Mobi-SAGE: A Sparse Additive Generative Model for Mobile App Recommendation

Hongzhi Yin† Liang Chen§ Weiqing Wang† Xingzhong Du† Quoc Viet Hung Nguyen† Xiaofang Zhou†
†*The University of Queensland, School of Information Technology and Electrical Engineering, Australia*
§*School of Data and Computer Science, Sun Yat-sen University, China*
†db.hongzhi@gmail.com §jasonclx@gmail.com †{weiqingwang,x.du,q.nguyen,uqxzhou}@uq.edu.au

***Abstract*—With the rapid prevalence of smart mobile devices and the dramatic proliferation of mobile applications (Apps), App recommendation becomes an emergent task that will benefit different stockholders of mobile App ecosystems. Unlike traditional items, Apps have privileges to access a user's sensitive resources (e.g., contacts, messages and locations) which may lead to security risk or privacy leak. Thus, users' choosing of Apps are influenced by not only their personal interests but also their privacy preferences. Moreover, user privacy preferences vary with App categories. In this paper, we propose a mobile sparse additive generative model (Mobi-SAGE) to recommend Apps by considering both user interests and category-aware user privacy preferences. We collected a real-world dataset from 360 App store - the biggest Android App platform in China, and conduct extensive experiments on it. The experimental results show that our Mobi-SAGE consistently and significantly outperforms the state-of-the-art approaches, which implies the importance of exploiting category-aware user privacy preferences.**

I. Introduction

Recent years have witnessed the rocketing development and prevalence of smart mobile devices, such as smart phones. An important driver behind the wide adoption of smart mobile devices is the emergence of application ("App") stores, where the third party developers publish mobile Apps that users can download to augment their mobile devices' functionality. As of November 2015, there were over 1.8 million Apps with over 60 billion cumulative downloads on Google Play (one of the largest App markets), and the number of Apps grew by around 80% between July 2013 and November 2015. With the dramatically increasing number of Apps, it is hard for users to explore the huge world of Apps and locate relevant Apps. Thus, it is urgent to develop effective personalized App recommendation systems. However, we discover that most mainstream App markets (e.g., Google Play and Apple App Store) currently do not provide the functionality of personalized App recommendation. In this paper, we focus on developing mobile App recommender system.

Unlike the traditional items such as music, movies, books and points of interest, Apps have privileges to access the users' personal private information such as location, contacts and messages. Android and Iphone use permission systems to control the privileges of Apps. Apps can only access privacy and security-relevant resources if the user approves an appropriate permission request. For example, an Android application can only send text messages if it has the permission "SEND_SMS". As reported by NBC news[1], users have grown very concerned about privacy on their mobile phones. For instance, many users have avoided downloading some mobile Apps which may have access to their personal data. Besides, different users might have different privacy preferences, e.g., user A tends to not allow the App to share contacts while user B does not expect his/her locations (e.g., home locations or workplaces) to be spied by the third party Apps. Moreover, user privacy preferences tend to vary with App categories/functionalities [2]. For instances, users tend to permit navigation Apps to access their locations, while they might forbid entertainment Apps from doing that.

In light of this, we propose a mobile sparse additive generative model (Mobi-SAGE) for mobile App recommendation in this paper, which jointly learns user interests and category-aware user privacy preferences in a unified way. However, given a user, his/her rated or downloaded Apps with a specific category are extremely sparse. It is very difficult to directly learn the user-category-specific privacy preferences without overfitting. To combat the data sparsity issue, we decompose user-category-specific privacy preferences into two parts: user-specific privacy preferences and category-specific privacy preferences. The first part captures the user's stable privacy preferences, while the second part exploits the public's collective privacy preferences for a specific category of Apps (i.e., the wisdom of crowds) that captures the common patterns in permission requests for Apps with that specific category. Thus, an App with "unusual" permission requests (e.g., malware Apps) will enjoy low priority to be recommended.

II. Preliminaries and Problem Formulation

In this section, we first present the private resources and permission system in mobile devices, and then formulate the problem of mobile App recommendation.

