132

Personalized Context-aware Collaborative Online Activity Prediction

YALI FAN†, School of Electronics and Information Technology, Sun Yat-Sen University, China
ZHEN TU†, Tencent Inc., China
YONG LI, Department of Electrical Engineering, Tsinghua University, China
XIANG CHEN*, School of Electronics and Information Technology, Sun Yat-Sen University, China
HUI GAO, Tencent Inc., China
LIN ZHANG, School of Electronics and Information Technology, Sun Yat-Sen University, China
LI SU, Department of Electrical Engineering, Tsinghua University, China
DEPENG JIN, Department of Electrical Engineering, Tsinghua University, China

With the rapid development of Internet services and mobile devices, nowadays, users can connect to online services anytime and anywhere. Naturally, user's online activity behavior is coupled with time and location contexts and highly influenced by them. Therefore, personalized context-aware online activity modelling and prediction is very meaningful and necessary but also very challenging, due to the complicated relationship between users, activities, spatial and temporal contexts and data sparsity issues. To tackle the challenges, we introduce offline check-in data as auxiliary data and build a user-location-time-activity 4D-tensor and a location-time-POI 3D-tensor, aiming to model the relationship between different entities and transfer semantic features of time and location contexts among them. Accordingly, in this paper we propose a transfer learning based collaborative tensor factorization method to achieve personalized context-aware online activity prediction. Based on real-world datasets, we compare the performance of our method with several state-of-the-arts and demonstrate that our method can provide more effective prediction results in the high sparsity scenario. With only 30% of observed time and location contexts, our solution can achieve 40% improvement in predicting user's $Top5$ activity behavior in new time and location scenarios. Our study is the first step forward for transferring knowledge learned from offline check-in behavior to online activity prediction to provide better personalized context-aware recommendation services for mobile users.

CCS Concepts: • **Information systems → Collaborative filtering**; **Personalization**; **Spatial-temporal systems**.

Additional Key Words and Phrases: Context-aware activity prediction, transfer learning, collaborative tensor factorization

ACM Reference Format:
Yali Fan†, Zhen Tu†, Yong Li, Xiang Chen, Hui Gao, Lin Zhang, Li Su, and Depeng Jin. 2019. Personalized Context-aware Collaborative Online Activity Prediction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article 132 (December 2019), 28 pages. https://doi.org/10.1145/3369829

*This is the corresponding author. † These two authors contributed equally to this research.

Authors' addresses: Yali Fan†, School of Electronics and Information Technology, Sun Yat-Sen University, China, fanyali@mail2.sysu.edu.cn; Zhen Tu†, Tencent Inc. China, cherrytu@tencent.com; Yong Li, Department of Electrical Engineering, Tsinghua University, China, liyong07@tsinghua.edu.cn; Xiang Chen, School of Electronics and Information Technology, Sun Yat-Sen University, China, chenxiang@mail.sysu.edu.cn; Hui Gao, Tencent Inc. China, huigaogao@tencent.com; Lin Zhang, School of Electronics and Information Technology, Sun Yat-Sen University, China, isszl@mail.sysu.edu.cn; Li Su, Department of Electrical Engineering, Tsinghua University, China; Depeng Jin, Department of Electrical Engineering, Tsinghua University, China.

2474-9567/2019/12-ART132 $15.00
https://doi.org/10.1145/3369829

1 INTRODUCTION

With increasing prevalence of mobile devices, nowadays users can connect to Internet services anytime and anywhere, thus generating a large amount of **online** activity data. Such data records users' activity behaviors interacting with the Internet, such as web browsing behavior and app usage behavior. Naturally, individual online activity contains spatial and temporal information and is highly influenced by these two contexts [54, 72]. For example, for mobile app usage behavior, which is the most representative online activity, in the morning users may use news apps more often while preferring video apps in the evening. As for location context, users are likely to use travel apps in scenic spots while using educational apps more frequently in campus. Since users tend to have different activity behaviors in different locations or at different time periods, context-aware online activity modelling and prediction becomes very necessary. Different from pure activity prediction, context-aware online activity prediction requires a better understanding of user's activity behavior at fine-grained level to provide accurate and customized prediction results. Absolutely, it will benefit mobile operators and app developers for guiding them to provide higher-quality online services and personalized recommendations [9, 78, 79].

In this paper, we focus on this important research question and aim to study personalized context-aware online activity prediction problem. More specifically, based on user online activity history with time and location context information, we want to predict his/her online activity in new scenarios, in other words, predict the user's activity under new time and location contexts. Though achieving this goal is quite meaningful, solving this problem is actually very difficult because online activity behavior is influenced by three different factors, *i.e.*, user interest, temporal context and spatial context. Considering all the factors together to make accurate prediction faces two critical challenges. **First**, spatial and temporal contexts jointly influence user's online activity behavior [54, 72], and their impacts are nonlinear and even not independent. It means that we need to consider time and location contexts at the same time and learn their coupling impact on online activity behavior. **Second**, context-aware activity prediction task needs to face worse data sparsity situation. On the one hand, for activity behavior, we may only observe a small part of online activity for each user, the sparsity of observed activity makes it very hard to learn user interest accurately. On the other hand, when further considering time and location features, the behavior data becomes sparser that each user's records only cover a small part of scenarios (*i.e.*, location and time pairs) compared with all the possible scenarios. Therefore, such high sparsity leads to great difficulty in learning user interest, spatial semantics and temporal semantics as well.

In order to tackle above challenges, in this paper we propose a transfer learning based collaborative tensor factorization method to achieve personalized context-aware online activity prediction. Specifically, for our first challenge, we build a user-location-time-activity 4D-tensor, a higher-order matrix, to model the relationship between multiple dimensions of users, locations, time periods and activities. And then we utilize tensor factorization [3, 80], a prevalent multi-dimensional collaborative recommendation method, to learn user interest, activity features, time and location semantics, and finally model the coupling impact of location and time contexts towards activity behavior. As for the second challenge, to make up for data sparsity problem, we find **offline** check-in data is very suitable to serve as auxiliary data, and help solve context-aware online activity prediction problem. Note that "**offline**" means offline activity. Besides, we regard the check-in data as offline activity because it records users' behaviors in the real physical world, *i.e.*, check-in data records users' visiting behaviors towards different POIs in specific locations and time slots. In our task, utilizing offline check-in data has many advantages: on the one hand, it records user's POI visiting behavior with time and location information [72], which makes it possible to learn semantics of location and time periods and transfer such knowledge to help our prediction task; on the other hand, it is public and easy to access, which means utilizing additional check-in data is a practical solution. Based on offline check-in data, we can also build a location-time-POI 3D-tensor. Combining 4D-tensor from online activity data with 3D-tensor from offline check-in data, we propose a collaborative tensor factorization method to learn latent feature matrices of users, locations, time slots, activities and POIs, by sharing the features of locations and time slots and transferring knowledge among them. Utilizing those latent matrices to reconstruct the targeted 4D-tensor, we are able to make predictions of activity behavior for each user under new time and location contexts. To sum up, the main contributions of this work are three-fold:

- To the best of our knowledge, we are the first to introduce offline check-in data as auxiliary data to learn semantic information of location and time contexts, to solve personalized context-aware online activity prediction problem, which helps predict online activity behavior of individuals under different contexts more accurately.
- Based on online activity data and offline check-in data, we propose a transfer learning based collaborative tensor factorization method to achieve personalized context-aware activity prediction, which jointly models the relationship of different entities and learns their latent feature matrices.
- Based on real world datasets, we evaluate our method's performance and demonstrate that our method outperforms the other state-of-the-arts in high sparsity scenario, $i.e.$, with 30% of observed time and location contexts, our solution can achieve 40% improvement than other baselines in predicting $Top5$ activity behavior under new time and location contexts. In addition, we also demonstrate that user factor, time and location contexts have quite different impacts on online activity prediction performance.

The rest of the paper is organized as follows. First we introduce related work in Section 2. Then Section 3 defines our investigated problem and motivates our study. Section 4 introduces our methodology and Section 5 evaluates its performance. Finally, Section 6 discusses its implication, and Section 7 concludes the paper.

2 RELATED WORK

2.1 Online Activity Modelling

In our daily life, there are many kinds of online activity for users and a variety of works focus on online activity modelling and behavior analysis. For email interaction [13, 39], Elsweiler et al. [13] presented a longitudinal, naturalistic study of email behavior and succeeded in isolating and understanding re-finding behavior in the email interaction logs. In addition, for social communication [18, 30], Kuan-Yu et al. [30] applied network externalities and motivation theory to explain why people continue to join SNS. As for web browsing behavior [12, 14, 49, 68], Su et al. [49] showed that it is possible to de-anonymize web browsing users with social networks.

Recently, with the highly convenience of mobile devices, many works have studied how individuals use their mobile phones as well as the mobile apps. Existing works focus on various dimensions such as user interactions [15, 46], network traffic [27, 52], and energy drain [10, 24]. A study [15] shows immense diversity of usage activities among users. Some works aim to find app usage patterns according to their app usage records [66, 78]. What's more, other works focus on app prediction or recommendation problems [22, 63, 69]. They generally make app recommendations to users based on their historical interactions with apps [7, 59, 64]. Besides, some researchers tend to satisfy extra requirements to make specific app recommendations, such as novelty and temptation [16, 70], privacy protecion [31, 83]. However, due to the large amount of apps, sparsity becomes a challenging problem when making personalized app prediction and recommendation. To solve this problem, one of the effective strategies is to introduce auxiliary information from other data sources, such as social relationship [11, 17, 29], user profiles [5, 28, 32], app's content description [2] and so on. For example, Costa et al. [17] considered the records of users sharing apps with others to see what apps are hot and should be recommended. Lin et al. [29] utilized social information from Twitter followers to solve the cold-start problem in the app recommendation task.

Different from existing works, we introduce a new kind of data, *i.e.*, offline check-in activity data as auxiliary information to tackle the sparsity challenge in personalized context-aware app prediction task. Indeed, many studies have shown that online activity behavior [36, 57] and offline behavior [41, 81] have certain correlations for they both reflect users' habits, lifestyles, interests. Therefore, our study aims to take a further step to better understand and predict user's online preferences with the help of offline behaviors.

2.2 Context-aware Recommendation

A context-aware recommendation is usually achieved by using additional information of location, time, and activity, etc [25, 33, 44, 80, 82]. Indeed, context information plays an important role in prediction and recommendation tasks. In many practical scenarios, prediction and recommendation tasks are often very complex because a variety of contexts need to be considered, *e.g.*, time, location, friends, ect. Related with our targeted problem, here we mainly focus on context-aware app recommendation studies. Many works study how to recommend suitable apps with a high likelihood of installation to users. Enrique et al. [7] presented an integrated app recommendation system by considering app ratings and tags, and contexts of location, time and weather. Wolfgang et al. [59] designed a hybrid system which recommends mobile apps to users based on what other users have installed in a similar context with location and POI information. Kuifei et al. [73] proposed to mine common context-aware preferences from many users' context logs through topic models and represent each user's personal context-aware preferences as a distribution of the mined common context-aware preferences, and then achieved personalized context-aware recommendation. Bohmer et al. [5] presented a system *AppFunnel* to track the performance of different recommender engines and found that a context-aware (*e.g.*, time, location and social network) recommender engine performs better in recommendation tasks for short-term usage.

In addition, some studies pay attention to app usage behavior, rather than installation behavior, and propose a variety of context-aware prediction methods. Within our scope, we mainly focus on spatiotemporal context-aware works [19, 25, 47, 72, 77]. Some researchers only consider one of spatial context and temporal context [47, 72, 77]. Shi et al. [47] considered location context information and proposed a context-aware recommendation approach based on tensor factorization for MAP maximization. Zhao et al. [77] considered temporal context and proposed a framework for next app usage prediction. Yu et al. [72] utilized static POI information in each location to help predict its app usage behavior at the group level. In addition, some other works considered both spatial context and temporal context. Han et al. [19] integrated the spatiotemporal information of app usage logs into a topic distribution model to recommend apps appropriate to current time and location of a user. Karatzoglou et al. [25] introduced a spatiotemporal context-aware algorithm for implicit feedback data.

However, the available data will become sparser when considering more dimensional context information, which makes it harder to learn the complex hidden correlation between app usage behavior and contextual factors. Thus it is important and necessary to consider how to make up for sparsity problem in such prediction tasks. Motivated by this, different from the works [19] and [25], we consider transfer extra spatiotemporal contextual knowledge from another data source (*i.e.*, offline check-in behavior) to further improve the prediction performance.

2.3 Tensor Factorization

Tensor is often utilized to solve context-aware prediction or recommendation problems [4, 8, 23, 37, 42, 51, 71, 80]. Zheng et al. [80] modeled the user-location-activity relations with a tensor representation to provide activity recommendations in certain locations for users. Yao et al. [67] exploited a tensor to model multi-dimensional contextual information to predict user's preferences towards POIs with the help of social regularization. Hong et al. [23] constructed a social context based tensor to discover latent interests of users in order to provide better movie recommendation. Ying et al. [71] took temporal factors into consideration in their POI recommendation system, where the temporal information is stored in a tensor. From these recent works, we can see that a great number of human activities are related to context information, such as temporal, spatial and social information.

