Modeling Spatio-Temporal App Usage for a Large User Population

HUANDONG WANG, YONG LI, and SIHAN ZENG, Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, China
GANG WANG, Department of Computer Science, Virginia Tech, USA
PENGYU ZHANG, Department of Computer Science, Stanford University, USA
PAN HUI, Department of Computer Science and Engineering, University of Helsinki, Finland
DEPENG JIN, Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, China

With the wide adoption of mobile devices, it becomes increasingly important to understand how users use mobile apps. Knowing when and where certain apps are used is instrumental for app developers to improve app usability and for Internet service providers (ISPs) to optimize their network services. However, modeling spatio-temporal patterns of app usage has been a challenging problem due to the complicated usage behavior and the very limited personal data. In this paper, we propose a Bayesian mixture model to capture when, where and what apps are used and predict future app usage. To solve the challenge of data sparsity, we apply a hierarchical Dirichlet process to leverage the shared spatio-temporal patterns to accurately model users with insufficient data. We then evaluate our model using a large dataset of app usage traces involving 1.7 million users over 3503 apps. Our analysis shows a clear correlation between the user's location and the apps being used. Extensive evaluations show that our model can accurately predict users' future locations and app usage, outperforming the state-of-the-art algorithms by 11.7% and 11.1%, respectively. In addition, our model can be used to synthesize app usage traces that do not leak user privacy while preserving the key data statistical properties.

CCS Concepts: • **Human-centered computing** → **User models**; • **Information systems** → *Spatial-temporal systems*; • **Networks** → *Network services*.

Additional Key Words and Phrases: app usage, Bayesian mixture model, spatio-temporal pattern

ACM Reference Format:
Huandong Wang, Yong Li, Sihan Zeng, Gang Wang, Pengyu Zhang, Pan Hui, and Depeng Jin. 2019. Modeling Spatio-Temporal App Usage for a Large User Population. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 1, Article 27 (March 2019), 23 pages. https://doi.org/10.1145/3314414

1 INTRODUCTION

The wide adoption of mobile devices and applications (apps) has enabled highly convenient and ubiquitous access to Internet services. For app developers and network service providers, it becomes increasingly important

Authors' addresses: Huandong Wang, whd14@mails.tsinghua.edu.cn; Yong Li, liyong07@tsinghua.edu.cn; Sihan Zeng, cengsh14@mails.tsinghua.edu.cn, Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, China; Gang Wang, gangwang@vt.edu, Department of Computer Science, Virginia Tech, USA; Pengyu Zhang, pyzhang@stanford.edu, Department of Computer Science, Stanford University, USA; Pan Hui, panhui@cse.ust.hk, Department of Computer Science and Engineering, University of Helsinki, Finland; Depeng Jin, jindp@tsinghua.edu.cn, Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, China.

2474-9567/2019/3-ART27 $15.00
https://doi.org/10.1145/3314414

to understand how users use mobile apps under various contexts. For example, knowing *when* and *where* users use certain apps is instrumental in improving app usability, optimizing network service quality, and providing context-aware recommendations and assistance [25, 55, 72]. Initial research already demonstrated strong location dependent patterns of app usage [22, 53], indicating the benefit of considering location contexts to understand such behaviors.

So far, our understanding of the mobile app usage over a large user population is still very limited due to a number of key challenges. First, it is extremely difficult to obtain unbiased app usage data at a large scale. Most existing studies [19, 22] collect data by recruiting crowdsourced volunteers, which are limited in the number of the participants. Second, app usage behavior is very complicated with thousands of apps being used over the city, which leads to a challenge of building an accurate model for all of them. Third, app usage is extremely heterogeneous where a small number of users contribute most of the app usage data [19, 53]. The resulting data sparsity makes it challenging to accurately model user behavior to cover a large population.

In this paper, we seek to build a scalable model for spatio-temporal mobile app usage through a data-driven approach for large scale populations. Our model seeks to provide a deeper understanding of when and where users use different apps, and predict app usage given the spatio-temporal contexts. In addition, our model enables producing synthesized traces, which enables researchers to study the mobile app usage at scale while preserving the privacy of original mobile users.

More specifically, we propose a Bayesian mixture model to capture app usage patterns in three domains simultaneously — location, time, and app. First, in the spatial domain, we leverage a multinomial distribution across locations to eliminate disruptive effect of the randomness of human mobility [21]. Second, in the temporal domain, we characterize the app usage pattern using Gaussian distribution and capture users' periodic app usage behavior. Finally, in the app domain, each app is regarded as a semantic word, and we build a model to capture the "topic" semantics of app usage based on these semantic words. We also apply multi-task learning (hierarchical Dirichlet process) to share the common apps' usage pattern across users to cover users with insufficient data. Finally, we evaluate our proposed model based on a large-scale app usage dataset containing 1.5 billion app usage records of 1.7 million users over 3503 apps. In summary, our paper makes the following contributions:

- We propose a novel Bayesian mixture model based on hierarchical Dirichlet process, which characterizes user, app, location, and time in a cohesive manner. In addition, we address the data sparsity challenge by sharing the statistical app usage patterns across users to model their behavior.
- Our model provides new understanding of the spatio-temporal patterns of app usage. The result shows a clear correlation between the point of interest (POI) distribution and the categories of the apps being used. Extensive evaluations confirm the high accuracy of our model in predicting users' location-based app usage. It improves the accuracy of user mobility prediction by 11.7% compared with the state-of-the-art algorithms, and outperforms the baseline algorithms in predicting app usage by 11.1%.
- Our model also helps to produce synthesized app usage traces to avoid user privacy leakage while allowing meaningful data minings. Results show that the average spatial and temporal gaps between the real and the synthesized traces are 6 km and 13 hours, indicating that users' true locations are hidden with privacy preserving. Meanwhile, the synthesized traces preserve key statistical characteristics such as the most frequently used apps, app usage entropy, and spatio-temporal distributions.

The rest of the paper is structured as follows. In Section 2, we present the mathematical model and formulate our problem, and give a high-level overview of our system. In Section 3, we introduce our Bayesian model for app usage traces. In section 4, we present location prediction, app prediction, and trace synthesizing algorithms based on the proposed model. In Section 5, we extensively evaluate the performance of our proposed algorithms compared with existing algorithms. After discussing related work in Section 6, we summarize our main findings in Section 7.

Table 1. A list of commonly used notations.

Notation	Description
$\mathcal{U}$	The set of all users.
$\mathcal{L}$	The set of locations.
$\mathcal{A}$	The set of apps.
$\boldsymbol{r}_i^u$	The ith records for user u.
R^u	The set of all records of user u.
t_i^u	The timestamp of the ith record of user u.
$\boldsymbol{x}_i^u$	The number of times user u visit different locations in the time bin t_i^u.
$\boldsymbol{a}_i^u$	The number of times user u uses different apps in the time bin t_i^u.
N_u	The number of records for user u.
K	The number of patterns/components.
n_k	The number of records for pattern k.
ξ_k	Parameter of pattern k, which consists of ζ_k, $\boldsymbol{\theta}_k$, and $\boldsymbol{\phi}_k$.
π_k	Mixture weight of pattern k, *i.e.*, prior probability of that an arbitrary record is generated by component k.
$\zeta_k = (\mu_k, \sigma_k)$	The mean and variance of Gaussian temporal distribution for pattern k.
$\boldsymbol{\theta}_{uk}$	Parameter of multinomial location distribution for pattern k and user u, which is an $\lvert\mathcal{L}\rvert$-sized vector.
$\boldsymbol{\phi}_{uk}$	Parameter of multinomial app distribution for pattern k and user u, which is an $\lvert\mathcal{A}\rvert$-sized vector.
$m_0, \kappa_0, \psi_0, \nu_0$	Hyper-parameters of NIG prior for $(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$.
β, γ	Hyper-parameters of Dirichlet prior for $\boldsymbol{\theta}_{uk}, \boldsymbol{\phi}_{uk}$.
α, ϵ	Concentration parameters of HDP.
$\boldsymbol{H}_w$	TF-IDF normalized POI distribution around location l.
$\boldsymbol{G}_w$	One-hot vector indicating the category of app w.