A. Private Resources and Permissions

We focus on Android Apps in this paper, although our approach is also applicable to other types of Apps (e.g., Iphone Apps). We follow the description of Android resources and permissions in [3]. Specifically, Android system is a permission-based framework to control the privileges of Apps. A permission consists of two elements: resources (e.g., Internet, contact, camera and location) and operations (e.g., read and write), and granting a permission to an App allows the App to operate the corresponding resource. For instance, giving the permission "READ_CONTACTS" to an App makes it authorized to read the user's contact data.

[1]http://www.nbcnews.com/

[2]App categories and functionalities are equivalent, as categories in the App stores are defined according to the Apps' functionalities.

2375-026X/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDE.2017.43

To use the functionality of an App, 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. A user with low privacy concerns might feel fine to use the App at the cost of his/her privacy, while a user with high privacy concerns might give up the App or transfer to another App that provides the same or similar functionality but uses less private resources.

B. Problem Definition

Notation. Through this paper, all vectors are column vectors and are denoted by bold lower case letters (e.g., $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$). We use calligraphic letters to represent sets, e.g., $\mathcal{U}$ and $\mathcal{A}$ represent the user set and App set. For simplicity, we use their corresponding normal letters to denote their cardinalities (e.g., $A = |\mathcal{A}|$).

Definition 1: **(Mobile App)** A mobile App is a computer program designed to run on mobile devices.

In our work, a mobile App has four attributes: identifier, description, images and permissions. We use a to represent an App identifier and $\mathcal{P}_a$ to denote a's permission request set. $\mathcal{T}_a$ and $\mathcal{V}_a$ are used to denote a's textual content and visual content, and we apply bag of words (BoW) to represent an App in both textual and visual spaces. Specifically, $\mathcal{T}_a$ is a collection of words (i.e., a textual document) extracted from a's description and name, and $\mathcal{V}_a$ is a collection of visual words (i.e., a visual document) extracted from the images associated with a using SIFT [5].

Definition 2: **(User Profile)** For each user u, we create a user profile $\mathcal{D}_u = \{(a,\ t)\}$, which is a set of u's downloading records. t is the timestamp of u downloading a. Thus, the whole App downloading dataset $\mathcal{D}$ consists of all user profiles, i.e., $\mathcal{D} = \{\mathcal{D}_u : u \in \mathcal{U}\}$ where $\mathcal{U}$ is the set of users.

Below we formally formulate our problem as:

Problem 1: **(Mobile App Recommendation)** Given a user set $\mathcal{U}$, a mobile App set $\mathcal{A}$ and a user downloading history dataset $\mathcal{D}$, our goal is to recommend top-k most relevant mobile Apps from $\mathcal{A}$ for each user $u \in \mathcal{U}$ by learning both user interests and category-aware user privacy preferences.

III. The Mobi-SAGE Model

To model mobile users' downloading behaviors on App stores, we propose a mobile sparse additive generative model (Mobi-SAGE) based on SAGE model [2]. Figure 1 shows the graphical representation of Mobi-SAGE. Our input data, that is, users' downloading profiles, are modeled as observed random variables, shown as shaded circles in Figure 1. Mobi-SAGE is a generative model jointly over the textual words, visual words, App-IDs and permission requests in the users' downloading profiles. It *discovers multi-view topics* and *learns user interests* and *topic-aware user privacy preferences* in a unified way. Below, we will describe each component.

User Interest Modeling. Intuitively, a user chooses an App by matching his/her personal interests with the content of that App. Inspired by the early work on user interest modeling [4], [12], [11], [9], [10], [13], Mobi-SAGE also adopts latent topics

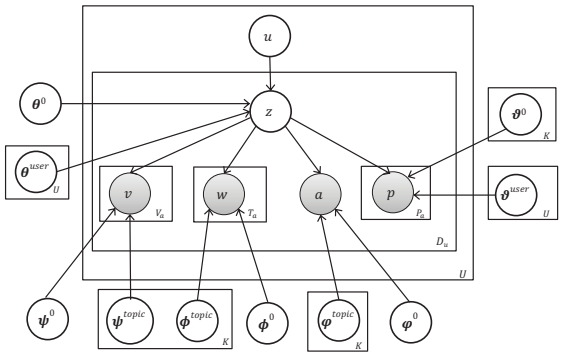

Fig. 1. The Graphical Representation of Mobi-SAGE.

to characterize users' interests. Specifically, we learn a topic-based vector representation for each user according to his/her downloaded Apps and their associated multi-modal content, denoted as $\boldsymbol{\theta}_u^{user}$. Besides, to alleviate the data sparsity and address the cold-start user problem, we also introduce a topic-based background vector $\boldsymbol{\theta}^0$ to capture the general public's interests (i.e., the common topics among all users). Another purpose of using a background model $\boldsymbol{\theta}^0$ is to make each user's interests $\boldsymbol{\theta}_u^{user}$ learned from the dataset more discriminative.