Tensor is regarded as an efficient strategy to store context information and represent complex relationship among multiple dimensions.

In tensor factorization models, sparsity is always a challenging problem [23, 53, 74, 80], especially in the case that more than one type of context need to be considered [3]. To solve this problem, many researchers introduce context information from other data resources, such as introducing social context information from social networks [67] and introducing location context information from Foursquare [3]. Based on these external information, they usually construct an additional matrix, containing one type of context, to join the tensor factorization.

However, in our case, user's online activity is influenced by both spatial and temporal contexts. To solve the sparsity problem, different from existing works, we construct another tensor containing both spatial and temporal contexts, to help solve our context-aware prediction task. Specifically speaking, we propose a new collaborative tensor factorization method to handle both constructed tensors and transfer knowledge among them.

3 PRELIMINARIES

In this section, we first introduce two datasets utilized in our study, then we provide a detailed description about our investigated problem, *i.e.*, personalized context-aware online activity prediction. After that, we discuss its potential challenges and come up with feasible strategies to overcome them accordingly, which motivates our proposed prediction model in next section.

3.1 Dataset

Telecom Dataset. This dataset is collected from a major mobile network operator in China, which contains the access records of the users when they issue a connection request to the cellular towers in a major metropolitan city of China. It records individual mobile app usage behaviors (online activities) under a variety of spatiotemporal scenarios. It covers 1.37 million users and 9.4 billion records during one week period on April 20-26, 2016. Each record contains the following information: anonymous user ID, timestamp, connected cellular tower ID and its GPS coordinates, and the used app ID and its category. What needs to be noted in particular is that, for privacy issues, these recorded users are anonymous, which means we researchers are unable to directly identify who the user is. In addition, all the used apps recorded in this dataset cover the most popular 2,000 apps across App Store and Google Play. Therefore, we can get information about spatiotemporal online activities among a large number of apps. In addition, this dataset is released in http://fi.ee.tsinghua.edu.cn/appusage/ and it has been utilized in previous researches [54, 62, 72].

Check-in Dataset. This dataset is collected from one of the most popular location-based service providers in China, which records a user's activity of visiting a specific location. It contains over 40 million records collected in Shanghai during three months. Each check-in record consists of anonymous user ID, check-in timestamp, check-in POI-category and its POI's GPS coordinate, where POIs are categorized into 16 categories, including *Shopping, Residence, Tourism, Education, Life-Service, Industry, Restaurant, Business, Medical-Service, Culture, Government, Hotel, Transportation, Entertainment, Fitness* and *Landmark*. It has been utilized in previous work [60].

Since we have a lot of terms to describe our datasets, in order to avoid confusion, we list the definitions of several terms used frequently in the whole paper as follows. For the Telecom Dataset, **app** means a specific app name, such as Wechat and Weibo. **location** means the area covered by an individual cellular tower. For the Check-in Dataset, **POI** means a specific POI, such as a specific school, but it is not recorded in this dataset. **POI-category** means the category of a POI, such as Shopping and Residence, which is recorded in this dataset. In addition, **time** means the timestamp in each record of both datasets, such as 20160420 15:15:15.

Ethics. When it comes to protecting privacy of mobile users, we also want to point out that we have taken active measures to realize it. **First**, before processing these two datasets, we have received the approval from their providers and signed a strict non-disclosure agreement. All of our processing procedure of these datasets is within the scope of the agreement and all the researchers take responsibility for the user's privacy. Moreover, these two datasets are stored in our private server protected by authentication mechanisms and firewalls, which physically guarantee their privacy. **Second**, these two datasets do not contain any personally identification information. In Telecom Dataset, the user ID has been anonymous (replaced with a random string) by our data providers, and we have no access to the true user ID. While in Check-in Dataset, we don't have the field of user ID. **Third**, for Check-in Dataset, in fact, the spatiotemporal information is shared by users themselves, which means they do not worry that such information will be seen by others. This further indicates that such check-in data does not threaten their true sensitive privacy (*e.g.* income, health) indeed. In addition, there are also some public check-in datasets, such as Gowalla and Foursquare. These check-in datasets are easy to access, which means utilizing additional check-in data is a practical solution. In a short summary, our processing of these two datasets is carried out with the agreement of users and we can not access the information of users' true ID at any time. Consequently, user privacy is well protected throughout our experiment.

3.2 Problem Definition

With the rapid development of Internet technology and mobile devices, nowadays we can get access to the online services anytime and anywhere through a variety of mobile apps. For example, we get fresh information through news apps, obtain route guidance through navigation apps, relax ourselves through music apps, etc. Indeed, mobile devices such as smartphones, have become the most important carrier to record online activity behavior of individuals, and one fundamental online activity is user mobile app usage, *i.e.*, when and where a user uses which app. For mobile app usage, generally speaking, users may use different apps in different locations or at different time periods, *e.g.*, in the morning users use weather apps more frequently while using more entertainment apps in the evening, and users often use shopping apps in shopping malls while using game apps at home. In other words, contexts of spatial and temporal information influence user's mobile app usage behavior a lot. Therefore, understanding context-aware app usage behavior and making the corresponding context-aware recommendation is very important, which benefits traffic resource allocation [50] and personalized recommendation [78].

In this paper, we take mobile app usage (*i.e.*, the most representative online activity) as a typical case and study personalized context-aware online activity prediction problem. Specifically, suppose we have collected some user's online activity records with context information, then we need to predict each user's online activity behavior under other unobserved contexts. In our case, the context refers to spatial and temporal information of online app usage behavior. Formally, we define our investigated problem as follows:

Suppose there are N_U users and we have observed part of their online (app) activity behavior, denoted as $R = \{R_1, R_2, ..., R_{N_U}\}, R_i = \{(R_{ij}^l, R_{ij}^t, R_{ij}^a)\}$, where R_i represents the i-th user's app usage records and $R_{ij}^l, R_{ij}^t, R_{ij}^a$ represent location, time and used app of the j-th record from the i-th user, respectively. Suppose all the records contain N_L different locations, N_T time slots and N_A distinctive apps. Then, we can construct a 4D-tensor to fully represent the observed online activity of users. Specifically, the user-location-time-activity tensor can be denoted as $\mathbf{X} = \{X_{u,l,t,a}\}$, with $X_{u,l,t,a}$ representing the usage frequency of app a in location l at time slot t for user u. When $X_{u,l,t,a} = 0$, it means that we haven't observed user u's usage of app a in location l at time slot t. For a user, note that we may only observe his/her online activity in very limited scenarios(*i.e.*, specific time and location pairs), thus online his/her activity in other new scenarios remains unknown and needs to be predicted. Specifically, since we have N_L different locations and N_T different time slots, we utilize $C_{all} = \{(l,t)|l = 1, ..., N_L, t = 1, ..., N_T\}$ to represent all the scenarios, and for user u, we utilize $C_u = \{(l,t)|\sum_{a=1}^{N_A} X_{u,l,t,a} > 0\}$ and $\overline{C}_u = C_{all} - C_u$ to represent all the observed scenarios and unobserved scenarios, respectively. Then the input and output of our targeted personalized context-aware online activity prediction problem are as follows:

Input: User online activity behavior in observed scenarios(*i.e.*, specific time and location pairs), denoted by the original user-location-time-activity tensor $\mathbf{X} = \{X_{u,l,t,a} | u = 1, ..., N_U, a = 1, ..., N_A, (l,t) \in C_u\}$.

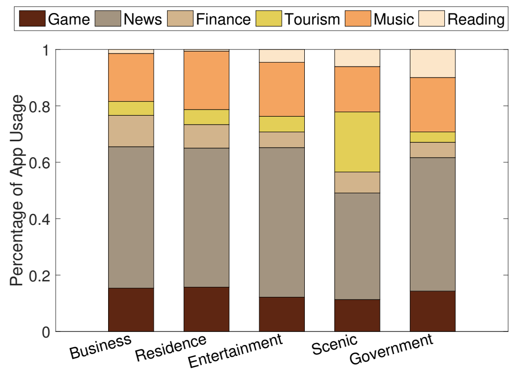
Output: User online activity behavior **in unobserved scenarios**, denoted by the recovered user-location-time-activity tensor $= \{\hat{X}_{u,l,t,a} | u = 1, ..., N_U, a = 1, ..., N_A, (l,t) \in \overline{C}_u\}$.

3.3 Challenges

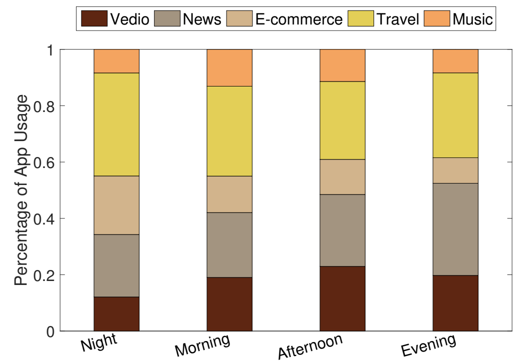
Though personalized context-aware online activity prediction is very meaningful, solving this problem faces two critical challenges, *i.e.*, the complicated interaction between user activity and spatiotemporal contexts, and the data sparsity issues. Now we provide detailed descriptions about these two challenges.

First, user's online activity behavior is naturally correlated with time and location contexts [54, 72]. For example, we often use shopping apps in shopping malls while using game apps at home, and we often use weather apps in the morning while using video apps in the evening. Since spatial and temporal contexts have an impact on user online activity behavior, context-aware online activity prediction becomes very necessary and we must fully understand the semantic meaning of spatial and temporal contexts to figure out how they influence user's online activity behavior, besides learning user interest and preference. Therefore, it will be difficult to build an effective prediction model when such multidimensional information needs to be considered and dealt with.

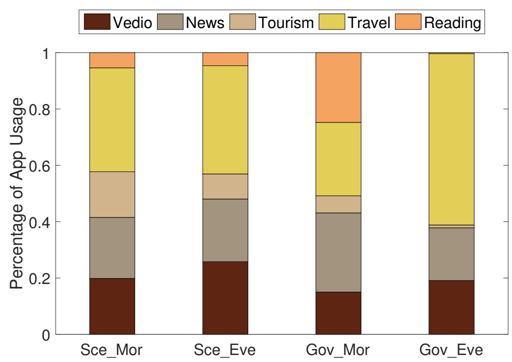
In order to better illustrate this challenge, we empirically analyze how different location and time contexts influence user's online activity behavior by utilizing a large-scale app usage dataset as a typical case. Specifically, we analyse the correlation between user's app usage behavior and three different kinds of contexts (*i.e.*, location context, time context, and both location and time contexts). The results are shown in Figure 1 (a)-(c) that we demonstrate the app usage frequency under different contexts. Note that we simply label each location by the most prevalent POI category within its coverage. In addition, for time context, we generally divide one day into four periods, *i.e.*, night 0:00-6:00, morning 6:00-12:00, afternoon 12:00-18:00 and evening 18:00-24:00. First, for location context in Figure 1 (a), we can find that finance apps are used the most frequently in business locations, tourism apps are used the most frequently in scenic locations and reading apps have the highest usage frequency in government locations. Note that we get these judgments just from the probability perspective. It doesn't mean that they will absolutely happen, but there is a high probability. For instance, based on the statistic results in Figure 1 (a), we find that users are more likely to use finance apps in business locations compared to other functional locations. However, we can not guarantee that users absolutely use finance apps in business locations. Obviously, app usage patterns are various and quite different in locations with different functions. Second, for time context in Figure 1 (b), likewise, we can find that e-commerce apps are used the most frequently at night, probably because people have some spare time shopping online before sleep. In addition, music apps are used very frequently in the morning, probably because people want to get rid of fatigue and get pleasure by listening to music in the morning. These results are consistent with user's daily habits. Third, when considering both location and time contexts, we choose two types of locations and two time periods and compare the difference of app usage behavior in different contexts, the results are shown in Figure 1 (c). Note that "Sce" means "Scenic", "Gov" means "Government", "Mor" means "Morning" and "Eve" means "Evening". From the figure, we can find that even with the same type of location, app usage pattern in the morning is quite different from that in the evening, so as the situation in different types of locations at the same time period. For example, for government location, reading apps are used more frequently in the morning, while travel apps are used more frequently in the evening. Moreover, in the morning, tourism apps are used more frequently in scenic locations, while news apps are used more frequently in government locations. What's more, with a fixed pair of time and location contexts, Figure 1 (d) plots the Cumulative Distribution Function (CDF) of Jaccard Distance [58] between pairwise users regarding their app usage behavior, including the difference of their app lists and category-level app lists. From the figure, as the red line shows, we can see that more than 90% pairwise users have a Jaccard Distance higher than 0.8, which