2 SYSTEM MODEL AND OVERVIEW

In this section, we propose a mathematical model and describe our problem. For readability, we summarize the major notations used throughout the paper in Table 1.

Mathematical Setup. In real life, users' app usage is unevenly distributed in the temporal dimension. For example, the duration of a continuous access to an app varies from 1 seconds to 2 days [53]. In order to capture the patterns without the influence of such temporal heterogeneity, we divide the time span into fixed sized time bins. App usage behavior in each time bin of user u is aggregated into a 3-tuple app usage record $\boldsymbol{r}_i^u = (t_i^u, \boldsymbol{x}_i^u, \boldsymbol{a}_i^u)$, where t_i^u represents the time of day (in time bins), $\boldsymbol{x}_i^u$ represents the user's location, and $\boldsymbol{a}_i^u$ represents the app usage information. Similar with the temporal dimension, we also divide locations into geographical regions. Let $\mathcal{L}$ denote the set of all geographical regions. Then, $\boldsymbol{x}_i^u$ is defined as an $\lvert\mathcal{L}\rvert$-sized vector, where $\boldsymbol{x}_i^u(l)$ represents the number of times user u visits location l within the time bin. We further denote $\mathcal{A}$ as the set of all apps. Then, $\boldsymbol{a}_i^u$ is defined as an $\lvert\mathcal{A}\rvert$-sized vector, where $\boldsymbol{a}_i^u(w)$ represents the number of times user u uses app w within the time bin. Finally, we define $\mathcal{U}$ as the set of all users. Given any user $u \in \mathcal{U}$, we define the set of all his app usage records as $R^u = \{\boldsymbol{r}_1^u, \boldsymbol{r}_2^u, ..., \boldsymbol{r}_{N_u}^u\}$, where N_u is the number of records belonging to user u.

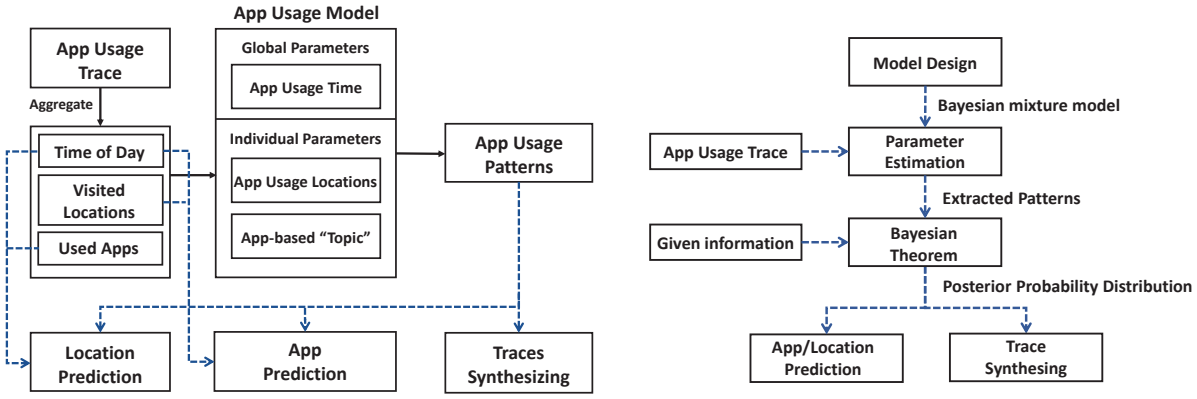
Problem Description. Based on the above mathematical definitions, the problem that we investigate can be expressed as following:

Spatio-Temporal App Usage Modeling:

Given: A set of users $\mathcal{U}$ and their app usage traces $\{R^u\}_{u \in \mathcal{U}}$.

Problem: For each user, discover the latent states that reflect the app usage behavior from a multidimensional view, *i.e.*, when and where the user is using what kind of apps.

System Overview. Fig. 1(a) shows the framework of our system, where arrows represent the flow of data and boxes represent data processing modules. In this system, we model the users' app usage behavior in three



(a) Framework (b) Workflow

Fig. 1. The framework and workflow of our system.

domain — location, time and app. First, we aggregate users' app usage traces in different time bins into app usage records. Then, from these records, we extract users' app usage patterns that characterize users' main app usage time, locations, and the "topic" of the used apps. Specifically, since different users may have different places of residence and office, and they may also prefer different apps, parameters describing location and app-based topic are separate for different users. On the other hand, most users have similar daily schedule, *e.g.*, sleep time, commute time, working time, etc. Thus, parameters describing app usage time are globally shared by different users. Overall, by using a Bayesian mixture model based on hierarchical Dirichlet process, we achieve extraction of users' app usage patterns. This model will be discussed in detail in Section 3.

A number of applications can be realized based on the obtained app usage patterns. We focus on three major applications in our work, including location prediction, app prediction, and app usage trace synthesizing. Utilizing the time and app usage information, we can predict users' location with more accuracy based on their historical patterns. Similarly, we can perform app usage prediction. Last but not least, the Bayesian framework in our system brings a strong ability to synthesize plausible app usage traces, which statistically resemble real traces while hide users' true location with privacy preserving. These applications will be discussed in detail in Section 4.

3 SPATIO-TEMPORAL APP USAGE MODEL

In this section, we design an app usage model that takes users' app usage records as input, and extracts their app usage patterns as output. Our intuition is that users tend to use different apps at different places. For example, business apps (e.g., email apps) are used more in central business districts compared with residential areas. Similarly, users also tend to use different apps at different time. Thus, we define an app usage pattern as *when and where a certain cluster of apps are used*. Then, the goal of the app usage model is to extract the clusters of apps corresponding to different app usage patterns, and describe the spatial and temporal distribution of these patterns.

The workflow of our proposed app usage model is shown in Fig. 1(b), which can be described as follows. First, we design a probabilistic model of users' app usage records, where each app usage pattern is highly abstracted by a set of parameters. Then, we use this model to fit the real app usage records, and estimate parameters of the model in this process. Finally, the app usage model outputs these parameters to represent the extracted app usage patterns, which can be utilized in different applications.

In this section, we first present an individual probabilistic model to describe single user's app usage patterns in Section 3.1. Then, in order to share similar patterns and parameters across users, we further leverage hierarchical Dirichlet process to propose a multi-user model in Section 3.2. Finally, we introduce how to estimate parameters of the proposed probabilistic model in Section 3.3.

3.1 Individual App Usage Model

We first focus on the individual usage behavior, and propose a probabilistic model to describe the records $R^u = \{\boldsymbol{r}_i^u\}_{i=1}^{N_u}$ of user $u \in \mathcal{U}$. Since we focus on individuals, we omit the superscript u of variables for simplicity in this section.

We adopt the Bayesian mixture model to describe single user's app usage patterns. It can be formally defined as a linear superposition of finite components, which can be represented as follows,

$$p(\boldsymbol{r}_i) = \sum_{k=1}^{K} \pi_k f(\boldsymbol{r}_i|\boldsymbol{\xi}_k), \tag{1}$$

where each density $f(\cdot|\boldsymbol{\xi}_k)$ is called a component and characterized by its own parameters $\boldsymbol{\xi}_k$, which is used to describe an app usage pattern of the target user. In addition, π_k is the mixture weight of component k, which satisfies $\sum_{k=1}^{K} \pi_k = 1$. As mentioned above, the goal of the app usage model is to estimate parameters $\boldsymbol{\xi}_k$ and π_k.

To make the problem of estimating parameters in the mixture model tractable, we introduce a latent discrete random variable z_i for the ith observed data point $\boldsymbol{r}_i$ to indicate which component/pattern it belongs to. Then, based on definitions, we have $p(z_i = k) = \pi_k$. If we have $z_i = k$, the ith data point is generated by the component k, and we have $p(\boldsymbol{r}_i|z_i = k) = f(\boldsymbol{r}_i|\boldsymbol{\xi}_k)$. Then, $f(\cdot|\boldsymbol{\xi}_k)$ is a spatio-temporal app usage pattern of the user, which describes when and where he is using what kind of apps. Thus, we decompose it into three factors, which can be expressed as follows:

$$p(\boldsymbol{r}_i|z_i = k) = f(\boldsymbol{r}_i|\boldsymbol{\xi}_k) = p(t_i|\boldsymbol{\zeta}_k)p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)p(\boldsymbol{a}_i|\boldsymbol{\phi}_k), \tag{2}$$

where $\boldsymbol{\xi}_k$ consists of $\boldsymbol{\zeta}_k$, $\boldsymbol{\theta}_k$, and $\boldsymbol{\phi}_k$. They are the parameters describing the pattern's temporal distribution, spatial distribution, and app distribution, respectively. Note that though in the probability density of each component shown in (2), time, location, and app usage are regarded to be independent with each others. In the probability density of their mixture (1), users' behavior in these three dimensions are highly correlated.

