},\mathbf{b}} \sum_{u,i\in\mathcal{D}} \ln\left(1+\exp\left(-\left(\mathbf{p}_u^\top \mathbf{q}_i + b_i\right)\right)\right)+\lambda_U \sum_{u\in\mathcal{U}} ||\mathbf{p}_u||_2^2+\lambda_I \sum_{i\in\mathcal{I}} ||\mathbf{q}_i||_2^2+\lambda_b \sum_{i\in\mathcal{I}} b_i^2$$

  where parameters λ_U, λ_I and λ_b are regularization weights for users, items, and item bias respectively.



(a) Category **(b) Games**

Figure 4.3: App distributions in the category hierarchy. (a) App distributions in app categories, and (b) App distributions in *Games* subcategory.

- Probabilistic Matrix Factorization with negative sampling ($\mathbf{PMF}_{\mathrm{Neg}}$). PMF is a widely used latent factor model for recommendations (Salakhutdinov & Mnih, 2008b). We adapt PMF to our problem, *i.e.*, we solve the following problem:

$$\arg\min_{\mathbf{P},\mathbf{Q},\mathbf{b}} \sum_{u,i\in\mathcal{D}} \Big(y_{ui} - \big(\mathbf{p}_u^\top \mathbf{q}_i + b_i\big)\Big)^2 + \lambda_U \sum_{u\in\mathcal{U}} ||\mathbf{p}_u||^2 + \lambda_I \sum_{i\in\mathcal{I}} ||\mathbf{q}_i||^2 + \lambda_b \sum_{i\in\mathcal{I}} b_i^2$$

  However, we only have positive adopted instance (u, i) which is treated as $y_{ui} = 1$. Thus, for each instance (u, i), we sample a certain number of negative instances $\{(u, j)\}$ and treat them as $y_{uj} = 0$. We denote this modified PMF as $\mathbf{PMF}_{\mathrm{Neg}}$. Note that $\mathbf{PMF}_{\mathrm{Neg}}$ is similar to the sample based one-class collaborative filtering methods (Pan et al., 2008; Y. Hu, Koren, & Volinsky, 2008b).

- **SVDFeature** (T. Chen et al., 2012). SVDFeature is a feature-based latent factor model for recommendation settings with auxiliary information. We use category information as the auxiliary information for SVDFeature.

- **LibFM** (Rendle, 2012, 2010). LibFM is a software implementation for factorization machines (FM) (Rendle, 2010) that models all interactions between variables (*e.g.*user, item and auxiliary information). We also use category information as the auxiliary information and choose the 2-way FM. One major difference between FM and SVDFeature is that SVDFeature only considers the interactions between user features and item features, while FM models all the interactions among all the available information.

- Bayesian Personalized Ranking (**BPR**) (Rendle, Freudenthaler, Gantner, &

Schmidt-Thieme, 2009). BPR was first proposed to model personalized ranking with implicit feedback by treating observed user-item pairs as positive instances and sampling some of the unseen user-item pairs as negative instances. Given the preference triples $\mathcal{D} = \{(u,i,j)|i \succ j\}$, where $i \succ j$ indicates user u prefers item i than item i, BPR aims at maximizing the following optimization criterion,

$$\begin{aligned}&\arg\max_{\mathbf{P},\mathbf{Q},\mathbf{b}} \left\{ \ln \prod_{(u,i,j)\in\mathcal{D}} \Pr((u,i,j)|i \succ_u j)\Pr(\Theta) \right\} \\ &= \arg\max_{\mathbf{P},\mathbf{Q},\mathbf{b}} \left\{ \ln \prod_{(u,i,j)\in\mathcal{D}} \sigma(y_{ui} - y_{uj}|\Theta)\Pr(\Theta) \right\}\end{aligned}$$

where $\sigma(\cdot)$ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, and $\Pr(\Theta)$ are Gaussian priors for the parameters.

- Collaborative Competitive Filtering (**CCF**) (Yang, Long, Smola, Zha, & Zheng, 2011): Given an offer set, CCF models user-item choice behavior by encoding a local competition effect to improve recommendation performances. For each instance (u,i), we sample a certain number of negative instances to formulate the offer sets $\{(u,i,\mathcal{A}(i))$ as described in (Yang et al., 2011). Given collections of choice decision making records $\mathcal{D} = \{(u,i,\mathcal{A}(i))\}$, CCF estimates the latent factors and the item bias terms by solving following optimization problem

$$\begin{aligned}\arg\min_{\mathbf{P},\mathbf{Q},\mathbf{b}} \Bigg\{ &\sum_{(u,i,\mathcal{A}(i))\in\mathcal{D}} \ln \left[\sum_{j\in\mathcal{A}(i)} \exp\left(\mathbf{p}_u^\top \mathbf{q}_j + b_j\right) \right] - \left(\mathbf{p}_u^\top \mathbf{q}_i + b_i\right) \\ &+ \lambda_U \sum_{u\in\mathcal{U}} ||\mathbf{p}_u||_2^2 + \lambda_I \sum_{i\in\mathcal{I}} ||\mathbf{q}_i||_2^2 + \lambda_b \sum_{i\in\mathcal{I}} b_i^2 \Bigg\}.\end{aligned}$$

Implementations, training, and testing: All models are implemented with a stochastic gradient ascent/descent optimization method with an annealing procedure to discount learning rate ϵ at the iteration nIter with $\epsilon^{\text{nIter}} = \epsilon\frac{\nu}{\nu+\text{nIter}-1}$ by setting $\nu = 50$. The learning rate ϵ and the regularization weights are set by cross validation. All parameters are initialized by a Gaussian distribution $\mathcal{N}(0, 0.1)$. We randomly sample 80% of adopted apps of each user as the training dataset, and we use the remaining adopted apps for testing.

4.5.3 Evaluation Metrics

In this implicit feedback app recommendation setting, we present each user with N apps that have the highest predicted affinity values but are not adopted by the user in the training phase, and we evaluate different approaches based on which of these apps were actually adopted by the user in the test phase. More specifically, we adopt a variety of widely used metrics to evaluate different approaches. In the following, we elaborate each metric.

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

$$\begin{aligned}\text{Precision@}N &= \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}|}{N} \\ \text{Recall@}N &= \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}|}{|C_{\text{adopted}}|},\end{aligned} \tag{4.28}$$

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

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

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \times \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}. \tag{4.29}$$

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

Mean Average Precision: Average precision (AP) is a ranked precision metric that gives larger credit to correctly recommended apps in higher positions. AP@N is defined as the average of precisions computed at all positions with an adopted app, namely,

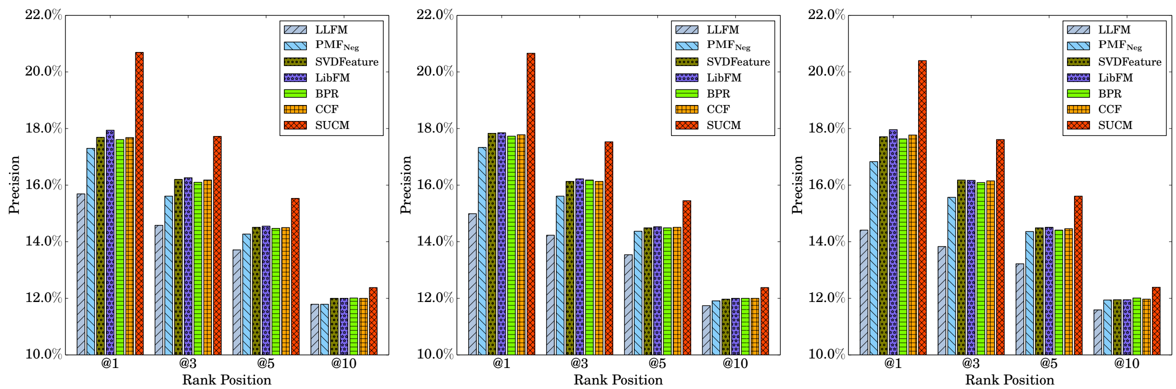
$$\text{AP@}N = \frac{\sum_{k=1}^{N} P(k) \times \text{rel}(k)}{\min\{N, |C_{\text{adopted}}|\}}, \tag{4.30}$$

where $P(k)$ is the precision at cut-off k in the top-N list $C_{N,\text{rec}}$, and $\text{rel}(k)$ is an indicator function equaling 1 if the app at rank k is adopted, otherwise zero. Finally, mean average precision (MAP@N) is defined as the mean of the average precision scores for all users.

Normalized Discounted Cumulative Gain: NDCG is a ranked precision metric that gives larger credit to correctly recommended apps in higher positions. Specifically, the discounted cumulative gain (DCG) given a cut-off N is calculated by

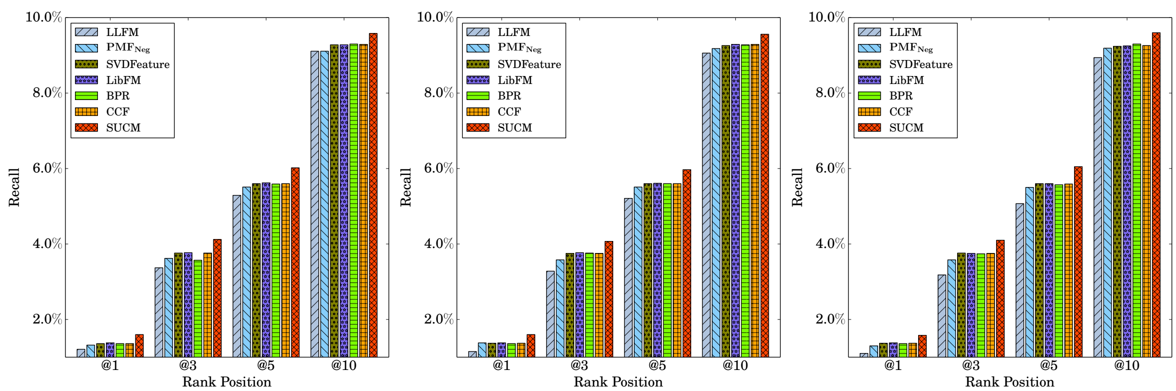
$$\text{DCG}_\text{N} = \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \tag{4.31}$$

where rel_i is is the relevance score, which is binary. Then the NDCG@N is computed as NDCG@N $= \frac{\text{DCG}_\text{N}}{\text{IDCG}_\text{N}}$, where IDCG_N is the DCG_N value of the ideal ranking list. The NDCG for the entire recommender system is computed by averaging the NDCG over all the users.



