From Fingerprint to Footprint: Cold-start Location Recommendation by Learning User Interest from App Data

ZHEN TU, Department of Electrical Engineering, Tsinghua University, China
YALI FAN, Sun Yat-Sen University, China
YONG LI*, Department of Electrical Engineering, Tsinghua University, China
XIANG CHEN, Sun Yat-Sen University, China
LI SU, Tsinghua University, China
DEPENG JIN, Beijing National Research Center for Information Science and Technology(BNRist), Department of Electrical Engineering, Tsinghua University, China

With increasing diversity of user interest and preference, personalized location recommendation is essential and beneficial to our daily life. To achieve this, the most critical challenge is the cold-start recommendation problem, for we cannot learn preference from cold-start users without any historical records. In this paper, we demonstrate that it is feasible to make personalized location recommendation by learning user interest and location features from app usage data. By proposing a novel generative model to transfer user interests from app usage behavior to location preference, we achieve personalized location recommendation via learning the interest's correlation between locations and apps. Based on two real-world datasets, we evaluate our method's performance with a variety of scenarios and parameters. The results demonstrate that our method outperforms the state-of-the-art solutions in solving cold-start problem, *i.e.*, when there are 60% cold-start users, we can still achieve a 77.0% hitrate in recommending the top five locations, which is at least 9.6% higher than the baselines. Our study is the first step forward for transferring user interests learning from online fingerprints to offline footprints, which paves the way for better personalized location recommendation services.

CCS Concepts: • **Information systems → Collaborative filtering**; **Personalization**; *Decision support systems.*

Additional Key Words and Phrases: Location recommendation, cold-start problem, generative model

ACM Reference Format:
Zhen Tu, Yali Fan, Yong Li, Xiang Chen, Li Su, and Depeng Jin. 2019. From Fingerprint to Footprint: Cold-start Location Recommendation by Learning User Interest from App Data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 1, Article 26 (March 2019), 22 pages. https://doi.org/10.1145/3314413

26

*This is the corresponding author.

This work was supported in part by the National Key Research and Development Program of China under grant 2017YFE0112300, the National Nature Science Foundation of China under 61861136003, 61621091 and 61673237, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.
Authors' addresses: Zhen Tu, Department of Electrical Engineering, Tsinghua University, China; Yali Fan, Sun Yat-Sen University, China, fanyali@mail2.sysu.edu.cn; Yong Li, Department of Electrical Engineering, Tsinghua University, Beijing, 100084, China, liyong07@tsinghua.edu.cn; Xiang Chen, Sun Yat-Sen University, China, chenxiang@mail.sysu.edu.cn; Li Su, Tsinghua University, China; Depeng Jin, Beijing National Research Center for Information Science and Technology(BNRist), Department of Electrical Engineering, Tsinghua University, China.

2474-9567/2019/3-ART26 $15.00
https://doi.org/10.1145/3314413

1 INTRODUCTION

In such an information exploding era, it is essential to learn user interest from their online behaviors to make recommendations. One of the most fundamental problems is location recommendation, which aims to predict which new locations the user may like [9, 23]. For example, many location based mobile applications (*i.e.*, Yelp and MaFengWo [30]) provide users new attractions to have fun, and such location recommendations should be customized to satisfy diverse user interests. For a young mother with a little baby, an appropriate recommendation may be a children's playground. While for a high school student, probably an educational museum is very suitable. Both mobile users and location owners can benefit from the effective location recommendation: for mobile users, they can more easily find favorite locations and have better life experiences; for location owners, they are able to target potential customers and attract them by precision marketing.

With the increasing diversity of user interest and activity, personalized location recommendation [15, 21, 22, 50, 51] becomes very necessary to provide satisfactory user services. To achieve this task, the most critical challenge is cold-start recommendation problem [33, 38, 44, 48], *i.e.*, providing recommendations for cold-start users or locations without any historical records. *User* cold-start recommendation problem [44, 48] aims to recommend locations to users who have not previously visited any locations. Due to the lack of interactions with locations, we cannot learn their potential interest. While location visiting patterns of different users are rather diverse, it is hard to predict suitable locations to achieve an effective recommendation. Likewise, in *location(item)* cold-start recommendation problem [33, 44], where a location has not been previously visited by anyone, learning its features and recommending it to potential users is also impossible [34]. To solve these cold-start problems, we need to look at other data sources, *e.g.*, learning user interest from other related behavior data or learning location features from other side information. Generally, one type of external information, *i.e.*, user side or item side, can only solve one type of problems, either user cold-start problem or location cold-start problem. In this paper, we consider utilizing human app usage data to assist location recommendation, which is feasible to solve these two types of cold-start problems together. On the one hand, using mobile app is now becoming the most important online activity in daily life, resulting in large amounts of app usage data. On the other hand, mobile app usage data can reflect user interests and correlates with location properties. Different users may use different apps according to their interests [2, 37, 46]. More importantly, app usage is closely correlated with locations, which have quite different usage patterns with distinct functions [42]. For example, people tend to use more office apps in their office. While staying at home, entertainment apps are used more frequently. These examples indicate that it is likely to learn user's interest from the app usage behaviors, and at the same time obtain the features of a location from the app usage information aggregated in it. Therefore, app usage data can be regarded as both user side and item side information, which has the potential to solve both user and location cold-start location recommendation problems. These two features provide us a unique opportunity to transfer knowledge from user app usage fingerprints to their footprints in the physical world.

In this paper, we propose a transfer learning based generative model for the personalized location recommendation problem, which transfers user interest and location features from app usage data to help cold-start location recommendation. By building user-location matrix, user-app matrix and location-app matrix from the data, we jointly learn the latent vectors of users, locations and apps, especially for cold-start users or locations. Based on the latent vectors, we estimate the user preferences towards the unobserved locations so as to recommend the $TopK$ ones. To sum up, the main contributions of this work are three-fold:

- To the best of our knowledge, we are the first to solve cold-start location recommendation problem by transferring app usage information, which enables learning user interest and location features to make up for the lack of historical records for cold-start users or locations, respectively.

- We propose a generative model based on collaborative filtering to achieve personalized location recommendation, which transfers knowledge from app usage domain into location visit domain by jointly learning the underlying interest correlations across these domains.
- We evaluate the performance of our model based on two real-world datasets with a variety of scenarios and parameters. The results demonstrate that our method outperforms the other state-of-the-arts in solving cold-start problem, *i.e.*, when there are 60% cold-start users, $Top5$ $Hitrate$ of our solution can achieve 77.0%, at least 9.6% higher than other baselines; when there are 30% cold-start locations, we can achieve 95.5% hitrate when recommending $Top5$ users to them.

The rest of the paper is organized as follows. Section 2 describes the dataset and Section 3 motivates our research. Section 4 defines the problem and introduces our methodology, and Section 5 evaluates its performance. Finally, Section 6 reviews the related work and discusses its implication, and Section 7 concludes the paper.

2 DATASET AND OBSERVATION

In our study, we aim to solve the cold-start location recommendation problem with the help of app usage data. In order to validate the feasibility of this idea, we use two real-world mobile app usage datasets with both location and app information. Now we introduce them in details.

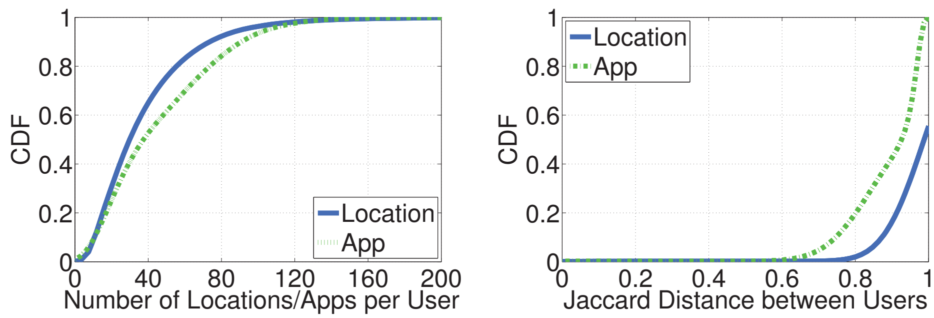
TalkingData Dataset (public): This dataset is collected by TalkingData SDK (*i.e.*, integrated within mobile applications) and released in the Kaggle website [14]. It records fine-grained app usage behavior and spatiotemporal information of mobile users in their devices. For each record, it contains the anonymized device ID, timestamp, longitude and latitude, and the active app ID. Since the location information is represented by a GPS coordinate and users are distributed all around China (too large), we perform pre-processing to make it suitable for our investigated problem. Firstly, we transfer GPS coordinates to grid IDs by dividing the coverage area into many 1km*1km grids. Secondly, we reduce the geographical coverage by choosing the densest 200km*200km area (located in Guangzhou, a major metropolitan city of China), and then extract all records of targeted users who have at least 30 records generated within the area. Thirdly, we filter out locations and apps with very few users for the sake of data sparsity. Through the process, finally we obtain a small-scale but fine-grained mobile app usage dataset with 256 users, 439 different locations and 689 distinct apps. On average, a user has visited 9 locations and used 27 apps. The key statistics of this dataset is shown in Table 1.

Telecom dataset: This dataset is a large-scale mobile app usage dataset, which is collected during April 20-26, 2016 from a major mobile network operator in China [42]. It contains the access records of the users when they issue a connection request to the cellular towers, covering 1.37 million users and 9.4 billion records during one week. Each record contains anonymized user ID, timestamp, connected cellular tower ID and its GPS location, and the used app ID and its category. In addition, all the used apps cover the most popular 2,000 apps across App Store (iOS apps) and Google Play (android apps). Based on the original data, we perform standard data pro-processing as follows: we first select active users with more than 10 visited locations and 5 used apps, after filtering out locations and apps with rare users. As shown in Table 1, the finally utilized dataset contains 10,000 users and 11,584 cellular towers with 1,327 apps. On average, a user has visited 40 locations and used 47 apps.

To further demonstrate the quality of the Telecom dataset, we show some basic statistic results in Figure 1. As for Figure 1 (a), we plot the Cumulative Distribution Function (CDF) of the number of locations and apps that a user visited and used, respectively. From the results, we can observe that only 30% of users have visited less than 20 different locations, which means most of users have been in dozens of locations in the one-week duration. In addition, 5% of users are very active and have visited more than 100 different locations. Likewise, for user app usage, we can observe that the number of used apps per user ranges from 5 to 200, and over 40% of users have used more than 50 different apps.