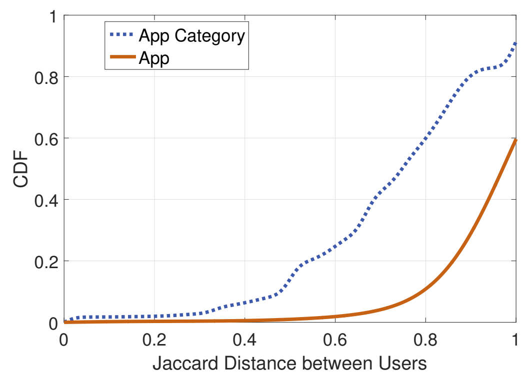
(a) Location context



(b) Time context



(c) Location and time contexts



(d) CDF of app usage difference between users

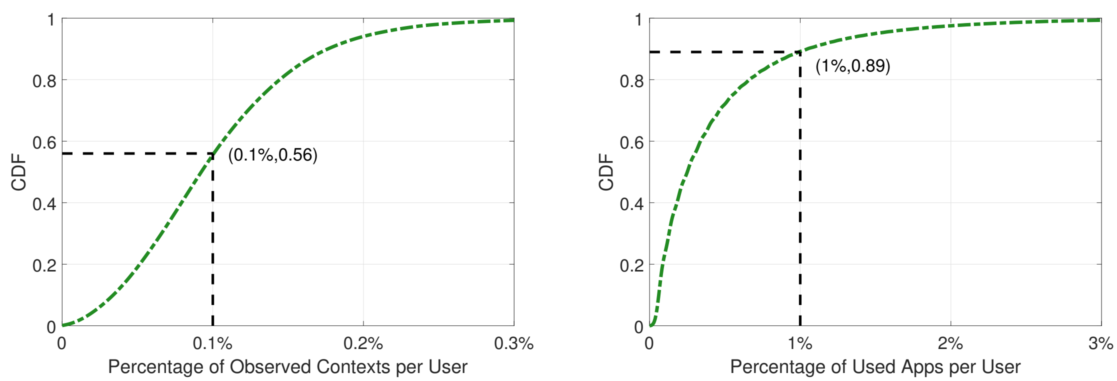
Fig. 1. (a)-(c) Illustration of the relationship between user's online activity behavior and location and time contexts. (d) Difference of online activity behavior between users under certain location and time contexts.

means they have more than 80% different apps. Even considering category-level app usage, there exists great difference between pairwise users that 75% users have a Jaccard distance more than 0.6. Different from Figure 1 (a)-(c), Figure 1 (d) shows that a single user's app usage behavior is quite different from that of others even under the same time and location contexts, showing the impact of user interest on app usage behavior besides the time and location contexts. All the results demonstrate the necessity of studying personalized context-aware online activity prediction problem, which means we should take fully consideration of user interest, time and location contexts to make the final decision. Absolutely, due to the complicated interaction among user app usage and contexts, it is a very challenging task to make accurate prediction for individual online activity.

Second, recorded online activity behavior of users is usually very sparse, the sparsity problem will become even worse when we consider more contexts such as time and location information. On the one hand, we can only observe a small part of used apps for each user, which makes it hard to learn user interest from such limited information. On the other hand, as for context-aware app usage behavior, indeed we only observe user's interaction with apps in very limited scenarios, which means that we need to predict user's app usage preference in unobserved scenarios with new time and location contexts just based on limited user app usage behavior under observed time and location contexts. Therefore, the sparsity issue is actually a critical challenge in context-aware online app activity prediction task, which prevents us from fully understanding user preference thus makes us unable to provide accurate prediction result.

Empirically, based on a real-world app usage dataset, we measure the sparsity level of the data from above two aspects. First, we calculate the percentage of observed scenarios (*i.e.*, the time and location pairs) for each user and plot its cumulative distribution in Figure 2 (a). Here we consider 7000∼ different locations and 10 distinctive time slots, so the number of different pairs is 70000∼. From the figure, we can see that the percentage of observed scenarios is quite low, *i.e.*, for 56% of users, their observed scenarios only cover less than 0.1% of all the possible pairs. In addition, the highest coverage rate is about 0.3%. Such sparsity level shows that the recorded interaction between a specific user and time and location contexts is very limited, which makes context-aware app usage prediction very challenging. In addition, Figure 2 (b) demonstrates the cumulative distribution function (CDF) of percentage of used apps per user in specific time and location contexts. In a scenario with fixed time and location contexts, we can see that 89% of users have used less than 1% apps in the whole app set. Therefore, the high sparsity level makes it very difficult to find a user's app usage preference under specific spatiotemporal context.

To sum up, the complex interaction between user app usage and spatiotemporal contexts and the data sparsity issue are two critical challenges for our targeted context-aware app usage prediction problem. In order to make accurate prediction, next we will introduce how we propose effective strategies to overcome these two challenges.



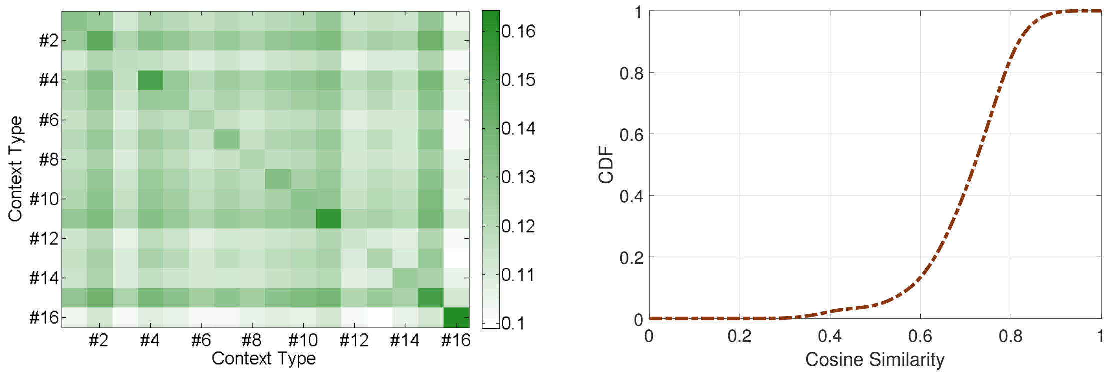
(a) Sparsity of covered contexts (b) Sparsity of used apps in specific contexts

Fig. 2. Illustration of the sparsity level of context-aware online activity data from two aspects.

3.4 Strategies

For our investigated context-aware prediction problem, one difficulty is how to model the complex relationship among users, activities and time, locations. Without prior knowledge, in fact it is very hard to build a suitable model to handle such high dimensional information. Recently, tensors, higher-order matrices, are proved to be able to model the relationship between multiple dimensions and tensor factorization is a prevalent multi-dimensional collaborative recommendation method [3, 80]. Therefore, we try to adopt tensor factorization to solve our context-aware prediction problem. In our case, we can build a user-location-time-activity tensor to model the complex correlation between users, locations, time and activities.

However, such a tensor is quite sparse because each user's spatiotemporal online activity is very limited, which makes it hard to learn features of users, activities, locations and time periods. In order to solve this problem, a common practice is to find other accessible external dataset and learn helpful knowledge and transfer it to benefit our prediction task. In our context-aware online activity prediction problem, learning the semantic meaning of time and location contexts is of essential importance. Therefore, public available check-in data seems to be a suitable helper. Check-in data records the frequency of category-level check-in POIs [72] in different locations and at different time periods, which can truly represent the function of different locations as well as its change during the day. From this aspect, it meets our requirement very well. Utilizing the offline check-in data, we can also build a location-time-POI tensor and learn useful information of time periods and locations, then transfer the knowledge to help our tensor factorization mainly based on above user-location-time-activity tensor.



(a) App usage similarity

(b) App usage vs POI visitation

Fig. 3. The correlation between spatiotemporal app usage and POI visitation behavior.

Further, in order to test whether the strategy is feasible, we measure the correlation between online activity behavior and offline check-in behavior in the same time and location context. First of all, we compute the average app usage similarity between pairwise contexts in the same type and in different types, where the context refers to a pair of location and time and the type is assigned by the most popular POI category under the specific context. To be specific, first, for each scenario $(l, t) \in C_{all}$, we can observe its feature from two aspects. One is from its app usage behavior (online behavior) based on Telecom Dataset, and the other is from its POI distribution (offline behavior) based on Check-in Dataset. Second, in order to learn the relationship between these two kinds of behavior, we first label each scenario (l, t) according to its POI distribution. Then we put all the app usage records with the same scenario label together as the app usage behavior group with this label. Finally, we compute the average Jaccard Similarity [58] between two app usage behavior groups with the same scenario label or different scenario labels. The Jaccard Similarity formulation is as follows:

$$J\left(V_i^{\#k_1}, V_j^{\#k_2}\right) = \frac{S_i^{\#k_1} \cap S_j^{\#k_2}}{S_i^{\#k_1} \cup S_j^{\#k_2}}, \tag{1}$$

where $\mathbf{V}_i^{\#k_1}$ represents the app usage frequency vector of i-th scenario labeld $\#k_1$, and $\mathrm{S}_i^{\#k_1}$ represents its corresponding app set. So as $\mathbf{V}_j^{\#k_2}$ and $\mathrm{S}_j^{\#k_2}$. The result is shown in Figure 3 (a). From the figure, we see that the app usage similarity in the same context type is higher than that in different context types, *i.e.*, for context type #16, the app similarity within the same context type is about 0.16, while the similarity decreases to 0.1 when compared with contexts of different types. Since the same type of context has the same most popular POI category, it shows that if two time and location contexts have similar POI visitation behavior, they may also have similar app usage behavior.

What's more, we directly measure the similarity between app usage behavior and POI visitation behavior in the same time and location context. Based on app usage frequency and POI visitation frequency, we utilize Cosine Similarity [38] to compute app similarity and POI similarity between pairwise contexts, respectively. In specific, we can obtain app usage behavior under each context (l, t) in C_{all} based on Telecom Dataset, and also can obtain these contexts' POI distribution based on Check-in Dataset. Therefore, we can get the Cosine Similarity between two different contexts $(l, t)_i$ and $(l, t)_j$ according to their app usage behavior. So as these two contexts' POI distribution. We use $\mathbf{E}_i = \{e_{ij}\}$ denoting app similarity between context i and other contexts, with e_{ij} representing app similarity between context i and j. So as the POI similarity vector $\mathbf{F}_i = \{f_{ij}\}$. If the app similarity between context i and context j and their POI similarity are closely related, the app vector $\mathbf{E}_i$ and POI vector $\mathbf{F}_i$, *i.e.*, the distribution of similarity, will have a strong correlation. Thus, to further quantify the relationship between apps and POIs, we still use Cosine Similarity to compute the correlation between app similarity vector $\mathbf{E}_i$ and POI similarity vector $\mathbf{F}_i$ for context i. The correlation S_i of $\mathbf{E}_i$ and $\mathbf{F}_i$ is computed as follows:

$$S_i = cos(\mathbf{E}_i, \mathbf{F}_i), \forall i, j = 1, ..., N_L N_P, \tag{2}$$

with $N_L N_P$ representing the total number of contexts. The CDF of $S = \{S_i\}$ is shown in Figure 3 (b). We can observe that for nearly 60% of contexts, the correlation between their app usage behavior and POI visiting behavior is more than 70%, which indicates that there is a strong correlation between app usage and POI visitation. Therefore, with the help of offline check-in data, we are able to learn more knowledge about the semantic meanings of location and time contexts, so as to make up the sparsity problem in context-aware online activity behavior.

All the above analysis demonstrates that POI visiting behavior in offline check-in data and app usage behavior in online activity data are strongly correlated, especially in the same time and location context. Therefore, it is feasible to predict what apps a user will use when given the time and location contexts with the help of check-in dataset and tensor factorization.

4 METHODOLOGY

In our study, we aim to solve the context-aware online activity problem. Adopting the strategies discussed in last section, we propose a transfer learning based collaborative tensor factorization method to solve our targeted problem. Now we first give an overview about our framework and then discuss the details about our proposed method.

4.1 Framework Overview

Figure 4 demonstrates our framework for context-aware online activity prediction task. Roughly, it can be divided into three parts: Data & Input, Generative Model, Features & Output.