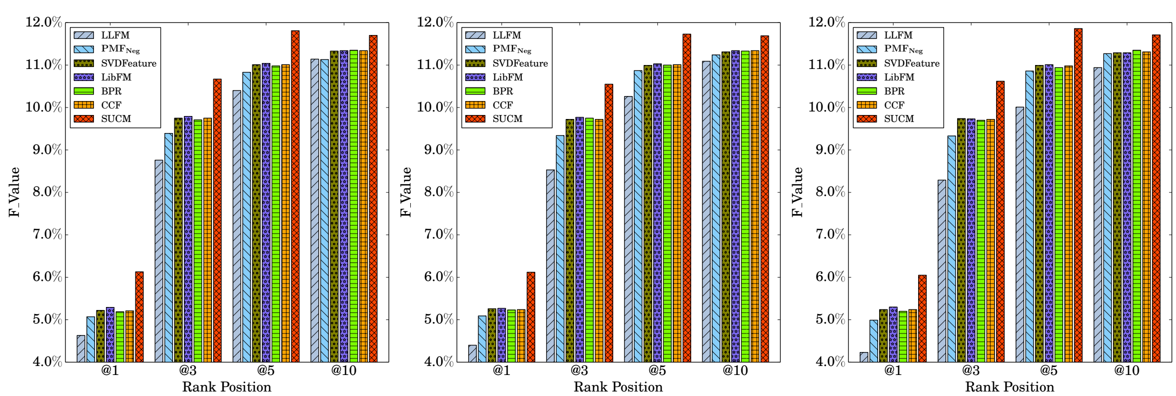
(a) $K = 20$ (b) $K = 30$ (c) $K = 50$

Figure 4.4: Precision @N with different latent dimensions K.



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

Figure 4.5: Recall @N with different latent dimensions K.



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

Figure 4.6: F-measure F_{β}@N with different latent dimensions K ($\beta = 0.5$).

4.5.4 Performance Comparisons

In this subsection, we present the performance comparisons on top-N performances between our proposed SUCM and the baseline methods. We compare various approaches with three latent dimensions $K = 20$, $K = 30$, and $K = 50$, and four top-N values $N = 1, 3, 5, 10$.

Figure 5.4, Figure 5.5, and Figure 5.6 respectively show the precision@N, recall@N, and F_β@N of all compared approaches on our dataset. We find that our approach consistently and substantially outperforms the previous methods for different N and different K. Moreover, we observe that negative sampling based methods $\mathrm{PMF_{Neg}}$ and BPR outperform LLFM which considers only positive instances for all the three considered number of latent dimensions. This is because LLFM polarizes towards the positive response values, and the learned recommendation model would predict positive for almost all unseen items and yield poor ranking performances. However, $\mathrm{PMF_{Neg}}$ and BPR mitigate the issue of LLFM via sampling unseen items as negative instances. Although $\mathrm{PMF_{Neg}}$ and BPR achieve close performances for top-1 recommendations, BPR works slightly better than $\mathrm{PMF_{Neg}}$ for top-3, top-5, and top-10 recommendations. Moreover, CCF further slightly outperforms BPR in most cases. This is because CCF captures the local competition by a softmax of the chosen item over the offer set. Our proposed SUCM further improves upon CCF with significant margins for all the three evaluation metrics. For example, SUCM improves upon CCF with around 3% in terms of top-1 precision.

Besides, previous work (T. Chen et al., 2012; Rendle, 2012, 2010) showed that latent factor based recommendation could be improved by incorporating auxiliary information such as item features and context information. However, we observe that, by treating category information as auxiliary information, these methods (*e.g*SVDFeature and LibFM) can only gain marginal improvements in terms of top N recommendation performances. Compared with counterpart method PMF_{Neg} without auxiliary information, SVDFeature can only gain around 0.4% improvement and LibFM can only gain around 0.6% improvement in terms of top-1 precision respectively. We argue that category information, treated as auxiliary feature, is not fine-grained enough to discriminate user preferences. Quite differently, SUCM leverage the hierarchical structure of category information to better profile user interest preferences.

Precision, recall, and F-measure do not consider the ranking positions of correctly recommended apps. So we further adopt MAP and NDCG to provide more fine-grained understanding of these recommendation approaches. Intuitively, MAP and NDCG give larger credits to correctly recommended apps that are in higher ranking positions. Table 4.3 and Table 4.4 respectively show the MAP@N and NDCG@N of all compared approaches. Again, we observe consistent and substantial improvements of our SUCM upon previous methods.

Summary: Through extensive evaluations, we found that our method SUCM consistently and substantially outperforms previous methods for a variety of evaluation metrics. We argue that SUCM achieves this performance gain by learning fine-grained user interest preferences via leveraging the hierarchical category tree of apps and capturing the competitions between apps.

Table 4.3: MAP@N with different latent dimensions K.

K	MAP	LLFM	$\mathrm{PMF}_{\mathrm{Neg}}$	SVDFeature	LibFM	BPR	CCF	SUCM
20	@1	15.69%	17.30%	17.69%	17.94%	17.61%	17.68%	**20.69%**
	@3	10.38%	11.29%	11.72%	11.82%	11.64%	11.69%	**13.40%**
	@5	8.17%	8.73%	9.00%	9.07%	8.95%	8.99%	**10.11%**
	@10	5.49%	5.69%	5.85%	5.88%	5.84%	5.85%	**6.40%**
30	@1	14.99%	17.33%	17.83%	17.85%	17.73%	17.78%	**20.66%**
	@3	10.07%	11.46%	11.71%	11.78%	11.72%	11.70%	**13.23%**
	@5	7.97%	8.89%	9.01%	9.05%	9.00%	9.00%	**9.97%**
	@10	5.38%	5.80%	5.85%	5.87%	5.86%	5.86%	**6.33%**
50	@1	14.41%	16.83%	17.71%	17.96%	17.63%	17.77%	**20.40%**
	@3	9.71%	11.23%	11.74%	11.77%	11.67%	11.72%	**13.25%**
	@5	7.71%	8.77%	9.01%	9.03%	8.94%	8.99%	**10.05%**
	@10	5.25%	5.76%	5.84%	5.86%	5.84%	5.85%	**6.36%**

4.6 Related Work

Our work is related with two research fields: personalized recommendation methodology and mobile app recommendation.

Recommendation methodology: The most popular model-based approaches are based on the latent factor models (Salakhutdinov & Mnih, 2008b; Koren et al., 2009; Agarwal & Chen, 2009). For the binary implicit feedback setting, models such as LLFM use cross-entropy loss (Agarwal & Chen, 2009), but it is still apt to obtain an estimator that would polarize toward the positive response values, thus leading to limited top N performances. Negative sampling provides an alternative by sampling a certain number of unseen items as negative samples. Then standard latent fac-

Table 4.4: NDCG@N with different latent dimensions K.

K	NDCG	LLFM	PMF_{Neg}	SVDFeature	LibFM	BPR	CCF	SUCM
20	@1	15.69%	17.30%	17.69%	17.94%	17.61%	17.68%	**20.69%**
	@3	14.83%	15.96%	16.54%	16.64%	16.45%	16.52%	**18.40%**
	@5	14.17%	14.94%	15.29%	15.36%	15.23%	15.28%	**16.72%**
	@10	12.69%	12.99%	13.27%	13.30%	13.26%	13.26%	**14.11%**
30	@1	14.99%	17.33%	17.83%	17.85%	17.73%	17.78%	**20.66%**
	@3	14.41%	16.00%	16.52%	16.59%	16.54%	16.52%	**18.24%**
	@5	13.89%	15.04%	15.29%	15.34%	15.28%	15.30%	**16.61%**
	@10	12.53%	13.10%	13.26%	13.29%	13.27%	13.28%	**14.07%**
50	@1	14.41%	16.83%	17.71%	17.96%	17.63%	17.77%	**20.40%**
	@3	13.96%	15.87%	16.54%	16.58%	16.47%	16.52%	**18.23%**
	@5	13.51%	14.96%	15.28%	15.33%	15.21%	15.26%	**16.68%**
	@10	12.29%	13.08%	13.24%	13.27%	13.26%	13.25%	**14.07%**

tor models such as probabilistic matrix factorization (PMF) (Salakhutdinov & Mnih, 2008b) can be adopted. Hu *et al* (Y. Hu et al., 2008b) proposed to treat implicit data as indication of positive and negative preference associated with vastly varying confidence levels on the objective function. Pan *et al* (Pan et al., 2008) used a similar strategy by applying weighted low rank approximation. Instead of optimizing point-wise loss function, Bayesian Personalized Ranking (BPR) (Rendle et al., 2009) optimizes a pairwise loss function to preserve the relative order of items for each user.

There are few works that adopt discrete choice models to model user item choices for recommendation (Yang et al., 2011) and for geographic ranking (Kumar, Mahdian, Pang, Tomkins, & Vassilvitskii, 2015). Discrete choice models (Luce, 1959; McFadden, 1973) are built on established theories on consumer preferences and utility, and have been widely used for understanding consumer behavior in different application domains such as travel (Ben-Akiva & Lerman, 1985), transportation (Train, 1978), and brand choice (Guadagni & Little, 1983). Based on discrete choice model, Yang *et al* (Yang et al., 2011) proposed a collaborative competitive filtering (CCF) model to learn user-item choice to improve recommendation performance. Given users' interaction with offer sets, CCF models user choice by a softmax function of the chosen item over the offer set. In this sense, CCF can also be categorized into sampling based method with samples given in the offer set. Though trying to model user choice process, CCF does not consider the structural dependence between items to be chosen for users, thus the choice model in CCF is "flat" rather then structural. We extend the "flat" choice model into structural user choice model to capture fine grained user preferences for mobile app recommendation.

Recently, there are a few works to explore the item hierarchy and side information for recommendation. For instance, Kanagal *et al* (Kanagal et al., 2012) and Ahmed *et al* (Ahmed et al., 2013) proposed to learn user preferences with additional hierarchical item relationship and other side information such as brand and temporal purchase sequence. Instead of using the predefined item hierarchy, some other works also try to learn the item taxonomy (Y. Zhang, Ahmed, Josifovski, & Smola, 2014). However, these works did not consider the competitions among similar items or among similar categories/subcategories. We do not compare our methods with them because they utilized more side information such as item brands and item semantics, but it is an interesting future work for us to extend our framework to incorporate similar side information in the domain of app recommendation. Besides, Ziegler *et al* (Ziegler et al., 2005) utilized taxonomy information to balance and diversify personalized recommendation lists. However, our work different from Ziegler *et al* (Ziegler et al., 2005) in both the purpose and the way of utilizing taxonomy information.