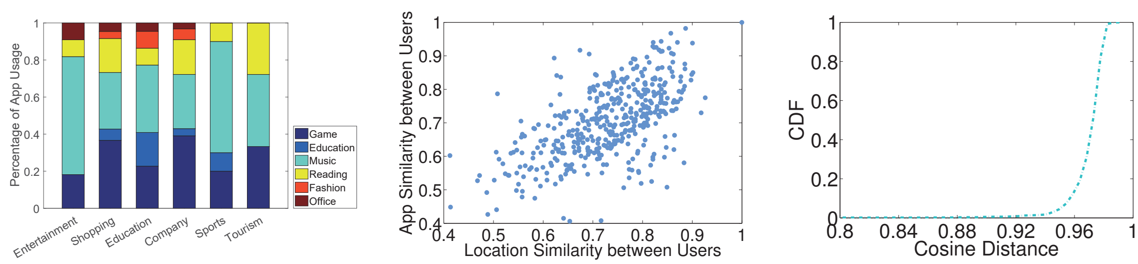
Table 1. Major statistics and key features of our utilized datasets.

Datasets & Metrics	TalkingData Dataset	Telecom Dataset
Source	Mobile application	Cellular network
City	Guangzhou, China	Shanghai, China
Time Duration	1st-7th, May, 2016	20th-26th, April, 2016
Records	180,106	40,470,865
Users	256	10,000
Locations	439	11,584
Apps	689	1,327



(a) Number of locations/apps per user (b) Location/app differences

Fig. 1. Illustration of the key statistical characteristics of our Telecom dataset.



(a) App usage in different locations (b) Location similarity vs app similarity (c) CDF of Cosine Distance

Fig. 2. (a) The distribution of different kinds of app usage in different locations. (b)-(c) The statistical correlation of location similarity and app similarity between pairwise users.

In addition, we measure the difference of user behaviors in this dataset from the perspective of location visit and app preference, which serves as a necessary precondition for our solution. In Figure 1 (b), we plot the Cumulative Distribution Function (CDF) of Jaccard Distance [3] of visited locations between users and used apps between users. We can find that for 90% of pairwise users, their Jaccard Distance of visited locations is more than 0.9, showing that their visited locations are rather different. As for app usage, we can observe that for 80% of pairwise users, their Jaccard Distance is more than 0.8. All these results demonstrate that this dataset contains rich information of users' location visit and app usage behaviors. More importantly, user preferences towards locations and apps are very diverse. On the one hand, it indicates that it is necessary to develop personalized location recommendation for each user since user location preferences are so different. On the other hand, it shows that it may be possible to learn different user interests from app usage data so as to help location recommendation.

According to above analysis, the two datasets have different data sources and user scales, which makes our investigation covering a broad range of scenarios and data quality. Testing recommendation effectiveness in two real-world datasets (one is public), also assesses the repeatability and generalization of our proposed method.

Ethics. We also want to point out that we are very aware of the privacy implications of using these two datasets for research and have taken active steps to protect privacy of mobile users. First, the app usage datasets do not contain any personally identification information or any user-level meta-data. The user ID has been anonymized (as a bit string) by our data providers, and we never have the access to the true user ID. Second, all the researchers are regulated by a strict non-disclosure agreement. Two datasets are stored in a server protected by authentication mechanisms and firewalls. This work also has received the approval from both data providers.

3 MOTIVATION

In this study, we intend to solve cold-start location recommendation problem, whose challenges come from two aspects. On the one hand, we need to recommend locations to cold-start users, who don't have any historical location visiting records for recommendation. On the other hand, we need to recommend users to cold-start locations, *i.e.*, a newly built POI or a location which has not been previously visited by anyone in the system. Since we cannot extract characteristics of such cold-start users or locations, it is challenging to provide personalized recommendations to them and satisfy their diverse requirements.

To tackle above challenges, we look at other data sources and find app usage data is very suitable to serve as external information, which can solve user and location cold-start problems together. On the one hand, app usage data is now becoming increasingly prevalent since this is the most important thing we are doing on the website. On the other hand, app usage preference does have a strong relationship with location visiting behavior. Next, we utilize the large-scale Telecom dataset to show statistic correlations between app usage and location visiting behaviors.

First, we investigate the correlation between apps and locations, $i.e.$, how apps are used in different locations. Usually, what app a user uses is often related to where she is. In Telecom dataset, we randomly select six locations with different functions, $i.e.$, entertainment, shopping, education, company, sports and tourism, then calculate the distribution of different kinds of apps used in these locations. Note that we simply label each location by crawling Point of Interest (POI) Information within the area and regarding the category of the most prevalent POI as the label. The results are shown in Figure 2 (a). From the results, we can find that in entertainment and sports locations, the most frequently used app's type is music and its proportion is much higher than other apps. As for educational location, obviously, users use the related educational apps more frequently. Comparing all locations, the distributions of different kinds of app usage are quite different, and more importantly specific apps are used much more frequently in semantically similar locations. All these results show that we are likely to differentiate different locations or find similar locations according to app usage information in distinct locations. Therefore, even for a cold-start location, we are able to utilize app usage behaviors nearby to learn its features, which absolutely help solve location cold-start problem.

Second, we study the correlation of user's location visitation and app usage behaviors at the individual level. Since we tend to utilize app usage information to help personalized location recommendation, one critical question is whether two users using similar apps tend to visit the same locations. In a case study, we find that 2 users with 8 common apps have the same 3 frequently-visited locations and 3 users sharing 4 common apps have visited the same 5 locations, etc. In order to answer this question comprehensively, we analyze the statistical correlation of location similarity and app similarity between pairwise users. Based on used apps and visited locations, we utilize Cosine Similarity [28] to compute app similarity and location similarity between pairwise users, respectively. In specific, app similarity vector $\mathbf{P}_i = \{p_{ij}\}$ denotes app similarity between user i and other users, with p_{ij} representing app similarity between user i and j. So as the location similarity vector $\mathbf{Q}_i = \{q_{ij}\}$. If the app similarity between user i and user j and their location similarity are closely related, the app vector $\mathbf{P}_i$ and location vector $\mathbf{Q}_i$, *i.e.*, the distribution of similarity, will have a strong correlation. Thus, to further quantify the relationship between apps and locations, we still use Cosine Similarity to compute the correlation between app similarity vector $\mathbf{P}_i$ and location similarity vector $\mathbf{Q}_i$ for user i. The correlation $\mathbf{C}_i$ of $\mathbf{P}_i$ and $\mathbf{Q}_i$ is computed as follows:

$$\mathbf{C}_i = cos(\mathbf{P}_i, \mathbf{Q}_i), \forall i, j = 1, ..., N, \tag{1}$$

with N representing the total number of users.

After formal definition, we demonstrate the results of the correlation between apps and locations in Figure 2 (b) and (c). In Figure 2 (b), we can obviously observe that app similarity and location similarity between different users are linear correlated. As for Figure 2 (c), we plot the Cumulative Distribution Function (CDF) of the correlation $\mathbf{C} = \{\mathbf{C}_i\}$. From the result, we can observe that for nearly all the users, the correlation between their app usage and location visit is more than 92%. In addition, 80% of users have a very high Cosine Similarity (over 96%) between their used apps and visited locations, which means most of users' app usage behaviors have a strong relationship with their mobility patterns. All these results indicate that individual app usage and mobility behavior are correlated strongly. Therefore, for a cold-start user without historical location visiting records, we can learn user interest from her app usage behaviors and choose suitable locations to recommend, which solves the user cold-start problem.

All the above analysis demonstrates that individual app usage behaviors and their location visiting patterns are strongly correlated, which provides a unique opportunity to learn characteristics of cold-start user or locations from the app usage data, so as to make location recommendations. Thus, it is feasible to utilize individual and location app usage data to improve the effectiveness of personalized location recommendation.

4 SOLUTION

In this section, we first formally define our investigated problem, then introduce our proposed personalized location recommendation system.

4.1 Problem Definition

Location recommendation is to predict what other locations the user will like to visit besides those are known from the observations. It requires us to learn user interest and location features accurately and predict user preferences towards different locations. Due to the sparsity of location visitation information, we may only learn a small part of information or even cannot obtain any knowledge about user or location features, leading to the failure of personalized location recommendation, especially for cold-start users or locations. Since we have verified the feasibility of transferring user interest and location characteristic from app usage data to help with personalized location recommendation, in this study we aim to learn more about user interest and location characteristic with the help of app usage data. Based on this targeted scenario, we formally define the investigated problem as follows.

Suppose there are N users, L locations and M apps, then we obtain a user-location matrix $\mathbf{X} = \{x_{ij}\}$ and a user-app matrix $\mathbf{Y} = \{y_{ik}\}$, with x_{ij} and y_{ik} representing the visited frequency of location j and usage frequency of app k for user i, respectively. In addition, based on aggregated app usage information, we can acquire a location-app matrix $\mathbf{Z} = \{z_{jk}\}$, where z_{jk} denotes the aggregated times of using app k in location j for all the users. For our targeted user-location matrix $\mathbf{X}$, $x_{ij} = 0$ means we haven't observed user i visited location j, indicating that user i's preference towards location j is still unknown and needs to predict. In the cold-start setting, for a cold-start user i, all the values in the i-th row of user-location matrix X are missing. Accordingly, for a cold-start location j, all the values in the j-th column of user-location matrix X are missing. In this study, we mainly focus on the recommendation effectiveness of these cold-start users and locations.

Therefore, our investigated problem is how to predict user preferences towards the unobserved locations after knowing user-location matrix, user-app matrix and location-app matrix. Formally, our targeted personalized location visit prediction problem can be defined as:

Input: Users' location visit information denoted by user-location matrix $\mathbf{X}$, users' app usage information denoted as user-app matrix $\mathbf{Y}$ and locations' app usage information denoted as location-app matrix $\mathbf{Z}$.

Output: A predicted denser user-location matrix $\hat{\mathbf{X}}$, *i.e.*, a personalized location ranking preference towards the unobserved locations for each user. More precisely, the prediction of the unobserved values (the zero values) in user-location matrix $\mathbf{X}$.