Data & Input: In our task, we consider two types of data (*i.e.*, online app data and offline check-in data) as our system's input. Specifically, app data records user online activity behavior in the targeted domain, which contains spatial and temporal app usage behavior of individuals. Based on four dimensional information of user, location, time and app activity, we can construct a 4D-tensor, denoted as $\mathbf{X} = \{X_{u,l,t,a}\}$, where $X_{u,l,t,a}$ represents app a's normalized usage frequency for user u at time slot t in location l. Likewise, check-in data records user offline activity behavior in the auxiliary domain, which reflects POI visitation behavior in different location and time scenarios. Based on three dimensional information of location, time and POI, we are able to construct a 3D-tensor, denoted as $\mathbf{Y} = \{Y_{l,t,p}\}$, where $Y_{l,t,p}$ represents POI category p's normalized visitation frequency at

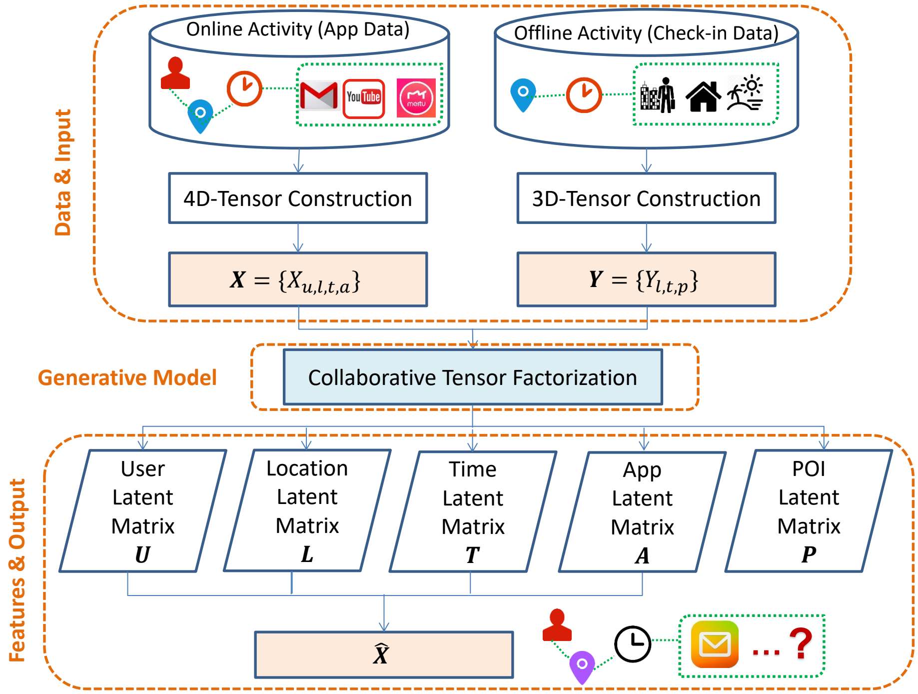


Fig. 4. Illustration of our framework for context-aware online activity prediction task.

time slot t in location l. Note that for tensor $\mathbf{Y}$, we utilize POI visitation behavior in the aggregated level rather than in the individual level, which means that our method doesn't necessarily require collecting offline check-in data from targeted users in the app data. Since we just want to learn semantic meaning of locations and time periods from check-in data, the aggregated category-level POI visitation behavior is enough and more easy to obtain in practice.

Generative Model: After obtaining required 4D-tensor $\mathbf{X}$ and 3D-tensor $\mathbf{Y}$, a generative model, *i.e.*, *Collaborative Tensor Factorization*, is used to model the relationship between multiple dimensions and transfer knowledge among locations and time contexts. The details of this generative model will be introduced later in subsection 4.2.

Features & Output: Learned from above collaborative tensor factorization method, this module receives all the latent matrices from users, locations, time slots, app activities and POIs, denoted by U, L, T, A and P. By utilizing these latent matrices, we are able to reconstruct the user-location-time-activity tensor by filling in all the empty values, and then predict each user's app preference when given new time and location contexts.

4.2 Prediction Method

In this section, we provide the details of our transfer learning based generative model, *i.e.*, *Collaborative Tensor Factorization*. Recently, tensors prove to be effective to model the relationship between multiple dimensions and tensor factorization is a prevalent multi-dimensional collaborative recommendation method [3, 80]. But existing tensor factorization models only handle one single tensor, which cannot meet our requirement of addressing two tensors with different dimensions. Therefore, for our specific task, we propose a novel collaborative tensor

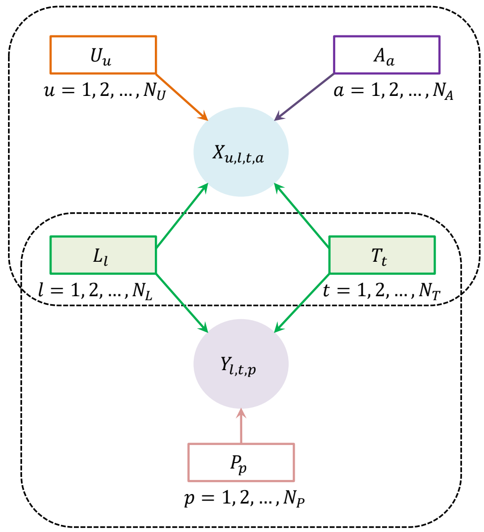


Fig. 5. Illustration of the generative model.

factorization method, which is able to combine both user-location-time-activity 4D-tensor $\mathbf{X}$ and location-time-POI 3D-tensor $\mathbf{Y}$.

Recall that we have two tensors denoted as $\mathbf{X} = \{X_{u,l,t,a}\}, \mathbf{Y} = \{Y_{l,t,p}\}, u = 1, ..., N_U, l = 1, ..., N_L, t = 1, ..., N_T, a = 1, ..., N_a, p = 1, ..., N_P$, respectively. In detail, $X_{u,l,t,a}$ represents app a's normalized usage frequency for user u at time slot t in location l, and $Y_{l,t,p}$ represents POI category p's normalized usage frequency at time slot t in location l.

Combining above two tensors, we propose a transfer learning [40, 61] based generative model, shown in Figure 5. There are five latent feature vectors: $U_u \in \mathbf{U} \in \mathbf{R}^{N_U \times H}, L_l \in \mathbf{L} \in \mathbf{R}^{N_L \times H}, T_t \in \mathbf{T} \in \mathbf{R}^{N_T \times H}, A_a \in \mathbf{A} \in \mathbf{R}^{N_A \times H}, P_p \in \mathbf{P} \in \mathbf{R}^{N_P \times H}$, where U_u, L_l, T_t, A_a, P_p represent the H-dimensional latent feature vectors of user u, location l, time t, app activity a and POI p, respectively. Accordingly, **U, L, T, A, P** are user latent feature matrix, location latent matrix, time latent matrix, app latent matrix and POI latent matrix, respectively. As shown in the figure, the idea is to share the latent vectors of locations and time slots (**L, T**), which aims to transfer the semantic meanings of locations and time slots between tensor $\mathbf{X}$ and $\mathbf{Y}$. Based on this generative model, we learn all the latent feature matrices by minimizing the following objective function through CANDECOMP/PARAFAC (CP) [6, 21] decomposition, which is a sum of squared errors with quadratic regularization terms:

$$\begin{aligned}\mathcal{L}(\mathbf{U}, \mathbf{L}, \mathbf{T}, \mathbf{A}, \mathbf{P}) &= \frac{1}{2}\|\mathbf{X} - [\![\mathbf{U}, \mathbf{L}, \mathbf{T}, \mathbf{A}]\!]\|_F^2 + \frac{\alpha}{2}\|\mathbf{Y} - [\![\mathbf{L}, \mathbf{T}, \mathbf{P}]\!]\|_F^2 \\ &+ \left(\frac{\lambda_{\mathrm{U}}}{2}\|\mathbf{U}\|_F^2 + \frac{\lambda_{\mathrm{L}}}{2}\|\mathbf{L}\|_F^2 + \frac{\lambda_{\mathrm{T}}}{2}\|\mathbf{T}\|_F^2 + \frac{\lambda_{\mathrm{A}}}{2}\|\mathbf{A}\|_F^2 + \frac{\lambda_{\mathrm{P}}}{2}\|\mathbf{P}\|_F^2\right),\end{aligned} \tag{3}$$

where all the latent feature matrices can be regarded as the set of column vectors that $\mathbf{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_H]$, $\mathbf{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_H]$, $\mathbf{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_H]$, $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_H]$, $\mathbf{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_H]$. Then we have $[\![\mathbf{U}, \mathbf{L}, \mathbf{T}, \mathbf{A}]\!] = \sum_{i=1}^{H} \boldsymbol{u}_i \circ \boldsymbol{l}_i \circ \boldsymbol{t}_i \circ \boldsymbol{a}_i$ and $[\![\mathbf{L}, \mathbf{T}, \mathbf{P}]\!] = \sum_{i=1}^{H} \boldsymbol{l}_i \circ \boldsymbol{t}_i \circ \boldsymbol{p}_i$, and "$\circ$" means the outer product. In addition, $\|\cdot\|_F^2$ denotes the Frobenius norm. For other parameters, α is the transfer weight of location-time-POI tensor, and $\frac{\lambda_U}{2}, \frac{\lambda_L}{2}, \frac{\lambda_T}{2}, \frac{\lambda_A}{2}, \frac{\lambda_P}{2}$ are regularization term coefficients for different latent feature matrices accordingly. This loss function consists of three parts. In the first part, we decompose the tensor **X** into four low-dimensional matrices and then calculate the errors after they are restored. In the second part, we decompose the tensor **Y** into three low-dimensional matrices and then also calculate the errors after they are restored. As for the last part, it contains several regularization terms in order to prevent over-fitting.

Since there are no closed form solution for $\mathcal{L}(\cdot)$, we choose to provide a numerical optimal approximation solution. In order to reduce the time complexity of the model training process, we adopt mini-batch gradient descent approach to learn the parameters. The first-order derivatives of these five matrices are as follows:

$$\begin{aligned}
\nabla_{\mathrm{U}}\mathcal{L} &= -\mathbf{X}^{(1)}(\mathbf{A}*\mathbf{T}*\mathbf{L}) + \mathbf{U}\left[(\mathbf{A}*\mathbf{T}*\mathbf{L})^{\mathrm{T}}(\mathbf{A}*\mathbf{T}*\mathbf{L})\right] + \lambda_{\mathrm{U}}\mathbf{U},\\
\nabla_{\mathrm{L}}\mathcal{L} &= -\mathbf{X}^{(2)}(\mathbf{A}*\mathbf{T}*\mathbf{U}) + \mathbf{L}\left[(\mathbf{A}*\mathbf{T}*\mathbf{U})^{\mathrm{T}}(\mathbf{A}*\mathbf{T}*\mathbf{U})\right] - \mathbf{Y}^{(1)}(\mathbf{P}*\mathbf{T}) + \mathbf{L}(\mathbf{P}*\mathbf{T})^{\mathrm{T}}(\mathbf{P}*\mathbf{T}) + \lambda_{\mathrm{L}}\mathbf{L},\\
\nabla_{\mathrm{T}}\mathcal{L} &= -\mathbf{X}^{(3)}(\mathbf{A}*\mathbf{L}*\mathbf{U}) + \mathbf{T}\left[(\mathbf{A}*\mathbf{L}*\mathbf{U})^{\mathrm{T}}(\mathbf{A}*\mathbf{L}*\mathbf{U})\right] - \mathbf{Y}^{(2)}(\mathbf{P}*\mathbf{L}) + \mathbf{T}(\mathbf{P}*\mathbf{L})^{\mathrm{T}}(\mathbf{P}*\mathbf{L}) + \lambda_{\mathrm{T}}\mathbf{T},\\
\nabla_{\mathrm{A}}\mathcal{L} &= -\mathbf{X}^{(4)}(\mathbf{T}*\mathbf{L}*\mathbf{U}) + \mathbf{A}\left[(\mathbf{T}*\mathbf{L}*\mathbf{U})^{\mathrm{T}}(\mathbf{T}*\mathbf{L}*\mathbf{U})\right] + \lambda_{\mathrm{A}}\mathbf{A},\\
\nabla_{\mathrm{P}}\mathcal{L} &= -\mathbf{Y}^{(3)}(\mathbf{T}*\mathbf{L}) + \mathbf{P}(\mathbf{T}*\mathbf{L})^{\mathrm{T}}(\mathbf{T}*\mathbf{L}) + \lambda_{\mathrm{p}}\mathbf{P}.
\end{aligned} \tag{4}$$

As the same as related work [80], $\mathbf{X}^{(k)}$ denotes the mode-k tensor unfolding, and each element X_{i_1,i_2,i_3,i_4} in the tensor **X** has a corresponding position (i,j) in each mode's unfolding. To be specific, the size of $\mathbf{X}^{(k)}$ and the corresponding values of i and j are as follows:

- For mode-1 ($i=1$): $\mathbf{X}^{(1)} \in \mathbf{R}^{N_U\times(N_L N_T N_A)}$, $i=i_1$, $j=(i_4-1)\,N_T N_L+(i_3-1)\,N_L+i_2$.
- For mode-2 ($i=2$): $\mathbf{X}^{(2)} \in \mathbf{R}^{N_L\times(N_U N_T N_A)}$, $i=i_2$, $j=(i_4-1)\,N_T N_U+(i_3-1)\,N_U+i_1$.
- For mode-3 ($i=3$): $\mathbf{X}^{(3)} \in \mathbf{R}^{N_T\times(N_U N_L N_A)}$, $i=i_3$, $j=(i_4-1)\,N_L N_U+(i_2-1)\,N_U+i_1$.
- For mode-4 ($i=4$): $\mathbf{X}^{(4)} \in \mathbf{R}^{N_A\times(N_U N_L N_T)}$, $i=i_4$, $j=(i_3-1)\,N_L N_U+(i_2-1)\,N_U+i_1$.