Mobile app recommendation: Recently, app recommendation has drawn an increasing number of attentions. Different from other domains such as movies (R. M. Bell & Koren, 2007), point-of-interests (B. Liu et al., 2013; B. Liu, Xiong, Papadimitriou, Fu, & Yao, 2015a), and musics (Aizenberg, Koren, & Somekh, 2012), app recommendation has its own characteristics. Yin *et al* (P. Yin et al., 2013) considered a trade-off between satisfaction and temptation for app recommendation with a special focus on the case that a user would like to replace an old app with a new one. Similarly, (Lin, Sugiyama, Kan, & Chua, 2014; Lin, 2014) considered app versions to improve app recommendation by incorporating features distilled from version descriptions. Karatzoglou *et al*(Karatzoglou et al., 2012) provided a context-aware recommendation using tensor factorization by including context information such as location, moving status, and time. Woerndl *et al*(Woerndl et al., 2007) applied a hybrid method for context-aware app recommendation. To address the cold-start problem for app recommendation, (Lin, Sugiyama, Kan, & Chua, 2013; Lin, 2014) proposed to leverage side information from Twitter. Specifically, information of followers of an app's official Twitter account is collected and utilized to model the app, providing an estimation about which users may like the app. Davidsson *et al*(Davidsson & Moritz, 2011) presented a context-based recommender prototype for cold-start user users. Zhu *et al.* (Zhu et al., 2014) proposed a mobile app ranking system by considering both the app's popularity and security risks. More recently, Liu *et al* (B. Liu, Kong, et al., 2015) studied personalized app recommendation by reconciling user functionality preferences and user privacy preferences. Baeza-Yates *et al*(Baeza-Yates et al., 2015) proposed a method to predict which app a user is going to use by leveraging spatio-temporal context features, and Park *et al*(Park et al., 2015) proposed a method to improve the accuracy of mobile app retrieval by jointly modeling app descriptions and user reviews using topic model. However, these works are orthogonal to ours because they use other auxiliary information such as app versions, app satisfaction and temptation, and app privacy, while our work focuses on leveraging app taxonomy to model structural user choices among competing apps.

4.7 Summary

In this chapter, we proposed a novel structural user choice model to learn fine-grained user preference via leveraging the tree hierarchy of apps and capturing competitions between apps for app recommendation. Specifically, given all apps in a centralized mobile app market organized as a category tree, we represented the structural user choice as a unique choice path, starting from the root till the user makes app adoption decision, over the category hierarchy. Then we captured the structural choice procedure by cascading user preferences over the choice path through a novel probabilistic model. We also designed an efficient learning algorithm to estimate the model parameters. Moreover, we collected a real-world large-scale user-app adoption dataset from Google Play and used it to evaluate our method and various previous methods. Our results demonstrated that our method achieved consistent and substantial performance improvements over previous methods. A few interesting future directions include exploring more semantic information in the hierarchical category tree to construct a more fine-grained model to capture user app choice process and applying our model to other domains where items are hierarchically categorized.

CHAPTER 5
PRIVACY AWARE RECOMMENDATION FOR MOBILE APPS

5.1 Introduction

Mobile devices are becoming more and more popular in the past few years. For instance, it was reported that the smartphone market was bigger than the PC market in 2011 for the first time in history (The Smartphone Market is Bigger Than the PC Market, 2011). Thereafter, the smartphone market has continued to increase dramatically, *e.g.*, the smartphones shipped in the third quarter of 2013 increased 44% year-on-year (Smartphone Sales in the Third Quarter of 2013, 2013). One of the reasons lies in the fact that users are able to augment the mobile devices' functions via taking advantage of various feature-rich third-party *applications* (or Apps for brevity), which can be easily obtained from centralized markets such as Google Play and App Store. However, the number of Apps has recently increased dramatically, which makes it hard for a user to locate relevant Apps. For instance, as of July 2013, Google Play had over 1 million Apps with over 50 billion cumulative downloads, and the number of Apps has reached over 1.2 million in June 2014 (Google Play Statistics, 2015); as the beginning of June 2014, App Store had 1.2 million Apps and a cumulative of 75 billion downloads (App Store Statistics, 2015). Therefore, it is urgent to develop effective *personalized* App recommendation systems.

Conventional recommender systems (Adomavicius & Tuzhilin, 2005; Koren et al., 2009; Salakhutdinov & Mnih, 2008b; Lee & Seung, 2000; Canny, 2004; Gopalan et al., 2013) essentially aim to learn the *interest* of each user and the *functionality* of each item (*e.g.*, an App in our problem), given the list of items used or rated by each user. Then, an item is recommended to a user if the item's functionality matches the user's interest. For instance, matrix-factorization-based approaches (Koren et al., 2009; Salakhutdinov & Mnih, 2008b) model a user's interest as a latent vector and an item's functionality as another latent vector; and an item is recommended to a user if the item's functionality vector is close to the user's interest vector in the latent space. Such *interest-functionality* driven recommendation systems have been successfully used to recommend products in e-commence (*e.g.*, Amazon) (Linden, Smith, & York, 2003), movies (*e.g.*, Netflix) (R. M. Bell & Koren, 2007), musics (Aizenberg et al., 2012), point-of-interests (Ye et al., n.d.; B. Liu et al., 2013), and used for link prediction and attribute inference (Gong et al., 2014).

However, these approaches are not appropriate for App recommendations. Specifically, unlike items such as music, movies, and point-of-interests, Apps could have privileges to access the user's personal information such as locations, contacts, and messages. Moreover, users might have different *privacy preferences*, *e.g.*, user A tends to not share contacts with the App while user B tends to not allow the App to access her/his locations. Although an App's functionality may matches a user's interest well, the user could still choose to not install it or dislike it if it does not respect the user's privacy preference. Indeed, according to a recent report (Boyles, Smith, & Madden, 2012), 54% of surveyed users have decided not to install Apps that want to access their sensitive personal information and 30% of users have uninstalled at least one App after they realized that the App was collecting unexpected personal information. Therefore, whether a user selects/likes an App is a result of the trade-off between two factors:

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

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

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

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

Our key contributions are summarized as follows:

- We provide the *first* systematic study on leveraging both interest-functionality expectation and user privacy preference to provide personalized App recommendations.

- We propose a new model to capture the trade-off between functionality and user privacy preference.

- We crawled a real-world dataset from Google Play, and we use it to comprehensively evaluate our approach and state-of-the-art methods and explore the impact of privacy levels on the performance of our method. We find that our method consistently and substantially outperforms state-of-the-art approaches.

5.2 Problem Formalization

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

Table 5.1: Six dangerous permissions. They manipulate sensitive information *locations*, *contacts*, and *messages*, respectively.

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

5.2.1 Trade-off between Functionality and Privacy

We focus on Android Apps, though our approach is also applicable to other types of Apps. Android system is a permission-based framework. A permission is related to a critical resource (*e.g.*, Internet, contact, and message) on the mobile device, and granting a permission to an App allows the App to either read or write the corresponding resource. Table 5.1 shows some permission examples and their corresponding descriptions. For instance, giving the permission READ_CONTACTS to an App makes it capable to read the user's contact data.

We identify that whether a user adopts an App is a result of the trade-off between the App's functionality and the user's privacy preference. To achieve the functionality desired by the user, the App might need to manipulate the user's certain type of private data through requesting the corresponding sensitive permissions. For instance, Google Map, a navigation App, requires the user's GPS location data and thus needs the ACCESS_FINE_LOCATION permission. Moreover, the App could also request other sensitive permissions intentionally (Shekhar, Dietz, & Wallach, 2012) or unintentionally (Felt, Chin, Hanna, Song, & Wagner, 2011) for non-functionality purposes such as advertisements. For instance, Shekhar *et al.* (Shekhar et al., 2012) found that around 25% of Android Apps access users' location data only for advertisements; Felt *et al.* (Felt et al., 2011) found that around 30% of Android Apps request sensitive permissions that are not used by them at all. A user with low privacy concerns with the requested permissions/resources might sacrifice its privacy for the App's functionality, while a user with high privacy concerns might sacrifice the App's functionality for privacy or might transfer to another App that provides the same functionality but uses less private resources.

5.2.2 User Privacy Levels

Different users could have different definitions on *private resources* and could have different privacy concerns to different operations (*e.g.*, read or write) on the resources. We define three privacy levels, each of which consists of a set of resources and corresponding operations. A user's privacy preference essentially characterizes the concerns for the operations on the private resources in a given privacy level. Figure 5.1 illustrates the hierarchy of the three privacy levels.

- **Level I:** This level considers 10 resources (*e.g.*, contact, message, and location) as private. The 10 resources are listed in the first column of Table 5.2. However, this level does not distinguish the operations that can be applied to the private resources. Thus, this privacy level is represented as a binary vector of the 10 resources. If a user does not concern a certain private resource such as message, the user would accept an App to read, write, or even send messages.

- **Level II:** This level considers the same 10 resources in Level I as private. However, this level explicitly distinguishes different operations that can be applied to the resources. In this level, a user could have a low privacy concerns on reading messages but a high privacy concern on writing messages. This level of privacy can be expressed by the set of Android permissions that are related to the 10 resources. In total, there are 23 such permissions. Level II is more fine-gained than level I, and Table 5.2 described mappings between level I and level II.

- **Level III:** This level considers all critical resources including the 10 resources in the Level I and II and other resources (*e.g.*, Internet and bluetooth) on a mobile device as private, and it also distinguishes different operations. This level is more complete and more fine-grained than the Level II, and it can be expressed as a binary vector of all dangerous Android permissions. In total, we identified 72 such permissions, which are a superset of level II permissions.

