CAP: Context-aware App Usage Prediction with Heterogeneous Graph Embedding

XINLEI CHEN, Carnegie Mellon University, USA
YU WANG, Carnegie Mellon University, USA
JIAYOU HE, Beijing Experimental High School Attached to Beijing Normal University, China
SHIJIA PAN, Carnegie Mellon University, USA
YONG LI, Tsinghua University, China
PEI ZHANG, Carnegie Mellon University, USA

Context-aware mobile application (App) usage prediction benefits a variety of applications such as precise bandwidth allocation, App launch acceleration, etc. Prior works have explored this topic through individual data profiles and contextual information. However, it is still a challenging problem because of the following three aspects: i. App usage behavior is usually influenced by multiple factors, especially temporal and spatial factors. ii. It is difficult to describe individuals' preferences, which are usually time-variant. iii. A single user's data is sparse on the spatial domain and only covers a limited number of locations. Prediction becomes more difficult when the user appears at a new location.

This paper presents *CAP*, a context-aware App usage prediction algorithm that takes both contextual information (location & time) and attribution (App with type information) into consideration. We find that the relationships between App-location, App-time, and App-App type are essential to prediction and propose a heterogeneous graph embedding algorithm to map them into the common comparable latent space. In addition, we create a user profile for each user with App usage and trajectory history to describe the individual dynamic preference for personalized prediction. We evaluate the performance of our proposed *CAP* with two large-scale real-world datasets. Extensive evaluations demonstrate that *CAP* achieves 30% higher accuracy than a state-of-the-art method Personalized Ranking Metric Embedding (*PRME*) in terms of Accuracy@5. In terms of mean reciprocal rank (MRR), *CAP* achieves 1.5× higher than the straightforward baseline *Sta* and 2× higher than *PRME*. Our investigation enables a range of applications to benefit from such timely predictions, including network operators, service providers, and etc.

CCS Concepts: • **Information systems → Spatial-temporal systems**; • **Human-centered computing → Mobile phones**; • **Computing methodologies → Machine learning algorithms**.

Additional Key Words and Phrases: Context Aware, Application Usage, Graph Embedding, Behaviour Modeling

Authors' addresses: Xinlei Chen, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, Pennsylvania, 15213, USA, xinlei.chen@sv.cmu.edu; Yu Wang, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, Pennsylvania, 15213, USA, yuwang4@adrew.cmu.edu; Jiayou He, Beijing Experimental High School Attached to Beijing Normal University, No.14, Erlong Road, XiCheng District, Beijing, Beijing, China, jiayouhejiayou@gmail.com; Shijia Pan, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, Pennsylvania, 15213, USA, shijiapan@cmu.edu; Yong Li, Tsinghua University, Beijing, China, liyong07@tsinghua.edu.cn; Pei Zhang, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, Pennsylvania, 15213, USA, peizhang@cmu.edu.

2474-9567/2019/3-ART4 $15.00
https://doi.org/10.1145/3314391

ACM Reference Format:
Xinlei Chen, Yu Wang, Jiayou He, Shijia Pan, Yong Li, and Pei Zhang. 2019. CAP: Context-aware App Usage Prediction with Heterogeneous Graph Embedding. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 1, Article 4 (March 2019), 25 pages.
https://doi.org/10.1145/3314391

1 INTRODUCTION

The smart device market has been showing continuous and rapid growth in the last decade. Smartphones alone show a projection of 2.53 billion worldwide users by 2020 [2]. These smart devices are mainly used with mobile Apps, which project revenues of around 189 billion US dollars by the year 2020 [1]. Currently, around 2.8 million and 2.2 million Apps have been developed and made available in Google Play and Apple App Store respectively.

With this explosive growth of the mobile App market, accurately predicting users' App usage is essential for carriers and consumers. Because the projection of mobile data usage by 2021 is 48 exabytes per month, carriers will need more accurate and dynamic bandwidth allocation schemes to increase bandwidth utilization efficiency [56]. Predicting consumers' App usage at specific times and locations helps carriers understand consumers' bandwidth needs more precisely for smart bandwidth allocation. For consumers, App usage prediction information not only helps accelerate App launching but also eases the inconvenience of searching for Apps. According to Yan etc., even simple Apps (e.g. weather report) need at least 10 seconds to reach a playable state. With prediction results, the smartphone is able to cache several potential Apps in memory, which decreases launch time and has been implemented by iOS, Android and WP [76]. In addition, to prevent too much energy waste, the mobile phone can decide pre-launching or not based on the accuracy of the App usage prediction. In [42], the authors find that the average number of Apps in a user's smartphone is around 56 and some users have up to 150 Apps. Such predictions enable the smartphone to show the potential several Apps on the main screen and users do not have to spend more time swiping screens and finding the Apps they want to use.

Prior works have attempted to predict mobile App usage [29, 32, 35]. Church et al. [24] summarized the challenges for mobile phone usage learning and analysis as well as a series of studies and applications on mobile phone usage, including App recommendation [88], launcher prediction [63], and battery management [31]. Various prediction algorithms have been explored to achieve that goal. Kostakos et al. [38] applied a Markov state transition model to predict the next screen event. Xu et al. [75] proposed a multi-faceted approach to predict App usage. The study focused on a small-scale dataset, posing a key challenge to understand and predict App usage behavior over a large user population. Shin et al. [63] predicted the App usage based on a personalized Naïve Bayes model for each user profiled from the usage data from their phones. Since their prediction is based on individual historical data and contextual information, it is limited by what a user has already experienced. In addition, various algorithms and information types have been explored for prediction and recommendation in different domains. The context-aware recommendation is usually achieved by using information about location, time, and activity [36, 47, 48, 92, 94]. The context-aware collaborative filtering and recurrent neural network are proposed for activity recommendation, App recommendation, and location prediction. In addition, Berkel et al. [71] looked into a different aspect of smartphone usage by classifying usage gaps to identify the user usage session. They also use user profile information to achieve personalized recommendations for news, blogs, Apps, and e-commerce items [11, 25, 43, 46, 59]. Until now, no research focuses on users' App usage prediction over a large scale population using both temporal and spatial information.

The goal of this paper is to consistently predict users' next App usage given time and location over a large scale user population. Despite the related work mentioned before, challenges remain. *i.* Mobile App usage behavior is complicated. What are the key factors that affect the prediction? It is also difficult to derive the importance of these factors. *ii.* A user's App usage preference is decided by multiple factors and time-variance, which is difficult to describe. *iii.* A user's data is sparse on the spatial domain. One user only covers a limited number of locations.

Prediction is difficult if the user appears at a new location. Accurate prediction is impossible without addressing these challenges.

In this work, we present the *CAP*, a context-aware App Usage prediction algorithm that handles the aforementioned challenges. To address the challenge *i.*, we propose a heterogeneous graph embedding algorithm, which maps time, location, App, and App type into one common latent space. The embedding catches the relationships between App-location, App-time, and App-App type. To address challenge the *ii.*, we design user profiles with users' past App usage and trajectory, which are affected by a time decay factor, to describe the individual dynamic preference. Finally, we adopt the history data of all users to construct a heterogeneous graph for an individual user to generate personal preference, which addresses the challenge *iii.* and keeps our prediction personalized. Our contributions are demonstrated as follows:

- We are the first to investigate the context-aware App-usage prediction problem over a large user population. We consider context information (time & location), attribute information (App & App type) and dynamic user preference.
- We find that the relationships between App-location, App-time, and App-App type are essential to prediction and propose a heterogeneous graph embedding algorithm to map them into one common comparable latent space. We propose a user profile with personal App usage & trajectory history affected by a time decay factor to achieve a personalized prediction. We extract both the common attribution of all users and individual user dynamic preferences to ensure sufficient training data without losing personalization.
- We evaluate our algorithm through two large-scale real-world datasets. The first one includes more than 6,000,000 mobile App usage records from 1788 individual users, while the second one includes 400,000 mobile App usage records with richer contextual information from 801 individual users. *CAP* demonstrates a significant improvement in the prediction accuracy compared to baselines.

The rest of the paper is organized as follows: Section 2 introduces the dataset collection and problem description. The algorithm design is described in Section 3. Then in Section 4, we evaluate our algorithm using a large-scale real-world dataset. Next, we discuss the related work in Section 5. Finally, we conclude this work in Section 6.

2 DATASET AND PROBLEM DESCRIPTION

This section introduces the preliminaries of the mobile App usage prediction. We first introduce two App usage record datasets used for prediction, one indirect App usage dataset from China Telecom and one direct App usage dataset from Talking Data platform. Then, we discuss the characteristics of the two datasets. Finally, we formally define the prediction problem.

2.1 Data Collection and Processing

2.1.1 Data Collected by China Telecom. The first App usage record dataset is collected with Deep Packet Inspection (DPI) appliances [4], through China Telecom, a major cellular network operator in China [3]. It records the spatiotemporal information of mobile subscribers when they access cellular network for App usage. Thus, the recorded locations are at the granularity of the cellular base station. In the dataset, each entry contains an anonymized user identification, timestamps of HTTP request or response, the length of the packet, the domain visited and the user-agent field. The data is collected in Shanghai, one of the largest city in China.

We extract the information of what App is used for network requests. In the HTTP header captured by our DPI, various fields are utilized as the identifiers of the Apps to communicate with their host servers or third party services. The hosting servers need to distinguish between different Apps in order to provide appropriate content. Therefore, we are able to identify the App making a network request by inspecting those HTTP header identifiers. We utilize a systematic framework for classifying network traffic generated by mobile Apps: SAMPLES [81]. It uses constructs of conjunctive rules against the App identifier found in a snippet of the HTTP header. The

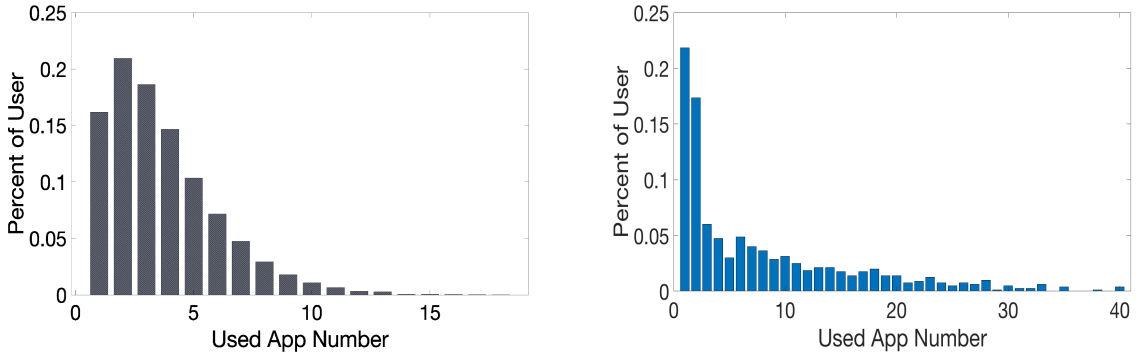
framework operates in an automated fashion through a supervised methodology over a set of labeled data streams. It has been shown to identify over 90% of these Apps with 99% accuracy on average [81]. In order to obtain the labeled dataset, we crawled the 2000 most popular Apps across Apple App Store (iOS Apps) and Google Play (Android Apps) and applies SAMPLES to generate conjunctive rules to match each App's network traffic. We manually verify the correctness of the matched Apps, which achieves about 97% accuracy. In addition, we first extract all cellular network connections for each App. Then, to avoid the repetitive count of App sessions, we adopt density-based spatial clustering of applications with noise (DBSCAN) to cluster the App sessions. Each cluster is regarded as one session [7].