Likewise, for the tensor Y, $\mathbf{Y}^{(k)}$ denotes the mode-k tensor unfolding, and each element Y_{i_1,i_2,i_3} in the tensor **Y** has a corresponding position (i,j) in each mode's unfolding. Likewise, the size of $\mathbf{Y}^{(k)}$ and the corresponding values of i and j are as follows:

- For mode-1 ($i=1$): $\mathbf{Y}^{(1)} \in \mathbf{R}^{N_L\times(N_T N_P)}$, $i=i_1$, $j=(i_3-1)\,N_T+i_2$.
- For mode-2 ($i=2$): $\mathbf{Y}^{(2)} \in \mathbf{R}^{N_T\times(N_L N_P)}$, $i=i_2$, $j=(i_3-1)\,N_L+i_1$.
- For mode-3 ($i=3$): $\mathbf{Y}^{(3)} \in \mathbf{R}^{N_P\times(N_L N_T)}$, $i=i_3$, $j=(i_2-1)\,N_L+i_1$.

In addition, "$*$" denotes the Khatri-Rao product [26]: for two matrices with the same number of columns $\mathrm{A} = \left[\boldsymbol{a}_1,\boldsymbol{a}_2,\ldots,\boldsymbol{a}_J\right] \in \mathbf{R}^{V\times J}$ and $\mathrm{B} = \left[\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_J\right] \in \mathbf{R}^{W\times J}$, their Khatri-Rao product is defined as $\mathrm{A} * \mathrm{B} = \left[\boldsymbol{a}_1\otimes\boldsymbol{b}_1,\boldsymbol{a}_2\otimes\boldsymbol{b}_2,\ldots,\boldsymbol{a}_J\otimes\boldsymbol{b}_j\right] \in \mathbf{R}^{VW\times J}$, where "$\otimes$" denotes the Kronecker product [56].

The gradient descent process of each matrix is as follows:

$$\mathbf{U} := \mathbf{U} - \eta\nabla_{\mathrm{U}}\mathcal{L},\ \mathbf{L} := \mathbf{L} - \eta\nabla_{\mathrm{L}}\mathcal{L},\ \mathbf{T} := \mathbf{T} - \eta\nabla_{\mathrm{T}}\mathcal{L},\ \mathbf{A} := \mathbf{A} - \eta\nabla_{\mathrm{A}}\mathcal{L},\ \mathbf{P} := \mathbf{P} - \eta\nabla_{\mathrm{P}}\mathcal{L}. \tag{5}$$

In conclusion, we propose a transfer learning based generative model to accomplish the personalized context-aware online activity prediction goal. After inputting a 4-D user-location-time-activity tensor **X** and a 3-D location-time-POI tensor **Y**, our method can reconstruct a denser 4-D user-location-time-activity tensor $\hat{\mathbf{X}}$ by sharing the latent feature matrices of location and time contexts.

5 EVALUATION

In this section, to evaluate the performance of our proposed personalized context-aware online activity prediction model from different aspects, we conduct a series of experiments to answer the following three research questions:

- How will our model perform with different levels of data sparsity, especially compared to a series of state-of-art algorithms?

- How the performance of our model will be affected under different hyper-parameter settings, *i.e.*, the transfer weight of auxiliary Check-in data and the dimension of latent feature vectors.
- What impact of different contexts and factors (*i.e.*, different time periods, location functions and user attributes) have on the performance of our model?

5.1 Data Pre-processing

In our study, we utilize two datasets to solve the personalized context-aware prediction problem, including the Telecom dataset and the Check-in dataset, where the latter is used as auxiliary information. Now we introduce their pre-processing process in details.

In order to construct our targeted tensor **X** from the Telecom Dataset, we need to do some pre-processing based on the raw data. First, for time granularity, we divide the time of one day into several time slots for convenience. In specific, we divide one day into 10 **time-slot**s, *i.e.*, duration 0:00-6:00 is regarded as one time-slot, and the rest duration 6:00-24:00 is divided into 9 time-slots by two-hour duration. As for location granularity, in this dataset, our "location" refers to the area covered by each cellular tower. Note that 50% of these locations' areas are less than 0.14 km^2, and 70% of them are less than 0.4 km^2, which means that we need to predict a certain user's online activity behavior in very fine-grained location level. Second, because users do not provide any explicit ratings to smartphone app, in our study, in order to measure the online activity preference of a certain user, we adopt the frequency metric, *i.e.*, how often the user interacts with a certain app. With above information, we can construct our targeted tensor **X**, as mentioned in last section. Further, in order to make sure the range of each element in tensor **X** is within [0, 1], we perform normalization to all elements in tensor **X**. In detail, we choose to perform normalization in the scenario (*i.e.*, specific spatial and temporal context) level, because our task is to predict individual online activity preference in unobserved scenarios. The specific formula is as follows:

$$X_{u,l,t,a_i} := \frac{\log\left(X^*_{u,l,t,a_i}\right)}{\max\left\{\log\left(X^*_{u,l,t,a_j}\right) | j = 1,\ldots,N_A\right\}}, \forall i = 1,\ldots,N_A, \tag{6}$$

where X^*_{u,l,t,a_i} represents the number of interactions for the i-th app of user u in location l at time slot t, X_{u,l,t,a_i} represents the corresponding normalized frequency and N_A denotes the number of apps.

As for the Check-in Dataset, in order to keep the same location granularity with the Telecom Dataset, we map each POI's GPS coordinate to its cellular tower through Voronoi diagram segmentation method [1]. As for consistent time granularity, the same time division scheme is applied to Check-in dataset. Based on this dataset, we can learn semantic meanings of locations and time slots, which is helpful to solve sparsity problem and explore users' online spatiotemporal activity preference. In addition, as the same as tensor **X**, for tensor **Y**, we also perform normalization on it as follows:

$$Y_{l,t,p_i} := \frac{\log\left(Y^*_{l,t,p_i}\right)}{\max\left\{\log\left(Y^*_{l,t,p_j}\right) | j = 1,\ldots,N_P\right\}}, \forall i = 1,\ldots,N_P, \tag{7}$$

with Y^*_{l,t,p_i} representing the number of times visited for the i-th POI category in location l at time slot t, Y_{l,t,p_i} representing the corresponding normalized frequency and N_P denoting the number of POI categories.

Now we have constructed tensor **X** and **Y**, which serve as the inputs of our proposed model. Based on this data, we are able to evaluate our model's performance.

5.2 Metrics and Baselines

In our study, we adopt two prevalent metrics, *i.e.*, *TopK Accuracy*, and $nDCG_K$, to evaluate the effectiveness of prediction results and compare with other baselines.

TopK Accuracy: This is a metric that measures the mean prediction accuracy on *TopK* prediction of online activity behavior under each unobserved scenario for each user, which can be expressed as follows:

$$TopKAccuracy = \frac{\sum_{i=1}^{N_U}\left[\sum_{j=1}^{N_{\overline{C}_i}}\left(L_{i,j}^p \cap L_{i,j}^t / K\right) / N_{\overline{C}_i}\right]}{N_U}, \tag{8}$$

where N_U denotes the number of users and $N_{\overline{C}_i}$ denotes the number of unobserved scenarios of the i-th user. $L_{i,j}^p$ refers to the set of predicted *TopK* apps of the i-th user under his j-th unobserved scenario and $L_{i,j}^t$ refers to the corresponding ground truth results in the testing set.

$nDCG_K$: This metric is a common measure of ranking quality for the relevance of *TopK* results. In our study, it can be expressed as follows:

$$nDCG_K = \frac{DCG_K}{IDCG_K} = \sum_{i=1}^{N_U}\left[\sum_{j=1}^{N_{\overline{C}_i}}\left(\frac{\sum_{k=1}^{K} rel_k^p / \log_2(k+1)}{\sum_{k=1}^{K} rel_k^t / \log_2(k+1)}\right) / N_{\overline{C}_i}\right] / N_U, \tag{9}$$

where rel_k^p denotes the relevance (the normalized usage frequency) of the j-th predicted app and rel_k^t denotes the relevance (the normalized usage frequency) of the j-th app in the testing set, for the i-th user under his j-th unobserved scenario. A higher $nDCG_K$ value indicates a better ranking result.

The above two different metrics can evaluate the accuracy and effectiveness of our model from different perspectives. In order to evaluate the performance of our model, we compare it with other seven state-of-art algorithms.

SCP [3]: SCP means Standard CP decomposition model, which only takes one tensor to decompose. Here this tensor refers to $\mathbf{X}$, which is equivalent to our model in the case that $\alpha = 0$ and $\lambda_P = 0$. Thus, this baseline only utilizes online activity information without any offline activity information. By compared with our model, we can further observe what role the external POI visiting information will play in our prediction task.

SoRec [20]: SoRec means Social Recommendation, where the word Social originally refers to social friendship information among users via a user-user matrix. In our case, to get such a user-user matrix, we first construct a user-app matrix from the tensor $\mathbf{X}$ and then obtain the user-user matrix by calculating the cosine similarity between pairwise users.

SR-U: SR [35] means Social Regularization, which integrates social information (the user-user matrix) via a social regularization term rather than a collaborative factorization matrix term. Here the letter U refers to user, and SR-U means that we utilize user-user information as the same as SoRec.

SR-T: Like SR-U, here the letter T refers to time, and SR-T means that we utilize time-time information obtained from the tensor $\mathbf{Y}$ to replace the social information. Similarly, we first obtain a time-POI matrix from the tensor $\mathbf{Y}$, and then calculate the cosine similarity between pairwise time slots.

SR-L: Similar to SR-T, we first obtain a location-POI matrix from the tensor $\mathbf{Y}$, then obtain the location-location matrix to replace the social information.

SoCo [76]: SoCo integrates social and contextual information into recommend system to divide the ratings in the user-item matrix into several clusters. Therefore, the ratings in the same cluster are more contextually similar and a new rating's predicting is only carried out in its contextually nearest cluster.

SVD-MFN [45]: In order to better predict the preference of a user to an item, Singular Value Decomposition with Multi-Factor Neighborhood (SVD-MFN) considers several context factors to help find the target item's nearest K neighbor items from the target user's historical interacted items. Here we consider spatial, temporal and social factors to find neighbors.

KNN: KNN can exploit similarity between users, which means recommending items for a certain user according to the interaction history of K users who have the closest interests with her. Based on the user-user matrix, for each user, KNN first finds the nearest K neighbor users, then predicts their $TopN$ most commonly used apps for her according to the user-app matrix.

POP: POP recommends the most popular items sorted by the training set for a certain user in the testing set. We obtain the sorted app popularity list according to the total usage frequency of each app in the training set.

From above descriptions, we can see that baselines SCP, SoRec, SR-U, KNN and POP provide prediction only through the online activity behaviors from the Telecom Dataset. While the other four baselines SR-T, SR-L, SoCo and SVD-MFN utilize both online activity information and offline activity information. In addition, KNN and POP can not realize context-aware prediction because they do not utilize any spatiotemporal information.

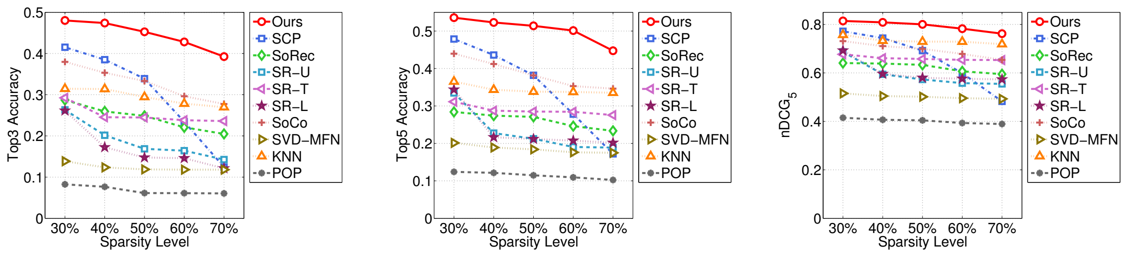
5.3 Parameter Setting

There are many hyper-parameters in our model: dimension of latent feature vector H, transfer weight for offline activity information α, regularization coefficients $[\frac{\lambda_U}{2}, \frac{\lambda_L}{2}, \frac{\lambda_T}{2}, \frac{\lambda_A}{2}, \frac{\lambda_P}{2}]$, learning rate η, and maximum iteration times I.

In order to determine the value of each hyper-parameter, we experiment with a sequence of settings and select the most appropriate one. In this way, we empirically set $H = 10$, $\alpha = 0.4$, $\eta = 0.01$, $I = 4000$ and $[\frac{\lambda_U}{2} = 0.01, \frac{\lambda_L}{2} = 0.04, \frac{\lambda_T}{2} = 0.01, \frac{\lambda_A}{2} = 0.08, \frac{\lambda_P}{2} = 0.01]$. In order to keep consistency and guarantee the comparability of results, for our baselines, we set the same values of hyper-parameters as our model if they have. More specifically, for SR-T and SR-L, we also set $\alpha = 0.4$ as their POI information's contribution. However, for SoRec and SR-U, the contribution of user-user information is not reflected in our method, so we again empirically set their weights $\frac{\alpha_1}{2} = 0.6$, $\frac{\alpha_2}{2} = 0.1$, respectively. In addition, for both SoCo and SVD-MFN, we integrate four context factors (*i.e.*, app-category, location, time-slot and social relationship) into each of them according to [76] and [45]. For SoCo, we empirically set the number of decision trees to 4 and the height of each tree to 1. For SVD-MFN, we empirically set the contribution factors [45] of those four context factors are 0.467, 0.003, 0.031 and 0.755 successively. Besides, for KNN, we set the number of nearest neighbors $K = 20$. In addition, users may use some popular apps in different locations during the whole data, like Wechat (the most popular social communication app in our dataset). Since such app's usage behavior actually cannot reflect personal interest and unique spatiotemporal context features, we exclude the top 30 apps and pay more attention to context-aware app usage behavior prediction. Note that we have 2000 apps in our dataset, so removing the top 30 apps will not make big difference on our evaluation results.