4.2 System Design

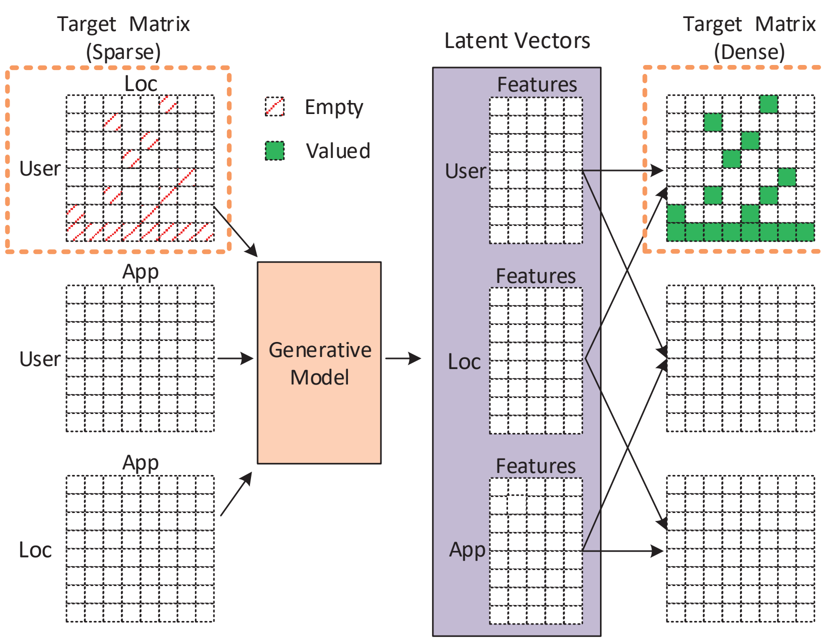


Fig. 3. The framework of our transfer learning based personalized location prediction system.

In recommendation system, transfer learning is a prevalent strategy to incorporate different data source information to accomplish a task. Beyond transfer learning models, matrix co-factorization is one of the most common and effective implement methods in personalized recommendation. In our personalized location recommendation task, we need to combine user-location matrix, user-app matrix and location-app matrix and transfer latent features of users, locations and apps beyond them, so we also adopt this methodology to solve our problem. Figure 3 is the framework of our designed transfer learning based and personalized location recommendation system.

As the figure shows, we first have a sparse user-location matrix $\mathbf{X}$ (even with some rows or columns empty), and obtain a user-app matrix $\mathbf{Y}$ and a location-app matrix $\mathbf{Z}$ as the input. Then, we assume features of users, locations and apps are shared in these matrices and design a generative model to learn their latent vectors. Finally we recover a dense user-location matrix $\hat{X}$ as the output, which achieves the target of predicting user preferences towards the unobserved locations. Now we introduce the key part – how we learn the latent representatives of three different entities.

In order to learn user-location, user-app and location-app information together, we use a generative model to transfer the knowledge among them. What we transfer among the user-location domain, user-app domain and location-app domain are the latent feature vectors of users, locations and apps. Specifically, we denote $\mathbf{U} \in \mathbf{R}^{H \times N}, \mathbf{L} \in \mathbf{R}^{H \times L}$ and $\mathbf{A} \in \mathbf{R}^{H \times M}$ to represent the latent vectors of user, location and app feature matrices respectively, with column vectors u_i, l_j, a_h representing the K-dimensional latent feature vector of user i, location j and app k respectively.

We propose a generative model to learn these vectors by maximize the log-posterior likelihood, which is equivalent to minimizing the following objective function, *i.e.*, a sum of squared errors with quadratic regularization terms:

$$\begin{aligned}\zeta(\mathbf{U},\mathbf{L},\mathbf{A}) = &\frac{1}{2}||I_X \circ \left(\mathbf{X} - g\left(\mathbf{U}^\top \mathbf{L}\right)\right)||_F^2 + \frac{\beta 1}{2}||I_Y \circ \left(\mathbf{Y} - g\left(\mathbf{U}^\top \mathbf{A}\right)\right)||_F^2 \\ &+ \frac{\beta 2}{2}||I_Z \circ \left(\mathbf{Z} - g\left(\mathbf{L}^\top \mathbf{A}\right)\right)||_F^2 + \left(\frac{\lambda_u}{2}||\mathbf{U}||_F^2 + \frac{\lambda_l}{2}||\mathbf{L}||_F^2 + \frac{\lambda_a}{2}||\mathbf{A}||_F^2\right),\end{aligned} \tag{2}$$

where the logistic function $g(x) = 1/(1 + exp(-x))$ bounds the matrix multiplication range within $[0, 1]$ interval, $\circ$ means the point-wise matrix multiplication, and I_X, I_Y, I_Z are denoted as flag matrices for user-location data, user-app data and location-app data respectively. If record of user i and location j is known, then $I_X(i, j) = 1$, otherwise $I_X(i, j) = 0$. I_Y and I_Z are defined in the similar ways. $\| \cdot \|_F^2$ denotes the Frobenius norm. β_1 is the weight of user-app data we use for transfer learning, β_2 means the weight of location-app data. The last three terms are regularization terms with coefficients $\frac{\lambda_u}{2}, \frac{\lambda_l}{2}, \frac{\lambda_a}{2}$, respectively. More details can be found in Section 7.

There exist several methods to reduce the time complexity of model training, and we adopted mini-batch gradient descent approach to learn the parameters. With random sampling, the cost of the gradient update no longer grows linearly in the number of entities related to latent feature vectors, but only in the number of entities sampled. The hyper-parameters, *i.e.*, number of latent features and regularization coefficient, are set by cross-validation.

In conclusion, we propose a transfer learning model to accomplish the personalized location prediction task, which inputs user-location matrix $\mathbf{X}$, user-app matrix $\mathbf{Y}$ and location-app matrix $\mathbf{Z}$, then outputs a denser user-location matrix $\hat{\mathbf{X}}$ by sharing the latent feature vectors of users, locations and apps.

5 EVALUATION

To evaluate the performance of our proposed personalized location recommendation system, we conduct a series of experiments to answer the following three key research questions:

- **RQ1**: Can our method outperform the state-of-art recommendation approaches in different cold-start scenarios, *i.e.*, user cold-start problem and location cold-start problem?
- **RQ2**: What performance can our method achieve under different levels of data sparsity?
- **RQ3**: How do different hyper-parameter settings, *i.e.*, two transfer weights and the dimension of latent feature vectors, affect the performance of our method?

5.1 Experimental Setting

5.1.1 Metrics. In order to compare the recommendation performance between our method and other baselines, we adopt three prevalent metrics, *i.e.*, *TopK Hitrate*, *TopK Accuracy*, and $nDCG_K$, to evaluate the accuracy of recommendation results.

TopK Hitrate: This metric measures the percentage of users whose Top-K locations are successfully predicted (correct for at least one location) in the test set, which is commonly used since the recommendation system usually recommends a list of items to expect users click at least one of them. The computation is as follows:

$$TopK\ Hitrate = \frac{\sum_{i=1}^{N}(|L_i^{pred} \cap L_i^{test}| \geqslant 1)}{N}, \tag{3}$$

where L_i^{pred} denotes the set of predicted $TopK$ locations and L_i^{test} denotes the most frequently visited K locations by the user among locations in the testing set, for each user $u_i \in \mathcal{U}$.

TopK Accuracy: This is a metric that measures the mean prediction accuracy on $TopK$ prediction of all users, which can be expressed as follows:

$$TopK\ Accuracy = \frac{\sum_{i=1}^{N}(|L_i^{pred} \cap L_i^{test}|/K)}{N}. \tag{4}$$

$nDCG_K$: This metric is a common measure of ranking quality which can measure the effectiveness of our location recommendation algorithm. It can be expressed as follows:

$$nDCG_K = \frac{DCG_K}{IDCG_K} = \sum_{i=1}^{N} \frac{\sum_{j=1}^{K} rel_j^{pred(i)}/log_2(j+1)}{\sum_{j=1}^{K} rel_j^{test(i)}/log_2(j+1)}/N, \tag{5}$$

where $rel_j^{pred(i)}$ denotes the relevance (the normalized usage frequency) of the j-th predicted app and $rel_j^{test(i)}$ denotes the relevance (the normalized usage frequency) of the j-th app in the testing set, for each user $u_i \in \mathcal{U}$.

Taking $TopK$ recommended locations into consideration, these three metrics can measure whether we recommend effective locations and how accurate our recommendation is. Therefore, they are enough to reflect the performance of a recommendation system.

5.1.2 Baselines. In order to investigate the performance of our model, we compare it with other seven state-of-art algorithms.

SVD [17]: In recommendation systems, SVD is used as a collaborative filtering (CF) algorithm, which predicts an item pair rating for a user based on the history of ratings given to the items by the user. Here the item refers to location. With no prior information, SVD only utilizes the user-location matrix.

MF [32]: MF also only utilizes the user-location matrix without any external information. Here we choose a popular low-rank factorization method to complete such a typical collaborative filtering task. MF is equivalent to our approach in the case that we set $\beta_1 = 0$ and $\beta_2 = 0$. We consider this baseline to show that with such sparse user location visit data, the quality of personalized location recommendation is poor if we do not transfer any information from other resources.

CMF-U: Considering external information from user side, CMF-U utilizes both user-location matrix and user-app matrix to do collaborative matrix factorization. CMF-U is equivalent to our approach with $\beta_2 = 0$.

CMF-L [36]: Considering external information from item side, CMF utilizes both user-location matrix and location-app matrix to do collaborative matrix factorization. CMF-L is equivalent to our approach with $\beta_1 = 0$.

KNN: Based on user-app matrix $\mathbf{Y}$, for each user, KNN first finds the nearest K neighbor users, then predicts the visited locations in the testing set according to these neighbors' location visiting behaviors. Assume the app usage similarity between user u_i and his K neighbors $\mathcal{U}_i^K$, denoted as $\{S_{ij}\}, \forall u_j \in \mathcal{U}_i^K$, then the location predication of user u_i can be expressed as follows:

$$\hat{\mathbf{X}}_i = \sum_j S_{ij}\mathbf{X}_j / \sum_j S_{ij}, \forall u_j \in \mathcal{U}_i^K. \tag{6}$$

SoRec [24]: Besides the user-location matrix, SoRec also introduces a user-user matrix to do collaborative matrix factorization. Originally, the user-user matrix is filled by social friendship information (whether friend or not). Here we replace it with the cosine similarity of app usage information between pairwise users. Note that for cold-start location recommendation, we replace user-user matrix with location-location matrix.

SR [25]: Compared with MF, it integrates social information to via a social regularization term which can limit the distance in latent space of users' embedding vectors with their friends. Again, we utilize the similarity of app usage to reflect the social relationship between users and the weight of the social regularization term is the cosine similarity of app usage between users.