In addition, we also adopt App types for attribute information. The App type represents the type of an App, which usually indicates its functionality and attribute. Each App is categorized into at least one of the nineteen App types, including game, video, news, social, E-shopping, finance, real estate, tourism, daily service, education, therapy, baby caring, taxi, vehicle relevance, music, map, reading, vogue, and office. We manually assign each App name an App ID and each App type name an App type ID for easy successive processing.

It is noticed that China Telecom dataset has an inherent limitation, which does not capture the App usage that makes network requests solely through WiFi or makes no requests. However, according to a recent report, Chinese users are accessing nearly 11 Apps daily on average [67], which is similar to the average number 9.2 in this dataset. This indicates that the number of Apps that do not request networks is non-trivial but negligible [86].

2.1.2 Data Collected by TalkingData. Since the first dataset indirectly indicates users' App usage behaviors, we adopt the second dataset, which directly reflects users' App usage behaviors. The second App usage record dataset is collected by TalkData, China's largest third-party mobile data platform [65]. The data is collected with TalkingData SDK integrated within mobile Apps, which runs background [66]. After agreeing on the terms, the users do not have to do anything when they are using mobile phones, which ensures no interruption on their App usage behavior. Full recognition and consent from individual users of those Apps have been obtained, and appropriate anonymization has been performed to protect privacy [65]. The data is collected on more than 50 phone brands. The SDK helps to record the App usage event on the mobile phone. Each event includes time, location (latitude and longitude), the App being run (foreground & background), the App foreground activation status, the App type, phone brand, etc. To keep spatial consistency, we first remove App usage events from the areas other than Shanghai. Then we map each (latitude, longitude) pair to a cellular tower ID, whose location is closest. Therefore, similar to China Telecom dataset, we also use cellular tower ID information as the location for TalkingData dataset. We also remove the inactive App usage events to track the real users' App usage. Thus, comparing to the first dataset, the second one has richer context information, which indicates the real users' activities when using the Apps.

2.1.3 Data Anonymization. It is worth pointing out that privacy issues of both datasets are carefully considered and measures are taken to protect the privacy of these mobile users. Both App usage record datasets do not contain any personally identifiable information, including age, gender, name, etc. The user identities have been anonymized as a bit string and do not contain any user meta-data. All of the individual data is stored in the provider's servers. It is the employees of the provider that process the raw data, and we as researchers only utilize aggregated pre-processed results for further analysis. Our research has been reviewed and approved by both the provider and our local university institutional board. China Data Protection Regulations (CDPR) are strictly followed. All the researchers are regulated by the strict non-disclosure agreement and the datasets are located in a secure offline server.



(a) Statistics of Data Collected by China Telecom

(b) Statistics of Data Collected by TalkingData Platform

Fig. 1. This figure shows statistics of mobile App usage number in 7 days from two different datasets.

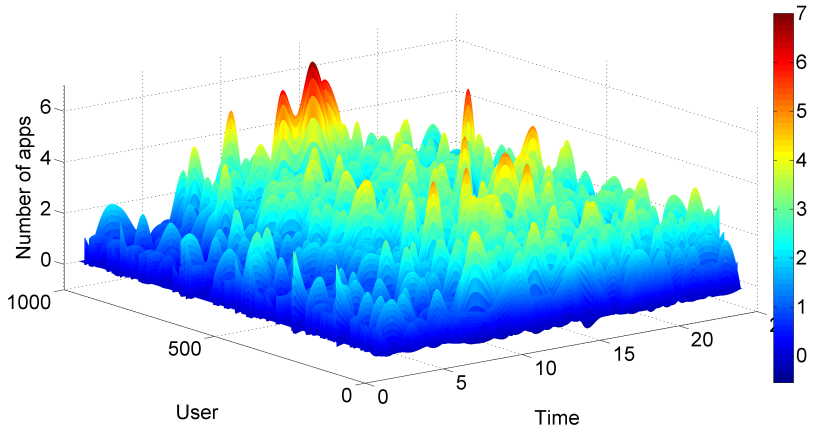
2.2 Dataset Characteristics

Figure 1 shows the statistics of the mobile App usage of two datasets. It is noticed that we first filter the original datasets and do the statistics. We filter out the records of two categories of Apps: 1) the Apps that are used at almost all locations and at all time; 2) the Apps that contains less than 5 records within the datasets. As results, 135 Apps are filtered from original 1633 Apps. As a result, 1498 Apps are remaining. In the China Telecom dataset, more than half of users in the China Telecom dataset access more than 4 Apps. This number increases to 5 in the TalkingData dataset. This is because the TalkingData SDK captures all App behaviors from users, while the App usage without cellular connection is missed in the China Telecom dataset.

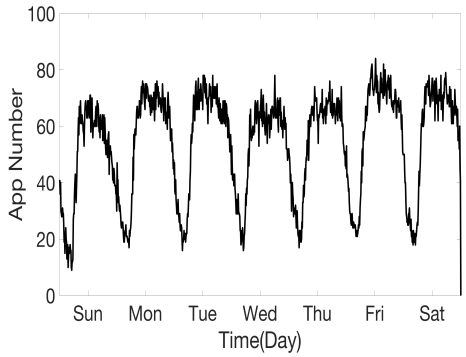
Figure 2(a) shows the number of mobile App used averaged over one week for 1000 randomly selected user at different time. The X axis and Y axis represent user id and time in a day respectively. Different users show variant mobile App usage number and peak number time. Figure 2(b) shows total mobile App usage number in 10 minutes used by same 1000 users. Mobile App usage is highly coupled with the human activity pattern, in which people use more Apps during daytime and fewer Apps during night. Figure 2(c) illustrates App types percent in different areas. The X axis and Y axis represent the App type percent and region types respectively. Apps of different types have different usage patterns in different region types. All these observations illustrate the potential to predict users' mobile App usage pattern with context information (location and time). In addition, mobile App usage is time-variant, user-variant and location-variant. As a result, it is necessary to figure out a novel solution to do context-aware mobile App usage prediction.

2.3 Problem Description

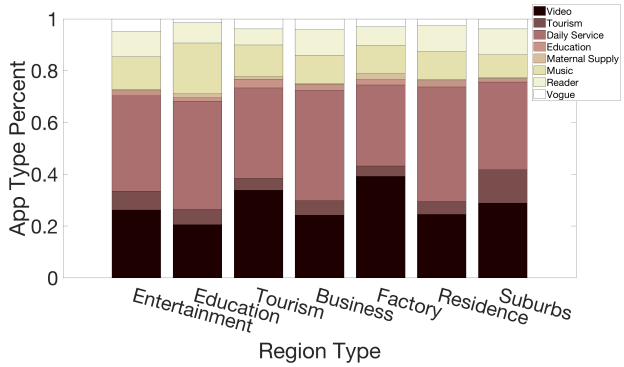
In order to predict context-aware App usage patterns, i.e. what App a user will use given the time and location, we first define the problem as follows. Let $\mathfrak{R}$ be a corpus of mobile user App usage records. We apply U, C, T, A, P to represent information of user identity, location, time, App and App type respectively. We use a subscript to denote the record id. For the kth record, it is a tuple $< U_k, C_k, T_k, A_k, P_k >$, where U_k and C_k are user ID and cellular tower ID, while T_k, A_k and P_k are time, App ID and App type ID. As mentioned in Section 2.1.2, to keep spatial consistency, we map each (latitude, longitude) pair in the TalkingData dataset to a cellular tower ID. As a result, the location information in both datasets is represented by the cellular tower ID. It is noticed that all records are sorted by ascending order of time. Smaller k means early time. We aim to correlate the mobile



(a) Mobile App usage number of 1000 users at different time



(b) Temporal distribution of total mobile App usage number with 10 minutes resolution



(c) Apps of different types have different usage patterns in different location types

Fig. 2. This figure shows (a) mobile App usage number of 1000 users at different time, (b) total number of App used every 10 minutes in one week, and (c) the usage of Apps at different types of locations.

App usage pattern to time and location with $\mathfrak{R}$, which consists of large amounts of App usage records. Given five different factors intertwined, an effective and fast model is needed to accurately capture the cross-modal correlation among C, T, A, P for each user.

Based on the App usage record dataset $\mathfrak{R}$, given a querying user u ($U_k = u$) with the context of time T_k and connected cellular towers C_k (query $q = (u, T_k, C_k)$), our algorithm predicts top N mobile Apps the user u will probably use. The prediction is based on the history mobile App usage of all users, i.e. all records in $\mathfrak{R}$, whose time is earlier than T_k. The algorithm outputs top N most possible mobile Apps user u will use. It is noticed that instead of only applying user u's own historical data, the algorithm adopts the history data of all users due to two reasons. User u may not have large amounts of historical data to figure out the mobile App usage pattern. In addition, the history data from only one user only contains a limited number of mobile Apps and locations. This leads to a wrong prediction if a user appears at a new location or uses a new mobile App.

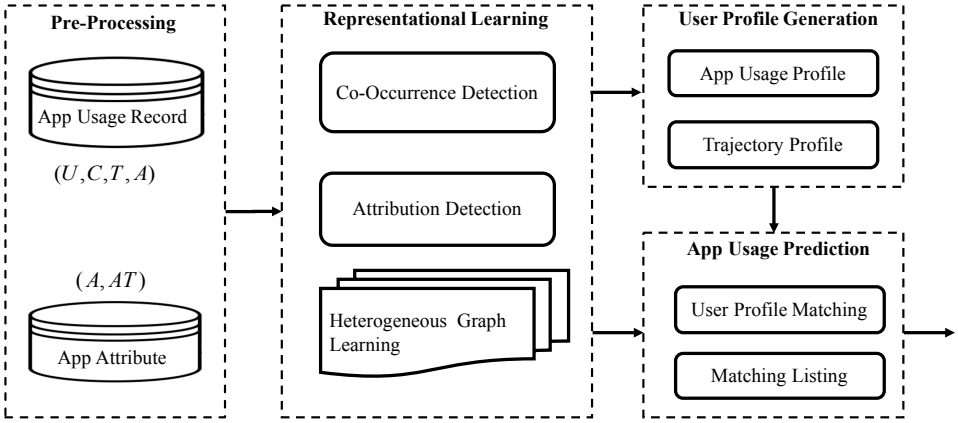


Fig. 3. The architecture of our algorithm for personalized context-aware App usage prediction. Information from the *Pre-Processing* module is mapped into a latent space through the *Representational Learning* module. The *User Profile Generation* module generates the user's profile based on their history App usage and trajectory data. The *App Usage Prediction* module lists top N possible Apps based on the user's profile and representational learning outputs matching.