Multi-View Topic Modeling. The quality of the discovered topics are very important for user interest modeling. To take full advantage of the strengths of both content-based and collaborative filtering-based recommendation methods, a topic z in our Mobi-SAGE model is associated with three vectors: a textual-word vector $\boldsymbol{\phi}_z^{topic}$, a visual-word vector $\boldsymbol{\psi}_z^{topic}$ and an App-ID vector $\boldsymbol{\varphi}_z^{topic}$. In other words, each topic z in our model is responsible for simultaneously generating textual words, visual words and App-IDs. This design enables $\boldsymbol{\phi}_z^{topic}$, $\boldsymbol{\psi}_z^{topic}$ and $\boldsymbol{\varphi}_z^{topic}$ to be mutually influenced and enhanced during the topic discovery process by associating them. Thus, the discovered topic z, on one hand, can cluster semantically and visually similar Apps together. On the other hand, it can also capture the App co-occurrence patterns to link relevant Apps together, similar to item-based collaborative filtering methods. Besides, through topic z, we capture the correlation between textual content and visual content (i.e., textual words and visual words), and provide a multi-view interpretation for the topics. By integrating and exploiting multi-modal contents associated with Apps, our Mobi-SAGE can effectively address the cold-start App problem. We also introduce three background models for textual words, visual words and App-IDs, respectively: $\boldsymbol{\phi}^0$, $\boldsymbol{\psi}^0$ and $\boldsymbol{\varphi}^0$. The purpose of using background models is to make the topics learned from the dataset more discriminative, since $\boldsymbol{\phi}^0$, $\boldsymbol{\psi}^0$ and $\boldsymbol{\varphi}^0$ assign high probabilities to non-discriminative and non-informative textual and visual words and Apps.

For each App a in the user profile $\mathcal{D}_u$, $\mathcal{T}_a$ and $\mathcal{V}_a$ are a bag of textual words and a bag of visual words describing App a in the semantic and visual spaces, respectively. We associate $\mathcal{T}_a$ and $\mathcal{V}_a$ with a latent variable z to indicate the topic of App a. By applying this constraint to our Mobi-SAGE model, we aim to build the potential one-to-one correspondence between the discovered latent topics and the categories defined by the App stores. As categories in the App stores are defined according to the Apps' functionalities, a discovered topic is expected

to effectively capture and describe the functionalities that the same type of Apps have.

Topic-Aware User Privacy Preference Modeling. Being different from the traditional items or products, the user's decision making for mobile Apps is also influenced by his/her privacy preferences. Moreover, user privacy preferences are not always stable and tend to vary with App categories. Therefore, we need to model topic-aware user privacy preferences in Mobi-SAGE. One alternative is to directly learn a user-category-specific vector $\boldsymbol{\vartheta}_{u,z}$ to capture u's privacy preferences under topic z. Considering that a user's downloaded Apps under a specific category are extremely sparse, and it is very difficult to learn $\boldsymbol{\vartheta}_{u,z}$ without overfitting. Therefore, to combat the data sparsity issue, we take advantage of the additivity in SAGE to decompose the user-category-specific privacy preferences $\boldsymbol{\vartheta}_{u,z}$ into two parts: user-specific privacy preferences $\boldsymbol{\vartheta}_u^{user}$ and category-specific privacy preferences $\boldsymbol{\vartheta}_z^0$ (i.e., $\boldsymbol{\vartheta}_{u,z} = \boldsymbol{\vartheta}_u^{user} + \boldsymbol{\vartheta}_z^0$). The first component $\boldsymbol{\vartheta}_u^{user}$ is used to capture the intrinsic privacy preferences of user u, while the second component $\boldsymbol{\vartheta}_z^0$ exploits the public's collective privacy preferences for a specific category of Apps (i.e., the wisdom of crowds) that captures the common patterns in permission requests for Apps with that specific category. Thus, an App with "unusual" permission requests (e.g., malwares) will enjoy low priority to be recommended.