5.1.3 Parameter Setting. In our method, we have the following hyper-parameters: dimension of latent feature vector H, weight for user-app matrix β_1, weight for location-app matrix β_2, regularization coefficients $[\frac{\lambda_u}{2}, \frac{\lambda_l}{2}, \frac{\lambda_a}{2}]$, learning rate η, and maximum iteration times T.

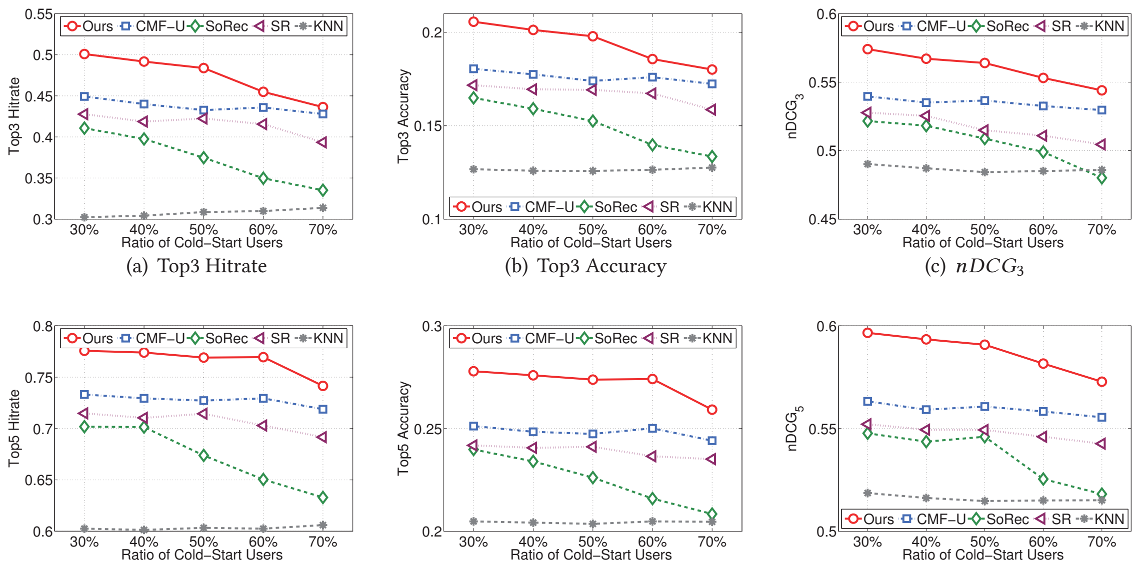
For the system based on Telecom dataset, to determine the dimension of latent feature vector, we experiment with a sequence of settings ranging from 5 to 50 and empirically select $H = 20$ as our default value. Likewise, we empirically set $\beta_1 = 0.7, \beta_2 = 0.07$. As for the other parameters, we set $[\frac{\lambda_u}{2} = 0.1, \frac{\lambda_l}{2} = 0.1, \frac{\lambda_a}{2} = 0.1]$, $\eta = 0.01$ and $T = 600$. In order to keep consistency and guarantee the comparability of results, we set the same dimension of latent feature vector, regularization coefficients, learning rate and maximum iteration times for our baselines. In addition, for SR, we set the weight for the social regularization term $\frac{\alpha_u}{2}$ to 0.1, and for KNN, we set the number of nearest neighbors $K = 20$.

Likewise, for the system based on TalkingData dataset, we set $H = 10$, $\beta_1 = 0.2$, $\beta_2 = 0.03$, $\eta = 0.8$ and $T = 200$. Besides, values of $[\frac{\lambda_u}{2}, \frac{\lambda_l}{2}, \frac{\lambda_a}{2}]$, and $\frac{\alpha_u}{2}$ for SR are the same with the system based on Telecom dataset. In addition, for KNN, we set the number of nearest neighbors $K = 5$.

5.2 Cold-start Problem Solving(RQ1)

In our targets, there are two types of cold-start problems. In these two scenarios, only the user-location matrix is not enough, thus our two baselines SVD and MF fail to work. To solve cold-start problems, we must transfer knowledge from either user side (CMF-U) or item side (CMF-L) external information. Moreover, our model (Ours) utilizes both side external information to solve cold-start problems.

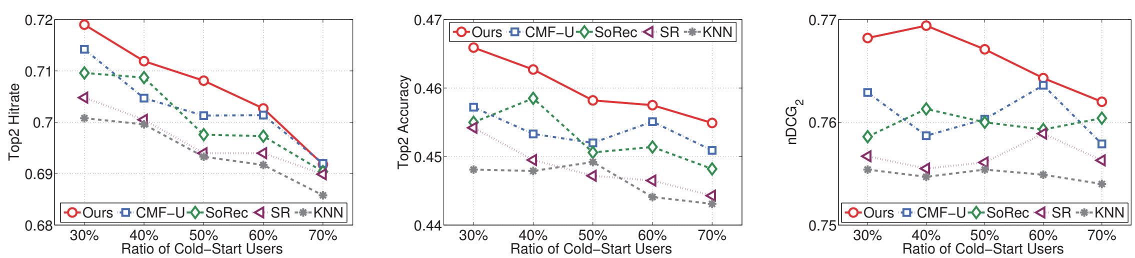
To investigate the performance in user cold-start scenario, we randomly sample some cold-start users by hiding all their location visit records and try to recommend locations to them by utilizing location visit records of the rest of users plus extra information from user-app matrix and location-app matrix. To be specific, we split the training set and test set as follows: Firstly, we hide values of some rows in user-location matrix $\mathbf{X}$ by random sampling, $i.e.$, the relevant users without any location records are regarded as cold-start users to form the test set. It means other users are regarded as the training users. Secondly, we input matrices of $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ into our system, then predict location preferences of the test users and evaluate the effectiveness of location recommendation to different proportions of cold-start users. The performance comparisons are shown in Figure 4 and Figure 5. From the results, we can observe that our method achieves higher hitrate and accuracy than other baselines in both datasets. For example, in Figure 4, considering the top 3 locations, with 30% cold-start users, we achieve 50.1% hitrate, which is 11.1% higher than baselines. With 60% cold-start users, $Top5\ Hitrate$ of our method reaches 77.0%, which is 5.5% higher than CMF-U, 6.6% higher than SR, 18.3% higher than SoRec, 27.7% higher than KNN. Moreover, $nDCG_5$ of our method is 58.1%, which is 4.9% higher than CMF-U, 6.4% higher than SR, 9.3% higher than SoRec, and 12.9% higher than KNN. In addition, even with 70% cold-start users, the hitrate and accuracy of recommending $Top3$ or $Top5$ locations still outperform other four baselines. The TalkingData dataset in Figure 5 also shows similar results. According to these results, we can infer that user app usage information is more beneficial for learning user interest so as to recommend locations to these cold-start users. In addition, our method, a combined utilization of user and location app usage information, can further improve the effectiveness of location recommendation greatly for cold-start users.



(a) Top3 Hitrate (b) Top3 Accuracy (c) $nDCG_3$

(d) Top5 Hitrate (e) Top5 Accuracy (f) $nDCG_5$

Fig. 4. The performance comparison between our method and two baselines for user cold-start problem on *Telecom* dataset.

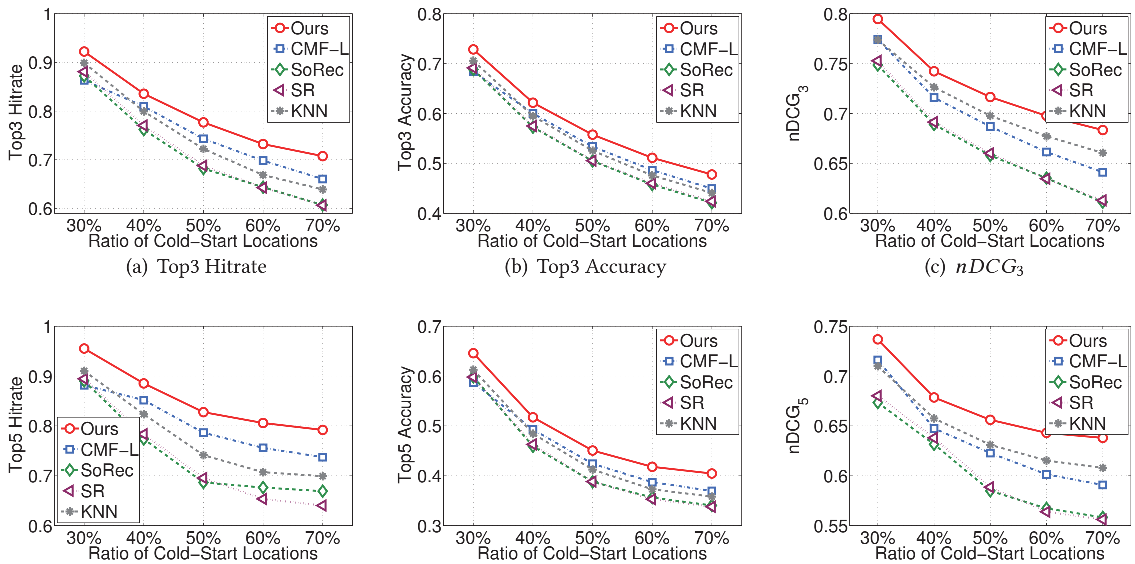


(a) Top2 Hitrate (b) Top2 Accuracy (c) $nDCG_2$

Fig. 5. The performance comparison between our method and two baselines for user cold-start problem on *TalkingData* dataset.

Likewise, in the location cold-start scenario, we also obtain some cold-start locations by hiding all their user interactions and recommend users to them. We show the performance of our method and baselines in Figure 6 and Figure 7. Obviously, our method has better performance than CMF-L, SR and SoRec and KNN. Take the Telecom dataset in Figure 6 for example, with 30% cold-start locations, $Top5\ Hitrate$ can reach 95.5% when recommending users to them, which is 5.0% and 6.8% higher than KNN and SR, respectively. Even with 70% cold-start locations, our model can also provide 13.5% improvement of $Top3\ Accuracy$ compared to SoRec. In addition, the $Top3\ Hitrate$ can also reach over 90%, which means that for those locations without user information, users recommended by our model are likely to visit the locations. So as the TalkingData dataset in Figure 7. Notice that we measure the accuracy of recommending Top2 locations because users have less visited locations in the dataset and predicting Top2 locations is meaningful. The performance comparison based on two datasets shows obviously that our method can effectively solve location cold-start problem, for better understanding of user interest and cold-start locations' characteristics from app usage information.