For temporal distribution $p(t_i|\boldsymbol{\zeta}_k)$, it has been found that the users' daily movements and app usage are fairly predictable with repeated patterns [47]. In order to capture this periodicity, we model the temporal distribution of each pattern in terms of the time of day by Gaussian distribution, which has been shown to have a good performance in a number of works [37, 66]. Specifically, we denote $\boldsymbol{\zeta}_k = (\mu_k, \sigma_k)$ to be its mean and variance. Then, the probability density of t_i can be represented as:

$$p(t_i|\boldsymbol{\zeta}_k) = \mathcal{N}(t_i|\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp(-\frac{(t_i - \mu_k)^2}{2\sigma_k^2}).$$

In terms of spatial dimension, it is possible for a user to visit multiple locations within the time bin t_i. Thus, we model $\boldsymbol{x}_i$ by a multinomial distribution over locations, which can be expressed as follow:

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}_k) = \prod_{l \in \mathcal{L}} \theta_k(l)^{x_i(l)}.$$

As for app dimension, we regard each app as a semantic word, and each pattern represents a "topic" for apps [8]. Thus, we model $\boldsymbol{a}_i$ by a multinomial distribution over apps, which can be expressed as follow:

$$p(\boldsymbol{a}_i|\boldsymbol{\phi}_k) = \prod_{w \in \mathcal{A}} \phi_k(w)^{a_i(w)}.$$

Since we adopt Bayesian framework, parameters in our model are regarded as random variables. We use the common conjugate prior distributions for these parameters. For parameters of category and multinomial distribution, *i.e.*, $\boldsymbol{\pi}$, $\boldsymbol{\theta}_k$, and $\boldsymbol{\phi}_k$, we use Dirichlet prior to model them. For parameters of Gaussian distribution, *i.e.*, $\boldsymbol{\zeta}_k = \{\mu_k, \sigma_k\}$, we use the Normal-inverse-gamma (NIG) distribution to model them. These conjugate priors enable us to use collapsed Gibbs sampling in parameter estimation, which will be introduced in detail in Section 3.3.

3.2 HDP-based Multi-User Model

The above introduced Bayesian mixture model provides an approximation of the individual app usage behavior. However, there are still two problems unsolved. The first one is that the number of components/patterns, *i.e.*, K is different for different users, and it has critical impact on correctly modeling the app usage behavior. The method should automatically determine K based on the records of the specific users. The second problem is data sparsity. According to [19], the active time of app usage per user is highly skewed: over 20% of users have active time less than less than 30 minutes per day, which making modeling their app usage behavior very challenging in practice. Motivated by this limit, we aim to share similar patterns and parameters among users to help modeling their app usage patterns more accurately, especially for the users with sparse data. In order to solve these two problems, we extend our model based on hierarchical Dirichlet process (HDP) [46]. Specifically, Dirichlet process (DP) is a Bayesian non-parametric approach that can estimate the number of components in users' records, and its extension – HDP further shares common patterns between different users to improve the performance.

Dirichlet Process App Usage Model: The mixture model based on Dirichlet process has a possibly infinite number of mixture components, and it defines a concentration parameter $\alpha > 0$, which impacts the number of components. We denote Dirichlet process with concentration parameter α as $DP(\alpha)$.

A widely employed metaphor for the Dirichlet process is the Chinese restaurant process (CRP). In CRP, each data point is regarded as a customer. When a new customer enters a Chinese restaurant, he sits down at a table with a probability proportional to the number of customers already sitting here. In addition, he opens a new table with a probability proportional to α. Then, by this process, the number of tables (components) can be determined by concrete customers (data points).

In our model, since we use conjugate priors for all distributions, we are able to perform inference of DP mixture model through collapsed Gibbs sampling [46]. It starts by randomly initializing all z_i, and then iterates sampling each z_i based on the following equations:

$$\begin{cases} p(z_i = k, k \leq K | z_{-i}, \alpha) \sim n_k \int f(\boldsymbol{r}_i | \xi_k) p(\xi_k | z_{-i}) d\xi_k, \\ p(z_i = K+1 | z_{-i}, \alpha) \sim \alpha \int f(\boldsymbol{r}_i | \xi) p(\xi) d\xi, \end{cases}$$

where $z_{-i} = \{z_j | j \neq i,\ 1 \leq j \leq N_u\}$, and n_k is the number of records assigned to component k. $p(z_i = k, k \leq K | z_{-i}, \alpha)$ is in proportion to the number of data points belonging to the component k and collapse probability density of the kth component around $\boldsymbol{r}_i$.

Hierarchical Dirichlet Process App Usage Model: The above introduced Dirichlet process only considers the single user. However, individual model may suffer from the sparsity of users' historical records. Motivated by this limit, we further leverage HDP to jointly model all users' spatio-temporal app usage over the city, and leverage the similar spatio-temporal app usage patterns among users to model users with insufficient data more accurately.

HDP consists of two levels of Dirichlet process. In the first level, the global patterns of all users are drawn from the Dirichlet process based on the prior distribution, while in the second level, patterns of each user are drawn from the global patterns. In this way, we achieve the goal that different users share the same patterns and

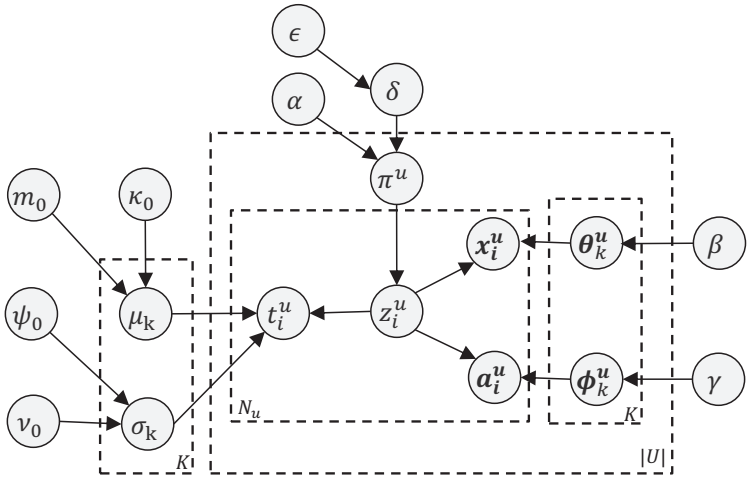


Fig. 2. Graphical model.

parameters. The collapsed Gibbs sampling of HDP can be expressed as follows [46]:

$$\begin{cases} p(z_i^u = k, k \leq K | z_{-ui}, \alpha) \sim (n_k^u + \alpha n_k) \int f(\boldsymbol{r}_i^u | \xi_{uk}) p(\xi_{uk} | z_{-ui}) d\xi_{uk}, \\ p(z_i^u = K+1, | z_{-ui}, \alpha) \sim \alpha\epsilon \int f(\boldsymbol{r}_i | \xi) p(\xi) d\xi, \end{cases} \tag{3}$$

where $z_{-ui} = \{z_j^w | w \in \mathcal{U}, 1 \leq j \leq N_w, j \neq i \text{ or } w \neq u\}$, n_k^u is the number of observed data points of user u assigned to component k, and n_k is the number of data points of all users assigned to component k.