3 ALGORITHM DESIGN

3.1 Algorithm Overview

This subsection introduces high-level algorithm design intuition to address the challenges described in Section 1. First, our algorithm should ensure the quantity and richness of information for training. Then, information from different dimensions should be comparable in our algorithm. Finally, the dynamic user preference should be reflected in mobile App usage prediction for different users. Based on these considerations, we design our algorithm architecture as shown in Figure 3.

Pre-Processing module prepares clean data for successive processing. As shown in Figure 3, each App usage record includes an anonymous user ID U, a connected cellular base station ID C, a time stamp T and an App ID A, which captures the information of user identity, location, time and App usage. The module first removes conflicting and redundant App usage records by checking the time stamp and cellular base station ID. Then, we remove the records of two categories of Apps: 1) the Apps that are used at almost all locations and at all time; 2) the Apps that contains few records within the datasets. As results, 135 Apps are removed from original 1633 Apps. We removed Apps of category 1) based on the following reasons: 1) A user can use them at any location and any time, which is difficult to predict. 2) Adding records of these Apps does not contain too much useful information. 3) Other predictable Apps' usage pattern will be overwhelmed by a large amount of these unpredictable App records. The removed Apps include WeChat (Chinese WhatsApp), WeiBo (Chinese Twitter) and etc. These popular social network Apps are used almost any location and at any time. We removed Apps of category 2) since they do not contain enough records to show temporal or spatial patterns. The Apps being removed include Haodaifu (looking for good doctors), Qichehui (car information), and etc., which are rarely used. With data collection of longer period in the future, these Apps will be contained. The remaining Apps are those with obvious temporal and spatial patterns, which include Wangzherongyao (a popular video game), Kuaishsou (a popular photo and video sharing App), and etc. In addition, the App attribute dataset includes the information of App ID A and App type ID P, which capture the common attribution of Apps. Each App usage record is associated with one App attribute record by matching the same App ID.

Representational Learning module gathers and maps information for training. This module first gathers information of different dimensions from all users' historical data. This ensures both quantity and richness of information for training. Then, information from different dimensions is mapped into a latent space based on two kinds of relationships *co-occurrence* and *attribution*. A heterogeneous graph-based learning method is designed to make information from different dimensions comparable. The module outputs the embedded vectors in the latent space and the details can be found in Section 3.2.

User Profile Generation module generates user profile to describe dynamic user preference for different users. The profile is based on two basic elements: a user's mobile App usage history and his/her past trajectory, both of which are affected by a time decay factor. This module outputs personalized user profiles expressed with vectors in the latent space. The details can be found in Section 3.3.

App Usage Prediction module predicts what App a user will use given the location C and time T. It matches the personalized user profile with history record vectors from *Representational Learning* in the latent space. Based on the matching scores, this module outputs the top N predicted mobile Apps. The details can be founded in Section 3.3.

3.2 Representational Learning with Embedding

In order to make the information of time, location, App, and App type comparable and figure out the importance and intersections of these factors, we adopt representational learning technology to map the information into a common latent space. To be more specific, a graph-based embedding method is designed, which brings the following benefits. First, it preserves the direct occurrence of interactions between factors, such as time, location and App in the same App usage record. Second, it also keeps indirect attribution interactions between records, through Apps belong to the same App type. Finally, it lowers the dimension needed to represent these factors by extracting two kinds of interactions. Take the App representation as an example, traditional one hot representation requires a vector of 2000 dimensions to represent 2000 Apps [70]. In contrast, embedding first extracts direct and indirect structures between these 2000 Apps given the users' history App usage records and the App attribute dataset. Then, these 2000 Apps are mapped into a latent space of much lower dimension based on the learned interactions. In our case, only 20 dimensions are needed.

3.2.1 Bipartite Graph Construction. High-quality embedding requires preserving both direct and indirect factor interactions, which are defined as *co-occurrence* and *attribution* respectively. The *co-occurrence* relationship captures the direct interaction between user, time, location and App, i.e. who use what App at what location and time. It preserves the information where and when an App is used. The *co-occurrence* relationship happens when two units shown in the same record. For example, *Pre-Processed Data* module outputs a record with a time unit (e.g. 6:50 PM), a location unit (e.g. cellular base station ID 272368) and an App (e.g. App ID 223). Two *co-occurrence* relationships reflect direct spatial and temporal usage correlation: App-location and App-time. The *attribution* reflects the indirect semantic interactions of these factors, i.e. correlating time, locations and Apps from different App usage records based on similar App attribute, which is expressed by App type information. The *attribution* relationship comes from the assumption that Apps belong to the same type share similar attribution and users tend to use Apps belong to the same type.

We use bipartite graphs to encode the *co-occurrence* and *attribution* relationships for further embedding learning, as shown in Figure 4. The graph has four different node types which correlate to four factors, App, location, time and App type respectively. The edges are constructed based on *co-occurrence* and *attribution* relationships.

App-Location Graph captures the spatial attribution of mobile App usage and is denoted as $G_{AC} = (A \cup C, \varepsilon_{AC})$, where A and C represent the mobile App Id and connected cellular base station ID. The edge ε_{AC} connects mobile App nodes and cellular base station nodes. Edge weights w_{AC} are set to the normalized *co-occurrence* counts.

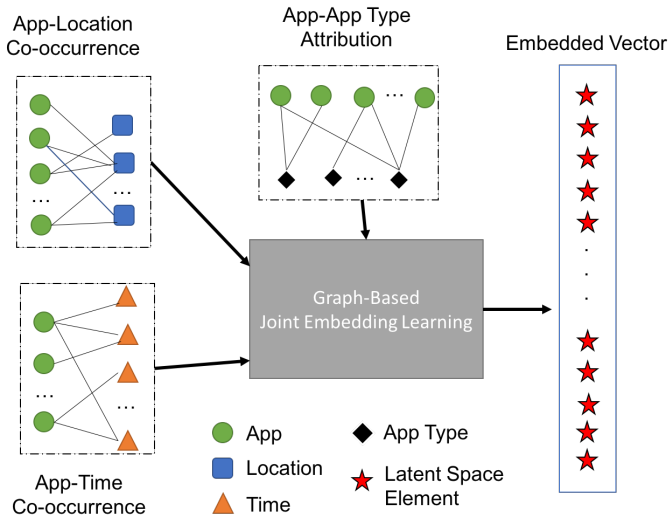


Fig. 4. This figure shows how we encode the *co-occurrence* and *attribution* relationships and adopt graph-based joint embedding learning to map these relationships into one latent space.

App-Time Graph captures the temporal attribution of mobile App usage and is denoted as $G_{AT} = (A \cup T, \varepsilon_{AT})$, where A and T represent the mobile App Id and time. The edge ε_{AT} connects mobile App nodes and time nodes. Edge weights w_{AT} are set to the normalized *co-occurrence* counts.

App-App Type Graph captures the interactions of Apps with similar functions, i.e. belong to the same App type, which is denoted as $G_{AP} = (A \cup P, \varepsilon_{AP})$. A and P represent the mobile App Id and mobile App type Id. If a mobile App $A_i \in P_j$, there is an edge $\varepsilon_{A_iP_j}$ connecting mobile App node A_i and mobile App type node P_j. In order to reflect the importance of different Apps, the TF-IDF is applied to derive edge weights $w_{A_iP_j}$ [9]. We treat Apps as a bag of words and App types as documents. We input counts of all Apps in all App types. Let $n_{i,j}$ be the count of the App i in the App type j, then we calculate the term frequency $TF_{i,j}$ of the App i in the App type j by $TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$. Let $|P|$ be the total number of App types, and p_j represent the App type j, the IDF for App i will be $IDF_i = \log \frac{|P|}{1+|p_j \in P: a_i \in p_j|}$. The final $TF\text{-}IDF$ value for the App i in the App type j is calculated as $TF\text{-}IDF_{i,j} = TF_{i,j} \times IDF_i$, which is used as the weight of the edge in App-App type graph.

The three graphs above capture the temporal, spatial and semantic effect of users' App usage respectively. Take the App-location graph as an example for interpretation. If a mobile App A_i is often used in location C_j, the edge weight w_{AC} is large. As a result, given a target user u at location C_j, he/she is most likely use mobile App A_i. These three graphs are embedded into a shared low dimensional latent space R^d, whose dimension is d. In the latent space, App, time, location, App type are represented as $\vec{a}$, $\vec{c}$, $\vec{t}$ and $\vec{p}$.

3.2.2 Heterogeneous Graph Learning. The goal of graph embedding learning is to represent the graph nodes in lower dimensional while preserving the structure. In addition, due to the heterogeneity of three graphs, how to decide the importance of them is important. We first model the emission probability distribution of each node according to the latent embeddings and then minimize the distance between the distributions and really observed distributions. The joint training described in Algorithm 1 is designed to iteratively optimize the overall loss function of three graphs.

Given a bipartite graph $G_{XY} = (X \cup Y, \varepsilon_{XY})$, where X and Y are two sets of nodes representing different information types and ε_{XY} is the set of edges connecting them, the likelihood of generating node j given node i is defined as

$$p(j|i) = \frac{\exp(-u_j^T \cdot v_i)}{\sum_{k \in X} \exp(-u_k^T \cdot v_i)}, \tag{1}$$

where u_j and v_i are embedded vectors of node j in Y and i in X respectively. It is noticed that each node i has two different embedding vectors according to their function: v_i when i acts as given node and u_i when i acts as emitted node. The true observed distribution of generating node j given node i is defined as

$$\hat{p}(j|i) = \frac{w_{ij}}{d_i}, \tag{2}$$

where w_{ij} is the edge weight, $d_i = \sum_{k \in X} w_{ik}$ and node i belongs to type X.

Before minimizing the distance between the embedding-based distributions and really observed distributions, we define the loss function for the graph G_{XY} as

$$L_{XY} = \sum_{i \in X} d_i KL(\hat{p}(\cdot|i)||p(\cdot|i)) + \sum_{j \in Y} d_j KL(\hat{p}(\cdot|j)||p(\cdot|j)), \tag{3}$$

where node i and j belong to type X and Y respectively. $KL()$ is Kullback-Leibler divergence [41]. Since there are three different graphs in our algorithm, the overall loss function is derived as

$$L = L_{AT} + L_{AC} + L_{AP}, \tag{4}$$

where L_{AT}, L_{AC} and L_{AP} represent loss function of App-location, App-time, and App-App type graph.