In a short summary, our approach performs very well in both user and location cold-start problems. Considering 60% cold-start users, $Top5\ Hitrate$ of our model is 77.0%, at least 9.6% higher than other baselines. In addition, for 30% cold-start locations, we can achieve 95.5% hitrate when recommending $Top5$ users to them.



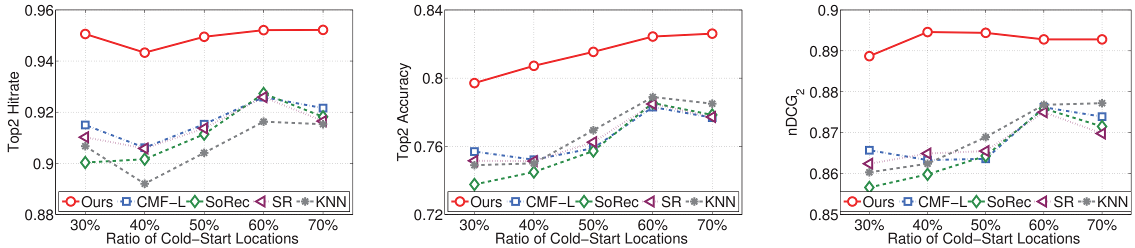
(a) Top3 Hitrate (b) Top3 Accuracy (c) $nDCG_3$

(d) Top5 Hitrate (e) Top5 Accuracy (f) $nDCG_5$

Fig. 6. The performance comparison between our method and two baselines for location cold-start problem on *Telecom* dataset.

5.3 Performance in Varying Data Sparsity(RQ2)

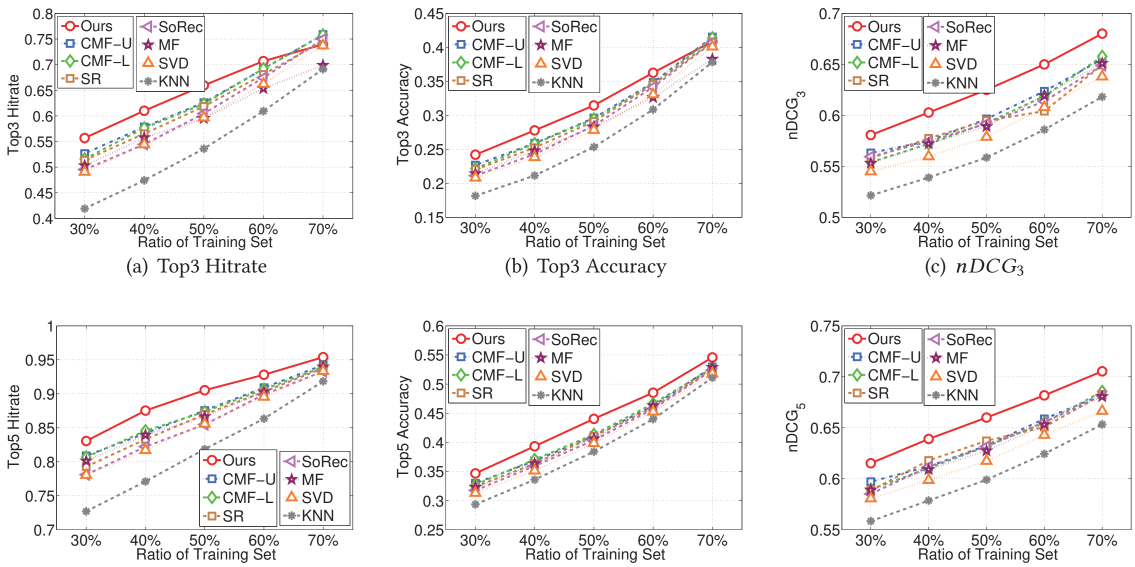
The above analyses have shown our significant advantages in solving the cold-start problem. Now, we investigate how our method performs under different data sparsity. In order to split training and test set, we randomly select a part of locations for each user and regard visit behaviors among these locations as the training set, then assume that the rest locations are unknown so as to form the test set. In addition, to simulate different levels of data sparsity, we extract different percentages of known locations from each user to form the training set. Thus, we select five different ratios of training set: 30%, 40%, 50%, 60% and 70%. We adopt six metrics, *i.e.*, $Top3\ Hitrate$,



(a) Top2 Hitrate (b) Top2 Accuracy (c) $nDCG_2$

Fig. 7. The performance comparison between our method and two baselines for location cold-start problem on *TalkingData* dataset.

*Top*3 *Accuracy*, *Top*3 *nDCG*, *Top*5 *Hitrate*, *Top*5 *Accuracy*, *Top*5 *nDCG* for Telecom dataset and three metrics, *i.e.*, *Top*2 *Hitrate*, *Top*2 *Accuracy*, *Top*2 *nDCG* for TalkingData dataset to evaluate location recommendation accuracy. Based on our two real-world datasets, we compare their performances and show the results in Figure 8 and Figure 9.

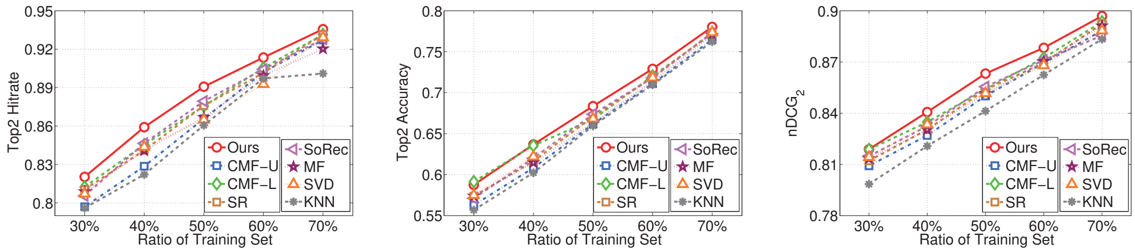


(a) Top3 Hitrate (b) Top3 Accuracy (c) $nDCG_3$

(d) Top5 Hitrate (e) Top5 Accuracy (f) $nDCG_5$

Fig. 8. The performance comparison between our method and several baselines under different data sparsity levels on *Telecom* dataset.

Firstly, let's look at the results of the Telecom dataset. From Figure 8 (a), we can observe that our method (Ours) outperforms the other baselines under different sparsity levels. By taking external information of user side or item side into consideration, CMF-U and CMF-L perform better than MF and SVD, indicating both user-app matrix and location-app matrix are useful for personalized location recommendation. In addition, SR and SoRec utilize user similarity information directly while CMF-U and CMF-L still outperform them, which shows that



(a) Top2 Hitrate (b) Top2 Accuracy (c) $nDCG_2$

Fig. 9. The performance comparison between our method and several baselines under different data sparsity on *TalkingData* dataset.

the method of CMF is more effective than considering user correlation directly in collaborative filtering or regularization term. Moreover, these above methods generally perform better than KNN, which shows that collaborative matix factorization is more effective than just considering several nearest neighbors. In terms of our designed method, with 30% to 50% training data, Our $Top3\ Hitrate$ is 9% to 13% higher than SVD and MF, and 5% to 6% higher than CMF-U and CMF-L. It shows adding user-app and location-app data information simultaneously provides progressive performance improvement. So as the $Top3\ Accuracy$. Under different levels of data sparsity, our recommendation is more accurate than others. Similar results have been shown in Figure 8 (d) and (e), which measure those methods' performances by $Top5\ Hitrate$ and $Top5\ Accuracy$. With 50% training data, the $Top5\ Hitrate$ of our approach is over 90%, which is about 3.4% higher than CMF-U, 4.1% higher than SR, 5.8% higher than SVD, 10.6% higher than KNN. As for $nDCG_5$, our method still outperforms other baselines.

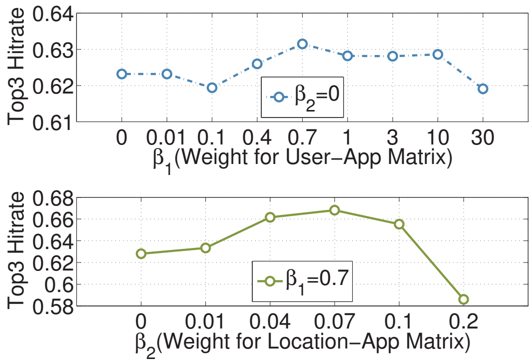
We also show the performance under the TalkingData dataset in Figure 9. The results demonstrate that our solution performs better than the other baselines in all cases. For example, when utilizing 50% training data, our method achieves 89.1% $Top2\ Hitrate$, which is 1.7% higher than CMF-L, 3.0% higher than MF, and 3.5% higher than KNN.

In summary, results on two datasets both demonstrate that our designed recommendation system outperforms the other baselines under different sparsity levels. With 30% training data, $Top3\ Hitrate$ of our method is 55.7%, providing 5.7% and 10.8% improvements compared to CMF-U and MF respectively. Moreover, our method has a 83.1% hitrate in predicting Top5 apps. All these results show that adding user-app and location-app matrices provides progressive performance improvement, especially in the sparse data.

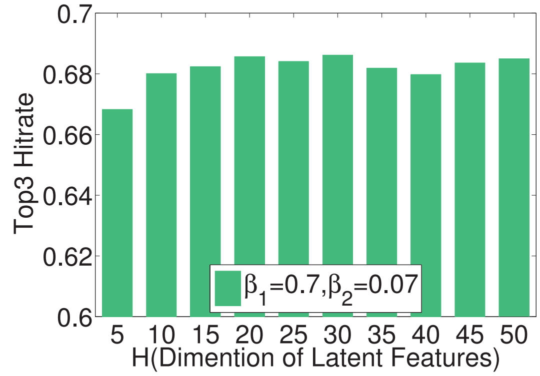
5.4 Hyper-parameter Impact(RQ3)

In this section, we measure the impact of different hyper-parameter settings and evaluate the impact of properties of user-app and location-app on the location prediction accuracy. More specifically, we will investigate the performance of our method when setting two transfer weights and the dimension of latent feature vector with different values. The results are shown in Figure 10 and Figure 11.