A. Generative Process of Mobi-SAGE

The generative process of the Mobi-SAGE model for a downloaded App a in the user profile $\mathcal{D}_u$ is as follows.

- Draw a topic index $z \sim P(z|\boldsymbol{\theta}^0, \boldsymbol{\theta}_u^{user})$
- For each textual word w in $\mathcal{T}_a$, draw $w \sim P(w|\boldsymbol{\phi}_z^0, \boldsymbol{\phi}_z^{topic})$
- For each visual word v in $\mathcal{V}_a$, draw $v \sim P(v|\boldsymbol{\psi}_z^0, \boldsymbol{\psi}_z^{topic})$
- Draw an App-ID $a \sim P(a|\boldsymbol{\varphi}_z^0, \boldsymbol{\varphi}_z^{topic})$
- For each permission request p in $\mathcal{P}_a$, draw $p \sim P(p|\boldsymbol{\vartheta}_z^0, \boldsymbol{\vartheta}_u^{user})$

For each downloaded App in $\mathcal{D}_u$, Mobi-SAGE first chooses the topic this App is about. To generate the topic index z, we utilize a multinomial model as follows:

$$P(z|\boldsymbol{\theta}^0, \boldsymbol{\theta}_u^{user}) = P(z|\boldsymbol{\theta}^0 + \boldsymbol{\theta}_u^{user}) = \frac{exp(\theta_z^0 + \theta_{u,z}^{user})}{\sum_{z'} exp(\theta_{z'}^0 + \theta_{u,z'}^{user})} \quad (1)$$

Here $\boldsymbol{\theta}^0$ and $\boldsymbol{\theta}_u^{user}$ are topic-based vectors that represent u's personal interests and the general public's interests. Once the topic z is generated, the App a and its associated textual and visual words are generated as expressed in Equations (2, 3, 4), respectively.

$$P(a|\boldsymbol{\varphi}_z^0, \boldsymbol{\varphi}_z^{topic}) = P(a|\boldsymbol{\varphi}_z^0 + \boldsymbol{\varphi}_z^{topic}) = \frac{exp(\varphi_a^0 + \varphi_{z,a}^{topic})}{\sum_{a'} exp(\varphi_{a'}^0 + \varphi_{z,a'}^{topic})} \quad (2)$$

$$P(w|\boldsymbol{\phi}_z^0, \boldsymbol{\phi}_z^{topic}) = P(w|\boldsymbol{\phi}_z^0 + \boldsymbol{\phi}_z^{topic}) = \frac{exp(\phi_w^0 + \phi_{z,w}^{topic})}{\sum_{w'} exp(\phi_{w'}^0 + \phi_{z,w'}^{topic})} \quad (3)$$

$$P(v|\boldsymbol{\psi}_z^0, \boldsymbol{\psi}_z^{topic}) = P(v|\boldsymbol{\psi}_z^0 + \boldsymbol{\psi}_z^{topic}) = \frac{exp(\psi_v^0 + \psi_{z,v}^{topic})}{\sum_{v'} exp(\psi_{v'}^0 + \psi_{z,v'}^{topic})} \quad (4)$$

Similarly, based on the sampled topic z, the permissions of App a are generated as follows:

$$P(p|\boldsymbol{\vartheta}_z^0, \boldsymbol{\vartheta}_u^{user}) = P(p|\boldsymbol{\vartheta}_z^0 + \boldsymbol{\vartheta}_u^{user}) = \frac{exp(\vartheta_{z,p}^0 + \vartheta_{u,p}^{user})}{\sum_{p'} exp(\vartheta_{z,p'}^0 + \vartheta_{u,p'}^{user})} \quad (5)$$

The above generative process applies to all user profiles in the dataset. The graphical representation of the generation process is shown in Figure 1.

We employ the Gibbs EM algorithm [8] to infer the model parameters.