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

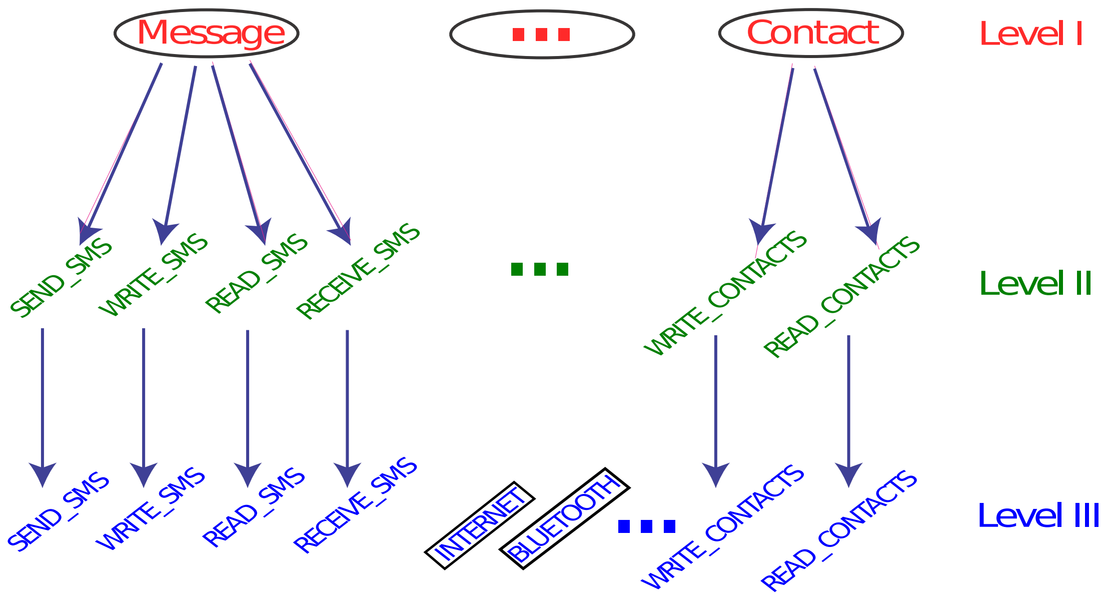


Figure 5.1: Illustration of the three privacy levels. Level I corresponds to *privacy-sensitive resources*; Level II corresponds to *privacy-sensitive permissions* (refer to Table 5.2); Level III is a superset of Level II.

5.2.3 Privacy-respect App Recommendation

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

Suppose we are given a sparse matrix of user-App response records (*e.g.*, ratings or likes) and the set of privacy-sensitive operations/resources of each App at a given

Table 5.2: ***Privacy-sensitive resources*** **(Level I) vs. corresponding** ***privacy-sensitive permissions*** **(Level II).**

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

privacy level, our goal is to recommend most relevant Apps for each user by learning both interest and privacy preference of each user and functionality of each App.

5.3 Proposed Method

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

5.3.1 General Idea

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

Modeling functionality match: Following the latent factor models in standard recommendation systems (Koren et al., 2009; Salakhutdinov & Mnih, 2008b), we model a user i's interest as a user latent vector $\mathbf{u}_i^{\text{interest}} \in \mathbb{R}^K$ and an App j's functionality as an App latent vector $\mathbf{v}_j \in \mathbb{R}^K$, where K is the number of latent dimensions of user interests and App functionalities. More specifically, each element $u_{ik} \in \mathbf{u}_i^{\text{interest}}$ encodes the preference of user i to "preference aspect" k, and each element $v_{ik} \in \mathbf{v}_j$ reflects the aspect affinity of App j to aspect k, where $k = 1, 2, \cdots, K$. Then the functionality match score $g_{\text{func},i,j}$ is modeled as:

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

Modeling privacy respect: We also adopt a latent factor model to describe user privacy preference and App's private information. This latent factor model assumes that it is possible to group users by a relatively small number of privacy profiles.

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

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

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

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

$$g_{i,j} = g_{\text{func},i,j} + \lambda g_{\text{privacy},i,j}, \tag{5.1}$$

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

Table 5.3: Mathematical Notations

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

5.3.2 Model Specifications

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

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

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

$$x_{ij} = \mathbf{u}_i^T \left(\mathbf{v}_j + \lambda \frac{1}{|\Pi_j|} \sum_{s \in \Pi_j} \mathbf{p}_s \right) \tag{5.2}$$

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

To model user profile and App profile, it is practical to formulate the user-App preference score x_{ij} to follow some probability distribution $\Pr(y_{ij}|x_{ij}, \Theta)$, then we can infer the latent factors $\mathbf{u}_i$, $\mathbf{v}_j$, and $\mathbf{p}_s$ through statistical inference methods. One most used probabilistic model, as used in probabilistic matrix factorization (PMF) (Salakhutdinov & Mnih, 2008b), is to assume $\Pr(y_{ij}|x_{ij}, \Theta)$ as a Gaussian distribution.

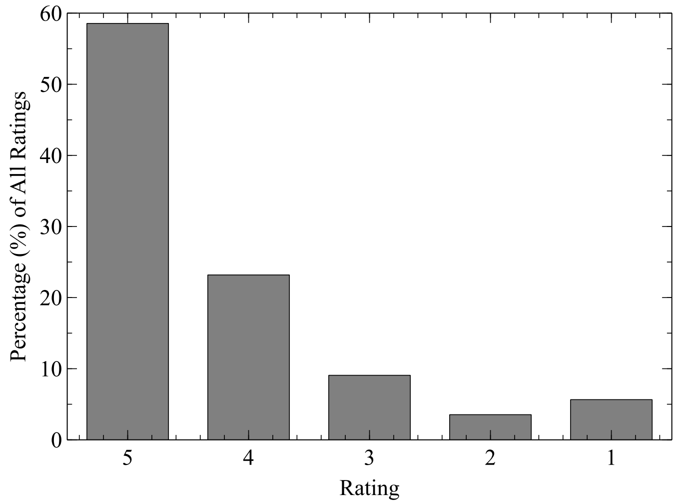


Figure 5.2: Rating distribution.

However, the rating distribution in an App dataset is polarized as shown in Figure 5.2, which indicates that Gaussian distribution would not be a good choice for our problem. Therefore, instead of using a Gaussian distribution, we adopt a Poisson distribution:

$$\begin{aligned} y_{ij} &\sim \text{Poisson}(x_{ij}) \\ \Pr(y_{ij}|x_{ij}) &= (x_{ij})^{y_{ij}} \frac{\exp\{-x_{ij}\}}{y_{ij}!}. \end{aligned} \tag{5.3}$$

As noticed by (Gopalan et al., 2013; Canny, 2004; B. Liu, Xiong, Papadimitriou, Fu, & Yao, 2015b), Poisson distribution is a better choice for modeling discrete user-item responses. Firstly, it better captures real user-item response data. By setting non-negative constrains on latent factors, Poisson latent factor model force response variables to be in a wider range than the rating based response. As a result, it can better capture preference order. Secondly, due to the form of Poisson distribution, only the observed part of user-item matrix needs to be iterated during modeling, which provides advantage for the sparsity of user-item matrix in recommendation problems. Therefore, we model user-App preference as:

$$\Pr(y_{ij}|u_i, v_i, p_s) = (x_{ij})^{y_{ij}} \frac{\exp\{-x_{ij}\}}{y_{ij}!}, \tag{5.4}$$

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

1. For each user i, generate user latent factor:

$$u_{ik} \sim \text{Gamma}(\alpha_U, \beta_U), \tag{5.5}$$

2. For each App j, generate App functionality latent factor:

$$v_{jk} \sim \text{Gamma}(\alpha_V, \beta_V), \tag{5.6}$$

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

$$p_{sk} \sim \text{Gamma}(\alpha_P, \beta_P), \tag{5.7}$$

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

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

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

5.3.3 Model Estimation

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

(MAP) rule, we need to maximize:

$$\begin{aligned}
&\max_{\mathbf{U},\mathbf{V},\mathbf{P}} \Pr(\mathbf{U},\mathbf{V},\mathbf{P}|\mathbf{Y},\Phi) \\
&\propto \max_{\mathbf{U},\mathbf{V},\mathbf{P}} \Pr(\mathbf{Y}|\mathbf{U},\mathbf{V},\mathbf{P})\Pr(\mathbf{U},\mathbf{V},\mathbf{P}|\Phi)
\end{aligned} \tag{5.8}$$

where $\Pr(\mathbf{u}_i, \mathbf{v}_j, \mathbf{p}_s|\alpha_u, \beta_u, \alpha_v, \beta_v, \alpha_s, \beta_s)$ are the prior distributions for $\mathbf{U}$, $\mathbf{V}$, $\mathbf{P}$ generated from Eqs.(5.5, 5.6, 5.7), and $\Pr(y_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j, \mathbf{p}_s)$ can be computed using Eq.(5.4).

Following the likelihood principle, we can determine the optimal solution for $\mathbf{U}, \mathbf{V}, \mathbf{P}$ to Eq.(5.8) by Maximum a Posteriori (MAP) estimation. Specifically, we have:

$$\begin{aligned}
\Pr(\mathbf{Y}|\mathbf{U},\mathbf{V},\mathbf{P}) &= \prod_{i=1}^{M}\prod_{j=1}^{N} (x_{ij})^{y_{ij}} \exp\left\{-x_{ij}\right\}/y_{ij}! \\
\Pr(\mathbf{U}|\alpha_U,\beta_U) &= \prod_{i=1}^{M}\prod_{k=1}^{K} \frac{u_{ik}^{\alpha_U-1}\exp(-u_{ik}/\beta_U)}{\beta_U^{\alpha_U}\Gamma(\alpha_U)} \\
\Pr(\mathbf{V}|\alpha_V,\beta_V) &= \prod_{j=1}^{N}\prod_{k=1}^{K} \frac{v_{jk}^{\alpha_V-1}\exp(-v_{jk}/\beta_V)}{\beta_V^{\alpha_V}\Gamma(\alpha_V)} \\
\Pr(\mathbf{P}|\alpha_P,\beta_P) &= \prod_{s=1}^{S}\prod_{k=1}^{K} \frac{p_{sk}^{\alpha_P-1}\exp(-p_{sk}/\beta_P)}{\beta_P^{\alpha_P}\Gamma(\alpha_P)},
\end{aligned} \tag{5.9}$$

where

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

Then the log-likelihood of Eq.(5.8) is given by

$$\begin{aligned}
\mathcal{L} &= \log \Pr(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{F} | \mathbf{Y}, \Phi) \\
&= \sum_{i=1}^{M} \sum_{k=1}^{K} \Big((\alpha_U - 1) \ln u_{ik} - u_{ik}/\beta_U \Big) \\
&+ \sum_{j=1}^{N} \sum_{k=1}^{K} \Big((\alpha_V - 1) \ln v_{jk} - v_{jk}/\beta_V \Big) \\
&+ \sum_{s=1}^{S} \sum_{k=1}^{K} \Big((\alpha_P - 1) \ln p_{sk} - p_{sk}/\beta_P \Big) \\
&+ \sum_{i=1}^{M} \sum_{j=1}^{N} (y_{ij} \ln x_{ij} - x_{ij}) + \text{const.}
\end{aligned} \tag{5.10}$$

Thus maximization of Eq.(5.8) w.r.t $\mathbf{U}, \mathbf{V}, \mathbf{P}$ is equivalent to maximization of Eq.(5.10). To control the model complexity, we further add a penalty term, then the objective function becomes

$$\mathcal{Q} = \mathcal{L} - \frac{\eta}{2} \left(||\mathbf{U}||_{\mathrm{F}}^2 + ||\mathbf{V}||_{\mathrm{F}}^2 + ||\mathbf{F}||_{\mathrm{F}}^2 \right) \tag{5.11}$$