It is noticed that it is computationally expensive to optimize loss Eq. function 3 since calculating the likelihood $p(j|i)$ requires sum over the entire set of node in X. To address this problem, a negative sampling approach is adopted, in which we sample multiple negative edges. The sampling is based on noisy distribution for each edge [53]. Then, asynchronous stochastic gradient (ASGD) algorithm is used for optimization [58]. For a directed edge w_{ij} , we randomly elect L nodes that do not connect to the node i. We consider node j as a positive example, and the L nodes as negative examples, then the loss function minimized will be organized as

$$F = -\log \sigma(u_j^T \cdot v_i) - \sum_{l=1}^{L} \log \sigma(-u_l^T \cdot v_i) \tag{5}$$

We repeat the sample and update process M times, which denotes the number of samples M.

Since three graphs in our algorithm are heterogeneous and cannot be optimized simultaneously by merging all the edges together, we adopt a joint training algorithm, as shown in Algorithm 1, to iteratively optimize the overall loss function Eq. 4 to get d dimensional embedded vectors for App, time, location and App type in the common latent space.

3.3 Context-aware App-usage Prediction

To achieve context-aware App-usage prediction, the algorithm needs not only to embed information from different spaces into one latent space, but also to generate a dynamic user profile for each user to describe his/her dynamic preference. The profile should also be mapped into the same latent space as App, location, time and App type for prediction.

User profile generation: We generate the user profile based on two observations: 1) A user's current App usage is related to his/her past App usage and the visited locations with high probability. 2) Recent App usage and

ALGORITHM 1: Joint training heterogeneous graph

Input: G_{AC}, G_{AT}, G_{AP}, number of samples M, number of negative samples L
Output: App embedded vector $\vec{a}$, location embedded vector $\vec{c}$, time embedded vector $\vec{t}$ and App type embedded vector $\vec{p}$
while $m \leq M$ **do**
  sample an edge from ε_{AC} and draw L negative edges, update App and location embedded vectors;
  sample an edge from ε_{AT} and draw L negative edges, update App and time embedded vectors;
  sample an edge from ε_{AP} and draw L negative edges, update App and App type embedded vectors;
  increase m by one;
end

visited locations should play more important roles than old ones. Therefore, given a time τ and a user u, we first extract his/her App usage records before τ, i.e. all records whose $U = u$ and $T < \tau$. These records form a sub-set $\mathfrak{R}_\tau^u \subset \mathfrak{R}$, which consists of records of (u, t_i, c_i, a_i). Then we define the profile of the user u at time τ as:

$$\vec{u}_\tau = \beta \sum_{(u,\vec{c}_i,\tau_i)\in\mathfrak{R}_\tau^u} e^{-(\tau-\tau_i)}\vec{c}_i + (1-\beta) \sum_{(u,\vec{a}_i,\tau_i)\in\mathfrak{R}_\tau^u} e^{-(\tau-\tau_i)}\vec{a}_i, \tag{6}$$

where $\vec{a}_i$ and $\vec{c}_i$ are the embedded vectors of the App a_i and the location c_i in $\mathfrak{R}_\tau^u$. $e^{-(\tau-\tau_i)}$ is a time decay factor indicating that older data has less influence. $\beta \in [0, 1]$ is the coefficient to tune the importance of App usage history and trajectory history.

Given a query $q = (u, \tau, c)$, we first obtain the user u's profile at time τ, expressed as $\vec{u}_\tau$. Then we compute scores of different possible mobile Apps a_j as:

$$S(q, a) = \vec{u}_\tau \cdot \vec{a}_j, \tag{7}$$

where $\vec{a}_j$ is the embedded vector of mobile App a_j and $\vec{u}_\tau$ is the user profile calculated as Eq. 6. The above score not only captures the App usage preference, but also captures the user history trajectory. In addition, the time decay effect is also considered. Based on the score ranking of different mobile Apps, the algorithm outputs N most possible mobile App predictions.

4 EVALUATION

In this section, we evaluate our algorithm with real world-collected mobile App usage data. We first introduce how we set up evaluation in Section 4.1. Then we analyze the algorithm performance in Section 4.2. The evaluation focuses on 1) comparing the performance of our algorithm with four different baselines and 2) investigating the influence of key parameters in the algorithm.

4.1 Evaluation Setup

Data Pre-Processing: The ChinaTelecom dataset used for evaluation contains more than 6 million mobile App usage logs from 1788 users in one week, April 19, 2016 - April 26, 2016. The TalkingData dataset used for evaluation contains more than 400,000 mobile App usage logs from 801 users in one week, May 01, 2016 - May 07, 2016, after removing inactive usage logs and conflict & unknown data. These logs record who uses what App at what time and locations. Each record includes an anonymous user ID, a connected cellular base station ID or latitude & longitude value, a time stamp and an App ID. Each App ID is associated with one or more App type IDs.

For both datasets, we sort the App usage records by ascending order of time respectively. We use the first 80% of the records as training data and the last 20% as ground truth for testing.

Algorithm Setting: We set the default embedding dimension d and importance coefficient of user profile β as 20 and 0.5 respectively. We will also check how these two key parameters affect algorithm performance in Section 4.2.

Performance Metric: We adopt *Accuracy@k* to evaluate prediction accuracy [74]. *Accuracy@k* is the statistical result of all test predictions, which is calculated with hit@k. The value of $hit@k$ for a single prediction equals 1 if the ground truth App appears in the top k predictions, or 0 if otherwise. The overall *Accuracy@k* is calculated as the average over all test cases:

$$\text{Accuracy@k} = \frac{\#hit@k}{|R_{test}|}, \tag{8}$$

where $\#hit@k$ and $|R_{test}|$ represent the number of hits in the whole test set and the number of test cases.

We also adopt mean reciprocal rank (MRR) to quantify the performance of our method and four baselines. MRR calculates reciprocal of the rank at which the first relevant document was retrieved [27]. Given the number of the testing set N, MRR is defined as:

$$MRR = \left(\sum_{i=1}^{N} \frac{1}{rank_i}\right) \Big/ N, \tag{9}$$

where r_i is the ranking of the ground truth for the $i-th$ prediction. Higher MRR means high prediction accuracy.

Baselines: In order to illustrate the advantages from our algorithm design, we compare our *CAP* with the following baselines.

- *Statistics (Sta):* This method counts the users' history of mobile App usage and selects the most frequently used ones. This is the straightforward method for prediction, which does not use time and location information.
- *Graph based embedding (GE):* This method adopts the graph-based embedding in a recent work [74]. Besides the three graphs in our algorithm, this paper also embeds App-App sequential relationships. In addition, this method also adopts user profiles, but ones generated by APP usage history, current location, and current time, i.e. without App time decay. By comparing this method with our *CAP*, we can check the performance improvement against our embedding method and user profile.
- *Modified graph based embedding (M-GE):* This method is a combinatorial scheme, which takes the same embedding method as our algorithm and same user profile generation method as *GE*. By comparing this method with our *CAP*, we can check the performance improvement against our user profile.
- *Personalized Ranking Metric Embedding (PRME):* This method jointly models the sequential transition of App usage and user profile [30]. PRME utilizes one sequential transition space and one user profile space [30].

4.2 Result Analysis

4.2.1 Performance on Two Datasets. In order to compare our *CAP* with baselines, we plot Accuracy@5 in Figure 5. Our *CAP* performs best, which achieves 84% in terms of Accuracy@5. *M-GE* and *PRME* rank second and third, both of which achieve around 30% lower accuracy than *CAP* in terms of Accuracy@5. *Sta* and *GE* achieve only 35% and 6% in terms of Accuracy@5 respectively. First, *Sta* does not get high accuracy since the simple statistical method cannot handle the cases when a user is going to open a new mobile App that never shows up in his/her training set. For example, in the testing set, user 0067461 opens App 59 at 7:43 pm at cellular tower 336271. The App 59 has never been used by this user and does not appear in the *Sta* prediction list. In contrast, the top 5 predictions from our *CAP* are App 179 59 241 125 144. This is because our *CAP* takes all users' history data and correlates App usage temporal, spatial and attribution characteristics through heterogeneous graph embedding. Second, *GE* performs worst since it includes App-App sequential relationship in embedding, which

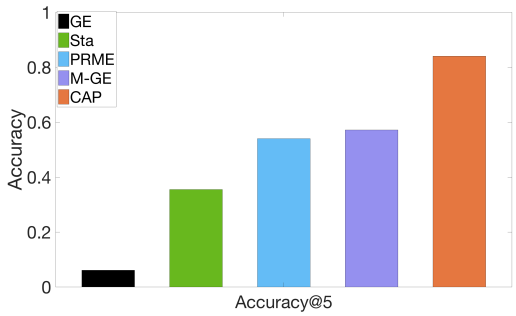


Fig. 5. This figure shows Accuracy@5 from our *CAP* and baselines with China Telecom dataset. Our *CAP* performs best, achieving 84% for Accuracy@5. *M-GE* and *PRME* rank second and third. *Sta* and *GE* achieve low accuracy.

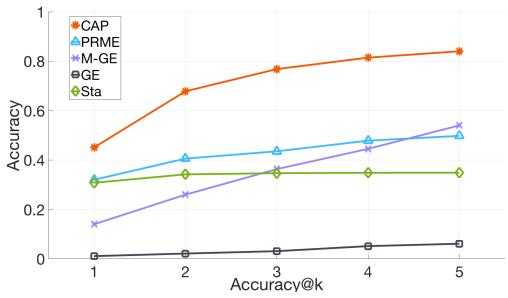


Fig. 6. This figure shows Accuracy@k with different k values for all methods with China TeleCom dataset. Our *CAP* achieves ~ 68% in terms of Accuracy@2, which is higher than Accuracy@5 in terms of all other methods.

brings wrong connections. This illustrates that adding wrong interactions between influential factors leads to serious performance deterioration. *M-GE* and *PRME* performing much better than *GE* also proves that what App a user is going to use is usually not decided by the previous used App, but decided by time and location. Third, the advantage of our *CAP* over *M-GE* validates the effectiveness of our user profile. The time-decay on both App usage and user's trajectory history help extract the dynamic user preference.

To further analyze the accuracy of different metrics for different methods, we plot Accuracy@k in Figure 6. Our *CAP* achieves ~ 68% in terms of Accuracy@2, which is higher than Accuracy@10 for all other methods. In other words, with two candidates, *CAP* is able to outperform other approaches with five candidates. Only *M-GE* and *PRME* achieve similar accuracy in terms of Accuracy@5. This illustrates that to achieve similar accuracy, our *CAP* needs much smaller prediction list.

Table 1 shows MRRs with different used App numbers (i.e. different numbers of Apps that the user uses) with China TeleCom dataset. First, MRR values decrease when users use more Apps. This is because that higher number of used Apps represents diverse App usage pattern, which is more difficult to predict. Second, for most cases, our *CAP* shows consistent advantages over other methods. This echoes our analysis on Accuracy@k in Fig. 5 and Fig. 6. Finally, when few Apps (less than 4) are used by users, the *Sta* is better than our *CAP*. This is because that *Sta* only get prediction from the past used App and it is easy to guess from only less than 4 candidates, while the prediction from our *CAP* considers more candidate Apps. For example, when user uses only 1 App, historical-based results will always be 100%, while with 4 Apps, the chances of getting the most frequently used App is quite high. When the number of Apps used increase, the advantage of our *CAP* over *Sta* increases to 27% (0.48 VS 0.37). Our *CAP* aims at the cases that people tend to have diverse App usage pattern and use more Apps. This is also the trend for App usage in the future. One should note that this result is biased toward the *Sta* history based method. This is because our method can be applied to predict usage of new Apps, whereas *Sta* does not. Furthermore, current Apps mostly are designed to be used all locations and time, thus favoring Sta. With the advent of temporary Apps (e.g. Android Instant Apps) more location and time specific Apps will become available [64]. Despite these, our *CAP* still shows a significant improvement over currently used *Sta* approach especially compared to other state-of-the-art methods.