Firstly, we evaluate the impact of the weight for user-app matrix β_1 and the weight for location-app matrix β_2 separately. We set the ratio of training set to be 50% and keep other parameters the same. The results are shown in Figure 10 (a) and Figure 11 (a), for Telecom dataset and TalkingData dataset respectively. now let's look at Figure 10 (a) based on Telecom dataset. From the results we can observe that $Top3\ Hitrate$ grows at the beginning then decreases when β_1 gradually increases from 0 to 30. It is because when β_1 is very small, the model cannot fully utilize user-app information to capture user interests and the relationship among different users. When β_1 becomes large, the user-app information dominates the object function, thus overwhelming the location visit information from user-location matrix. With $\beta_1 = 0.7$, the system finds a balance and achieves

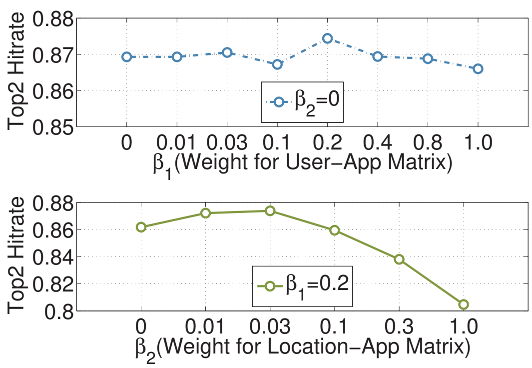


(a) β_1, β_2

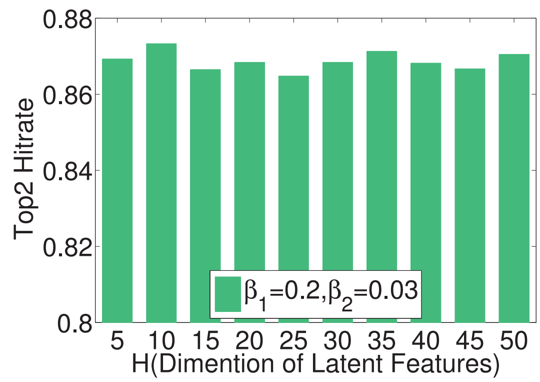


(b) H

Fig. 10. (a) The impact of transfer weights on the location recommendation accuracy, based on *Telecom* dataset. (b) The impact of the dimension of latent feature vector on the location recommendation accuracy, based on *Telecom* dataset.



(a) β_1, β_2



(b) H

Fig. 11. (a) The impact of transfer weights on the location recommendation accuracy, based on *TalkingData* dataset. (b) The impact of the dimension of latent feature vector on the location recommendation accuracy, based on *TalkingData* dataset.

the best performance, with a 63.2% $Top3\ Hitrate$, which provides 3.2% improvement than the case that $\beta_1 = 0$. So as β_2, the weight for location-app matrix. We find the most suitable value of β_2 is 0.07, in this case, the best $Top3\ Hitrate$ are 66.8%, which provides 6.4% improvement than the case that $\beta_2 = 0$ with $\beta_1 = 0.7$. We can also find the same trend of β_1 and β_2 in Figure 11 (a) which is based on TalkingData dataset. Similarly, with $\beta_1 = 0.2$ and $\beta_2 = 0.03$, the system achieves the best performance. These results validate our intuition that user-app and location-app correlation information are useful thus making full use of them can benefit personalized location recommendation.

Secondly, we evaluate the impact of the dimension of latent feature vector H. We change it from 5 to 50, and the results are shown in Figure 10 (b) and Figure 11 (b), for Telecom dataset and TalkingData dataset respectively. When $H = 20$ for Telecom dataset and $H = 10$ for TalkingData dataset, their corresponding systems perform the best, but the performance does not change too much indeed, showing our model is very robust and performs equally well under various H values. Thus we set $H = 20$ for Telecom dataset and $H = 10$ for TalkingData dataset in our evaluation.

In conclusion, our proposed personalized location recommendation system is very robust and outperforms other state-of-art algorithms in different cold-start and data sparsity settings, indicating that the introduction of app usage data is very beneficial, and making full use of them can greatly improve the performance of cold-start location recommendation.

6 DISCUSSION AND RELATED WORK

In this study, we make full use of individual app usage data to infer user interest and extract location features, then further recommend locations to cold-start users or help cold-start locations target potential visitors, *i.e.*, solving cold-start location recommendation problem. Indeed, the considered scenario is practical in our daily life, for people now highly rely on recommendation services of mobile applications, in order to reduce the cost of information acquisition in such an information explosion era. Many mobile applications, such as Yelp and MaFengWo [30], provide users personalized location recommendations (*e.g.*, restaurants and attractions) mainly based on their collected online activity data, (*e.g.*, user app usage fingerprints), with the user agreement [1, 5, 11, 16]. Therefore, the need for utilizing online fingerprints to predict offline footprints does exist and application vendors usually own the required data base. Under these conditions, our study is very meaningful.

Our idea of transferring online app usage information to help predict offline location visitation behaviors, is actually a first successful attempt of "from online fingerprints to offline footprints" application. Nowadays, people use mobile applications everyday and everywhere, thus many offline activities are recorded by online fingerprints. Online behaviors, such as mobile app usage, have rich and diverse contents and can absolutely reflect user interests and characteristics [8, 20, 37, 46], therefore, posing the possibility of predicting user offline behaviors. Quantitative analysis in this study has verified the strong correlation of online app usage and offline location visiting patterns. Moreover, our performance evaluation, based on both a large-scale dataset and a public dataset, has shown the generalization of domain knowledge transformation from online fingerprints to offline footprints.

More generally, our work reveals the huge benefits of transfer learning applications by combining user's online and offline behaviors. Since the diverse activities in smart phones make users generate a great deal of online behavior data, such as app usage information and web browsing history [6, 7, 13]. Many studies have shown that these fine-grained and informative data can capture user's habits, preferences and requirements [2, 26, 31, 37, 46]. While for offline behaviors, many location based services [22, 30, 39, 51], like trajectory prediction and location recommendation, require precise user portraits and suffer from limited information about user behavior. Therefore, it is essential and effective to boost offline prediction and recommendation applications by learning knowledge from online user behaviors. Our work takes a first step forward to promote a better understanding of the intrinsic correlation between individual's online and offline behaviors and the benefits of combining them. This is an important research area of ubiquitous computing and a vibrant research topic in the UbiComp community and beyond. In addition, during the past few years, many works [2, 19, 37, 42, 45, 46] in UbiComp have realized the importance and great value of app usage data and conducted different studies to explore its applications. Our work, is also a follow-up to show the possibility of utilizing app usage data to help personalized location recommendation, which poses a brand-new angle of applying app data into practice.

6.1 Location Recommendation and Prediction

Location recommendation systems have been widely used and a wide range of approaches have been proposed, which are usually achieved by using additional information of time, activities, etc. [15, 21, 22, 50, 51]. Zheng et al. and Karatzoglou et al. presented collaborative filtering based recommendation algorithm and used a large-scale user data pool to collaboratively filter the like-minded users at different locations or activities [15, 50]. Zhu et al. [51] focused on the problem of insufficient information from individual users by learning the common location preference of many users. Kostakos et al. [18] applied a Markov state transition model to predict next screen event. Zhao et al. [47] proposed a spatial-temporal latent ranking method to explicitly model the interactions among user, POI, and time. Liu et al. [22] proposed a neural network solution for location recommendation. Other works also concern about the problem of location prediction, which aims to predict the future visited locations based on the historical data. Considering the mobility similarity between user group, Zhang et al. [43] proposed GMove to share significant movement regularity among users. Moreover, pattern-based methods [10, 27, 41] were also utilized to predict the mobility based on these popular patterns.

Existing works mainly utilize spatiotemporal information to discover similar users and find some common interested locations to recommend. However, due to data sparsity problem, it is very hard to find actually similar users only based on the mobility trajectories. In this paper, we consider utilizing a new data source – app usage information to help location recommendation. App usage behavior can directly reflect user interest and preference, therefore, it is more promising and effective to filter similar users and discover potential locations to recommend, which makes up for the sparsity of location visiting data.

6.2 Cold-start Problems

To solve the cold-start problem, the common practice is to find addition datasets to obtain necessary information for the cold-start users/items without any records in the target domain. In this paper, we focus on the cold-start problem of location recommendation. Recently, many studies concern about this issue. Xie et al. [38] proposed a generic graph-based embedding model, taking sequential, geographical, temporal and semantic factors into consideration. Gao et al. [9] addressed the cold-start location recommendation problem by utilizing both social and geographical relationships among users. Long and Joshi [23] proposed a HITS-based POI recommendation algorithm to recommend POIs to LBSN users considering their social relationships. [34] et al. utilized basic demographic data (gender, age, location) or social network information (Facebook friends or page likes) to solve cold-start problems. In addition, [25] integrated social information to recommender system via a social regularization term which can limit the distance in latent space of users' embedding vectors with their friends when performing the recommendation task.

Note that many existing works require user profiles or social relationship of users, compared with these works, our study first introduces online app usage data to learn the "interest relationship" between different users and verifies its effectiveness to boost cold-start location recommendation. More importantly, with app usage data, we are able to learn both user's interest and location's features and solve user and location cold-start recommendation problem together.

6.3 App Usage Modelling

Recent works have studied how users use mobile apps by focusing on three aspects: user interactions, network traffic, and energy drain [6, 7, 13]. Church et al. [4] summarized the challenges for mobile phone usage learning and analysis, as well as a series of studies and applications on mobile phone usage. Falaki et al. [6] discovered immense diversity usage activities among users. Another related work [2, 26, 31, 37] reveals that users can be identified through the sets of apps they use. Other studies cluster mobile users according to their app usage records[46]. Moreover, users' mobility patterns can impact the way that the apps are used [49]. Context such as location and time are shown to have an impact on app usage [12, 35]. A multi-faceted approach to predict app usage is developed in [40].

Most studies focus on discovering app usage patterns to boost the understanding of users. While in our paper, we make full use of app data to serve as an additional but important information to help solve the cold-start problem of location recommendation. On the one hand, app usage information can reflect user interest so as to measure user similarity, which can solve cold-start recommendation problem of users with no location records. On the other hand, app usage information can reflect the functional attributes of locations to some degree, which can help the recommendation of cold-start locations. To the best of our knowledge, it is the first work to show the possibility of utilizing app usage data to help personalized location recommendation, which poses a brand-new angle of applying app data into practice.