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

$$\begin{aligned}
\frac{\partial \mathcal{Q}}{\partial u_{ik}} &= \frac{\alpha_U - 1}{u_{ik}} - \frac{1}{\beta_U} - \eta \times u_{ik} \\
&+ \sum_{j=1}^{N} \left(\frac{y_{ij}}{x_{ij}} - 1 \right) \left(v_{jk} + \frac{\lambda}{|\Pi_j|} \sum_{s \in \Pi_j} p_{sk} \right) \\
\frac{\partial \mathcal{Q}}{\partial v_{jk}} &= \frac{\alpha_V - 1}{v_{jk}} - \frac{1}{\beta_V} - \eta \times v_{jk} + \sum_{i=1}^{M} \left(\frac{y_{ij}}{x_{ij}} - 1 \right) u_{ik} \\
\frac{\partial \mathcal{Q}}{\partial p_{sk}} &= \frac{\alpha_P - 1}{p_{sk}} - \frac{1}{\beta_P} - \eta \times p_{sk} \\
&+ \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\frac{y_{ij}}{x_{ij}} - 1 \right) \lambda u_{ik} \frac{\mathbb{I}(s \in \Pi_j)}{|\Pi_j|}
\end{aligned} \tag{5.12}$$

We adopt the ascending gradient method (Bertsekas, 1999) to infer the latent factors. Specifically, parameters θ are updated by the following equation:

$$\theta \leftarrow \theta + \epsilon \times \frac{\partial \mathcal{Q}}{\partial \theta}, \tag{5.13}$$

where θ is an element in $\{\mathbf{U}, \mathbf{V}, \mathbf{P}\}$, $\frac{\partial \mathcal{Q}}{\partial \theta}$ is the derivatives according to Equation (5.12), and ϵ is the learning rate. The algorithms iterate over $\{\mathbf{U}, \mathbf{V}, \mathbf{P}\}$ until one of the following termination conditions is reached (a) the value of objection function $\mathcal{Q}$ in Equation (5.11) keeps stable, or (b) the maximum number of iterations is reached.

5.4 Experiments

5.4.1 Experimental setup

We aim to answer the following two questions:

- **Question 1**: Whether, and to what extent, our privacy-respect App recommendation model improves upon previous recommendation approaches that do not consider user privacy preferences?

- **Question 2**: How privacy levels (as introduced in Section 5.2.2) influence the performance of our approach?

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

5.4.2 Data Collection

We collected our dataset from Google Play. On Google Play, a user's ratings about Apps he/she used are publicly available. Once we obtain the Google ID of a user, we can locate all Apps the user has rated. Therefore, we first obtain a list of Google user IDs from Gong et al. (Gong et al., 2012) and write a crawler to retrieve all rated

Table 5.4: Data Description

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

Apps of these users. Moreover, for each retrieved App, we crawled its permissions from Google Play. Our crawls were performed during June and July of 2014.

We remove unpopular Apps and users with too few rated Apps from our collected dataset to avoid *cold start* problem. Specifically, we first remove Apps with less than 5 users and then exclude users with less than 10 Apps. After this preprocessing step, our dataset has $16,344$ users, $6,157$ Apps, and $263,054$ rating observations. The user App rating matrix has a sparsity as high as 99.74%. Each user rated 16.09 Apps on average, which is a very small fraction of all the Apps. Table 5.4 shows some basic statistics of our preprocessed dataset.

Figure 5.3 shows the top 20 most frequently used permissions and the number of Apps that require those permissions. While permissions such as `INTERNET` might not lead to privacy concerns for most users, some permissions (*e.g.*, `READ_CONTACTS`, `ACCESS_FINE_LOCATION`, and `CALL_PHONE`) could raise serious privacy concerns depending on how they are used by the Apps.

5.4.3 Compared Approaches

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

- Singular Value Decomposition (SVD) (Koren et al., 2009): SVD is a low dimension decomposition based recommendation method.

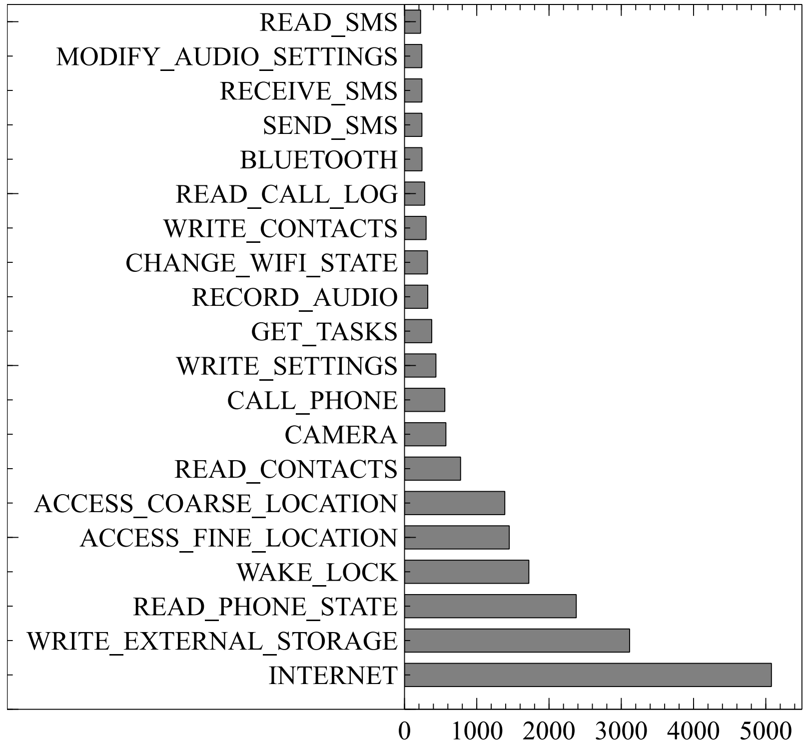


Figure 5.3: The top 20 most frequently used permissions and the number of Apps (out of the $6,157$ Apps in our dataset) that require those permissions.

- Probabilistic Matrix Factorization (PMF) (Salakhutdinov & Mnih, 2008b): PMF extends SVD to a probabilistic framework.

- Non-negative Matrix Factorization (NMF) (Lee & Seung, 2000): Similar to SVD, but NMF requires the latent vectors to be non-negative.

- Poisson Factor Model (Poi-FM) (Gopalan et al., 2013; Canny, 2004): Poisson factor model provides an alternative for latent factor model for different applications, and previous work (Gopalan et al., 2013) shows that Poisson factor model outperforms Gaussian based PMF.

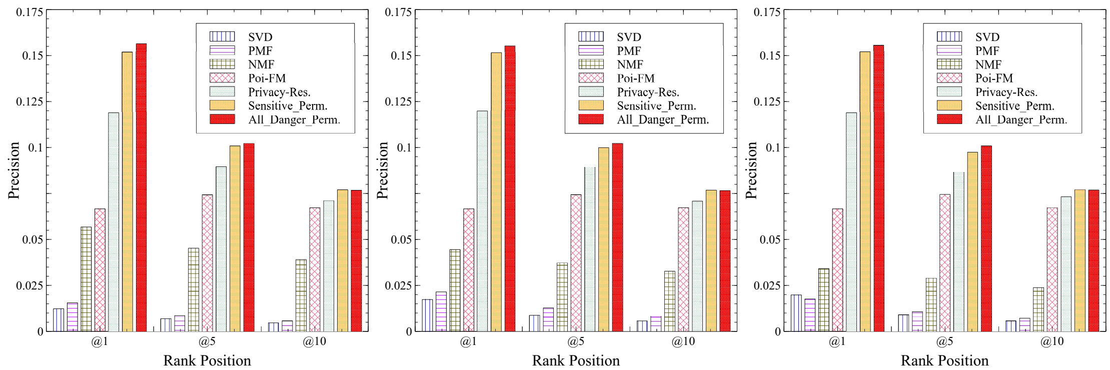
Besides comparing to previous methods that do not consider user privacy preferences, we also investigate how different privacy levels impact the performance of our method. Specifically, we compare the following three variants of our method:

- **Privacy_Res.**: Privacy-respect App recommendation with Level I as the privacy level. Recall that this level considers the 10 resources listed in Table 5.2 as private data and do not distinguish different operations. Thus, each App's permissions are transformed to a 10-dimensional binary vector, which represents the private resources used by the App.

- **Sensitive_Perm.**: Privacy-respect App recommendation with Level II as the privacy level. Level II considers the 23 dangerous permissions that are related to the 10 sensitive resources. Therefore, we have a 23-dimensional binary vector to represent the private resources used by an App.

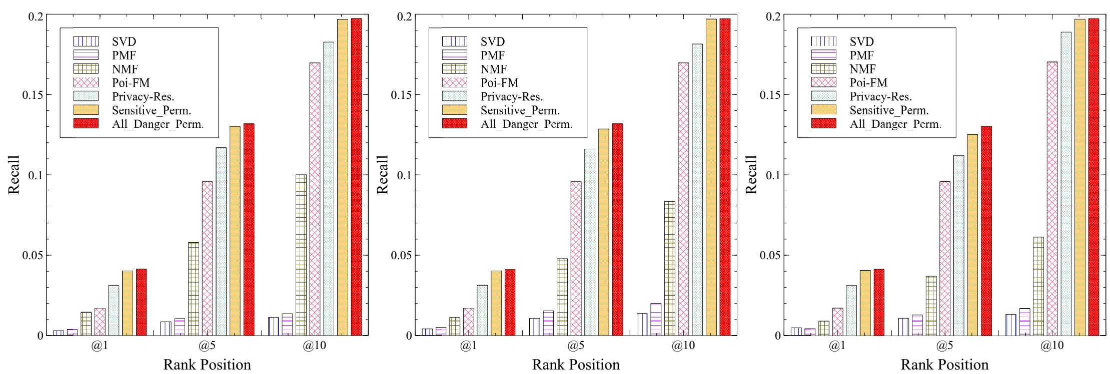
- **All_Danger_Perm.**: Privacy-respect App recommendation with Level III as the privacy level. Level III considers all critical resources (corresponding to 72 dangerour Android permissions) on a mobile device as private. Therefore, we have a 72-dimensional binary vector to represent the private resources used by an App.