We also evaluate the performance of our *CAP* and baselines with the data collected by TalkingData platform. Figure 7 shows Accuracy@5 for our *CAP* and other 4 baselines. Our *CAP* performs best, which achieves 50%. The straightforward baseline *Sta* and *PRME* rank second and third, both of which achieve more than 10% lower

Table 1. This table shows MRR values under different numbers of used Apps with China TeleCom dataset.

Methods / N_{Apps}	2	4	6	8	10
CAP	0.85	0.75	0.60	0.55	0.48
PRME	0.26	0.45	0.54	0.44	0.25
Sta	0.94	0.79	0.57	0.49	0.38
M-GE	0.09	0.09	0.09	0.09	0.04
GE	0.03	0.01	0.01	0.01	0.01

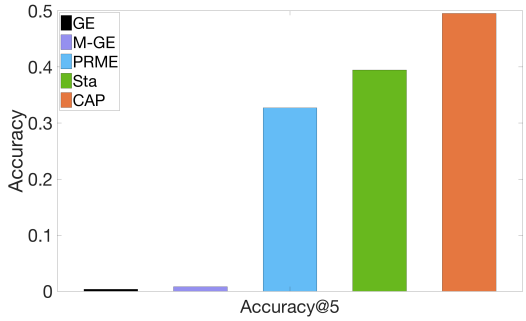


Fig. 7. This figure shows Accuracy@5 from our *CAP* and baselines with the TalkingData dataset. Our *CAP* performs best, achieving 50% Accuracy@5. *Sta* and *PRME* rank second and third. *GE* and *M-GE* get very low accuracy.

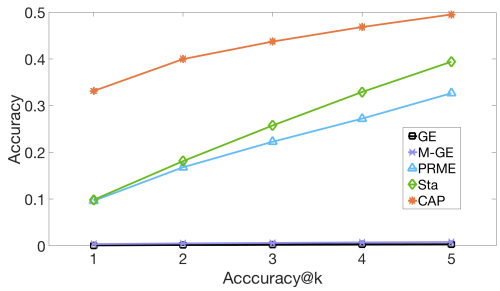


Fig. 8. This figure shows Accuracy@k with different k values for all methods with the dataset collected by TalkingData platform. Our *CAP* achieves ~ 40% in terms of Accuracy@2, which is higher than Accuracy@5 in terms of all other methods.

accuracy than *CAP*. *GE* and *M-GE* achieve only less than 1% accuracy. The improvements of our *CAP* over the *M-GE* and *GE* illustrate the importance of user profile and the embedding method designed in *CAP*.

To analyze the accuracy of different metrics for different methods with the TalkingData dataset, we plot Accuracy@k in Figure 8. Our *CAP* shows consistent advantages over all baselines. At accuracy@1, our *CAP* achieves more than 20% accuracy than other methods. Although accuracy improvement of our *CAP* over other methods decreases with larger k value, our CAP still achieves more than 10% accuracy than the second best method Sta. This echoes to performance run on the China TeleCom dataset and proves that our CAP maintains advantage over baselines on different datasets.

Table 2 shows MRRs with different used App numbers on TalkingData dataset, which shows similar results with those on China TeleCom dataset. First, MRR values of all methods decrease when users use more Apps since higher number of used Apps represents diverse App usage pattern, thus causing more difficult prediction. Second, for most cases, our *CAP* shows consistent advantages over other methods, which echoes our analysis on Figure 7 and Figure 8. Finally, when users only use few Apps (less than 4), the *Sta* is better than our *CAP*, since *Sta* only get prediction from the past used App and it is easy to guess from only less than 4 candidates, while the prediction from our *CAP* considers more candidate Apps. When the number of Apps used increase, our *CAP* shows consistent advantages over all baselines, i.e. 1.5× over *Sta* and 2× over *PRME*, sine our *CAP* aims at the cases that people tend to have diverse App usage pattern and use more Apps.

4.2.2 Performance of Variants of Algorithm. Based on the performance analysis on two datasets, we can draw several conclusions. First, the result trends from both datasets are similar. Our *CAP* keeps performance advantages

Table 2. This table shows MRR values under different numbers of used Apps with TalkingData dataset.

Methods / N_{Apps}	2	4	6	8	10
CAP	0.32	0.19	0.17	0.17	0.17
PRME	0.12	0.11	0.09	0.06	0.08
Sta	0.61	0.20	0.14	0.12	0.13
M-GE	0.02	0.02	0.02	0.01	0.01
GE	0.02	0.01	0.01	0.003	0.001

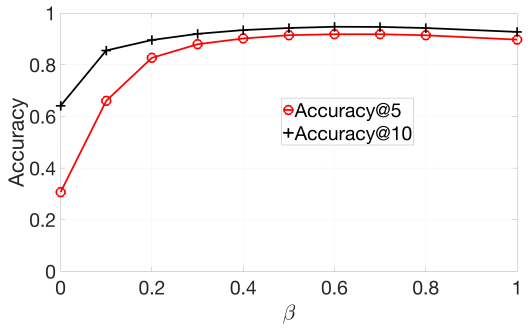


Fig. 9. This figure shows the Accuracy@5 and Accuracy@10 with different user profile importance coefficient β values using our *CAP*. For both Accuracy@5 and Accuracy@10, the peak values show when $\beta = 0.5$. The accuracy at $\beta = 1$ is much higher than accuracy at $\beta = 0$.

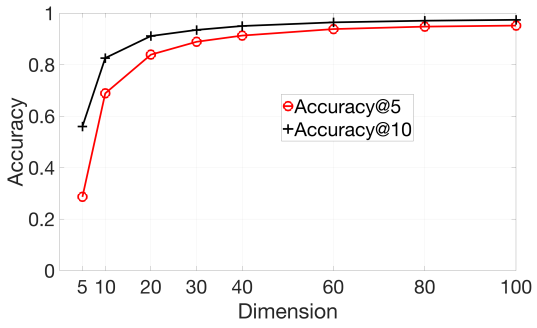


Fig. 10. This figure shows Accuracy@5 and Accuracy@10 with different embedding dimensions using our *CAP*.

over all baselines on both datasets. This proves the robustness of our *CAP* on different datasets. Second, all methods perform worse on the TalkingData dataset, since users in the TalkingData dataset show more diverse App usage behavior. In contrast, the China TeleCom dataset only captures the App usage with network connection. As a result, more diverse App usage behavior in the TalkingData dataset increases the prediction difficulty. Finally, on both datasets, *Sta* performs better than our *CAP* when the number of used App is less than 4. This is because our *CAP* aims at the cases that people tend to have diverse App usage pattern and is able to predict the usage of new Apps. In the future, a new algorithm can be designed, which combines the *Sta* and our *CAP* to achieve better performance.

In order to illustrate how the user profile importance coefficient β in Eq.6 affects our algorithm performance, we plot the Accuracy@5 and Accuracy@10 with different β values in Figure 9. For both Accuracy@5 and Accuracy@10, the peak values show when $\beta = 0.5$. This means that in the optimal solution of user profile, App usage and user's trajectory history play equally important role. This validates our selection of user profile combination. In addition, accuracy at $\beta = 1$, when only user's trajectory history is adopted, is much higher than accuracy at $\beta = 0$, when only user's App usage history is adopted. This means that if only one factor can be included to represent a user's dynamic preference, his or her trajectory history is more important. This is because mobile App usage is more related to locations.

Figure 10 shows how embedding dimension affects our algorithm Accuracy@5 and Accuracy@10. First, obviously, high embedding dimension leads to high accuracy. Second, Accuracy@5 and Accuracy@10 saturates at 80 embedding dimensions, which achieves 94% and 97% accuracy respectively. This means that 80 dimensions

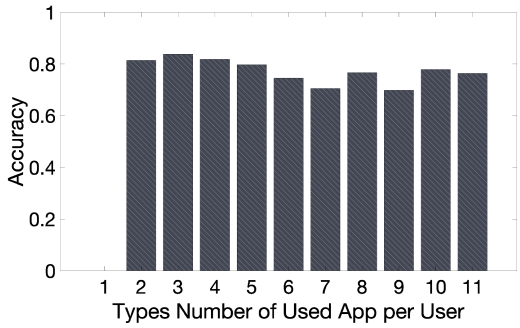


Fig. 11. This figure shows Accuracy@5 with different user types. We classify users according to how many App types they use in the whole dataset. All user types achieve more than 70% in terms of Accuracy@5, which illustrates the robustness of our *CAP*.

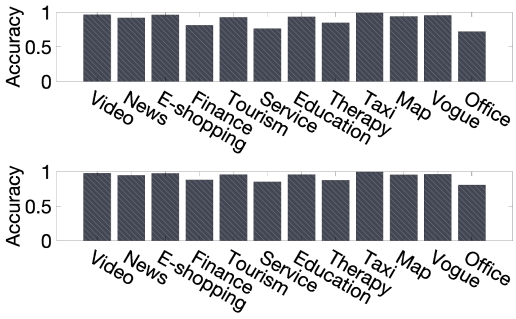


Fig. 12. This figure shows Accuracy@5 and Accuracy@10 of different App type predictions with our *CAP*. All App types achieve more than 70% in terms of Accuracy@5 and 80% in terms of Accuracy@10. Taxi App ranks highest since users usually take taxis at regular location and regular time.

are large enough to embed information. Third, the accuracy improvement shows an obvious difference before and after 20 embedding dimensions. From 5 embedding dimensions to 20 embedding dimensions, the prediction accuracy improves 56% (from 28% to 84%) and 35% (from 56% to 91%) in terms of Accuracy@5 and Accuracy@10 respectively. In contrast, from 20 embedding dimensions to 100 embedding dimensions, the prediction accuracy improves no large than 10% for both Accuracy@5 and Accuracy@10. Considering the tradeoff between accuracy and computing complexity, we adopt 20 embedding dimensions as our algorithm default setting.

In order to check the prediction accuracy of different users with our algorithm, we plot Accuracy@5 with different user types in Figure 11. We classify users according to how many App types they use in the whole dataset. All user types achieve more than 70% in terms of Accuracy@5, which illustrates the robustness of our *CAP* on predicting for different user types. The robustness comes from our dynamic user profile, which includes the App usage and trajectory history.