6.4 Limitations

Our work has some limitations. First, app usage data has not been published extensively, which throws the doubt on its accessibility and seems to limit the utility of our proposed method. However, it is encouraging that the situation now begins to change. Recently, some mobile app usage datasets [14, 29] have been published. In addition, a study [42] shows it is possible to utilize POI information to infer which types of apps users use given a particular location. That means, even the app usage dataset is not available, we can still utilize POI information to infer app usage information. Moreover, as we mentioned before, in industry, many mobile operators and application vendors [2, 37, 42, 46] do have such online behavior data to learn user interest and provide personalized recommendation services with the user agreement. Therefore, we believe the situation will be gradually improved and our study is meaningful and prospective. Second, our recommendation is static rather than a dynamic one, since we don't consider the temporal factor in our model. Indeed, this work takes the first step to show the benefits that app usage information can bring to location recommendation. And we leave further explorations of time-variant personalized location recommendation to future work.

7 CONCLUSION

In this paper, we demonstrate the feasibility of making personalized location recommendation by transferring app usage information, especially in solving the cold-start problem. Accordingly, we propose a generative model to transfer knowledge from app usage behaviors into location visiting preference. Based on two real-word mobile app usage datasets, we evaluate the performance of our method and find it outperforms the other four state-of-the-art algorithms in both user and location cold-start problems. Moreover, our method also shows a great performance under different levels of data sparsity, indicating our method's effectiveness and robustness. Our study is the first step forward for transferring user interests learning from online fingerprints to offline footprints, which paves the way for providing better personalized location recommendation services for mobile users.

APPENDIX: THE GENERATIVE MODEL

We define the conditional distribution over the user-location matrix $\mathbf{X}$, user-app matrix $\mathbf{Y}$, and location-app matrix $\mathbf{Z}$ as follows:

$$\begin{aligned}&\rho(\mathbf{X},\mathbf{Y},\mathbf{Z}|\mathbf{U},\mathbf{L},\mathbf{A},\sigma_1^2,\sigma_2^2,\sigma_3^2)=\rho(\mathbf{X}|\mathbf{U},\mathbf{L},\sigma_1^2)\rho(\mathbf{Y}|\mathbf{U},\mathbf{A},\sigma_2^2)\rho(\mathbf{Z}|\mathbf{L},\mathbf{A},\sigma_3^2)\\&=\prod N(x_{i,j}|g((\mathbf{u}_i)^\top\mathbf{l}_j),\sigma_1^2)\times\prod N(y_{i,k}|g((\mathbf{u}_i)^\top\mathbf{a}_k),\sigma_2^2)\times\prod N(z_{j,k}|g((\mathbf{l}_j)^\top\mathbf{a}_k),\sigma_3^2)\ ,\end{aligned}\tag{7}$$

where we consider the most common Gaussian distribution and $N(\cdot|\mu,\sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2. The function $g(x)$ is the logistic function $1/(1+exp(-x))$ to bound the range within $[0,1]$ interval, since our data are composed of implicit feedbacks. From the conditional distribution above, we can observe that the latent feature vectors of users, locations and apps are shared in user-location domain, user-app domain and location-app domain. We also place spherical Gaussian priors on

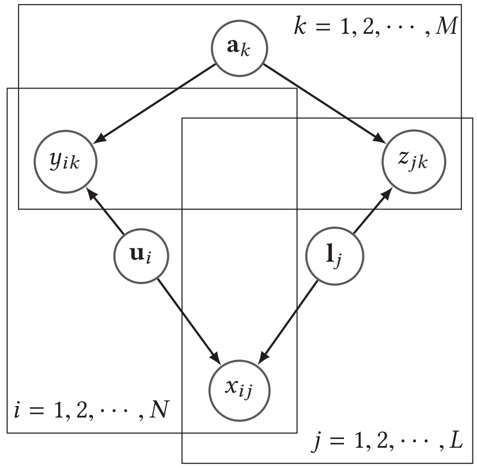


Fig. 12. The graphical representation of our generative model.

user, location and app feature vectors, which are defined as follows:

$$\rho(\mathbf{U}|\sigma_u^2) = \prod_{i=1}^{N} N(\mathbf{u}_i|0, \sigma_u^2 I),\ \rho(\mathbf{L}|\sigma_l^2) = \prod_{j=1}^{L} N(\mathbf{l}_j|0, \sigma_l^2 I),\ \rho(\mathbf{A}|\sigma_a^2) = \prod_{k=1}^{M} N(\mathbf{a}_k|0, \sigma_a^2 I). \tag{8}$$

The graphical representation of our model is illustrated in Figure 12. Its generative process runs as follows:

- For each user i, draw the vector as $\mathbf{u}_i \sim N(0, \sigma_u^2 I)$;
- For each location j, draw the vector as $\mathbf{l}_j \sim N(0, \sigma_l^2 I)$;
- For each app k, draw the vector as $\mathbf{a}_k \sim N(0, \sigma_a^2 I)$;
- For each user-location pair (i, j), draw the value $x_{i,j} \sim N\left(g((\mathbf{u}_i)^\top \mathbf{l}_j), \sigma_1^2\right)$;
- For each user-app pair (i, k), draw the value $y_{i,k} \sim N\left(g((\mathbf{u}_i)^\top \mathbf{a}_k), \sigma_2^2\right)$;
- For each location-app pair (j, k), draw the value $z_{j,k} \sim N\left(g((\mathbf{l}_j)^\top \mathbf{a}_k^2), \sigma_3^2\right)$.

Through Bayesian inference, the posterior probability of the latent feature vector sets U, L and A can be obtained as follows:

$$\begin{aligned}
&\rho(\mathbf{U}, \mathbf{L}, \mathbf{A}|\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_u^2, \sigma_a^2, \sigma_k^2) \propto \rho(\mathbf{X}, \mathbf{Y}, \mathbf{Z}|\mathbf{U}, \mathbf{L}, \mathbf{A}, \sigma_1^2, \sigma_2^2, \sigma_3^2)\rho(U|\sigma_u^2)\rho(L|\sigma_l^2)\rho(A|\sigma_a^2) \\
&= \prod N(x_{i,j}|g((\mathbf{u}_i)^\top \mathbf{l}_j), \sigma_1^2) \prod N(y_{i,k}|g((\mathbf{u}_i)^\top \mathbf{a}_k), \sigma_2^2) \times \prod N(z_{j,k}|g((\mathbf{l}_j)^\top \mathbf{a}_k), \sigma_3^2) \\
&\times \prod_{i=1}^{N} N(\mathbf{u}_i|0, \sigma_u^2 I) \prod_{j=1}^{L} N(\mathbf{l}_j|0, \sigma_l^2 I) \prod_{k=1}^{M} N(\mathbf{a}_k|0, \sigma_w^2 I).
\end{aligned} \tag{9}$$

The log of posterior distribution over the user, app and word latent feature vector is calculated as follows:

$$\begin{aligned}
&\ln \rho(\mathbf{U}, \mathbf{L}, \mathbf{A} | \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_u^2, \sigma_l^2, \sigma_a^2) \\
&= -\frac{1}{2\sigma_1^2} \sum_{i,j} \left[x_{i,j} - g((\mathbf{u}_i)^\top \mathbf{l}_j)\right]^2 - \frac{1}{2\sigma_2^2} \sum_{i,k} \left[y_{i,k} - g((\mathbf{u}_i)^\top \mathbf{a}_k)\right]^2 - \frac{1}{2\sigma_3^2} \sum_{j,k} \left[z_{j,k} - g((\mathbf{l}_j)^\top \mathbf{a}_k)\right]^2 \\
&- \frac{1}{2\sigma_u^2} \sum_{i=1}^{N} \|\mathbf{u}_i\|_2^2 - \frac{1}{2\sigma_a^2} \sum_{j=1}^{L} \|\mathbf{l}_j\|_2^2 - \frac{1}{2\sigma_a^2} \sum_{k=1}^{M} \|\mathbf{a}_k\|_2^2 \\
&- \frac{1}{2}(NL \ln \sigma_1^2 + NM \ln \sigma_2^2 + LM \ln \sigma_3^2) - \frac{1}{2} H(N \ln \sigma_u^2 + L \ln \sigma_l^2 + M \ln \sigma_a^2) + C,
\end{aligned} \tag{10}$$

where C is a constant that does not depend on the parameters. $\| \cdot \|_F^2$ denotes the Frobenius norm. Keeping the parameters, $i.e.$, observation noise variance and prior variance, fixed, maximizing the log-posterior over the latent feature of users, apps and locations is equivalent to minimizing the following objective function, which is a sum of squared errors with quadratic regularization terms:

$$\begin{aligned}
\zeta(\mathbf{U}, \mathbf{L}, \mathbf{A}) = \frac{1}{2} ||I_X \circ \left(\mathbf{X} - g\left(\mathbf{U}^\top \mathbf{L}\right)\right)||_F^2 + \frac{\beta 1}{2} ||I_Y \circ \left(\mathbf{Y} - g\left(\mathbf{U}^\top \mathbf{A}\right)\right)||_F^2 + \frac{\beta 2}{2} ||I_Z \circ \left(\mathbf{Z} - g\left(\mathbf{L}^\top \mathbf{A}\right)\right)||_F^2 \\
+ \left(\frac{\lambda_u}{2} ||\mathbf{U}||_F^2 + \frac{\lambda_l}{2} ||\mathbf{L}||_F^2 + \frac{\lambda_a}{2} ||\mathbf{A}||_F^2\right),
\end{aligned} \tag{11}$$

where $\circ$ means the point-wise matrix multiplication. I_X, I_Y, I_Z are denoted as flag matrices for user-location data, user-app data and location-app data respectively. If record of user i and location j is known, then $I_X(i, j) = 1$, otherwise $I_X(i, j) = 0$. I_Y and I_Z are defined in the similar ways. β_1 is the weight of user-app data we use for transfer learning, β_2 means the weight of location-app data. Specifically, $\beta_1 = \sigma_1^2/\sigma_2^2, \beta_2 = \sigma_1^2/\sigma_3^2$ and $\lambda_u = \sigma_1^2/\sigma_u^2$, $\lambda_l = \sigma_1^2/\sigma_l^2$, $\lambda_a = \sigma_1^2/\sigma_a^2$.