B. Efficient Mobile App Recommendation

Once we have trained the model Mobi-SAGE, given a target user u, we compute the probability for each unrated App a, as in Equation (6), and then select the top-k ones with highest probabilities as recommendations.

$$\begin{aligned} & P(a, \mathcal{T}_a, \mathcal{V}_a, \mathcal{P}_a|u, \ominus) \\ &= \sum_{z \in \mathcal{K}} P(a, z, \mathcal{T}_a, \mathcal{V}_a, \mathcal{P}_a|u, \ominus) \\ &= \sum_{z \in \mathcal{K}} P(z|\boldsymbol{\theta}^0, \boldsymbol{\theta}_u^{user}) P(a|\boldsymbol{\varphi}^0, \boldsymbol{\varphi}_z^{topic}) P(\mathcal{T}_a|\boldsymbol{\phi}^0, \boldsymbol{\phi}_z^{topic}) \\ &\quad \times P(\mathcal{V}_a|\boldsymbol{\psi}^0, \boldsymbol{\psi}_z^{topic}) P(\mathcal{P}_a|\boldsymbol{\vartheta}_z^0, \boldsymbol{\vartheta}_u^{user}) \\ &= \sum_{z \in \mathcal{K}} \zeta_{u,z} \beta_{z,a} \Big(\prod_{w \in \mathcal{T}_a} \gamma_{z,w} \Big)^{\frac{1}{T_a}} \Big(\prod_{v \in \mathcal{V}_a} \xi_{z,v} \Big)^{\frac{1}{V_a}} \Big(\prod_{p \in \mathcal{P}_a} \eta_{u,z,p} \Big)^{\frac{1}{P_a}} \end{aligned} \quad (6)$$

IV. Experiments

In this section, we first describe the setup of experiments and then demonstrate the experimental results.

A. Dataset

Our data comes from a leading Android app store in China, called 360 Mobile Assistant [3]. The 360 platform has over 275 million active users. We randomly sampled 100000 users and collected their App download records in the recent one year. Totally, there are 50217 Apps and 10,203,755 download records in our dataset. For each App, we collected its name, description, screenshots and permissions.

B. Comparison Approaches

SPAR. SPAR [14] is the first security-aware mobile App recommendation approach. It first computed a security score for each App by a regularization approach, and then used a modern portfolio theory based method to rank Apps by striking a balance between the Apps' popularity and their security scores. It is a non-personalized recommendation approach.

LIBFM. LIBFM [6] is a state-of-the-art feature based factor model library. Based on it, we build a factorization model by incorporating all side information of Apps including textual, visual and permissions with the collaborative filtering.

BPR. BPR [7] is the state-of-the-art of personalized ranking model for implicit feedback datasets.

PBPR. Based on BPR, we implement a stronger baseline, the privacy-aware BPR, to integrate user privacy preferences with user interests, following the method developed in [3].

[3] http://zhushou.360.cn/

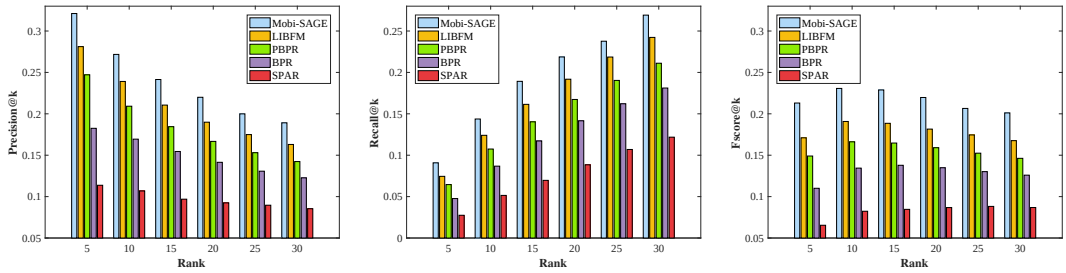

Fig. 2. Recommendation Effectiveness on 360 Mobile App Dataset.

C. Evaluation methods

Dataset Splitting. Given a user profile $\mathcal{D}_u$, namely a collection of downloading records, we first rank the records according to their downloading timestamps. Then, we use the 70-th and 80-th percentiles as the cut-off points so that the first 70% records are used for training, and the last 20% are marked off as the testing data. The remaining 10% are used as the validation data to tune the model hyper-parameters. Thus, we split the dataset $\mathcal{D}$ into the training set $\mathcal{D}^{training}$, the validation set $\mathcal{D}^{validation}$ and the test set $\mathcal{D}^{test}$.