Training and testing: We sample 80% of rated Apps of each user uniformly at random as training data, and we use the remaining rated Apps for testing. All the latent factor models are implemented with stochastic gradient ascent/descent optimization method with an annealing procedure to discount learning rate ϵ at iteration nIter with $\epsilon^{\text{nIter}} = \epsilon \frac{\nu}{\nu + \text{nIter} - 1}$ by setting $\nu = 50$. For SVD, PMF, and NMF, learning rates are set as $1e^{-3}$; learning rates for Poisson based methods are empirically set



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

Figure 5.4: Precision @N with different latent dimensions K.



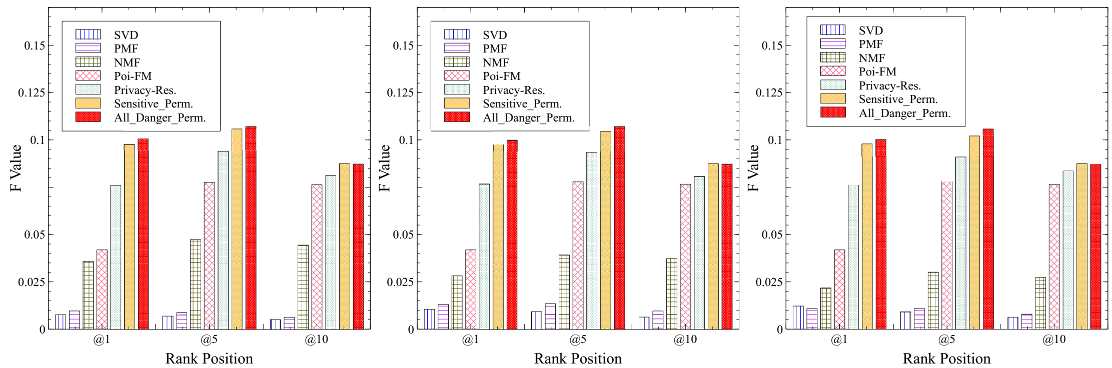
(a) $K = 20$ **(b)** $K = 30$ **(c)** $K = 50$

Figure 5.5: Recall @N with different latent dimensions K.

as $1e^{-4}$. For Poisson based latent factor models including both baseline Poi-FM and our proposed model, we set $\alpha_U = \alpha_V = \alpha_P = 20$ and $\beta_U = \beta_V = \beta_P = 0.5$; penalty weight η is set as $1e^{-5}$; functionality-privacy trade-off weight λ is empirically set as 1.

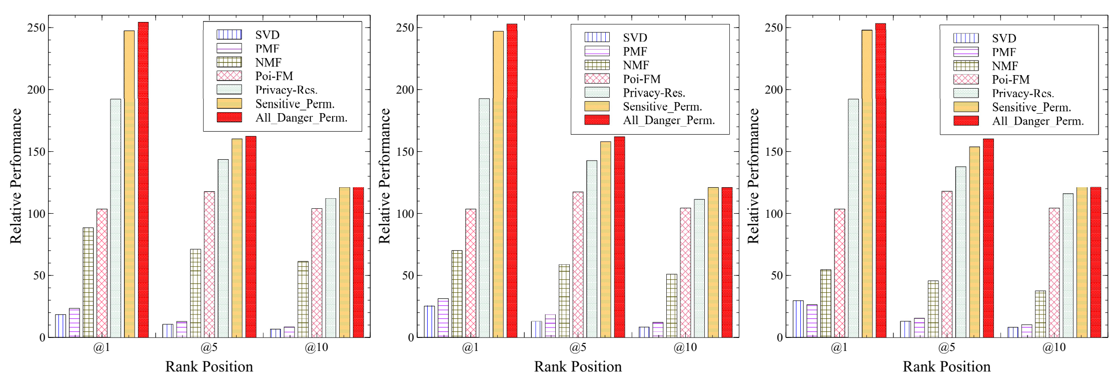
5.4.4 Evaluation Metrics

In App recommendation, we present to the user a list of recommendations, thus we evaluate the models in terms of ranking. Specifically, we present each user with N Apps that have the highest predicted values but are not rated by the user in the



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

Figure 5.6: F_β **@N with different latent dimensions** K **(**$\beta = 0.5$**).**



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

Figure 5.7: Relative performance @N with different latent dimensions K**.**

training phase, and we evaluate different approaches based on which of these Apps were actually adopted by the user in the test phase.

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

$$\begin{aligned} \text{Precision@}N &= \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}|}{N} \\ \text{Recall@}N &= \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}|}{|C_{\text{adopted}}|}, \end{aligned} \tag{5.14}$$

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

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

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \times \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}. \tag{5.15}$$

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

r-Recall and r-Precision: Similar to Yin et al. (P. Yin et al., 2013), we also compare the different models in terms of relative precision and recall. Let C denote the candidate Apps, the precision and recall in a top-N list of a random recommender system are $\frac{|C_{\text{adopted}}|}{|C|}$ and $\frac{N}{|C|}$, respectively. Then, the relative precision and recall (P. Yin et al., 2013) are defined as

$$\text{rPrecision@}N = \frac{\text{Precision@}N}{|C_{\text{adopted}}|/|C|} = \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}| \cdot |C|}{|C_{\text{adopted}}| \cdot N}$$
$$\text{rRecall@}N = \frac{\text{Recall@}N}{N/|C|} = \frac{|C_{N,\text{rec}} \bigcap C_{\text{adopted}}| \cdot |C|}{|C_{\text{adopted}}| \cdot N} \tag{5.16}$$

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

5.4.5 Comparison Results

Figure 5.4, Figure 5.5, Figure 5.6, and Figure 5.7 respectively show the precision@N, recall@N, F_β@N, and relative performance@N of all compared approaches on our dataset, where $N = 1, 5, 10$. For each approach, we explore 3 latent dimension settings, *i.e.*, $K = 20$, $K = 30$, and $K = 50$.

Comparing Our Method with Previous Approaches

We find that our approach consistently outperforms previous methods for different N and different K. Specifically, we observe that NMF outperforms both SVD and PMF, and that Poisson based factor model Poi-FM can further improve upon NMF with all the three considered number of latent dimensions. Moreover, our privacy-respect App recommendation methods with the three privacy levels all further improve upon Poi-FM with significant margins for all the four evaluation metrics. To better demonstrate the improvements, Table 5.5 shows the absolute improvements of our proposed method, with different privacy levels, as compared to the best baseline method Poi-FM when $K = 30$ and $N = 1$. We can see significant improvement margins. While precision and recall may change with different N, F-measure provides a stable com-

Table 5.5: Absolute improvements of our proposed method with different privacy levels as compared to the best baseline method Poi-FM ($K = 30$, $N = 1$).

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

parisons as it is the harmonic mean of precision and recall. In terms of F-measure, we can observe a 3.46% improvement for **Privacy_Res.**, a 5.56% improvement for **Sensitive_Perm.**, and a 5.80% improvement for **All_Danger_perm.**.

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

Impact of Privacy Levels

We find that our method with Level II privacy information (*i.e.*, **Sensitive_Perm.**) can improve upon our method with Level I privacy information (*i.e.*, **Privacy_Res.**) with notable margins. This implies that users treat different operations (*e.g.*, read and write) on the 10 private resources with different privacy concerns. However, **Sensitive_Perm.** and **All_Danger_Perm.** have very close performances consistently for all settings we considered. Figure 5.8 shows a zoom in comparisons of the three levels. Our observations indicate that users probably do not treat resources (*e.g.*, Internet, Bluetooth) other than the 10 resources in the privacy level Level I and Level II as private resources, and thus accessing those resources would not influence whether a user likes/adopts an App.

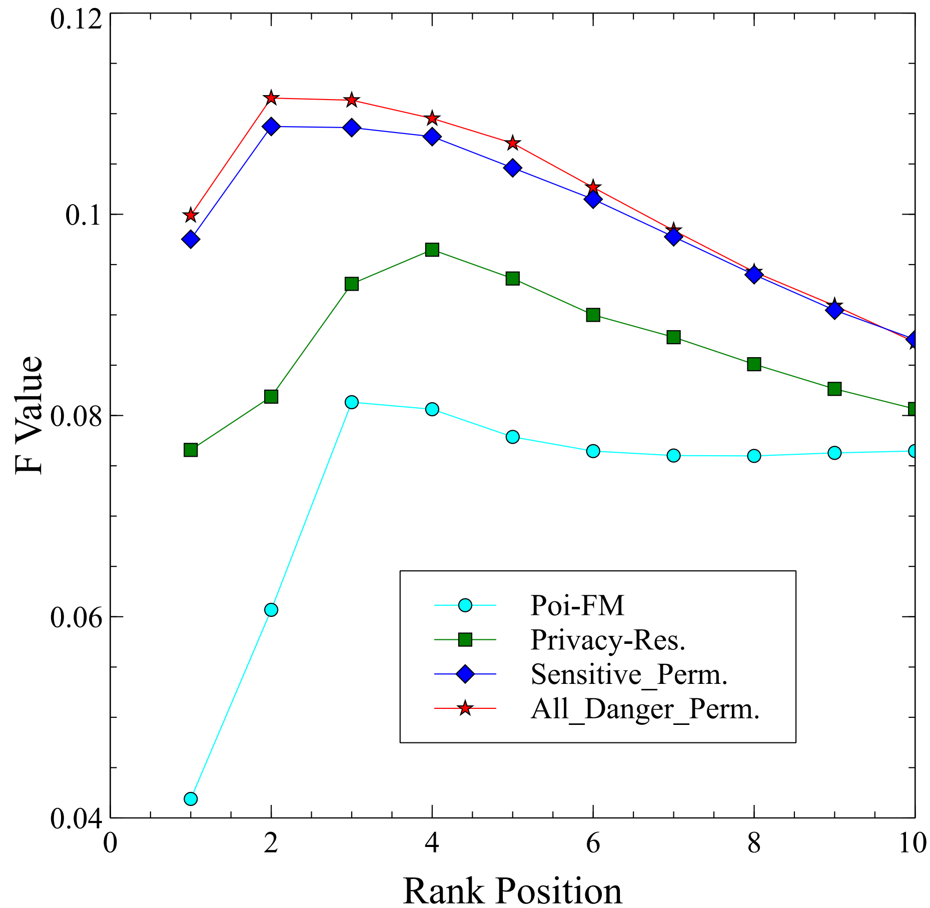


Figure 5.8: Zoom in comparisons between our methods with the three privacy levels (i.e., Privacy_Res., Sensitive_Perm., and All_Danger_perm.) and the best baseline method (i.e., Poi-FM) for $K = 30$**.**

Summary

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

5.5 Related Work

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

Personalized Recommendation: Latent factor models have become popular and been widely used in recommendation systems. These work include matrix factorization (Koren et al., 2009), Probabilistic Matrix Factorization (PMF) (Salakhutdinov & Mnih, 2008b) and its Bayesian version (Salakhutdinov & Mnih, 2008a), and their other variants (Koren, 2008; Agarwal & Chen, 2009; Kong, Zhang, & Ding, 2013). In our case and many cases alike, the recommendation system needs to infer user preferences from user feedbacks. Latent factor models, which are suitable for better capturing preference order are preferred and followed in our approach. One option in designing latent factor model is to set non-negative constrains on latent factors to force response variables to be in a wider range than the rating based response, which is normally limited to a certain range of integers. As a result, non-negative matrix factorization based methods are widely used (Lee & Seung, 2000; S. Zhang et al., 2006; Gu et al., 2010) due to this advantage.