Then, we perform gradient descent on $\mathbf{u}_i, \mathbf{a}_j, \mathbf{L}_k$ for all users, apps and locations to get a local minimum of the objective function. The formulas run as follows:

$$\begin{aligned}
\frac{\partial \zeta}{\partial \mathbf{u}_i} &= \sum_j \left[g((\mathbf{u}_i)^\top \mathbf{l}_j) - x_{i,j}\right] g'((\mathbf{u}_i)^\top \mathbf{l}_j)\mathbf{l}_j + \beta 1 \sum_k \left[g((\mathbf{u}_i)^\top \mathbf{a}_k) - y_{i,k}\right] g'((\mathbf{u}_i)^\top \mathbf{a}_k)\mathbf{a}_k + \lambda_u \mathbf{u}_i; \\
\frac{\partial \zeta}{\partial \mathbf{l}_j} &= \sum_i \left[g((\mathbf{u}_i)^\top \mathbf{l}_j) - x_{i,j}\right] g'((\mathbf{u}_i)^\top \mathbf{l}_j)\mathbf{u}_i + \beta 2 \sum_k \left[g((\mathbf{l}_j)^\top \mathbf{a}_k) - y_{i,k}\right] g'((\mathbf{l}_j)^\top \mathbf{a}_k)\mathbf{a}_k + \lambda_l \mathbf{l}_j; \\
\frac{\partial \zeta}{\partial \mathbf{a}_k} &= \beta 1 \sum_i \left[g((\mathbf{u}_i)^\top \mathbf{a}_k) - y_{i,k}\right] g'((\mathbf{u}_i)^\top \mathbf{a}_k)\mathbf{u}_i + \beta 2 \sum_j \left[g((\mathbf{l}_j)^\top \mathbf{a}_k) - z_{j,k}\right] g'((\mathbf{l}_j)^\top \mathbf{a}_k)\mathbf{l}_j + \lambda_a \mathbf{a}_k;
\end{aligned} \tag{12}$$

where $g'(x)$ is the derivative of the logistic function and $g'(x) = exp(-x)/(1 + exp(-x))^2$.

REFERENCES

[1] Apple. 2018. *Licensed Application End User License Agreement.* https://www.apple.com/legal/internet-services/itunes/dev/stdeula/.

[2] Konrad Blaszkiewicz, Konrad Blaszkiewicz, Konrad Blaszkiewicz, and Alexander Markowetz. 2016. Differentiating smartphone users by app usage. In *Proc. ACM Ubicomp.* 519–523.

[3] Sung-Hyuk Cha. 2007. Comprehensive survey on distance/similarity measures between probability density functions. *City* 1, 2 (2007), 1.

[4] Karen Church, Denzil Ferreira, Nikola Banovic, and Kent Lyons. 2015. Understanding the challenges of mobile phone usage data. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services.* ACM, 504–514.

[5] Nield David. 2017. *All the Ways Your Smartphone and Its Apps Can Track You.* https://gizmodo.com/all-the-ways-your-smartphone-and-its-apps-can-track-you-1821213704.

[6] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. 2010. Diversity in smartphone usage. In *Proc. ACM MobiSys.* 179–194.

[7] Denzil Ferreira, Jorge Goncalves, Vassilis Kostakos, Louise Barkhuus, and Anind K Dey. 2014. Contextual experience sampling of mobile application micro-usage. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices and services*. ACM, 91–100.

[8] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, and Tat-Seng Chua. 2019. Neural Multi-Task Recommendation from Multi-Behavior Data. In *ICDE*.

[9] Huiji Gao, Jiliang Tang, and Huan Liu. 2015. Addressing the cold-start problem in location recommendation using geo-social correlations. *Data Mining and Knowledge Discovery* 29, 2 (2015), 299–323.

[10] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. 2007. Trajectory Pattern Mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*. 330–339.

[11] Google. 2018. *Google's Privacy Policies*. https://policies.google.com/privacy#infosharing

[12] Ke Huang, Chunhui Zhang, Xiaoxiao Ma, and Guanling Chen. 2012. Predicting mobile application usage using contextual information. In *Proc. ACM UbiComp*. 1059–1065.

[13] Simon L Jones, Denzil Ferreira, Simo Hosio, Jorge Goncalves, and Vassilis Kostakos. 2015. Revisitation analysis of smartphone app use. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 1197–1208.

[14] Kaggle. 2018. TalkingData Mobile User Demographics. (April 2018). Retrieved Oct 11, 2018 from https://www.kaggle.com/c/talkingdata-mobile-user-demographics

[15] Alexandros Karatzoglou, Linas Baltrunas, Karen Church, and Matthias Böhmer. 2012. Climbing the app wall: enabling mobile app discovery through context-aware recommendations. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2527–2530.

[16] Conger Kate. 2017. *Researchers: Uber's iOS App Had Secret Permissions That Allowed It to Copy Your Phone Screen*. https://gizmodo.com/researchers-uber-s-ios-app-had-secret-permissions-that-1819177235.

[17] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).

[18] Vassilis Kostakos, Denzil Ferreira, Jorge Goncalves, and Simo Hosio. 2016. Modelling smartphone usage: a markov state transition model. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 486–497.

[19] Yuanchun Li, Yao Guo, and Xiangqun Chen. 2016. PERUIM:understanding mobile application privacy with permission-UI mapping. In *Proc. ACM Ubicomp*. 682–693.

[20] Tzu-Heng Lin, Chen Gao, and Yong Li. 2018. Recommender Systems with Characterized Social Regularization. In *CIKM*. 1767–1770.

[21] Qi Liu, Haiping Ma, Enhong Chen, and Hui Xiong. 2013. A survey of context-aware mobile recommendations. *International Journal of Information Technology and Decision Making* 12, 01 (2013), 139–172.

[22] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts.. In *AAAI*. 194–200.

[23] Xuelian Long and James Joshi. 2013. A HITS-based POI recommendation algorithm for Location-Based Social Networks. In *Ieee/acm International Conference on Advances in Social Networks Analysis and Mining*. 642–647.

[24] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 931–940.

[25] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. (2011), 287–296.

[26] Eric Malmi and Ingmar Weber. 2016. You Are What Apps You Use: Demographic Prediction Based on User's Apps. (2016).

[27] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. 2009. WhereNext: A Location Predictor on Trajectory Pattern Mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*. 637–646.

[28] Hieu V Nguyen and Li Bai. 2010. Cosine similarity metric learning for face verification. In *Asian conference on computer vision*. Springer, 709–720.

[29] Department of Computer Science and University of Helsinki. 2017. Context Recognition by User Situation Data Analysis (Context). (Oct. 2017). Retrieved Oct 11, 2018 from https://www.cs.helsinki.fi/group/context/#data

[30] Google Play. 2017. *Mafengwo in the Google Play*. https://play.google.com/store/apps/developer?id=mafengwo.mobile.

[31] Zhen Qin, Yilei Wang, Yong Xia, and Hongrong Cheng. 2014. Demographic information prediction based on smartphone application usage. In *International Conference on Smart Computing*. 183–190.

[32] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000. *Application of dimensionality reduction in recommender system-a case study*. Technical Report. Minnesota Univ Minneapolis Dept of Computer Science.

[33] Martin Saveski and Amin Mantrach. 2014. Item cold-start recommendations: learning local collective embeddings. In *RecSys*.

[34] Suvash Sedhain, Darius Braziunas, Darius Braziunas, Jordan Christensen, and Jordan Christensen. 2014. Social collaborative filtering for cold-start recommendations. In *ACM Conference on Recommender Systems*. 345–348.

[35] Choonsung Shin, Jin-Hyuk Hong, and Anind K Dey. 2012. Understanding and prediction of mobile application usage for smart phones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 173–182.

[36] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 650–658.

[37] Zhen Tu, Runtong Li, Yong Li, Gang Wang, Di Wu, Pan Hui, Li Su, and Depeng Jin. 2018. Your apps give you away: distinguishing mobile users by their app usage fingerprints. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 138.

[38] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 15–24.

[39] Fengli Xu, Pengyu Zhang, and Yong Li. 2016. Context-aware real-time population estimation for metropolis. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 1064–1075.

[40] Ye Xu, Mu Lin, Hong Lu, Giuseppe Cardone, Nicholas Lane, Zhenyu Chen, Andrew Campbell, and Tanzeem Choudhury. 2013. Preference, context and communities:a multi-faceted approach to predicting smartphone app usage patterns. In *Proc. ACM ISWC*. 69–76.

[41] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. 2011. Semantic Trajectory Mining for Location Prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11)*. 34–43.

[42] Donghan Yu, Yong Li, Fengli Xu, Pengyu Zhang, and Vassilis Kostakos. 2018. Smartphone App Usage Prediction Using Points of Interest. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 174.

[43] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T Hanratty, and J. Han. 2016. GMove: Group-Level Mobility Modeling Using Geo-Tagged Social Media. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1305–1314.

[44] Mi Zhang, Jie Tang, Xuchen Zhang, and Xiangyang Xue. 2014. Addressing cold start in recommender systems: a semi-supervised co-training algorithm. In *SIGIR*.

[45] Shuo Zhang, Khaled Alanezi, Mike Gartrell, Richard Han, Qin Lv, and Shivakant Mishra. 2017. Understanding Group Event Scheduling via the OutWithFriendz Mobile Application. In *Proc. ACM Ubicomp*.

[46] Sha Zhao, Julian Ramos, Jianrong Tao, Ziwen Jiang, Shijian Li, Zhaohui Wu, Gang Pan, and Anind K. Dey. 2016. Discovering different kinds of smartphone users through their application usage behaviors. In *Proc. ACM UbiComp*. 498–509.

[47] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. 2016. STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation.. In *AAAI*. 315–322.

[48] Wayne Xin Zhao, Sui Li, Yulan He, Edward Y. Chang, Ji-Rong Wen, and Xiaoming Li. 2016. Connecting Social Media to E-Commerce: Cold-Start Product Recommendation Using Microblogging Information. *IEEE Transactions on Knowledge and Data Engineering* 28 (2016), 1147–1159.

[49] Xiaoxing Zhao, Yuanyuan Qiao, Zhongwei Si, Jie Yang, and Anders Lindgren. 2016. Prediction of user app usage behavior from geo-spatial data. In *Proc. ACM GeoRich*. 1–6.

[50] Vincent Wenchen Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative Filtering Meets Mobile Recommendation: A User-Centered Approach.. In *AAAI*, Vol. 10. 236–241.

[51] Hengshu Zhu, Enhong Chen, Kuifei Yu, Huanhuan Cao, Hui Xiong, and Jilei Tian. 2012. Mining personal context-aware preferences for mobile users. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 1212–1217.

Received August 2018; revised November 2018; accepted January 2019