According to the strict definition of HDP, the parameters of components, *i.e.*, ξ_{uk} are global parameters independent with u. However, in a metropolis, there are thousands of buildings and streets, and thousands of apps being used all over the city. Different users have different residence, office, and prefer different apps. Thus, sharing $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ between different users leads to a large number of patterns. On the other hand, the computational complexity of collapsed Gibbs sampling is in proportion to the number of patterns. Thus, in order to avoid too many patterns, we only share temporal parameters between different users. That is, $\xi_{uk} = \{\zeta_k, \boldsymbol{\theta}_{uk}, \boldsymbol{\phi}_{uk}\}$, where $\boldsymbol{\theta}_{uk}$ and $\boldsymbol{\phi}_{uk}$ are different for different users, while they share the same ζ_k. We denote our proposed model as App Usage Model (AUM). The generative process of AUM is shown in Algorithm 1, and the corresponding graphical model is shown in Fig. 2.

ALGORITHM 1: Generative Process for AUM

Hyper-parameters: $\epsilon, \alpha, m_0, \kappa_0, \psi_0, \nu_0, \beta, \gamma$;
$\boldsymbol{\delta} \sim DP(\epsilon)$;
for $k \in \{1, 2, ..., K\}$ **do**
  $(\mu_k, \sigma_k) \sim \text{NIG}(.|m_0, \kappa_0, \psi_0, \nu_0)$;
for $u \in \mathcal{U}$ **do**
  $\boldsymbol{\pi}^u \sim DP(\alpha, \boldsymbol{\delta})$;
  for $k \in \{1, 2, ..., K\}$ **do**
    $\boldsymbol{\theta}_{uk} \sim Dirichlet(.|\beta), \boldsymbol{\phi}_{uk} \sim Dirichlet(.|\gamma)$;
  for $i \in \{1, 2, ..., N_u\}$ **do**
    $z_i^u \sim Category(.|\pi^u)$, $t_i^u \sim \mathcal{N}(.|\mu_{z_i^u}, \sigma_{z_i^u})$;
    $\boldsymbol{x}_i^u \sim Multinormial(.|\boldsymbol{\theta}_{u,z_i^u})$;
    $\boldsymbol{a}_i^u \sim Multinormial(.|\boldsymbol{\phi}_{u,z_i^u})$;

3.3 Parameter Estimation

Since we use conjugate priors, $\boldsymbol{\zeta}_k|z_{-ui}$ follows NIG distribution, and $\boldsymbol{\theta}_i^u|z_{-ui}$, $\boldsymbol{\phi}_i^u|z_{-ui}$ follow Dirichlet distribution. Then, the collapsed probability density of component k at $\boldsymbol{r}_i^u$ can be expressed as follows:

$$\int f(\boldsymbol{r}_i^u|\xi_{uk})p(\xi_{uk}|z_{-ui})d\xi_{uk} = \int p(t_i^u|\zeta_k)p(\zeta_k|z_{-ui})d\zeta_k \int p(\boldsymbol{x}_i^u|\boldsymbol{\theta}_k^u)p(\boldsymbol{\theta}_k^u|z_{-ui})d\boldsymbol{\theta}_k \int p(\boldsymbol{a}_i^u|\boldsymbol{\phi}_k^u)p(\boldsymbol{\phi}_k^u|z_{-ui})d\boldsymbol{\phi}_k, \quad (4)$$

where we omit the hyper-parameters $\{m_0, \kappa_0, \psi_0, \nu_0, \beta, \gamma\}$ for simplicity. For the first part, according to [40], we have:

$$\int p(t_i|\zeta_k)p(\zeta_k|z_{-ui})d\zeta_k = t_{2\nu_k}(t_i|m_k, \frac{\psi_k(\kappa_k+1)}{\kappa_k\nu_k}), \quad (5)$$

where $t_{2\nu_k}(\cdot)$ is the probability density function of multivariate t-distribution, and $\boldsymbol{\zeta}_k|z_{-ui}$ follows NIG distribution with parameters $(m_k, \kappa_k, \psi_k, \nu_k)$, which can be calculated as:

$$\begin{cases} m_k = (\kappa_0 m_0 + n_k\bar{t}_k)/(\kappa_0+n_k), \\ \psi_k = \psi_0 + \frac{1}{2}\sum_{z_j=k, j\neq i}(t_j-\bar{t}_k)^2 + \frac{1}{2}\frac{\kappa_0 n_k}{\kappa_0+n_k}(m_0-\bar{t}_k)^2, \\ \kappa_k = \kappa_0 + n_k,\ \ \nu_k = \nu_0 + \frac{1}{2}n_k, \end{cases} \quad (6)$$

where $\bar{t}_k$ is the mean value of t_j assigned to component k, *i.e.*, $\bar{t}_k = \frac{1}{n_k}\sum_{z_j=k, j\neq i} t_j$.

As for the second part of (4), we have

$$\int p(\boldsymbol{x}_i^u|\boldsymbol{\theta}_k^u)p(\boldsymbol{\theta}_k^u|z_{-ui}) = \frac{B(\boldsymbol{b}_k^u + \boldsymbol{x}_i^u)}{B(\boldsymbol{b}_k^u)}, \quad (7)$$

ALGORITHM 2: Collapsed Gibbs Sampling for AUM

Input: M, $\{R^u\}_{u\in\mathcal{U}}$, $\{\epsilon, \alpha, m_0, \kappa_0, \psi_0, \nu_0, \beta, \gamma\}$, K_0.
Output: K, $\{m_k, \kappa_k, \psi_k, \nu_k\}_K$, $\{n_k^u, \boldsymbol{b}_k^u, \boldsymbol{c}_k^u\}_{\mathcal{U}\times K}$.
Initialize: $K \leftarrow K_0$. Randomly initialise $z_i^u \in \{1, ..., K\}$.
for *iter* $\in \{1, ..., M\}$ **do**
  for $u \in \mathcal{U}$ **do**
    for $i \in \{1, ..., N_u\}$ **do**
      Remove $\boldsymbol{r}_i^u$ from pattern z_i^u.
      for $k \in \{1, ..., K\}$ **do**
        Update n_k^u and n_k;
        Update $m_k, \kappa_k, \psi_k, \nu_k, \boldsymbol{b}_k^u, \boldsymbol{c}_k^u$ based on (6)~(9);
        Calculate $\int f(\boldsymbol{r}_i^u|\xi_{uk})p(\xi_{uk}|z_{-ui})d\xi_{uk}$ based on (4)~(9);
        $l_k \leftarrow (n_k^u + \alpha n_k)$;
        $p_k \leftarrow l_k \int f(\boldsymbol{r}_i^u|\xi_{uk})p(\xi_{uk}|z_{-ui})d\xi_{uk}$;
      $p_{K+1} \leftarrow \alpha\epsilon \int f(\boldsymbol{r}_i^u|\xi)p(\xi)d\xi$;
      $c = \sum_{k=1}^{K+1} p_k$;
      With probability p_k/c **do**
        $z_i^u \leftarrow k$;
      if $z_i^u = K+1$ **then**
        $K \leftarrow K+1$;

where $B(\cdot)$ is the multivariate Beta function, and $\boldsymbol{\theta}_k^u|z_{-ui}$ follows Dirichlet distribution with parameter $\boldsymbol{b}_k^u$, which can be calculated as follows:

$$\boldsymbol{b}_k^u = \sum_{z_j^u=k, j\neq i} \boldsymbol{x}_j^u + \beta \cdot \mathbf{1}, \tag{8}$$

where $\mathbf{1}$ is the $|\mathcal{L}|$-sized vector with all elements to be 1.

As for the third part of apps in (4), we define $\boldsymbol{c}_k^u$ as follow,

$$\boldsymbol{c}_k^u = \sum_{z_j^u=k, j\neq i} \boldsymbol{a}_j^u + \gamma \cdot \mathbf{1}. \tag{9}$$

Since apps in our model is symmetrical with locations, we can just use $\boldsymbol{a}_i^u$, $\boldsymbol{\phi}_k^u$, and $\boldsymbol{c}_k^u$ to replace $\boldsymbol{x}_i^u$, $\boldsymbol{\theta}_k^u$, and $\boldsymbol{b}_k^u$ in (7), and obtain the expression of the third part.