Furthermore, most of these latent factors based studies along this line of research assume that the user response follows Gaussian distribution with expectation from the product of user and item latent factors. However, some recent work (Gopalan et al., 2013; Canny, 2004; B. Liu, Xiong, et al., 2015b) have pointed out that Poisson distribution could be a better choice for modeling user response. Firstly, it better captures real consumption data; secondly, due to the form of Poisson distribution, only the observed part of user-item matrix need to be iterated during modeling. This is a big advantage considering the usually extreme sparsity of user-item matrix in recommendation problems, providing better scalability.

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

Mobile App Ranking and Recommendation: There have been a few previous works on App recommendation, but these works only focused on recommending the most relevant Apps to a user without considering *user privacy preference*. Work from (Karatzoglou et al., 2012) provides a context-aware recommendation using tensor factorization by including context information such as location, moving status and time. To address the cold-start problem for App recommendation, (Lin et al., 2013) proposed to incorporate side information from Twitter. Information of followers of the App's official Twitter account is collected and utilized to model the App, providing an estimation about which users may like the App, even when the App still has no official rating yet. Yin *et al.* (P. Yin et al., 2013) considered a trade-off between satisfaction and temptation for App recommendation with a special focus on the case that a user would like to replace an old App with a new one. An interesting dataset is collected via an App from users, revealing users' process of choosing a new App after comparing it with those already obtained ones. And the satisfaction and temptation of an App is evaluated and used to facilitate the recommendation algorithm.

More recently, Zhu *et al.* (Zhu et al., 2014) proposed a mobile App recommendation (more precisely, *ranking*) system by considering both the App's popularity and security risks. They provide an identical global ranking of Apps to every user, and thus their work is not *personalized* App recommendation.

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

5.6 Summary

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

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

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this dissertation, we identified several unique challenges for recommendation in mobile business environments, and then introduced how we use advanced data mining techniques to address these challenges.

First, we studied the POI recommendation problem in LBSNs by exploiting textual information as well as regional word-of-mouth opinions. We proposed a *topic and location aware* POI recommender system by exploiting rich textual and context information associated with POIs in location based services. The location-aware aggregated LDA recommendation approach allows to profile user interests by performing topic modeling of the users' historical textual information. This provides a way to match the user interests to the POI topic, and thus alleviate the cold start problem in recommendation. Also, the proposed recommendation method can strike a balance between the use of individual information and the use of location-aware word-of-mouth opinions.

Second, many mobile services are location-dependent, which means a users choice can be influenced by location-dependent factors, in particular are the *user mobility* and *geographical influence*. We presented an integrated analysis of the joint effect of multiple factors which influence the decision process of a user choosing a POI and proposed a general framework to learn geographical preferences for POI recommendation in LBSNs. The proposed geographical probabilistic factor analysis framework strategically takes all these factors, which influence the user check-in decision process, into consideration.

Third, we leveraged hierarchy structure such as category hierarchy to capture fine-grained user choice behavior. We then introduce a structural user choice model (SUCM) to learn fine-grained user choice patterns by exploiting hierarchy structure. Moreover, we designed an efficient learning algorithm to estimate the parameters for the SUCM model. Evaluation on an app adoption data demonstrated that our approach can better capture user choice patterns and thus improve recommendation performance.

Finally, we investigated privacy issues for mobile app adoption, in which mobile apps could have privileges to access a user's sensitive resources (e.g., contact, message, and location). We presented the *first* systematic study on incorporating both interest-functionality interactions and users' privacy preferences to perform personalized app recommendations. Moreover, we explored the impact of different levels of privacy information on the performances of our method, which gives us insights on what resources are more likely to be treated as private by users and influence users' behaviors at selecting apps.

Future Work

Accurate modeling of user behavior will be of critical importance for most business domains. In particular, the continuing spread of mobile technology will have a dramatic impact on the way mobile users interact with different kind of services. Following are some future directions to explore: First, modeling spatial-temporal user preference: In our previous work of POI recommendation in LBSN, we mainly exploited geographical influences and user mobility to better model geographical user preferences. However, temporal factor is also critical for user-POI choice selection. Second, location-based advertising: it interesting to explore location-based advertising by considering user preference, user mobility, and geographical influences. Third, it will be interesting to explore more semantic information in the hierarchical category tree to construct a more fine-grained structural choice model to capture user app choice process. Finally, it will be interesting to apply random filed study to investigate users' privacy concerns in mobile app adoptions.

BIBLIOGRAPHY

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

Agarwal, D., & Chen, B.-C. (n.d.). flda: matrix factorization through latent dirichlet allocation. In (pp. 91–100).

Agarwal, D., & Chen, B.-C. (2009). Regression-based latent factor models. In *Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining* (pp. 19–28).

Ahmed, A., Kanagal, B., Pandey, S., Josifovski, V., Pueyo, L. G., & Yuan, J. (2013). Latent factor models with additive and hierarchically-smoothed user preferences. In *Proc of the sixth acm int'l conf. on web search and data mining* (pp. 385–394).

Aizenberg, N., Koren, Y., & Somekh, O. (2012). Build your own music recommender by modeling internet radio streams. In *Proc. of the 21st int'l conf. on world wide web* (pp. 1–10).

Andrieu, C., De Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to mcmc for machine learning. *Machine learning*, *50*(1-2), 5–43.

App Store Statistics. (2015). *http://en.wikipedia.org/wiki/App_Store_(iOS).*

Baeza-Yates, R., Jiang, D., Silvestri, F., & Harrison, B. (2015). Predicting the next app that you are going to use. In *Proceedings of the eighth acm international conference on web search and data mining* (pp. 285–294).

Bell, R., Koren, Y., & Volinsky, C. (2007). Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proc. of the 13th acm sigkdd conf. on knowledge discovery and data mining* (pp. 95–104).

Bell, R. M., & Koren, Y. (2007, December). Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.*, *9*(2), 75–79.

Ben-Akiva, M. E., & Lerman, S. R. (1985). *Discrete choice analysis: theory and application to travel demand* (Vol. 9). MIT press.

Berjani, B., & Strufe, T. (2011). A recommendation system for spots in location-based online social networks. In *4th workshop on social network systems.*

Bertsekas, D. P. (1999). *Nonlinear programming.* Athena Scientific.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003, March). Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*, 993–1022. Retrieved from `http://dl.acm.org/citation.cfm?id=944919.944937`

Bottou, L. (2010a). Large-scale machine learning with stochastic gradient descent. In Y. Lechevallier & G. Saporta (Eds.), *Proceedings of the 19th international conference on computational statistics (compstat'2010)* (pp. 177–187). Retrieved from `http://leon.bottou.org/papers/bottou-2010`

Bottou, L. (2010b). Large-scale machine learning with stochastic gradient descent. In *Proceedings of compstat'2010* (pp. 177–186).

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

Canny, J. (2004). Gap: A factor model for discrete data. In *Proc. of the 27th acm sigir conf. on research and development in information retrieval* (pp. 122–129).

Chen, T., Zhang, W., Lu, Q., Chen, K., Zheng, Z., & Yu, Y. (2012, December). Svdfeature: A toolkit for feature-based collaborative filtering. *J. Mach. Learn. Res.*, *13*(1), 3619–3622.

Chen, Y., Kapralov, M., Pavlov, D., & Canny, J. (2009). Factor modeling for advertisement targeting. In (pp. 324–332).

Cheng, C., Yang, H., King, I., & Lyu, M. R. (2012). Fused matrix factorization with geographical and social influence in location-based social networks. In *Aaai.*

Cheng, Z., Caverlee, J., Kamath, K. Y., & Lee, K. (2011). Toward traffic-driven location-based web search. In (pp. 805–814).

Cheng, Z., Caverlee, J., Lee, K., & Sui, D. Z. (2011). Exploring millions of footprints in location sharing services. In *Icwsm.*

Cho, E., Myers, S. A., & Leskovec, J. (2011). Friendship and mobility: user movement in location-based social networks. In *Proc. of the 17th acm sigkdd int'l conf. on knowledge discovery and data mining* (pp. 1082–1090).

Davidsson, C., & Moritz, S. (2011). Utilizing implicit feedback and context to recommend mobile applications from first use. In *Proceedings of the 2011 workshop on context-awareness in retrieval and recommendation* (pp. 19–22).

Farrahi, K., & Gatica-Perez, D. (2011, January). Discovering routines from large-scale human locations using probabilistic topic models. *ACM Trans. Intell. Syst. Technol.*, *2*(1), 3:1–3:27.

Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011). Android permissions demystified. In *Ccs.*

Ferrari, L., Rosi, A., Mamei, M., & Zambonelli, F. (2011). Extracting urban patterns from location-based social networks. In *3rd acm sigspatial workshop on location-based social networks* (pp. 9–16).

Ge, Y., Liu, Q., Xiong, H., Tuzhilin, A., & Chen, J. (2011). Cost-aware travel tour recommendation. In *Kdd* (p. 983-991).

Ge, Y., Xiong, H., Tuzhilin, A., Xiao, K., Gruteser, M., & Pazzani, M. (2010). An energy-efficient mobile recommender system. In *Kdd* (pp. 899–908).

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

Gong, N. Z., Xu, W., Huang, L., Mittal, P., Stefanov, E., Sekar, V., & Song, D. (2012). Evolution of social-attribute networks: measurements, modeling, and implications using google+. In *Proc of the 2012 acm conf on internet measurement conference* (pp. 131–144).

Google Play Statistics. (2015). *http://en.wikipedia.org/wiki/Google_Play.*

Gopalan, P., Hofman, J. M., & Blei, D. M. (2013). Scalable recommendation with poisson factorization. *CoRR*, *abs/1311.1704*.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *PNAS*, *101*(suppl. 1), 5228-5235.

Gu, Q., Zhou, J., & Ding, C. H. (2010). Collaborative filtering: Weighted non-negative matrix factorization incorporating user and item graphs. In *Sdm* (pp. 199–210).