Figure 12 shows Accuracy@5 and Accuracy@10 of different App type predictions with our *CAP*. All App types achieve more than 70% in terms of Accuracy@5 and 80% in terms of Accuracy@10. This illustrates the robustness of our *CAP* on predicting different App types. Taxi App ranks highest since users usually take taxis at regular locations and time, such as 8:00am from home, 3:00pm from school, 6:00pm from company etc. Office App ranks lowest accuracy since people in Shanghai, a big city in China, have high pressure on working and they could work at multiple time and locations.

In order to show how a user's data sparsity affects accuracy, we plot Accuracy@5 from different methods in Figure 13. More data helps improve accuracy for all methods. This is because more history information gets better training, thus better predictions. Our *CAP* saturates at 8000, which achieves 92% Accuracy@5. This illustrates that if all users have more than 8000 history records, our *CAP* can achieve up to 92% accuracy with five prediction candidates. On the contrary, *Sta*, *M-Ge* and *GE* saturate to get Accuracy@5 of $\sim$ 70%, $\sim$ 40% and $\sim$ 10%.

4.2.3 Spatial-Temporal Analysis. To check if the features are selected appropriately in our *CAP*, we compare the performance (Accuracy@5) of 4 different feature combinations with TalkingData in Figure 14. Besides the three features in our *CAP*, i.e. App-Time (A-T), App-Location (A-L), App-App Type (A-AT), one extra feature, App-App (A-A), is also investigated. First, using all four features leads to lowest accuracy since the App-App sequential relationship brings wrong connections. This illustrates that adding wrong feature (interactions between influential factors) leads to serious performance deterioration. Second, the algorithm achieves 11% for Accuracy@5 when

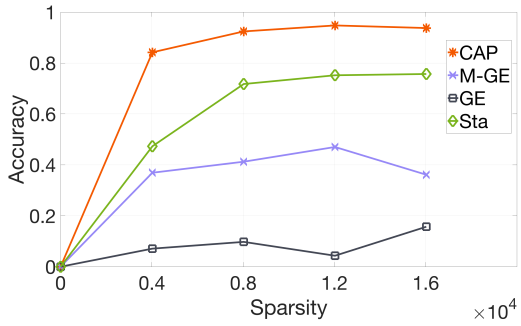


Fig. 13. This figure shows Accuracy@5 with different user's data sparsity from different methods. More data help improve accuracy for all methods. Our *CAP* saturates at 8000 with 92% Accuracy@5.

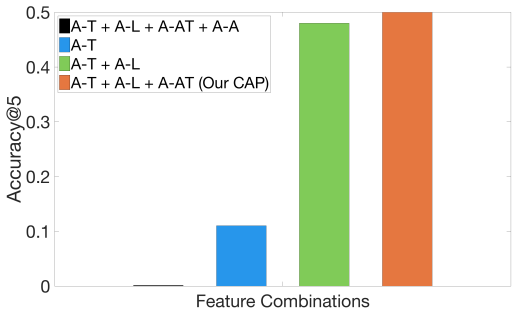


Fig. 14. This figure shows how different combinations of features affect the performance of our *CAP*. The Accuracy@5 values are derived with the TalkingData dataset.

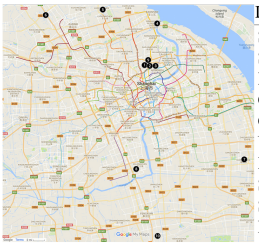
adopting only one feature (App-Time), which shows the necessity to include more features. Third, adding the feature of App-Location improves the Accuracy@5 a lot (from 11% to 48%), which shows the validation of this feature. Finally, the highest Accuracy@5 of three features (App-Time, App-Location, App-App Type) demonstrates the effectiveness of the designed feature choice in our *CAP*.

To verify whether the learned embeddings captures the correlations between the units of location, time and App, we visualize several illustrative cases. We show three examples of querying single units: location, time and App. For better illustration and understanding, we relate each location to one point of interest (POI), which reflects the function and interest of the region.

Figure 15 shows the top 10 results of querying LuXu Park, a scenery spot location. Tourists go to LuXu Park for travelling while local people go to LuXu Park for enjoying life, such as jogging or walking after dinner. The top 10 queried location's POIs are mostly related to enjoying life (2 catering, 1 shopping and 1 entertainment) or travelling (4 scenery spots). Eight queried time periods are after work when people tend to enjoy their life, while the rest two (15:30 and 16:30) are more likely to related to tourists. Among the top 10 queried Apps, five are related to travelling, including XieChengLvXing, MaFengWo, TongChengLvYou, TuNiu and YiLong [10, 28, 68, 69, 82]. Four are related to enjoying life, ShenMeZhiDeMai MeiLiHhui and XiaoHongShu for shopping and KuanDaiShan for crowd-sourced reviews about local businesses [39, 52, 61, 73].

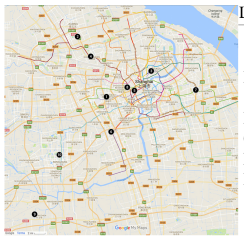
Figure 16 shows the top 10 results of querying an after work time 20:00. The queried top 10 time periods are all close to 20:00 in an ascending way. The top 10 queried location's POIs are all about enjoying life, such as scenery spot, shopping, entertainment, catering. The top 10 queried Apps are also related to having fun. For example, KuGouMusic is a Chinese music streaming and download service and ChangBa is a Chinese music streaming App where users can upload and share their own-recorded songs [12, 40].

Figure 17 shows the top 10 results of querying Zhihu App, which is a famous Chinese question-and-answer App mainly aiming at young people, where users create, answer, edit questions [93]. YouDaoDict, which is an online dictionary-like and translation App, ranks second on the queried list [85]. Both Zhihu and YouDaoDict help people acquire new knowledge. The rest of top 10 queried Apps are all commonly used by young people. WangYiMusic is a popular music App for young people [55]. Young people can also broadcast their music and talkshow through this App. Bilibili is a video sharing App focusing on animation, comic, and game for young people [6]. The sharing and discussion mode from both WangYiMusic and Bilibili are similar to Zhihu's question-and-answer interactive mode. All of these three Apps aim at young people in China. The top 10 queried time periods are



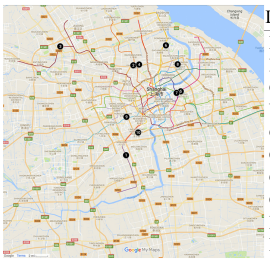
Location (POI)	Time	App
Scenery Spot	22:30	XieChengLvXing
Scenery Spot	23:00	MaFengWo
Entertainment	15:30	ShenMeZhiDeMai
Catering	17:00	MeiLiHui
Catering	21:30	TongChengLvYou
Unknown	22:00	TuNiu
Entertainment	16:30	YiLong
Shopping	19:00	LianJia
Scenery Spot	24:00	KuanDaiShan
Business	21:00	XiaoHongShu

Fig. 15. This figure shows the top 10 results of querying LuXu Park, a scenery spot location. Location, time and App query results are listed. Locations are related to their POIs.



Location (POI)	Time	App
Scenery Spot	20:00	KuGouMusic
Shopping	19:30	WifiWanNengYaoShi
Entertainment	20:30	ChangBa
Shopping	19:00	WanNianLi
Entertainment	21:00	MoGujie
Catering	18:30	58TongCheng
Residence	21:30	360YunPan
Entertainment	18:00	QiHuSouSuo
Scenery Spot	22:00	AliBaBa
Shopping	22:30	iBooks

Fig. 16. This figure shows the top 10 results of querying an after work time 20:00. Location, time and App query results are listed. Locations are related to their POIs.



Location (POI)	Time	App
Entertainment	17:00	ZhiHu
Education	8:00	YouDaoDict
Catering	8:30	QiJiNuanNuan
Unknown	15:30	WangYiMusic
Catering	9:30	Yume100
Office	9:00	TencentDongMan
Catering	17:30	BoHe
Office	16:30	Bilibili
Entertainment	16:00	KuanDaiShan
Education	7:30	XiaoHongShu

Fig. 17. This figure shows the top 10 results of querying Zhihu App, a famous Chinese question-and-answer App mainly aiming at young people. Location,time and App query results are listed. Locations are related to their POIs.

all within or slightly before working time (18:00). This is because people usually use Zhihu to get answers for questions in their work or study. Among top 10 queried locations, 2 are education areas and 2 are office areas, where people usually work and study. It is noticed that there are 2 entertainment and 3 catering areas. These areas usually include places like StarBucks where people like to work and study there in China.

Based on the observations from Figure 15 to Figure 17, high correlations are shown between App usage and spatial-temporal features. The App usage is affected by both temporal and spatial factors. This also validates the idea of considering time and location in both embedding and user profile in our *CAP* for App usage prediction.

In conclusion, this section evaluates the performance of our *CAP* and four baselines to show the advantages of our algorithm design. First, the extracted three interactions (App-time, App-location, App-App type) are proved to be valid on App usage prediction. The interactions between the influential factors need to be carefully designed. Adding App-App sequential interaction seriously deteriorate the performance. Second, the heterogeneous graph embedding is proved to successfully map the influential factors from different spaces into the same latent space and figure out the importance of these factors, which help predict App usage. Finally, the proposed user profile is shown to help improve the prediction accuracy by our method's accuracy improvement over *M-GE*.

5 RELATED WORK AND DISCUSSION

5.1 App Usage Behavior Modelling

Recent works have studied how users use mobile Apps by focusing on three aspects: user interactions, network traffic, and energy drain [29, 32, 35]. Church et al. summarized the challenges for mobile phone usage learning

and analysis, as well as a series of studies and applications on mobile phone usage [24]. Falaki et al. discover immense diversity usage activities among users[29]. Another related work [8] reveals that users can be identified through the sets of Apps they use. Other studies cluster mobile users according to their App usage records[89]. Moreover, users' mobility patterns can impact the way that the Apps are used [91]. Context such as location and time are shown to have impact on App usage [63][34]. A multi-faceted approach to predict App usage is developed in [75]. Most studies focus on small-scale datasets, posing a key challenges to understand and predict the App usage behavior over a large user population.

5.2 Recommendation Methods

Recommendation systems have been widely used and a wide range of approaches have been proposed. Context-aware recommendation is usually achieved by using additional information of location, time, and activity [36, 47, 48, 92, 94]. Zheng et al. and Karatzoglou et al. presented collaborative filtering based recommendation algorithm and use a large-scale user data pool to collaboratively filtering the like-minded users at different locations or activities [36, 92]. Zhu et al. focused on the problem of insufficient information from individual users by learning the common context-aware preference of many users, and the context sensors they targeted are spatio sensors such as GPS and accelerometer [94]. Kostakos et al. applied a Markov state transition model to predict next screen event [38]. Based on our study, the both spatio and temporal contextual information matters in the user behavior prediction. However, these prior works mostly limited the contextual information to location and activity. Zhao et al. [90] proposed a spatial-temporal latent ranking (STELLAR) method to explicitly model the interactions among user, POI, and time. Liu et al. considered both spatio and temporal contextual information and extended the RNN model to Spatial Temporal Recurrent Neural Networks (ST-RNN) with a time-specific transition matrices and a distance-specific transition matrices [48]. None of these works focus on App usage patterns. Compared to these prior works, our algorithm CAP is able to project both context and attribute information into comparable spaces, hence achieving a better integration.