5.4 Performance in Different Data Sparsity Levels

Before we show the performance of our model in varying data sparsity levels, we first introduce how we divide our dataset into training set and testing set because the ratio of training set indeed reflects the data sparsity. In order to assure the fairness of each experiment, we choose to keep the size of testing data the same for different training ratios. Thus, based on 5-fold cross validation, we randomly split the whole scenarios of each user into 5 folds. Then each time, for one fold, we regard the scenarios in it as the unobserved ones, while the rest 4 folds data as the **100%** training data, from which we randomly choose a certain ratio (*e.g.*, 30% of **100%**, while the rest 70% of **100%** unused) scenarios as the observed scenarios for training. In our case, our aim is to predict user's app usage preference in unobserved scenarios. In order to simulate different levels of data sparsity, we choose different ratios of observed scenarios to form the training set. In our experiment, we select five different ratios of training set: 30%, 40%, 50%, 60% and 70%. The smaller the ratio is, the higher the level of data sparsity will be. For example, if the ratio of training set (*i.e.*, the ratio of observed data) is 30%, we regard the sparsity level as 70% . To measure the quality of our prediction, we adopt three metrics, *i.e.*, *Top*3 *Accuracy*, *Top*5 *Accuracy*, $nDCG_5$, to evaluate the performance of our model. The results are shown in Figure 6.

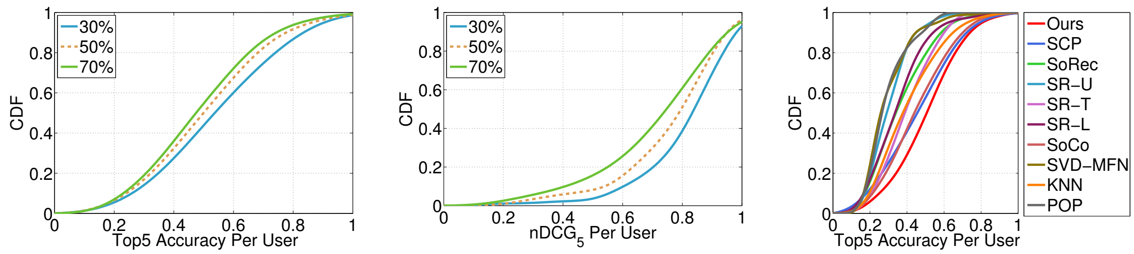


(a) Top3 Accuracy (b) Top5 Accuracy (c) $nDCG_5$

Fig. 6. The performance comparison between our method and several baselines under different data sparsity levels.

From the results, we can observe that in all sparsity levels, the prediction accuracy of our model (Ours) is always the highest, showing that our prediction model outperforms other baselines. In Figure 6 (a), taking the metric *Top*3 *Accuracy* for example, our model can achieve at least 23% improvement with 40% sparsity level, 34% improvement with 50% sparsity level and 45% improvement with 60% sparsity level. Likewise, for *Top*5 *Accuracy*, when the sparsity increases from 40% to 60%, our model's improvement increases from 20% to 42%. For $nDCG_5$, our model provides at least 5% improvement with 30% sparsity level and 10% improvement with 50% sparsity level. Thus, we can see that the prediction performance can get a rapid boost when the data is very sparse. This trend indicates that our auxiliary offline POI visiting information becomes more helpful in high sparsity scenario. In addition, comparing SR-T with SR-U and SR-L, we can observe that with 50% sparsity level, *Top*3 *Accuracy* of these three baselines are 24.5%, 16.8% and 14.7%, respectively. The difference between these three baselines is that they introduce different types of context, *i.e.*, time context, social context and location context. Comparing their prediction performances, we can find that auxiliary information of different types of context may have different effects on the prediction quality. According to the results, external time context information seems to be more useful in our task. Furthermore, the prediction accuracy of each of these three baselines is lower than our model, indicating that in such multi-context prediction problem, only introducing a single type of external context information as auxiliary information is not enough. Besides, we also observe that the performances of SoCo and SVD-MFN are lower than Ours at all sparsity levels. For example, with 50% sparsity level, the *Top*5 *Accuracy* of Ours is 35% higher than SoCo and 1.8 times higher than SVD-MFN. This indicates that it is more effective to directly integrate the context information into higher-order dimension tensor than just utilize it as an auxiliary tool for 2D user-item matrix factorization. This also demonstrates that with the same context information, our model can learn the user preference better.

In addition, in order to show personalization (*i.e.*, the detailed performance of different users), we also show the distribution of the prediction results for different users. As shown in Figure 7, we present several CDFs from several different aspects. **First**, Figure 7 (a) shows the individual's *Top*5 *Accuracy* of our model at three different sparsity levels. We can observe that the *Top*5 *Accuracy* value is from 0 (close to 0) to 1 (close to 1), which indicates that different users have different prediction performance. This shows that for different users, the difficulty to learn their preference is different. Besides, we can also see that when the sparsity level changes from 30% to 70%, the percentage of users whose *Top*5 *Accuracy* is over 60% changes from 60% to 71%. With the sparsity level increasing, the CDF curve gradually moves up. This shows that it becomes more difficult to learn the user's



(a) Individual Top5 Accuracy of Our Model (b) Individual $nDCG_5$ of Our Model (c) Individual Top5 Accuracy of Different Models

Fig. 7. Individual performance of different models.

preference with less observed data for training. **Second**, like Figure 7 (a), Figure 7 (b) shows the individual's $nDCG_5$ of our model at three different sparsity levels. We can also see the trend that with the sparsity level increasing, the CDF curve gradually moves up. For example, when the sparsity level changes from 30% to 70%, the percentage of users whose $nDCG_5$ is over 80% changes from 40% to 60%. **Third**, in order to compare the individual performance of our model with other methods', we present their CDFs of $Top5\ Accuracy$ per user with 50% sparsity level. We can see that the CDF of our model is at the bottom, which means that with the same $Top5\ Accuracy$ value, we have the most users whose $Top5\ Accuracy$ is over it. This indicates that although at the individual level, our model can provide better prediction quality than other methods. Therefore, from the perspective of individual prediction performance, our model can also preforms better than other baselines.

In addition, in order to measure the effect of the number of excluded popular apps on the online activity prediction quality, we also compare the performance of our model (Ours) with SCP in the case that different numbers of apps are excluded. The results are shown in Table 1. First, when the number of excluded popular apps ranges from 10 to 30, we can see that our model's performances are better than SCP with 70% sparsity level (*i.e.*, 30% observed data). Second, when we exclude top 10 popular apps, our performance gain still exists with 50% sparsity level. When the number of excluded apps increases to 30, our performance gain still exists with 30% sparsity level. The performance gain means that auxiliary check-in data has a positive impact in our online activity prediction task. Since the auxiliary check-in data contains rich semantic information of spatial and temporal contexts, it is reasonable that the auxiliary data becomes more helpful when we remove a bit more popular apps and the usage behavior of the rest apps shows a stronger personalized context-aware characteristic. Above results explicitly show the conditions that our transfer learning model will be more helpful.

In a short summary, the evaluation results have shown that our collaborative tensor factorization model has the best performance in different sparsity levels. In particular, our performance gain will be larger in higher sparsity scenario, showing that our auxiliary POI visiting information is helpful to provide external information of time and location contexts to make up for the sparsity. In addition, we find that when excluding a bit more popular apps, the auxiliary data will be more helpful to improve the online activity prediction quality, for usage behavior of the rest apps shows a stronger personalized context-aware characteristic.

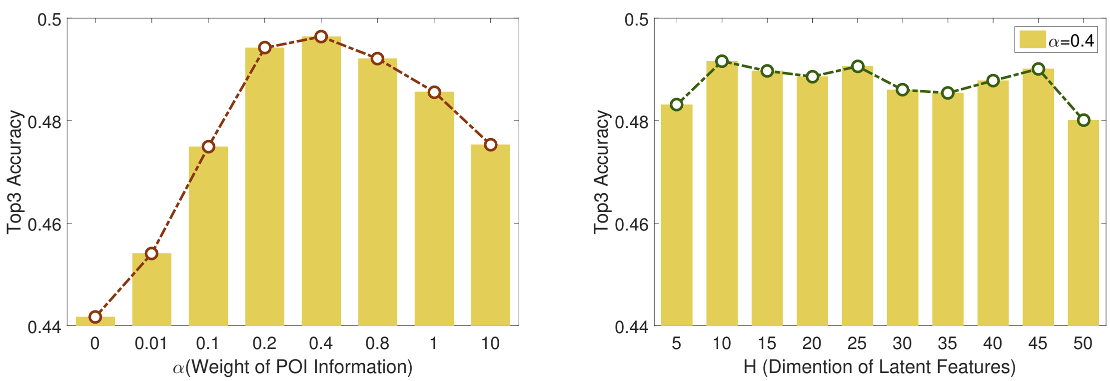
5.5 The Impact of Different Hyper-parameters

In our study, there are several hyper-parameters as mentioned before. Here we consider two important hyper-parameters (α and H) and evaluate their impact in our model. α reflects the transfer weight of POI information from Check-in dataset, and H represents the dimension of latent feature vectors. In order to measure the effect of each hyper-parameter, we experiment with a sequence of settings and compare their corresponding prediction

Table 1. The comparison of prediction accuracy when excluding different numbers of the most popular apps.

Number of Excluded Apps	Method	Top5 Accuracy				
		Sparsity Level				
		30%	40%	50%	60%	70%
10	Ours	0.5759	0.5693	**0.5558**	0.5517	**0.5841**
	SCP	0.5736	0.5658	**0.5443**	0.5064	**0.4612**
20	Ours	0.5755	**0.5680**	0.5605	0.5453	**0.5279**
	SCP	0.5610	**0.5576**	0.5325	0.4890	**0.4172**
30	Ours	**0.5358**	0.5230	0.5140	0.5013	**0.4474**
	SCP	**0.4787**	0.4360	0.3818	0.2782	**0.1721**

accuracy. When we change the target hyper-parameter's value, we keep the values of other hyper-parameters unchanged. In this experiment, we set the ratio of training set to be 50% and select *Top*3 *Accuracy* as the metric. The results are shown in Figure 8 (a) and Figure 8 (b) to show the impact of α and K, respectively.



(a) The impact of α (b) The impact of H

Fig. 8. (a) The impact of transfer weight on the prediction accuracy. (b) The impact of dimension of latent features on the prediction accuracy.

First, let's look at Figure 8 (a). We can observe that *Top*3 *Accuracy* grows at the beginning then decreases when α gradually increases from 0 to 10. It may be when α is very small, the model cannot fully utilize the POI information to capture spatiotemporal semantics and the relationship among different spatiotemporal contexts. When α becomes larger, the POI information gradually becomes dominate rather than the app usage information. When $\alpha = 0.4$, the influence of these two kinds of information becomes balanced and our model achieves the best performance. Therefore, in our model, we set $\alpha = 0.4$. Second, Figure 8 (b) shows the effect of the dimension of latent feature vectors H. We can observe that the *Top*3 *Accuracy* does not exhibit significant change when the value of H varies from 0 to 50 and our model performs equally well under various H values, which indicates the robustness of our model. Therefore, we roughly choose $H = 10$ which makes our model work best.

To sum up, in this section we determine our hyper-parameters through experimenting with a sequence of settings. These experimental results show that POI information is helpful to improve the prediction accuracy and our model is very robust under various values of dimension of latent vectors.

5.6 The Influence of Contexts and User Factor

So far, based on the fine-grained data, we have verified that our proposed model outperforms the state-of-art algorithms under different sparsity levels, specially in high sparsity scenario. Now we look at the influence of different contexts and factors on our model. Specifically, how the prediction quality will be for locations with different functions, different time periods, and different groups of users? Since our data contains information of multiple dimensions (user, location and time), we extract different contexts and factors from these three dimensions, and evaluate the prediction performance in different scenarios, the results are shown in Figure 9 and Figure 10.