We show the detailed process of collapsed Gibbs sampling for AUM in Algorithm 2. It takes the initial number of components K_0, app usage records of users, and hyper-parameters as the input. Then, it starts by randomly initializing all z_i^u, and iterates sampling z_i^u for each $u \in \mathcal{U}$ and $i \in \{1, ..., N_u\}$ based on (3)~(9). After each time of sampling, it recomputes the statistics $\{m_k, \kappa_k, \psi_k, \nu_k\}$ and $\{n_k^u, \boldsymbol{b}_k^u, \boldsymbol{c}_k^u\}$ for each user and each pattern. After M iterations, this algorithm outputs the statistics as the final results.

Computational complexity of Algorithm 2 is $O(MK|R^{\mathcal{U}}| \cdot (|\mathcal{L}| + |\mathcal{A}|))$, where $|R^{\mathcal{U}}|$ is the number of app usage records of all users in $\mathcal{U}$, M is the number of iterations, and K is the number of mixture components. Note that $|\mathcal{L}|$, $|\mathcal{A}|$ and M are fixed values. K is limited by the periodicity of human behavior, which have moderate values in practice. Thus, the computational complexity roughly grows linearly with $|R^{\mathcal{U}}|$, which is feasible in practice.

4 APPLICATIONS

As shown in Fig. 1(b), based on the estimated parameters of the app usage model, given an arbitrarily part of an app usage record, we can obtain the posterior probability of the rest part based on Bayesian theorem, which can be used for prediction. In addition, we can also draw samples from the obtained probabilistic distribution to synthesize plausible app usage traces. Thus, in this paper, we mainly consider three major applications including location prediction, app prediction, and app usage trace synthesizing, which are introduced in detail in the following sections.

Location Prediction: Accurate human mobility prediction is important for many applications including energy optimization [14], POI recommendation [61], etc. In most existing works, users' future location is predicted only based on his historical locations [34, 36]. Since users' app usage behavior is highly correlated with their mobility [22, 53], we can further use app information to aid to predict users' future location based on our proposed model.

Given the target user u, time of day t_i, and app usage vector $\boldsymbol{a}_i$, the problem of location prediction is to estimate the true location distribution among geographical regions, *i.e.*, $\boldsymbol{x}_i$.

In order to achieve this goal, we first calculate the probability of that user u is in the pattern k according to t_i and $\boldsymbol{a}_i$ based on (3) and (4). Denote this probability as $p(z_i = k)$, which can be calculated as follow:

$$p(z_i = k) = \frac{n_k^u + \alpha n_k}{\sum_k (n_k^u + \alpha n_k)} \int p(t_i|\boldsymbol{\zeta}_k) p(\boldsymbol{a}_i|\boldsymbol{\phi}_{uk}) d\boldsymbol{\zeta}_k d\boldsymbol{\phi}_{uk}, \tag{10}$$

Then, $\boldsymbol{x}_i$ is estimated by the probabilistic combination of the location parameter $\boldsymbol{\theta}_{uk}$ of K patterns. By using the expectation $\boldsymbol{b}_k^u / ||\boldsymbol{b}_k^u||_1$ to replace $\boldsymbol{\theta}_{uk}$, the location prediction results can be expressed as:

$$\hat{\boldsymbol{x}}_i = \sum_{k=1}^{K} p(z_i = k) \boldsymbol{b}_k^u / ||\boldsymbol{b}_k^u||_1. \tag{11}$$

App Usage Prediction: App usage prediction is also an important problem for its applications in improving user experience [22, 55]. Except for users' historical app usage behavior, users' location information can also help to predict users' app usage behavior in our framework.

The target of app usage prediction is to estimate the app usage vector $\boldsymbol{a}_i$ based on the time t_i and location vector $\boldsymbol{x}_i$. Since location and app information are symmetrical in our model, by replacing $(\boldsymbol{x}, \boldsymbol{b}_k^u, \boldsymbol{\theta}_{uk})$ in (10) and (11) with $(\boldsymbol{a}, \boldsymbol{c}_k^u, \boldsymbol{\phi}_{uk})$, we obtain the expression of the estimated results of our model.

App Usage Trace Synthesizing: App usage traces are instrumental for applications such as improving app usability, optimizing network service quality, and providing context-aware recommendations [25, 55]. However, utilizing app usage traces in research or business also raises privacy concerns, since personal information is possible to be exposed through them. For example, users do not want others to know *where they are* and *what apps they use* at certain time.

Many techniques have been proposed to protect users' privacy by adding random noise to user data [1, 2, 18]. However, these techniques also reduce the utility of the dataset [5]. At the same time, other techniques seek to protect users' privacy by synthesizing plausible traces [6, 23, 27, 42, 59]. The synthesized traces are able to statistically resemble real traces, *i.e.*, protect the utility of the dataset to assist various applications. In addition, the synthesized traces also protect users' privacy. Here we consider a scenario that the adversary wants to extract users' true spatio-temporal location from the synthesized traces, *i.e.*, *where they are* at certain time. Thus, the synthesized trace should be far away from the real trace at least in one dimension between time and space.

Inspired by this idea, in this work, we seek to synthesize app usage traces through our model. Specifically, the synthesized traces should resemble real traces in terms of both the location-related statistical metric and the app-related statistical metric. In addition, the synthesized traces also hide users' true app usage time and location to protect their privacy. We show the process of synthesizing app usage traces in Algorithm 3, where the number of records of each day and each user remains unchanged. This algorithm takes statistics of app usage patterns as the input, which are the output of Algorithm 2. Then, for each user u, it repeats for N_u times to draw u's app usage traces from distributions introduced in Section 3.

5 EVALUATION

In this section, we first introduce the utilized dataset. Then, we interpret the patterns extracted from our system in terms of multi-dimensional view. Further, we evaluate our model against state-of-the-art algorithms in location prediction and app usage prediction. Finally, we show the trace synthesizing ability of our proposed model.

ALGORITHM 3: Synthesizing App Usage Trace

Input: $\{m_k, \kappa_k, \psi_k, \nu_k\}_K$, $\{n_k^u, \boldsymbol{b}_k^u, \boldsymbol{c}_k^u\}_{\mathcal{U}\times K}$, α.
Output: $\{(\tilde{t}_i^u, \tilde{l}_i^u, \tilde{w}_i^u)\}_{\mathcal{U}\times N_u}$.
for $u \in \mathcal{U}$ **do**
  for $i \in \{1, ..., N_u\}$ **do**
    Draw z_i^u based on $p(z_i^u = k) \sim n_k^u + \alpha n_k$.
    Draw a timestamp $\tilde{t}_i^u$ from $\mathcal{N}(\cdot|m_{z_i^u}, \psi_{z_i^u}/\nu_{z_i^u})$.
    Draw a location $\tilde{l}_i^u$ from $Category(\boldsymbol{b}_{z_i^u}^u/||\boldsymbol{b}_{z_i^u}^u||_1)$.
    Draw an app $\tilde{w}_i^u$ from $Category(\boldsymbol{c}_{z_i^u}^u/||\boldsymbol{c}_{z_i^u}^u||_1)$.

Table 2. Dataset summary.

# Records	# Users	# Identified Apps	# Cellular Tower
1,548,972,010	1,731,070	3,503	10,875

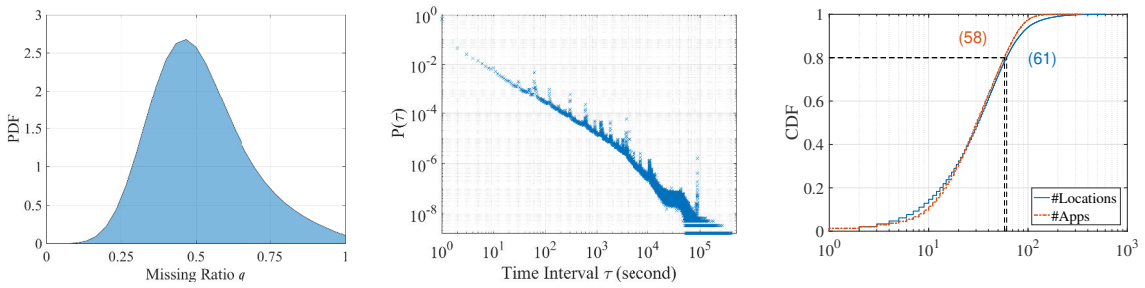
5.1 Dataset

To understand the spatio-temporal patterns of app usage, we collect a large dataset by collaborating with a major cellular network operator in China (Telecom). Our dataset was collected during one week (April 19–26) in 2016 covering the whole metropolitan area of Shanghai, one of the largest city of China. The dataset covers 2,140,327 users and their complete access logs to the cellular base stations during the data collection period. Each access record is characterized by an anonymized userID, timestamp, the cellular base station and its GPS location and the meta data of the connection. For HTTPS connection, we have the destination IP address. For HTTP connection, we have the destination IP and domain and the user agent information. Note that the userID is mapped to a mobile device (*e.g.*, smartphone, iPad). It is possible for one person to have multiple devices and we will treat them as different users, which allows us to perform more fine-grained modeling.