Guadagni, P. M., & Little, J. D. (1983). A logit model of brand choice calibrated on scanner data. *Marketing science*, *2*(3), 203–238.

Heinrich, G. (2009). *Parameter estimation for text analysis* (Tech. Rep.). Fraunhofer IGD.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004, January). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, *22*(1), 5–53.

Hong, L., Ahmed, A., Gurumurthy, S., Smola, A. J., & Tsioutsiouliklis, K. (2012). Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on world wide web* (pp. 769–778).

Hu, B., & Ester, M. (2013). Spatial topic modeling in online social media for location recommendation. In *Proc. of the 7th acm conf. on recommender systems* (p. 25-32).

Hu, Y., Koren, Y., & Volinsky, C. (2008a). Collaborative filtering for implicit feedback datasets. In (pp. 263–272).

Hu, Y., Koren, Y., & Volinsky, C. (2008b). Collaborative filtering for implicit feedback datasets. In *Data mining, 2008. icdm'08. eighth ieee int'l conf on* (pp. 263–272).

Kanagal, B., Ahmed, A., Pandey, S., Josifovski, V., Yuan, J., & Garcia-Pueyo, L. (2012, June). Supercharging recommender systems using taxonomies for learning user purchase behavior. *Proc. VLDB Endow.*, *5*(10), 956–967.

Karatzoglou, A., Baltrunas, L., Church, K., & Böhmer, M. (2012). Climbing the app wall: Enabling mobile app discovery through context-aware recommendations. In *Proc of the 21st acm int'l conf. on information and knowledge management* (pp. 2527–2530).

Kong, D., Zhang, M., & Ding, C. H. Q. (2013). Minimal shrinkage for noisy data recovery using schatten-p norm objective. In *ECML/PKDD 2013* (pp. 177–193).

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In (pp. 426–434).

Koren, Y. (2010). Collaborative filtering with temporal dynamics. *Commun. ACM*, *53*(4), 89–97.

Koren, Y., Bell, R., & Volinsky, C. (2009, August). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Kumar, R., Mahdian, M., Pang, B., Tomkins, A., & Vassilvitskii, S. (2015). Driven by food: Modeling geographic choice. In *Proceedings of the eighth acm international conference on web search and data mining* (pp. 213–222).

Kurashima, T., Iwata, T., Hoshide, T., Takaya, N., & Fujimura, K. (2013). Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *Wsdm* (p. 375-384).

Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *NIPS* (p. 556-562).

Liang Xiong, T.-K. H. J. S. J. G. C., Xi Chen. (2010). Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of siam data mining.*

Lin, J. (2014). *Mobile app recommendation.* Unpublished doctoral dissertation, National University of Singapore.

Lin, J., Sugiyama, K., Kan, M.-Y., & Chua, T.-S. (2013). Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *Proc. of the 36th int'l acm sigir conf. on research and development in information retrieval* (pp. 283–292).

Lin, J., Sugiyama, K., Kan, M.-Y., & Chua, T.-S. (2014). New and improved: Modeling versions to improve app recommendation. In *Proc of the 37th int'l acm sigir conf on research and development in information retrieval* (pp. 647–656).

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

Liu, B., Fu, Y., Yao, Z., & Xiong, H. (2013). Learning geographical preferences for point-of-interest recommendation. In *Proc. of the 19th acm sigkdd int'l conf. on knowledge discovery and data mining* (pp. 1043–1051).

Liu, B., Kong, D., Cen, L., Gong, N. Z., Jin, H., & Xiong, H. (2015). Personalized mobile app recommendation: Reconciling app functionality and user privacy preference. In *Proc. of the 8th int'l conf. on web search and data mining* (pp. 315–324).

Liu, B., & Xiong, H. (2013). Point-of-interest recommendation in location based social networks with topic and location awareness. In (pp. 396–404).

Liu, B., Xiong, H., Papadimitriou, S., Fu, Y., & Yao, Z. (2015a). A general geographical probabilistic factor model for point of interest recommendation. *Knowledge and Data Engineering, IEEE Transactions on*, *27*(5), 1167–1179.

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

Liu, C., Yang, H.-c., Fan, J., He, L.-W., & Wang, Y.-M. (2010). Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce. In *Proc. of the 19th int'l conf. on world wide web* (pp. 681–690).

Liu, Q., Ge, Y., Li, Z., Chen, E., & Xiong, H. (2011). Personalized travel package recommendation. In *Icdm* (p. 407-416).

Luce, R. D. (1959). *Individual Choice Behavior: A theoretical analysis.* Wiley.

Ma, H., Liu, C., King, I., & Lyu, M. R. (2011). Probabilistic factor models for web site recommendation. In *Proc. of the 34th 'l acm sigir conf. on research and development in information retrieval* (pp. 265–274).

Manski, C. F. (1977). The structure of random utility models. *Theory and decision*, *8*(3), 229–254.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior. In P. Zarembka (Ed.), *Frontiers in econometrics.* New-York: Academic Press.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Morin, F., & Bengio, Y. (2005). Hierarchical probabilistic neural network language model. In *Proc of the tenth int'l workshop on artificial intelligence and statistics* (pp. 246–252).

Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., & Yang, Q. (2008). One-class collaborative filtering. In *Data mining, 2008. icdm'08. eighth ieee int'l conf on* (pp. 502–511).

Park, D. H., Liu, M., Zhai, C., & Wang, H. (2015). Leveraging user reviews to improve accuracy for mobile app retrieval. In *Proceedings of the 38th international acm sigir conference on research and development in information retrieval* (pp. 533–542).

Pennacchiotti, M., & Gurumurthy, S. (2011). Investigating topic models for social media user recommendation. In *Www 2011* (pp. 101–102).

Rendle, S. (2010). Factorization machines. In *Data mining (icdm), 2010 ieee 10th international conference on* (pp. 995–1000).

Rendle, S. (2012, May). Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.*, *3*(3), 57:1–57:22. Retrieved from `http://doi.acm.org/10.1145/2168752.2168771` doi: 10.1145/2168752.2168771

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: bayesian personalized ranking from implicit feedback. In *Proc of the 25th conf on uncertainty in artificial intelligence* (pp. 452–461).

Salakhutdinov, R., & Mnih, A. (2008a). Bayesian probabilistic matrix factorization using markov chain monte carlo. In (pp. 880–887).

Salakhutdinov, R., & Mnih, A. (2008b). Probabilistic matrix factorization. In *Nips* (Vol. 20).

Scellato, S., Noulas, A., & Mascolo, C. (2011). Exploiting place features in link prediction on location-based social networks. In *Proc. of the 17th acm sigkdd int'l conf. on knowledge discovery and data mining* (pp. 1046–1054).

Schmidt, M. N., Winther, O., & Hansen, L. K. (2009). Bayesian non-negative matrix factorization. In (pp. 540–547).

Shekhar, S., Dietz, M., & Wallach, D. S. (2012). Adsplit: Separating smartphone advertising from applications. In *Usenix security.*

Sizov, S. (2010). Geofolk: latent spatial semantics in web 2.0 social media. In (pp. 281–290).

Smartphone Sales in the Third Quarter of 2013. (2013). `http://www.finfacts.ie/irishfinancenews/article_1026800.shtml`.

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

Tobler, W. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, *46*(2), 234–240.

Train, K. (1978). A validation test of a disaggregate mode choice model. *Transportation research*, *12*(3), 167–174.

Wang, C., & Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. In *Kdd 2011* (pp. 448–456).

Woerndl, W., Schueller, C., & Wojtech, R. (2007). A hybrid recommender system for context-aware recommendations of mobile applications. In *Data engineering workshop, 2007 ieee 23rd international conference on* (pp. 871–878).

Yang, S.-H., Long, B., Smola, A. J., Zha, H., & Zheng, Z. (2011). Collaborative competitive filtering: Learning recommender using context of user choice. In *Proc of the 34th int'l acm sigir conf on research and development in information retrieval* (pp. 295–304).

Ye, M., Shou, D., Lee, W.-C., Yin, P., & Janowicz, K. (2011). On the semantic annotation of places in location-based social networks. In *Kdd 2011* (pp. 520–528).

Ye, M., Yin, P., & Lee, W.-C. (2010). Location recommendation for location-based social networks. In *Proc. of the 18th sigspatial int'l conf. on advances in geographic information systems* (pp. 458–461).

Ye, M., Yin, P., Lee, W.-C., & Lee, D.-L. (n.d.). Exploiting geographical influence for collaborative point-of-interest recommendation. In (pp. 325–334).

Yin, P., Luo, P., Lee, W.-C., & Wang, M. (2013). App recommendation: a contest between satisfaction and temptation. In (pp. 395–404).

Yin, Z., Cao, L., Han, J., Zhai, C., & Huang, T. (n.d.). Geographical topic discovery and comparison. In (pp. 247–256).

Zhang, L., Agarwal, D., & Chen, B.-C. (2011). Generalizing matrix factorization through flexible regression priors. In *Proc. of the fifth acm conf. on recommender systems* (pp. 13–20).

Zhang, S., Wang, W., Ford, J., & Makedon, F. (2006). Learning from incomplete ratings using non-negative matrix factorization. In *Sdm'06.*

Zhang, Y., Ahmed, A., Josifovski, V., & Smola, A. (2014). Taxonomy discovery for personalized recommendation. In *Proc. of the 7th int'l conf. on web search and data mining* (pp. 243–252).

Zheng, V. W., Cao, B., Zheng, Y., Xie, X., & Yang, Q. (2010). Collaborative filtering meets mobile recommendation: A user-centered approach. In *Aaai.*

Zheng, V. W., Zheng, Y., Xie, X., & Yang, Q. (n.d.). Collaborative location and activity recommendations with gps history data. In (pp. 1029–1038).

Zheng, Y., & Xie, X. (2011, January). Learning travel recommendations from user-generated gps traces. *ACM Trans. Intell. Syst. Technol.*, *2*(1), 2:1–2:29.

Zheng, Y., Zhang, L., Xie, X., & Ma, W.-Y. (2009). Mining interesting locations and travel sequences from gps trajectories. In *Www 2009* (pp. 791–800).

Zhou, M., & Carin, L. (n.d.). Negative binomial process count and mixture modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on.*

Zhou, M., Hannah, L., Dunson, D. B., & Carin, L. (2012). Beta-negative binomial process and poisson factor analysis. In *Aistats* (p. 1462-1471).

Zhu, H., Xiong, H., Ge, Y., & Chen, E. (2014). Mobile app recommendation with security and privacy awareness. In *Proc. of the 20th acm int'l conf. on knowledge discovery and data mining.*

Ziegler, C.-N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on world wide web* (pp. 22–32).