First, for users, many previous works have studied the relation between user's attributes and their mobility patterns, showing that the user's habit can be reflected from her mobility pattern. For example, the range of living area where the user has appeared can reflect her mobility pattern. In detail, if the user lives in the suburbs while working in the city or if the user likes to participate in a variety of activities, the range of her living area will be very large. While if her work place is very close to her living place, the range will be very small. According to the feature, in our study we divide users into several group. Generally, user's online activity patterns may vary among people with different moving ranges. For instance, those who have large moving ranges are more likely to use travel apps and shopping apps. In our experiment, we adopt the metric *Radius of Gyration* [43] to measure the living range of a user. For an individual, the *Radius of Gyration* of her trajectory is calculated as follows:

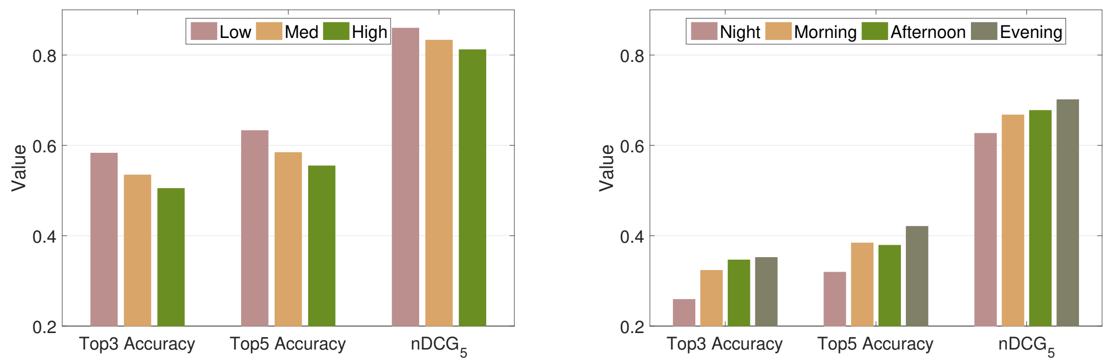
$$R_g = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (r_i - r_{\text{mean}})}, \tag{10}$$

where N denotes the number of the visited locations, r_i represents the position of the i-th location and r_{mean} is the mean position of all the visited locations. Note that a larger radius of gyration value means a larger living range of the user.

According to user's radius of gyration values, we divide them into three groups and all the groups have equal number of users. We use "Low", "Med" and "High" to represent user groups with different radii of gyration. The online activity prediction results of these three user groups are shown in Figure 9 (a). We can observe that when the user group varies from "Low" to "High", the $Top3$ $Accuracy$ ranges from 63% to 55%, showing that when the user's moving range becomes smaller, the prediction performance of our model will be better. Such trend also exists in $Top5$ $Accuracy$ and $nDCG_5$, which indicates that it is easier to predict app usage preference of those users with small moving ranges. This may be for the users with small moving ranges, their mobility patterns are relatively stable so that their online interaction with apps is correlated stronger with the locations they appear thus their personal spatiotemporal activity preferences are easier to learn. While for the users with large moving ranges, they appear in more diverse types of locations and have more complicated interactions, thus it will be more difficult to learn their spatiotemporal online activity interest.

Second, for time context, we divide one day into 4 time periods, *i.e.*, night 0:00-6:00, morning 6:00-12:00, afternoon 12:00-18:00 and evening 18:00-24:00. The online activity prediction results during these 4 time periods are shown in Figure 9 (b). We can observe that the $Top5$ $Accuracy$ of these four time periods are 31.9%, 38.4%, 37.9% and 42.1%, respectively, showing the prediction performance in the evening is better than the other three time periods, especially night. We think the reason may be that at night, users may have urgent tasks to address and need to make a sudden response, so the similarity of app behavior among users will be very low at night. While in the evening, most of the users are in leisure state, and the similarity of app behavior among users will be very strong.

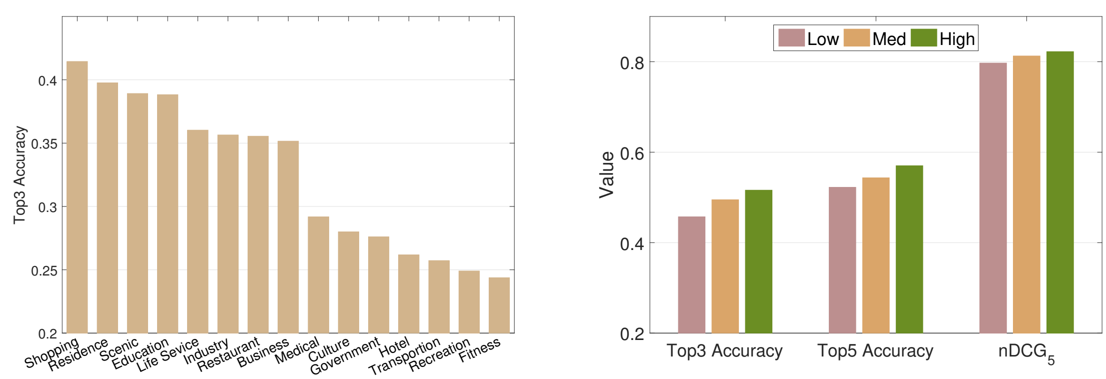
Third, for location context, considering the POI information of the check-in dataset, we try to group locations in two different ways. On the one hand, we want to compare the prediction quality of different functional areas because their online activity patterns may be different. For example, in entertainment areas, user's activities are



(a) User factor

(b) Time context

Fig. 9. (a) The prediction performance of our model for different user groups. (b) The prediction performance of our model under different time contexts.



(a) Location context with different functions

(b) Location context with different entropy

Fig. 10. The prediction performance of our model under different location contexts, regarding the function and entropy.

very diverse and they may use various kinds of apps to complete various tasks, such as shopping, watching videos or having a meal. However, in scenic areas, users may use travel apps and photo apps more often. To this end, intuitively, we can label each location by the category of its most prevalent POI. Thus, we group all the locations into 15 functional areas, *i.e.*, *shopping, residence, scenic, education, life service, industry, restaurant, business, medical service, culture, government, hotel, transportation, recreation* and *fitness*, respectively. The online activity prediction results of these 15 function areas are shown in Figure 10 (a). We can observe that the prediction performances of different functional areas are very different, ranging from 24% to 41%. In addition, the top 4 functional areas with the best performance are *shopping, residence, scenic* and *education*, while the top 4 functional areas with the lowest prediction accuracy are *hotel, transportation, recreation* and *fitness*.

Furthermore, for a location, the entropy of distribution of category-level POI visiting frequency is another metric to represent the location context. We explore how our model will perform under such kind of different location context, so we evaluate our model's performance in locations with different POI entropy values. In the previous location division scheme, we just simply adopt the most popular POI category as the location's function label. Here we take the visiting frequency of other POI categories into consideration, so we consider the location's POI entropy, which is calculated as follows:

$$E_l = -\sum_{i=1}^{C_l} \left(\frac{N_{l,i}}{\sum_{j=1}^{C_l} N_{l,j}} \right) \log \left(\frac{N_{l,i}}{\sum_{j=1}^{C_l} N_{l,j}} \right), \forall l = 1, \ldots, N_L, \tag{11}$$

where E_l denotes the entropy of location l based on distribution of category-level POI visiting frequency, C_l denotes the number of POI categories of location l, $N_{l,i}$ refers to the number of i-th categorized POI in location l and N_L denotes the number of locations.

According to the entropy calculation results, we group these locations into 3 groups, *i.e.*, Low, Med, High, *e.g.*, "Low" represents the location group with small entropy values. The online activity prediction results of these groups are shown in Figure 10 (b). We can observe that when the location group varies from "Low" to "High" entropy values, their $Top5\ Accuracy$ values range from 52.3% to 57.0%, showing that there exists an increasing trend when the entropy increases. There are similar trends for the metric $Top3\ Accuracy$ and $nDCG_5$. These results means that the larger the POI entropy of a location is, the more accurate the prediction result will be.

In a short summary, we investigate the prediction quality under different kinds of contexts and factors. In terms of user factor, our model performs better for users with small moving ranges, for they will have interaction with less types of locations and their activity behavior is easier to predict. For temporal context, the prediction accuracy of user's online activity behavior is better in the evening, mainly because most of the users are in leisure state in the evening and they will have more similar app usage behavior. For spatial context, we study its impact from two different aspects: first, we find online activity prediction in locations with shopping, residence, scenic functions is more accurate than that in other functional areas; second, if a location's category-level POI entropy is very large, online activity prediction accuracy of users in that location will be also higher.

In conclusion, we evaluate our method's performance under a variety of scenarios and explore the factors influencing it. First, considering different sparsity levels, our model can always outperforms several state-of-art algorithms, especially in high sparsity scenario. In addition, we also evaluate the impact of hyper-parameters to show our prediction method is quite robust. Besides, we also investigate the influence of user factor and different spatial and temporal contexts. All these results indicate that our collaborative tensor factorization method is very effective and the auxiliary check-in data plays an important role in improving the performance of our context-aware online activity prediction task.

6 DISCUSSION

In this section, we will discuss the following three important issues. We will first discuss about the benefit of combining online activity and offline activity, which has inspired our study. Then, we will discuss several potential applications of our work and our contribution to the UbiComp community.

6.1 Benefit of Combining Online Activity with Offline Activity

In our paper, our extensive evaluation results have shown that it is effective to transfer knowledge from offline check-in data to improve the prediction performance of user's online app usage activity, which has proved the benefit of combining online activity and offline activity. Indeed, some other researchers also benefited from combining offline activity (*e.g.* check-in records, mobility trajectories, traffic flow) with online app usage activity, such as improving app recommendation performance [54], solving cold-start problems [72], studying area economic development level [34] and so on. In addition, besides online app usage behavior, other types of online activity also can be combined with users' offline activity to benefit some other tasks, such as attendance click [45, 76]. All the above works have clearly demonstrated the mutual influence factor between users' online activity and offline activity, and their good performances have verified that there is a strong correlation between users' online activity and offline activity. The underlying reason relies on that whether it is online activity or offline activity, it is the embodiment of the user's current state and personal interests. In other words, we can learn users' interests or preferences from any of them. As a result, when dealing with tasks about online activity or offline activity, we can turn to the other one for help and transfer knowledge between them. This may provide a new angle to solve problems in online activity or offline activity domain.

6.2 Potential Applications

Our study has many potential applications in practice. First of all, our scheme of utilizing check-in data to help online activity prediction is quite practical for check-in datasets are easy to access, and there are also a number of public check-in datasets, such as Gowalla and Foursquare. Such available auxiliary datasets guarantee the feasibility of our method, which is the basis of further potential applications: First, from the user perspective, with the predicted app list that a user will use, we can prepare the apps in the background in advance, which can save response time and battery power [48, 65]. Second, for advertisers, after knowing what apps one use, they can choose different apps for different users at the right time [66] to provide better recommendations. Third, for mobile operators, they can provide personalized traffic plans for different users according to their spatiotemporal app usage patterns.

6.3 Contribution to the UbiComp Community

During the past few years, the research for online activity has attracted more and more researchers, and become an important research area of ubiquitous computing and a meaningful topic in the UbiComp community and beyond. At the same time, there exist many previous works exploring the relationship between individual's online activity and offline activity, such as [54, 55, 57, 72, 75, 78]. Our work goes along with this topic and takes a further step towards the benefits of combining offline check-in activity and online app usage activity for personalized context-aware prediction task, which poses a brand-new angle of solving personalized context-aware activity prediction problem.

7 CONCLUSION

In this paper, we aim to study personalized context-aware online activity prediction problem with the help of offline check-in data. Combined the user-location-time-activity 4D-tensor with location-time-POI 3D-tensor, we propose a transfer learning based collaborative tensor factorization method, which transfers the semantics of location and time contexts among them and achieves personalized context-aware online activity prediction. Based on real-world datasets, we evaluate our method's performance and find it outperforms the other state-of-the-arts in high sparsity scenario. Moreover, we also demonstrate that user factor, time and location contexts have quite different impacts on online activity prediction performance. Our study is the first step forward for transferring knowledge learned from offline activity behavior to online activity prediction for providing better personalized context-aware recommendation services for mobile users.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under grant SQ2018YFB180012, the States Key Project of Research and Development Plan under grant 2017YFE0121300-6, the

National Nature Science Foundation of China under 61971267, 61972223, 61861136003, 61621091 and 61673237, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, the research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology, and the MOE-CMCC Joint Research Fund of China under MCM20160101.

REFERENCES

[1] Franz Aurenhammer. 1991. Voronoi diagramsâĂŤa survey of a fundamental geometric data structure. *Acm Computing Surveys* 23, 3 (1991), 345–405.

[2] Upasna Bhandari, Kazunari Sugiyama, Anindya Datta, and Rajni Jindal. 2013. Serendipitous recommendation for mobile apps using item-item similarity graph. In *Asia Information Retrieval Symposium*. Springer, 440–451.

[3] Preeti Bhargava, Thomas Phan, Jiayu Zhou, and Juhan Lee. 2015. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 130–140.

[4] Jennifer Blaze, Arun Asok, and Tania L. Roth. 2014. Content-based tag propagation and tensor factorization for personalized item recommendation based on social tagging. *Acm Transactions on Interactive Intelligent Systems* 3, 4 (2014), 26.

[5] Matthias Böhmer, Lyubomir Ganev, and Antonio Krüger. 2013. Appfunnel: A framework for usage-centric evaluation of recommender systems that suggest mobile applications. In *Proceedings of the 2013 international conference on Intelligent user interfaces*. ACM, 267–276.

[6] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun Ichi Amari. 2009. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*.