Identifying Apps. From the raw access log, we then infer what apps that users are using based on various indicators of the app identity. Unfortunately, due to the limited metadata from HTTPS, we can only identify the apps for HTTP traffic. In our dataset, 95% of traffic is in HTTP and we rely on the URL domain and the user-agent field from the HTTP header to identify specific apps. We adopt an existing tool called SAMPLES [57] which uses supervised learning to automatically classify the network traffic generated by different mobile apps. In order to obtain the labeled dataset, we crawled the 3,503 most popular apps across App Store (iOS apps) and Google Play (Android apps) based on their download count, and applied SAMPLES to generate conjunctive rules to match each app's network traffic. Based on the obtained rules, we identify traffic of the 3,503 apps that cover more than 85 percent records of the whole dataset. We manually verified the correctness of the matched apps and filter out users who do not have any matched apps (20% of the users). After the filtering, the final dataset contains 1,731,070 users and 1,548,972,010 access records. These app usage records cover 10,875 distinct cellular base stations in the city. Table 2 presents a summary of the dataset.

Ethics. We are very aware of the privacy implications of using ISP dataset for research and have taken active steps to protect mobile users. First, the app usage traces do not contain any personally identifiable information or any user-level metadata. The userID has been anonymized (as a bit string) by the ISP, and we never have the access to the true userID. Second, all the researchers are regulated by a strict non-disclosure agreement. The dataset is stored in a server protected by authentication mechanisms and firewalls. Our collaborator from the ISP oversees the data processing on the server. This work has received the approval from both the ISP and the authors' local institution.

Characteristics. To provide contexts for the dataset, we further analyze the metrics that are related to the data quality in Fig. 3. First, we examine the potential missing parts in a user's trace that our dataset does not cover. We define missing ratio q as the portion of the time bin when we do not have any data about the user's app usage. The probability density function (PDF) of missing ratio q is presented in Fig. 3(a). We observe that the missing ratio follows a normal-like distribution ranging from 0.1 to 1 and the distribution is slightly skewed to the right. This suggests some level of data sparsity – users do not use mobile apps all the time, which is a challenge need to be solved in our model. Fig. 3(b) examines the time intervals of two consecutive records, which follows a power law distribution. The majority of the time intervals are less than 1000 seconds, with an average 222 seconds. This indicates that our dataset is fine-grained in time dimension. Regarding the spatial and app aspect, Fig. 3(c) shows the distribution of the number of distinct locations visited by each user and the number of distinct apps used by each user. We can observe that 80% of the users are recorded in less than 61 locations,



(a) Distribution of missing ratio

(b) Distribution of time interval

(c) Distribution of distinctive locations visited and apps used

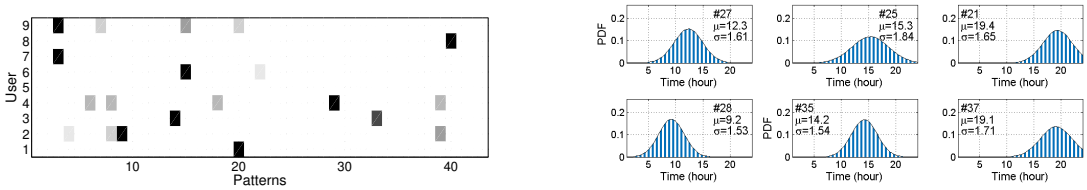
Fig. 3. Characteristics of the collected dataset.

and 80% of the users use less than 58 apps unique apps. These two curves follow very similar shapes, suggesting a potential correlation between the app usage pattern and the spatio-temporal pattern.

Data Limitations. Although the scale of the dataset is massive, in terms of individual user and duration, it has limitations. More specifically, the dataset does not cover the following four situations. 1) apps that do not make any network requests; 2) apps out of the identified 3,503 apps; 3) apps that use HTTPS for *every single Internet request*; 4) app usage behavior out of the one-week time window covered by the dataset. The first case is a natural limitation of using ISP datasets. For the second case, the identified 3,503 apps have included the most popular iOS apps and Android apps. In addition, based on our measurement, they have covered more than 85 percent records of our whole dataset, which are enough to characterize users' main app usage patterns. For the third case of apps that use HTTPS, we observe many of these apps still use HTTP for parts of the Internet requests, while they use HTTPS only to transmit sensitive information, making them identifiable. In addition, the HTTP traffic takes more than 95% of all the traffic within our data collection period. For the fourth case, though the duration of our dataset is only one week, our model mainly focus on capturing the principle components of users' app usage, *i.e.*, users' predictable daily app usage patterns. Therefore, we characterize these app usage patterns by focusing on the time of day in the temporal dimension. The dataset with duration of one week is enough to characterize users' predictable daily app usage patterns. Overall, we believe this dataset provides a meaningful representation of users' app usage.

5.2 Pattern Interpretations

We train our model based on the above introduced dataset by Algorithm 2. We set the number of iterations M as 50, and find it is enough to reach convergence for all users [37, 67]. We obtain 44 app usage patterns in total. Then, we randomly select 9 users, and show their distribution over different patterns in Fig. 4(a). Specifically, the vertical axis represents different users u, while the horizontal axis represents different patterns k. The shade of the grids describes the probability of an app usage record generated by user u belonging to pattern k, *i.e.*, π_k^u, where the deeper grey means the larger π_k^u. We can observe that each user's app usage behavior can be described by a small number of patterns. The largest number of patterns that one user have is less than 5, e.g., User 2 and 4. Results indicate that though the total number of patterns might be increased by sharing patterns across users, the number of individual's dependent patterns is still small. Thus, from the individual perspective, the number of effective variables to learn is not increased, demonstrating the effectiveness of our proposed app usage model.



(a) Users' distribution over obtained patterns

(b) Temporal parameters of obtained patterns

Pattern \ POI	Entertain.	Education	Scenery	Business	Industry	Residence	Suburb
#27	0.3483	0.0720	0.0281	0.1047	0.1112	0.2667	0.0971
#28	0.3296	0.0822	0.0353	0.1137	0.0622	0.2328	0.0521
#25	0.3962	0.0408	0.0420	0.1157	0.0989	0.2615	0.0999
#35	0.2861	0.0462	0.0351	0.0984	0.0490	0.2711	0.1438
#21	0.3300	0.0378	0.0225	0.0967	0.0640	0.3378	0.0531
#37	0.3382	0.0495	0.0275	0.0967	0.0544	0.2928	0.0687

(c) POI distribution corresponding to obtained patterns

Pattern \ APP	Games	Video	News	Social	Shopping	Lifestyle	Map	Music
#27	0.0062	0.0122	0.0222	0.2999	0.0222	0.0332	0.0927	0.0274
#28	0.0046	0.0099	0.0176	0.1936	0.0173	0.1482	0.1959	0.0868
#25	0.0161	0.0392	0.0250	0.2151	0.0106	0.0387	0.1549	0.0420
#35	0.0073	0.0096	0.0130	0.2692	0.0067	0.0471	0.0476	0.0153
#21	0.0169	0.0155	0.0274	0.2965	0.0463	0.0561	0.0674	0.0262
#37	0.0323	0.0148	0.0212	0.2667	0.0183	0.0521	0.0523	0.0332

(d) App category distribution corresponding to obtained patterns

Fig. 4. Interpretation for patterns obtained in our model.

Then, we focus on interpreting the obtained patterns in three domains – location, time and app. However, since the dimensionality of locations and apps are very high (3,503 apps and 10,875 locations in our dataset), it is hard to interpret the correlation of them directly. Thus, we reduce their dimensionality by utilizing point of interests (POIs) and app categories, which have strong correlation to location and app, and have rich semantic information to help us understand the results.