Other than contextual information, user profile information is also used to achieve personalized recommendation [11, 25, 43, 46, 59]. Rendle et al. presented their Factorizing Personalized Markov Chains (FPMC) model that subsumes both a common Markov chain and the normal matrix factorization model to profile the user. However, these personalized recommendation largely depends on the personal profiling, which can be biased and may not capture the local trend of the App usage. Liu et al. used a mobile Customized Content Service (m-CCS) to filter blog articles to mobile users based on the trend of time-sensitive popularity of weblogs and the users' browsing logs to determine their interests [46]. Costa et al. monitors the users' interaction and made recommendation based on the users' friends, similar behavior users, and the similarity between Apps [25]. Similarly, Bohmer et al. leveraged the insights of users' engagement with particular applications to achieve recommendation [11]. Lin et al. presented PRemiSE, which takes into account potential influencers on virtual social networks extracted from implicit feedbacks for recommendation [43]. Our work, compared to these prior works, allows better real-time modeling on personal choice of the App usage by combining the spatial and temporal contextual information in both group and individual level.

App similarity that is important for recommendation is usually calculate by graph [5] or kernel function [14, 15], which is utilized in Ranking [80] and popularity [95] based recommendation. When the user data is sparse, new challenges emerge. Problems of data sparsity [62] and cold-start [44] have been studied by using specific Apps' features of similarity. CAP handles the data sparsity by taking into account the data from many users in both spatio and temporal contextual information. Other recommendation has been done with a different focus from ours, which is privacy and security awareness [45, 83, 96]. These works demonstrate the possibility the secure aspects of the recommendation systems like ours.

To summarize, none of the existing works focus on App usage prediction over a large population using both temporal and spatial information. These works have not explored the key influential factors & their interactions

for App usage prediction, nor have they investigated the appropriate method to extract the importance of this information. In addition, the personalized factor of users has not been studied for App usage prediction. CAP achieves real-time personalized App usage prediction for multiple commercial applications as we discussed in Section 1. Context information (time & location), attribute information (App & App type) and the dynamic user preference are considered. We find that the relationships of App-location, App-time, and App-App type are essential to prediction and propose a heterogeneous graph embedding algorithm to map them into one common comparable latent space. We propose a user profile with his/her App usage & trajectory history affected by time decay factor, to achieve personalized prediction. We extract both the common attribution of all users and individual user dynamic preference to ensure sufficient training data without losing personalization.

5.3 User Behavior Prediction

User behavior prediction is very important in the applications like smart health [22], smart home [20], new generation wireless communication [21, 49] and etc. With the popularity of social networks, users leave a large volume of digital footprints online. By analyzing re-post behavior in social networks, Lu et al. [50] predicted the content dissemination trends. However, in traditional social networks, the user behavior such as posting blogs, sharing photos and uploading videos, does not necessarily reflect their daily activities. Location-based social networks (LBSNs), where users can share their real-time activities by checking in at POIs, provide a novel data source to study the collective behavior, and collective behavior analysis in LBSNs has gained increasing popularity in academia. For example, Cheng et al. [23] investigated 22 million checkins across 220,000 users and report a quantitative assessment of human mobility patterns by analyzing the spatial, temporal, social, and textual aspects associated with these footprints. Noulas et al. [57] conducted an empirical study of geographic user activity patterns based on check-in data in Foursquare. Cranshaw et al. [26] studied the dynamics of a city based on user collective behavior in LBSNs. Wang et al. [72] investigated the community detection and profiling problem using users' collective behaviors in LBSNs. In addition, the analysis of collective behavior in LBSNs can also enable various applications. For example, by analyzing users' check-in data in LBSNs, Yang et al.[78, 79] studied the personalized location based services such as POI recommendation and search. Sarwat et al. [60] introduced the Plutus framework that assists different POI (e.g., restaurants or shopping malls) owners in growing their business by recommending potential customers. Yang et al. [77] studied the large-scale collective behavior by introducing the NationTelescope platform to collect, analyze and visualize the user check-in behavior in LBSNs on a global scale. However, traditional LBSN cannot get access to mobile application data, so that predict the App usage is a novel contribution of our paper. Our *CAP* can also be potentially used to predict the behaviors of micro-aerial vehicles [16, 17] and mobility of taxis [18, 19].

Mobility prediction is also widely studied. Markov model and its variations are common models to predict human mobility. Markov model [13, 37] consider the probability to capture the unobserved characteristics between location transition, i.e., Mathew et al. [51] cluster the locations from the trajectories and then train a Hidden Markov Model for each user. Considering the mobility similarity between user group, Zhang et al. [87] propose GMove to share significant movement regularity among users.Moreover, pattern-based methods [33, 54, 84] also utilized to predict the mobility based on these popular patterns. All these mobility prediction techniques only deal with the two dimensional location and time information. However, in our context-aware App usage prediction, high-dimensional dataset are needed to be considered, which is a much more challenging problem.

5.4 Limitations

Our work has a number of limitations. First, the China TeleCom dataset was collected passively and anonymously. Thus, App usage activities that make no network requests, request by HTTPS, or connect through WiFi were not captured in our dataset. Second, in the China TeleCom dataset, we are unable to distinguish between Apps that made a network request after direct user input, and Apps that run in the background and make network

requests automatically. The ambiguity comes down to the definition of what does it mean to "use" an App. In our analysis, we assume that "use" means that the App is running on a user's phone, which does not imply that the user is explicitly interacting with the App. However, in the TalkingData dataset, these two limitations on data collection do not exist and we prove the advantage of our *CAP* with both datasets. Third, the assumption in the research is that a user's App usage habit does not change too much over time. Therefore, our method adopts the history data from all users for training. However, if the mobile App usage habit of the user to be predicted is different from the history, predicting performance may deteriorate. This inspires a new research problem of predicting users' changing App usage trend. In addition, if a user does not have enough history data to generate user profile, the prediction will also be inaccurate. In the future, this could be solved by exploring similar patterns between users and borrow others' history data for training. Finally, our prediction does not include all possible factors that influence users' App usage behavior, such as screen lock, social context, battery life, etc. The lack of the information leads to incapability to perfectly describe and predict a users' App usage pattern. However, the information is implicitly included in a user's most recently App usage. For example, a user who usually used to open Apps frequently, but stop using Apps at the same time and location, may probably suffer from low battery life. As a result, the users' App usage history information may implicitly cover part of such information. In the future, we will try to collect the aforementioned information, and explore how they help the prediction.

6 CONCLUSION

This paper presents *CAP*, a context-aware personalized App usage prediction algorithm that takes both contextual information (location & time) and attribution (App & App type) information into consideration. We find that the relationships between App-location, App-time, and App-App type are essential to this prediction and we propose a heterogeneous graph embedding algorithm to map them into one common comparable latent space. We design a user profile with users' historical App usage and trajectory to describe individual dynamic preferences. We evaluate our algorithm based on two large-scale real-world datasets. The evaluation validates 1) the designated three direct and indirect interactions between influential factors of App usage, 2) adopting heterogeneous graph embedding to map these influential factors and 3) the proposed user profile. The results show that *CAP* achieves 30% higher accuracy than the state-of-the-art method *PRME* in terms of Accuracy@5. In terms of MRR, *CAP* achieves 1.5× higher than the straightforward baseline *Sta* and 2× higher than *PRME*. At the same time, prediction with two candidates using our *CAP* achieves higher accuracy than prediction with five candidates using all the other baselines.

REFERENCES

[1] Mobile App Usage - Statistics & Facts. https://www.statista.com/topics/1002/mobile-app-usage/, 2017.

[2] Smartphones industry: Statistics & Facts. https://www.statista.com/topics/840/smartphones/, 2017.

[3] China TeleCom Webpage. http://www.chinatelecom-h.com/en/global/home.php/, 2018.

[4] Hal Abelson, Ken Ledeen, and Chris Lewis. Just deliver the packets, in. *Essays on Deep Packet Inspection", Ottawa". Office of the Privacy Commissioner of Canada. Retrieved*, pages 01–08, 2010.

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

[6] Bilibili. Bilibili, a video sharing website themed around anime, manga, and game fandom based in china, https://www.bilibili.com/, January 2018.

[7] Derya Birant and Alp Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.

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

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[10] Bloomberg. Company overview of beijing mafengwo network technology co.,ltd., https://www.bloomberg.com/research/stocks/private/snapshot.asp?privcapid=142130262, January 2018.
[11] Matthias Böhmer, Lyubomir Ganev, and Antonio Krüger. 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*, pages 267–276. ACM, 2013.
[12] ChangBa. Changba, a chinese music streaming app where users can upload and share their own-recorded songs, https://www.changba.com, January 2018.
[13] Meng Chen, Yang Liu, and Xiaohui Yu. Nlpmm: A next location predictor with markov modeling. 8444:186–197, 2014.
[14] Ning Chen, Steven CH Hoi, Shaohua Li, and Xiaokui Xiao. Simapp: A framework for detecting similar mobile applications by online kernel learning. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 305–314. ACM, 2015.
[15] Ning Chen, Steven CH Hoi, Shaohua Li, and Xiaokui Xiao. Mobile app tagging. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 63–72. ACM, 2016.
[16] Xinlei Chen, Aveek Purohit, Carlos Ruiz Dominguez, Stefano Carpin, and Pei Zhang. Drunkwalk: Collaborative and adaptive planning for navigation of micro-aerial sensor swarms. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 295–308. ACM, 2015.
[17] Xinlei Chen, Aveek Purohit, Shijia Pan, Carlos Ruiz, Jun Han, Zheng Sun, Frank Mokaya, Patric Tague, and Pei Zhang. Design experiences in minimalistic flying sensor node platform through sensorfly. *ACM Transactions on Sensor Networks (TOSN)*, 13(4):33, 2017.
[18] Xinlei Chen, Xiangxiang Xu, Xinyu Liu, Hae Young Noh, Lin Zhang, and Pei Zhang. Hap: Fine-grained dynamic air pollution map reconstruction by hybrid adaptive particle filter. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 336–337. ACM, 2016.
[19] Xinlei Chen, Xiangxiang Xu, Xinyu Liu, Shijia Pan, Jiayou He, Hae Young Noh, Lin Zhang, and Pei Zhang. Pga: Physics guided and adaptive approach for mobile fine-grained air pollution estimation. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 1321–1330. ACM, 2018.
[20] Xinlei Chen, Yulei Zhao, and Yong Li. Qoe-aware wireless video communications for emotion-aware intelligent systems: A multi-layered collaboration approach. *Information Fusion*, 47:1–9, 2019.
[21] Xinlei Chen, Yulei Zhao, Yong Li, Xu Chen, Ning Ge, and Sheng Chen. Social security aided d2d communications: Performance bound and implementation mechanism. *IEEE Journal on Selected Areas in Communications*, 2018.
[22] Xinlei Chen, Zheqi Zhu, Min Chen, and Yong Li. Large-scale mobile fitness app usage analysis for smart health. *IEEE Communications Magazine*, 56(4):46–52, 2018.
[23] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z Sui. Exploring millions of footprints in location sharing services. *ICWSM*, 2011:81–88, 2011.
[24] Karen Church, Denzil Ferreira, Nikola Banovic, and Kent Lyons. Understanding the challenges of mobile phone usage data. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 504–514. ACM, 2015.
[25] Enrique Costa-Montenegro, Ana Belén Barragáns-Martínez, and Marta Rey-López. Which app? a recommender system of applications in markets: Implementation of the service for monitoring users' interaction. *Expert systems with applications*, 39(10):9367–9375, 2012.
[26] Justin Cranshaw, Raz Schwartz, Jason I Hong, and Norman Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city. 2012.
[27] Nick Craswell. Mean reciprocal rank. In *Encyclopedia of Database Systems*, pages 1703–1703. Springer, 2009.
[28] Ctrip. Xiechenglvxing, a chinese provider of travel services including accommodation reservation, transportation ticketing, packaged tours and corporate travel management, https://www.ctrip.com/, January 2018.
[29] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. Diversity in smartphone usage. In *Proc. ACM MobiSys*, pages 179–194, 2010.
[30] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, pages 2069–2075, 2015.
[31] Denzil Ferreira, Eija Ferreira, Jorge Goncalves, Vassilis Kostakos, and Anind K Dey. Revisiting human-battery interaction with an interactive battery interface. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 563–572. ACM, 2013.
[32] Denzil Ferreira, Jorge Goncalves, Vassilis Kostakos, Louise Barkhuus, and Anind K Dey. Contextual experience sampling of mobile application micro-usage. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pages 91–100. ACM, 2014.
[33] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 330–339, New York, NY, USA, 2007. ACM.
[34] Ke Huang, Chunhui Zhang, Xiaoxiao Ma, and Guanling Chen. Predicting mobile application usage using contextual information. In *Proc. ACM UbiComp*, pages 1059–1065, 2012.