[7] Enrique Costa-Montenegro, Ana Belén Barragáns-Martínez, and Marta Rey-López. 2012. Which App? A recommender system of applications in markets: Implementation of the service for monitoring usersâĂŹ interaction. *Expert systems with applications* 39, 10 (2012), 9367–9375.

[8] Tiago Cunha, Carlos Soares, and AndrÃľ C. P. L. F. Carvalho. 2017. Metalearning for Context-aware Filtering: Selection of Tensor Factorization Algorithms. In *Eleventh Acm Conference on Recommender Systems*.

[9] Trinh Minh Tri Do, Jan Blom, and Daniel Gatica-Perez. 2011. Smartphone usage in the wild: a large-scale analysis of applications and context. In *Proc. ACM ICMI*. 353–360.

[10] Mian Dong and Zhong Lin. 2011. Sesame: Self-Constructive System Energy Modeling for Battery-Powered Mobile Systems. (2011).

[11] Xixi Du, Huafeng Liu, and Liping Jing. 2017. Additive Co-Clustering with Social Influence for Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*.

[12] Peter Eckersley. 2010. How Unique Is Your Web Browser? *Lecture Notes in Computer Science* 6205 (2010), 1–18.

[13] David Elsweiler, Morgan Harvey, and Martin Hacker. 2011. Understanding re-finding behavior in naturalistic email interaction logs. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 35–44.

[14] Elena Viorica Epure, Benjamin Kille, Jon Espen Ingvaldsen, Rebecca Deneckere, Camille Salinesi, and Sahin Albayrak. 2017. Recommending Personalized News in Short User Sessions. (2017).

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

[16] Wang Fei, Zhang Zhe, Hailong Sun, Richong Zhang, and Xudong Liu. 2013. A Cooperation Based Metric for Mobile Applications Recommendation. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence*.

[17] Andrea Girardello and Florian Michahelles. 2010. AppAware: which mobile applications are hot?. In *Conference on Human-computer Interaction with Mobile Devices Services*.

[18] Keith Hampton, Lauren Sessions Goulet, Lee Rainie, and Kristen Purcell. 2011. Social networking sites and our lives. *Pew Internet & American Life Project* 16 (2011), 1–85.

[19] Yong Jin Han, Seong Bae Park, and Se Young Park. 2017. Personalized App Recommendation Using Spatio-Temporal App Usage Log. *Inform. Process. Lett.* 124 (2017), 15–20.

[20] Ma Hao, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: Social recommendation using probabilistic matrix factorization.

[21] R. A Harshman. 1970. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-model factor analysis. *Ucla Working Papers in Phonetics* 16 (1970).

[22] Cao Hong and Lin Miao. 2017. Mining smartphone data for app usage prediction and recommendations: A survey. *Pervasive Mobile Computing* 37 (2017), 1–22.

[23] Minsung Hong and Jason J. Jung. 2018. Multi-Sided Recommendation based on Social Tensor Factorization. *Information Sciences* 447 (2018), S0020025518301968.

[24] Junxian Huang, Feng Qian, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. 2012. A close examination of performance and power characteristics of 4G LTE networks. In *International Conference on Mobile Systems*.

[25] 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.

[26] C. G. Khatri and C. Radhakrishna Rao. 1968. Solutions to Some Functional Equations and Their Applications to Characterization of Probability Distributions. *SankhyÄĄ: The Indian Journal of Statistics, Series A (1961-2002)* 30, 2 (1968), 167–180.

[27] Kyunghan Lee, Injong Rhee, Joohyun Lee, Yung Yi, and Chong Song. 2010. Mobile Data Offloading: How Much Can WiFi Deliver?. In *International Conference*.

[28] Chen Lin, Runquan Xie, Xinjun Guan, Lei Li, and Tao Li. 2014. Personalized news recommendation via implicit social experts. *Information Sciences* 254 (2014), 1–18.

[29] Jovian Lin, Kazunari Sugiyama, Min Yen Kan, and Tat Seng Chua. 2013. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *International Acm Sigir Conference on Research Development in Information Retrieval*.

[30] Kuan-Yu Lin and Hsi-Peng Lu. 2011. Why people use social networking sites: An empirical study integrating network externalities and motivation theory. *Computers in human behavior* 27, 3 (2011), 1152–1161.

[31] Bin Liu, Deguang Kong, Cen Lei, Neil Zhenqiang Gong, and Xiong Hui. 2015. Personalized Mobile App Recommendation: Reconciling App Functionality and User Privacy Preference. In *Eighth Acm International Conference on Web Search Data Mining*.

[32] Duen-Ren Liu, Pei-Yun Tsai, and Po-Huan Chiu. 2011. Personalized recommendation of popular blog articles for mobile applications. *Information Sciences* 181, 9 (2011), 1552–1572.

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

[34] S. Luo, F Morone, C Sarraute, M Travizano, and H. A. Makse. 2017. Inferring personal economic status from social network location. *Nature Communications* 8 (2017), 15227.

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

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

[37] Stathis Maroulis, Ioannis Boutsis, and Vana Kalogeraki. 2016. Context-aware point of interest recommendation using tensor factorization. In *IEEE International Conference on Big Data*.

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

[39] Byung-Won On, Ee-Peng Lim, Jing Jiang, Amruta Purandare, and Loo-Nin Teow. 2010. Mining interaction behaviors for email reply order prediction. In *2010 International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 306–310.

[40] Weike Pan and Qiang Yang. 2013. Transfer learning in heterogeneous collaborative filtering domains. *Artificial intelligence* 197 (2013), 39–55.

[41] Liu Qiang, Wu Shu, Wang Liang, and Tieniu Tan. 2016. Predicting the next location: a recurrent model with spatial and temporal contexts. In *Thirtieth Aaai Conference on Artificial Intelligence*.

[42] Dimitrios Rafailidis and Petros Daras. 2013. The TFC Model: Tensor Factorization and Tag Clustering for Item Recommendation in Social Tagging Systems. *IEEE Transactions on Systems Man Cybernetics Systems* 43, 3 (2013), 673–688.

[43] A. Ravve. 1982. *Principles of Polymer Chemistry*.

[44] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 635–644.

[45] Du Rong, Zhiwen Yu, Mei Tao, Zhitao Wang, and Bin Guo. 2014. Predicting activity attendance in event-based social networks: Content, context and social influence. In *Acm International Joint Conference on Pervasive Ubiquitous Computing*.

[46] Elaine Shi, Niu Yuan, Markus Jakobsson, and Richard Chow. 2011. *Implicit Authentication through Learning User Behavior*.

[47] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Alan Hanjalic, and Nuria Oliver. 2012. TFMAP: optimizing MAP for top-n context-aware recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 155–164.

[48] Choonsung Shin, Jin-Hyuk Hong, and Anind K. Dey. 2012. Understanding and prediction of mobile application usage for smart phones. In *Proc. ACM Ubicomp*.

[49] Jessica Su, Ansh Shukla, Sharad Goel, and Arvind Narayanan. 2017. De-anonymizing Web Browsing Data with Social Networks. In *Proceedings of the 17th international conference on World Wide Web (WWW)*. 1261–1269.

[50] Zhou Su, Qichao Xu, Fen Hou, Qing Yang, and Qifan Qi. 2017. Edge caching for layered video contents in mobile social networks. *IEEE Transactions on Multimedia* 19, 10 (2017), 2210–2221.

[51] Panagiotis Symeonidis, Alexis Papadimitriou, Yannis Manolopoulos, Pinar Senkul, and Ismail Toroslu. 2011. Geo-social recommendations based on incremental tensor reduction and local path traversal. In *Acm Sigspatial International Workshop on Location-based Social Networks*.

[52] Vincent F. Taylor, Riccardo Spolaor, Mauro Conti, and Ivan Martinovic. 2017. Robust Smartphone App Identification via Encrypted Network Traffic Analysis. *IEEE Transactions on Information Forensics Security* 13, 1 (2017), 63–78.

[53] Domonkos Tikk. 2012. Fast ALS-Based tensor factorization for context-aware recommendation from implicit feedback. In *Joint European Conference on Machine Learning Knowledge Discovery in Databases*.

[54] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 26.

[55] 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. *Proc. ACM UbiComp* 2, 3 (2018), 138.

[56] Ledyard R Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31, 3 (1966), 279–311.

[57] Pascal Welke, Ionut Andone, Konrad Blaszkiewicz, and Alexander Markowetz. 2016. Differentiating Smartphone Users by App Usage. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*.

[58] Wikipedia. 2017. *Jaccard index*. https://en.wikipedia.org/wiki/Jaccard_index.

[59] Wolfgang Woerndl, Christian Schueller, and Rolf Wojtech. 2007. A hybrid recommender system for context-aware recommendations of mobile applications. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*. IEEE, 871–878.

[60] Tong Xia and Yong Li. 2019. Revealing Urban Dynamics by Learning Online and Offline Behaviours Together. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 1 (March 2019), 30:1–30:25.

[61] Lin Xiao, Zhang Min, Zhang Yongfeng, Liu Yiqun, and Ma Shaoping. 2017. Learning and transferring social and item visibilities for personalized recommendation. In *Proc. ACM CIKM*. ACM, 337–346.

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

[63] Yanan Xu, Yanmin Zhu, Yanyan Shen, and Jiadi Yu. 2018. Leveraging app usage contexts for app recommendation: a neural approach. *World Wide Web* 8 (2018), 1–25.

[64] Bo Yan and Guanling Chen. 2011. AppJoy: Personalized Mobile Application Discovery. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*. ACM, New York, NY, USA, 113–126. https://doi.org/10.1145/1999995.2000007

[65] Tingxin Yan, David Chu, Deepak Ganesan, Aman Kansal, and Liu Jie. 2012. Fast app launching for mobile devices using predictive user context. In *International Conference on Mobile Systems*.

[66] Zhixian Yan, Lai Wei, Yunshan Lu, Zhongqiang Wu, and Bo Tao. 2017. You Are What Apps You Use: Transfer Learning for Personalized Content and Ad Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*.

[67] Lina Yao, Quan Z. Sheng, Yongrui Qin, Xianzhi Wang, Ali Shemshadi, and Qi He. 2015. Context-aware Point-of-Interest Recommendation Using Tensor Factorization with Social Regularization. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*.

[68] Ting Fang Yen, Yinglian Xie, Fang Yu, Roger Peng Yu, and Martin Abadi. 2012. Host Fingerprinting and Tracking on the Web:Privacy and Security Implications. 11, 1 (2012), 111 – 124.

[69] Hongzhi Yin, Chen Liang, Weiqing Wang, Xingzhong Du, Quoc Viet Hung Nguyen, and Xiaofang Zhou. 2017. Mobi-SAGE: A Sparse Additive Generative Model for Mobile App Recommendation. In *IEEE International Conference on Data Engineering*.

[70] Peifeng Yin, Luo Ping, Wang Chien Lee, and Wang Min. 2013. App recommendation: A contest between satisfaction and temptation. In *Acm International Conference on Web Search Data Mining*.

[71] Yuankai Ying, Chen Ling, and Gencai Chen. 2017. *A temporal-aware POI recommendation system using context-aware tensor decomposition and weighted HITS*.

[72] 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.

[73] Kuifei Yu, Baoxian Zhang, Hengshu Zhu, Huanhuan Cao, and Jilei Tian. 2012. Towards personalized context-aware recommendation by mining context logs through topic models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 431–443.

[74] Duoduo Zhang, Yang Ning, and Yuchi Ma. 2016. *Explicable Location Prediction Based on Preference Tensor Model*.

[75] Shuo Zhang, Khaled Alanezi, Mike Gartrell, Richard Han, Lv Qin, Shivakaht Mishra, Shuo Zhang, Khaled Alanezi, Mike Gartrell, and Richard Han. 2017. Understanding Group Event Scheduling via the OutWithFriendz Mobile Application. (2017).

[76] Shuo Zhang and Qin Lv. 2017. Hybrid EGU-based Group Event Participation Prediction in Event-based Social Networks. *Knowledge-Based Systems* 143 (2017), S0950705117305749.

[77] Sha Zhao, Zhiling Luo, Ziwen Jiang, Haiyan Wang, Feng Xu, Shijian Li, Jianwei Yin, and Gang Pan. 2012. AppUsage2Vec: Modeling Smartphone App Usage for Prediction. (2012).

[78] 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.

[79] 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.

[80] Vincent Wenchen Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative Filtering Meets Mobile Recommendation: A User-Centered Approach. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010.*

[81] Hengshu Zhu, Enhong Chen, Hui Xiong, Kuifei Yu, Huanhuan Cao, and Jilei Tian. 2014. Mining Mobile User Preferences for Personalized Context-Aware Recommendation. *ACM Trans. Intell. Syst. Technol.* 5, 4 (Dec. 2014).

[82] Hengshu Zhu, Enhong Chen, Kuifei Yu, Huanhuan Cao, Hui Xiong, and Jilei Tian. 2012. Mining personal context-aware preferences for mobile users. In *Proc. IEEE ICDM*. 1212–1217.

[83] Hengshu Zhu, Xiong Hui, Ge Yong, and Enhong Chen. 2014. Mobile app recommendations with security and privacy awareness. (2014).