Specifically, we interpret patterns in spatial dimension by studying the point of interest (POI) distribution associated to them, and in app dimension we consider the categories of associated apps of each pattern, which will be introduced in detail in the following parts of this section. Overall, we elaborately select 6 representative patterns with most distinct distribution in terms of POI and categorie of apps associated with them, and show their temporal parameters in Fig. 4(b). We can observe that each pattern is characterized by a distinct distribution of the app usage during different time of a day.

As for spatial dimension, we study the POI distribution for each pattern. POI is a specific point location of a certain function such as restaurant or shopping mall. POI can reflect function of a region, and can be open accessed through the APIs of map service providers. Thus, we crawl 0.75 million POIs of Shanghai city, and

Table 3. The utilized POI categories and taxonomies.

ID	Function	Utilized POI
#1	Entertainment	food, hotel, gym, shopping, leisure.
#2	Education	school, campus.
#3	Scenery	scenery spot.
#4	Business	finance, office building, company, trading area.
#5	Industry	factory, industrial estate, economic development zone.
#6	Residence	residence, life services.
#7	Suburb	villages, towns.

divide them into 7 categories according to [39, 64], *i.e.*, residence, entertainment, business, industry, education, scenery and suburb, shown in Table 3. To be better compared with, the number of POI is normalized by the term frequency-inverse document frequency (TF-IDF) [15]. Denote $\boldsymbol{H}_l$ as the TF-IDF normalized POI distribution around location l. Then, the weighted average POI distribution of pattern k can be expressed as follows:

$$\boldsymbol{H}_k = \sum_{u \in \mathcal{U}} \frac{\pi_k^u}{\sum_{u \in \mathcal{U}} \pi_k^u} \sum_{l \in \mathcal{L}} \theta_k^u(l) \boldsymbol{H}_l.$$

The results are shown in Fig. 4(c), where we highlight the maximum value of each column with red color and the minimum value of each column with blue color. As we can observe, compared with other patterns, app usage patterns in the early morning (*e.g.* #27, #28) have higher POI distribution around "education" and "industry". In addition, compared with other patterns, app usage patterns in the late-night (*e.g.* #21, #37) have higher POI distribution around "residence". These results interpret where users are corresponding to the different time of the day.

In terms of the apps being used, we consider their categories. We crawl the category information of the identified apps from the app market, and divide them into 19 categories. Denote $\boldsymbol{G}_w$ as the one-hot vector indicating the category of app w. Then, the weighted average category distribution of each pattern can be expressed as follow:

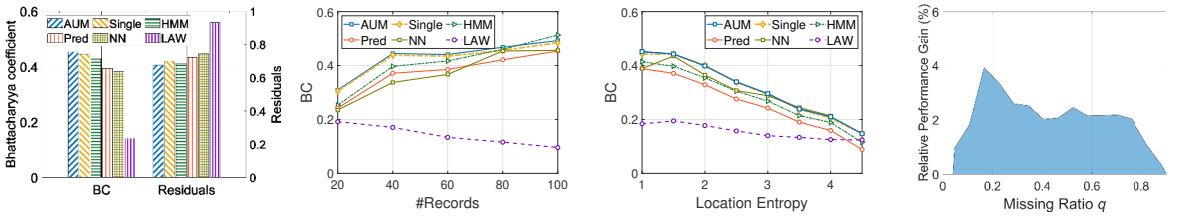
$$\boldsymbol{G}_k = \sum_{u \in \mathcal{U}} \frac{\pi_k^u}{\sum_{u \in \mathcal{U}} \pi_k^u} \sum_{w \in \mathcal{A}} \phi_k^u(w) \boldsymbol{G}_w.$$

Similarly with location, we also apply TF-IDF normalization to the results for better comparison. The results are shown in Fig. 4(d). For simplicity, we only list 8 categories with the most variance. We find that patterns in the early-morning (*e.g.* #27, #28) have higher category distribution of "maps", while patterns in the late-night (*e.g.* #21, #37) have higher app category distribution around "game". Take the pattern #21 and #37 for example. They have the second largest and the largest distribution around "game". In addition, the means of their temporal distribution are all after 7PM, indicating that people tend to use more "game" apps in the late-night. All these results interpret what users are doing corresponding to different time and location patterns. Overall, the obtained patterns well interpret and capture when, where, and what apps are used, which help us to understand users' app usage behavior substantially.

5.3 Location Prediction

Baseline Method: We compare our system with five state-of-the-art algorithms as follows: (1) **HMM** is a well-known approach for analysis sequential data [36], which assumes observations are generated by a Markov process among unobserved states. Specifically, the emission distribution, *i.e.*, the distribution of observations conditioned on a specific state is modelled to be multinomial distribution to fit our dataset. (2) **Pred** is original from the human mobility model proposed in [28], which predicts future locations based on the trip time and marginal distribution of users' personal destinations. (3) **NN** is based on the neural network classifier [43], which uses the last visited location and time as features. (4) **LAW** is original from the human mobility model proposed in [10]. It predicts future locations based on the travelling distance, which is modeled as a Lévy flight with long-tailed distribution. (5) **Single** is a simplified version of our method, which uses Dirichlet process instead of hierarchical Dirichlet process. In addition, we denote our proposed model as **AUM**.

Evaluation Metrics: In order to measure the correctness of the estimated distribution among locations $\hat{\boldsymbol{x}}_i$ compared with the true number of times users visit different locations $\boldsymbol{x}_i$. We use two well-known metrics including Bhattacharyya coefficient and residual.



(a) Overall performance (b) Performance vs. #records (c) Performance vs. location entropy (d) Relative performance gain vs. missing ratio

Fig. 5. Performance of location prediction.

Bhattacharyya coefficient has been widely used in object tracking [17, 41]. It can be expressed as:

$$BC(\hat{\boldsymbol{x}}_i, \boldsymbol{x}_i) = \sum_{l \in \mathcal{L}} \sqrt{\frac{\hat{x}_i(l)}{\|\hat{\boldsymbol{x}}_i\|_1} \cdot \frac{x_i(l)}{\|\boldsymbol{x}_i\|_1}}, \tag{12}$$

where $\|\cdot\|_1$ is the 1-norm, which can be calculated as $||\boldsymbol{x}||_1 = \sum_{l \in \mathcal{L}} |x(l)|$. Bhattacharyya coefficient is a measurement of the amount of overlap between two statistical samples, which ranges from 0 to 1. If two samples are equal, *i.e.*, $x_1(l) = x_2(l)$ for all $l \in \mathcal{L}$, BC will be 1. It will be 0 if two distributions have no overlap at all.

Another metric is the residuals [44], which can be defined as:

$$Residual(\hat{\boldsymbol{x}}_h, \boldsymbol{x}_h) = \frac{1}{2} \|\hat{\boldsymbol{x}}_h - \boldsymbol{x}_h\|_1 . \tag{13}$$

It measures the summation of difference around all dimensions between two distributions, which ranges from 0 to 1. Different from Bhattacharyya coefficient, a smaller residual indicates a better prediction result.

Parameter Settings: We implement a 5-fold leave-one-out cross validation [26] in our experiments. Specifically, we divide the whole dataset into 5 sets with equal size based on the time. We use each set (20% of all records) as the testing set based on parameters estimated from the left 4 sets as training set. Specifically, in the testing set, we assume time and used apps are given, while locations are unknown, which need to be predicted. Then, we take the average Bhattacharyya coefficient and residual of prediction results of all sets as the final performance.

For our proposed AUM, we set the number of iterations M as 50, which is enough to reach convergence for all users [37]. In addition, we set $\alpha = 1/N$ and $\epsilon = 50$, where N is the number of records in the training set. As for the temporal parameters, we set $m_0 = 12$, $\kappa_0 = 0.1$, $\nu_0 = N/40$, $\psi_0 = 2\nu_0$. As for the spatial and app parameters, we set $\beta = \gamma = 0.1$.