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

[36] Alexandros Karatzoglou, Linas Baltrunas, Karen Church, and Matthias Böhmer. 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*, pages 2527–2530. ACM, 2012.

[37] Marc Olivier Killijian. Next place prediction using mobility markov chains. In *The Workshop on Measurement, Privacy, and Mobility*, page 3, 2012.

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

[39] KuanDaiShan. Kuandaishan, a chinese mobile app for crowd-sourced reviews about local businesses, http://club.kdslife.com/f_15.html, January 2018.

[40] KuGouMusic. Kugoumusic, a chinese music streaming and download service, http://www.kugou.com/, January 2018.

[41] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[42] Zhung-Xun Liao, Shou-Chung Li, Wen-Chih Peng, S Yu Philip, and Te-Chuan Liu. On the feature discovery for app usage prediction in smartphones. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1127–1132. IEEE, 2013.

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

[44] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 283–292. ACM, 2013.

[45] Bin Liu, Deguang Kong, Lei Cen, Neil Zhenqiang Gong, Hongxia Jin, and Hui Xiong. Personalized mobile app recommendation: Reconciling app functionality and user privacy preference. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 315–324. ACM, 2015.

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

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

[48] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.

[49] Yu Liu, Xinlei Chen, Yong Niu, Bo Ai, Yong Li, and Depeng Jin. Mobility-aware transmission scheduling scheme for millimeter-wave cells. *IEEE Transactions on Wireless Communications*, 17(9):5991–6004, 2018.

[50] Xinjiang Lu, Zhiwen Yu, Bin Guo, and Xingshe Zhou. Predicting the content dissemination trends by repost behavior modeling in mobile social networks. *Journal of Network and Computer Applications*, 42:197–207, 2014.

[51] Wesley Mathew, Ruben Raposo, and Bruno Martins. Predicting future locations with hidden markov models. In *ACM Conference on Ubiquitous Computing*, pages 911–918, 2012.

[52] MeiLiHui. Meilihui, a chinese online e-business platform for luxuries, http://www.mei.com/index.html, January 2018.

[53] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[54] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. Wherenext: A location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 637–646, New York, NY, USA, 2009. ACM.

[55] NetEase. Netease music, https://music.163.com/, January 2018.

[56] Cisco Visual networking Index. Forecast and methodology, 2016-2021, white paper. *San Jose, CA, USA*, 2016.

[57] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. An empirical study of geographic user activity patterns in foursquare. *ICwSM*, 11:70–573, 2011.

[58] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, pages 693–701, 2011.

[59] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *International Conference on World Wide Web*, pages 811–820, 2010.

[60] Mohamed Sarwat, Ahmed Eldawy, Mohamed F Mokbel, and John Riedl. Plutus: leveraging location-based social networks to recommend potential customers to venues. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, volume 1, pages 26–35. IEEE, 2013.

[61] ShenMeZhiDeMai. Shenmezhidemai, a chinese online e-business platform based on recommendation mode, https://www.smzdm.com/, January 2018.

[62] Kent Shi and Kamal Ali. Getjar mobile application recommendations with very sparse datasets. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 204–212. ACM, 2012.
[63] Choonsung Shin, Jin-Hyuk Hong, and Anind K Dey. Understanding and prediction of mobile application usage for smart phones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 173–182. ACM, 2012.
[64] M. Simon. Google's instant apps hit the play store as a big step toward a future of cloud-based apps. https://www.pcworld.com/article/3234330/android/google-android-instant-apps-play-store.html, 2017.
[65] TalkingData. Talkingdata: China's leading third-party data intelligence solution provider. https://www.talkingdata.com/, 2018.
[66] TalkingData. Talkingdata mobile user demographics. https://www.kaggle.com/c/talkingdata-mobile-user-demographics/data, 2018.
[67] techcrunch. Report: Smartphone owners are using 9 apps per day, 30 per month. https://techcrunch.com/2017/05/04/report-smartphone-owners-are-using-9-apps-per-day-30-per-month/, 2017.
[68] TongCheng. Tongchenglvyou, a chinese traveling service provider, including accommodation reservation, transportation ticketing, packaged tours and corporate travel management, https://www.ly.com/, January 2018.
[69] TuNiu. Tuniu, a leading online leisure travel company in china that offers a large selection of packaged tours, http://www.tuniu.com/, January 2018.
[70] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.
[71] Niels van Berkel, Chu Luo, Theodoros Anagnostopoulos, Denzil Ferreira, Jorge Goncalves, Simo Hosio, and Vassilis Kostakos. A systematic assessment of smartphone usage gaps. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4711–4721. ACM, 2016.
[72] Zhu Wang, Daqing Zhang, Xingshe Zhou, Dingqi Yang, Zhiyong Yu, and Zhiwen Yu. Discovering and profiling overlapping communities in location-based social networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(4):499–509, 2014.
[73] XiaoHongShu. Xiaohongshu, a chinese overseas shopping tip app, https://www.xiaohongshu.com/, January 2018.
[74] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 15–24. ACM, 2016.
[75] Ye Xu, Mu Lin, Hong Lu, Giuseppe Cardone, Nicholas Lane, Zhenyu Chen, Andrew Campbell, and Tanzeem Choudhury. Preference, context and communities:a multi-faceted approach to predicting smartphone app usage patterns. In *Proc. ACM ISWC*, pages 69–76, 2013.
[76] Tingxin Yan, David Chu, Deepak Ganesan, Aman Kansal, and Jie Liu. Fast app launching for mobile devices using predictive user context. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 113–126. ACM, 2012.
[77] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. *Journal of Network and Computer Applications*, 55:170–180, 2015.
[78] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 119–128. ACM, 2013.
[79] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhiwen Yu. Fine-grained preference-aware location search leveraging crowdsourced digital footprints from lbsns. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 479–488. ACM, 2013.
[80] Dragomir Yankov, Pavel Berkhin, and Rajen Subba. Interoperability ranking for mobile applications. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 857–860. ACM, 2013.
[81] Hongyi Yao, Gyan Ranjan, Alok Tongaonkar, Yong Liao, and Zhuoqing Morley Mao. Samples: Self adaptive mining of persistent lexical snippets for classifying mobile application traffic. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 439–451. ACM, 2015.
[82] YiLong. Yilong, a chinese mobile and online travel agency, http://www.elong.com/, January 2018.
[83] Peifeng Yin, Ping Luo, Wang-Chien Lee, and Min Wang. App recommendation: a contest between satisfaction and temptation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 395–404. ACM, 2013.
[84] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 34–43, New York, NY, USA, 2011. ACM.
[85] Youdao. Netease youdao dictiontion, https://play.google. com/store/apps/details?id=com.youdao.dict&hl=en, January 2018.
[86] Donghan Yu, Yong Li, Fengli Xu, Pengyu Zhang, and Vassilis Kostakos. Smartphone app usage prediction using points of interest. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):174, 2018.
[87] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T Hanratty, and J. Han. Gmove: Group-level mobility modeling using geo-tagged social media. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1305–1314, 2016.
[88] Chunhui Zhang, Xiang Ding, Guanling Chen, Ke Huang, Xiaoxiao Ma, and Bo Yan. Nihao: A predictive smartphone application launcher. In *International Conference on Mobile Computing, Applications, and Services*, pages 294–313. Springer, 2012.

[89] Sha Zhao, Julian Ramos, Jianrong Tao, Ziwen Jiang, Shijian Li, Zhaohui Wu, Gang Pan, and Anind K. Dey. Discovering different kinds of smartphone users through their application usage behaviors. In *Proc. ACM UbiComp*, pages 498–509, 2016.
[90] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. Stellar: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *AAAI*, pages 315–322, 2016.
[91] Xiaoxing Zhao, Yuanyuan Qiao, Zhongwei Si, Jie Yang, and Anders Lindgren. Prediction of user app usage behavior from geo-spatial data. In *Proc. ACM GeoRich*, pages 1–6, 2016.
[92] Vincent Wenchen Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *AAAI*, volume 10, pages 236–241, 2010.
[93] Zhihu. Zhihu, chinese famous question-and-answer website, https://www.zhihu.com/signin?next=%2f, January 2018.
[94] Hengshu Zhu, Enhong Chen, Kuifei Yu, Huanhuan Cao, Hui Xiong, and Jilei Tian. Mining personal context-aware preferences for mobile users. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1212–1217. IEEE, 2012.
[95] Hengshu Zhu, Chuanren Liu, Yong Ge, Hui Xiong, and Enhong Chen. Popularity modeling for mobile apps: A sequential approach. *IEEE transactions on cybernetics*, 45(7):1303–1314, 2015.
[96] Hengshu Zhu, Hui Xiong, Yong Ge, and Enhong Chen. Mobile app recommendations with security and privacy awareness. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 951–960. ACM, 2014.

Received August 2018; revised November 2018; accepted January 2019