Evaluation Metrics. In the mobile App recommendation, we present a list of Apps to the user, thus we evaluate the recommender models in terms of personalized ranking. Following [3], we employ three IR metrics: *Precision*, *Recall* and F_β metric with $\beta = 0.5$ since we are more concerned about "Precision" in the recommendation scenario.

D. Recommendation Effectiveness

In this part, we present the effectiveness of the recommendation methods with well-tuned parameters. Figure 2 reports the comparison results between our proposed model Mobi-SAGE and other competitor methods.

Clearly, our proposed Mobi-SAGE model significantly outperforms other competitor models consistently in all the three metrics, and the relative improvements, in terms of Fscore@10, are 20.9%, 38.8%, 71.5% and 180% compared with LIBFM, PBPR, BPR and SPAR, respectively. Several observations are also made from the results: (1) PBPR achieves higher recommendation accuracy than BPR, showing the benefit brought by considering user privacy preferences. (2) Both Mobi-SAGE and LIBFM outperform PBPR. It might be because Mobi-SAGE and LIBFM leverage visual and textual contents of Apps to alleviate the issue of data sparsity. (3) Our Mobi-SAGE beats LIBFM due to that LIBFM cannot model and capture the subtle dependence between App functionalities and user privacy preferences. Another possible reason is that LIBFM is a general-purpose factor model that utilizes auxiliary information in a feature-engineering way and treats each feature equally, while our Mobi-SAGE is able to treat each App feature in a more reasonable way and thus improve the recommendation quality. (4) SPAR drops behind all other models, as it is a non-personalized recommendation method and tends to recommend the most popular Apps for each user, ignoring the uniqueness of user interests and tastes [1].

V. Conclusions

In this paper, we proposed a mobile sparse additive generative model (Mobi-SAGE) to recommend Apps by considering both user interests and functionality-aware user privacy preferences. To evaluate the performance of our proposed recommender model Mobi-SAGE, extensive experiments were conducted on a large App dataset. The experimental results revealed that our Mobi-SAGE significantly outperforms the state-of-the-art approaches, which implies the importance of exploiting functionality-aware user privacy preferences.

VI. Acknowledgement

The work described in this paper is partially supported by ARC Discovery Early Career Researcher Award (DE160100308), ARC Discovery Project (DP170103954) and the National Natural Science Foundation of China (61572335).

References

[1] Y.-A. de Montjoye, L. Radaelli, V. K. Singh, and A. ". Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539, 2015.

[2] J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse additive generative models of text. In *ICML*, pages 1041–1048, 2011.

[3] B. Liu, D. Kong, L. Cen, N. Z. Gong, H. Jin, and H. Xiong. Personalized mobile app recommendation: Reconciling app functionality and user privacy preference. In *WSDM*, pages 315–324, 2015.

[4] B. Liu and H. Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. In *SDM*, pages 396–404, 2013.

[5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.

[6] S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[8] H. M. Wallach. Topic modeling: Beyond bag-of-words. In *ICML*, pages 977–984, 2006.

[9] W. Wang, H. Yin, S. Sadiq, L. Chen, M. Xie, and X. Zhou. Spore: A sequential personalized spatial item recommender system. In *ICDE*, pages 954–965, 2016.

[10] H. Yin, B. Cui, X. Zhou, W. Wang, Z. Huang, and S. Sadiq. Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. *ACM Trans. Inf. Syst.*, 35(2):11:1–11:44, Oct. 2016.

[11] H. Yin, Z. Hu, X. Zhou, H. Wang, K. Zheng, Q. V. H. Nguyen, and S. Sadiq. Discovering interpretable geo-social communities for user behavior prediction. In *ICDE*, pages 942–953, 2016.

[12] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: A location-content-aware recommender system. In *KDD*, pages 221–229, 2013.

[13] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen. Adapting to user interest drift for poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2566–2581, 2016.

[14] H. Zhu, H. Xiong, Y. Ge, and E. Chen. Mobile app recommendations with security and privacy awareness. In *KDD*, pages 951–960, 2014.