As for the parameter of baselines, for HMM, NN, and Single, the number of number of iterations is also set to be 50. Specifically, for Single, we change $\alpha = 1$ and $\nu_0 = 2$, and other parameters are the same as those used in AUM. In NN, the utilized neural network is composed of two hidden layers, of which the dimensions are set to be 20. Other parameters are set by following the default setting in original literatures [10, 28, 36, 43].

Result Comparison: We present the location prediction results of the different algorithms in Fig. 5. Fig. 5(a) shows the prediction performance in terms of Bhattacharyya coefficient and residual. We can observe that our method achieves highest BC and lowest residual, which proves the feasibility and superiority of our proposed model. Specifically, the AUM outperforms the Single by 2.3%, which demonstrates the advantage of sharing patterns across users. AUM performs better than HMM, Pred, NN, and LAW by over 2.6%, 5.9%, 7.2%, and 31.2% in terms of BC, respectively. In addition, HMM and Pred have the best performance within baseline algorithms, which both predict users' movement based on personalized probabilistic models. Different with them, LAW is an aggregated probabilistic model without considering any personal information. On the other hand, the reason

of the under-performance of NN is the sparsity of app usage records per user. Thus, compared with HMM or Pred, the larger number of parameters in NN cannot be well estimated. In addition, we show the prediction performance of user groups with different number of records and location entropy in Fig. 5(b) and (c), respectively. Specifically, for a user u, its location entropy $E_L(u)$ can be calculated by $E_L(u) = -\sum_{l \in \mathcal{L}} P_l(u) \log P_l(u)$, where $P_l(u)$ is the probability of visiting location l by u. It describes the regularity of traces in spatial dimension [16]. As we can observe, compared with HMM, the performance gain of our proposed algorithm is larger for users with less app usage records and smaller location entropy. Then, in order to show the performance gain obtained from sharing patterns across users, the relative performance gain of AUM algorithm compared with Single is plotted as the function of missing ratio q of each user in Fig. 5(d). As we can observe, the maximum relative performance gain is over 4.2%, which is reached for users with missing ratio around 0.2. In addition, though it is hard to improve the performance for users with missing ratio close to 1, the performance gain of our proposed is still high (over 2.2%) for users with missing ratio q ranging from 0.7 to 0.8, indicating the effectiveness of our proposed algorithm on users with sparse data.

Although the duration of our dataset is only one week, the number of users in our dataset is massive, which cover users with diversified app usage patterns and mobility patterns. Thus, there are enough samples to evaluate the performance of our proposed method and baseline algorithms. In addition, we evaluate our proposed model compared with baselines in terms of multiple metrics and different user groups. Overall, the large-scale dataset guarantees the credibility and validity of the performance evaluation, and our proposed AUM performs better than other algorithms under the majority of the situations, demonstrating the effectiveness of our model.

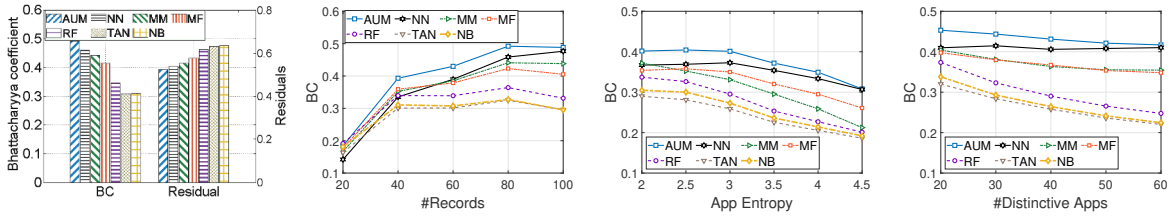
5.4 App Prediction

Baseline Method: As for app prediction, we compare our system with following methods: (1) **MM** is a Markov model [34], which is widely used to predict human behavior. It regards all the used apps as states and builds a transition matrix to capture the first order transition probabilities between them. (2) **NB** is a Naive-Bayes prediction model [22], that uses location, time-of-day and last-used-app as features. (3) **RF** is based on classical classification algorithm [70], which uses the same set of features as Bayesian and adopts the random forest classifier. (4) **NN** is based on the neural network classifier [43], which uses the last used app and time as features. (5) **TAN** is based on existing work [3] that uses spatio-temporal context and embedding vector of last-used-app based on word2vec [38] as features to implement a tree augmented naive Bayesian network. (6) **MF** is a model that always predicts the app usage to be the mean app usage of the user's historical records [13].

Evaluation Metrics: We use the same metrics as in location prediction. Specifically, for two app usage vectors $\hat{\boldsymbol{a}}_i$ and $\boldsymbol{a}_i$, their Bhattacharyya coefficient can be expressed by $BC(\hat{\boldsymbol{a}}_i, \boldsymbol{a}_i) = \sum_{w \in \mathcal{A}} \sqrt{\frac{\hat{a}_i(w)}{\|\hat{\boldsymbol{a}}_i\|_1} \cdot \frac{a_i(w)}{\|\boldsymbol{a}_i\|_1}}$. In addition, their residual can be calculated by $Residual(\hat{\boldsymbol{a}}_i, \boldsymbol{a}_i) = \frac{1}{2} \|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i\|_1$.

Parameter Settings: Similarly with location prediction, we implement a 5-fold leave-one-out cross validation [26] in app prediction. Specifically, in the testing set, we assume time and visited locations are given, while used apps are unknown. In addition, parameters of AUM is similar with those used in location prediction. Other parameters are set by following the default setting in original literatures [3, 13, 22, 34, 43, 70].

Result Comparison: We follow the same experiment setting as in location prediction, and show the results in Fig. 6. Similar to the location prediction, Fig. 6(a) shows the overall performance of app prediction in terms of BC and residual. Results show that our method achieves the best performance. The BC of our method is 3.2% higher than NN, 5.0% higher than MM, 7.8% higher than MF, 14.6% higher than RF, and over 18.2% higher than TAN and NB. In addition, we can observe that TAN and NB have similar performance, indicating that the structure of Bayesian network is not the bottleneck of performance. Instead, the more important challenge is



(a) Overall performance (b) Performance vs. #records (c) Performance vs. app entropy (d) Performance vs. #distinctive apps

Fig. 6. Performance of app prediction.

the data sparsity issue. On the other hand, NN has better performance in app prediction compared with location prediction, indicating that the influence of the last used app to future app usage is more significant than that of the last visited location to future movement. Then, we show the app prediction performance as the function of the number of records in Fig. 6(b). As we can observe, our proposed AUM outperforms baseline algorithms in most cases, indicating the advantage of AUM.

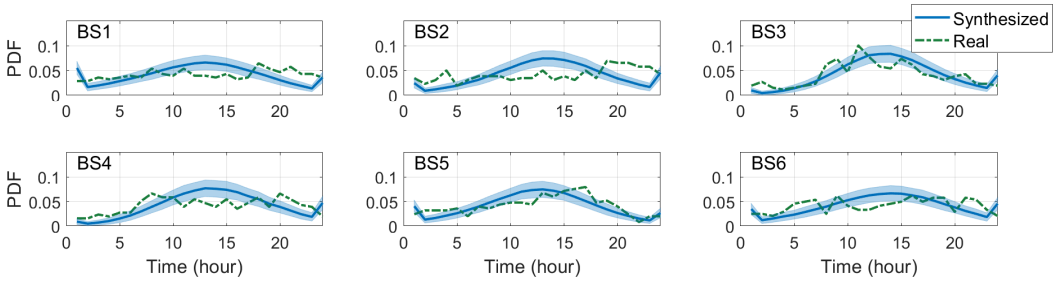
We also show the prediction performance as the function of the app usage entropy and number of distinctive apps used by each user in Fig. 6(c) and (d), respectively. Specifically, app usage entropy $E_A(u)$ is defined as $E_A(u) = -\sum_{w \in \mathcal{A}} P_w(u) \log P_w(u)$, where $P_w(u)$ is the probability of using app w by u. It describes the regularity of using apps [16]. As we can observe, users with larger app usage entropy and more used apps are predicted with worse performance, and app usage entropy exhibits larger influence than the number of distinctive apps. Overall, our method outperforms baseline algorithms in most situations with the maximum performance gap over 10%, demonstrating the superiority of the proposed method based on our model.

5.5 App Usage Trace Synthesizing

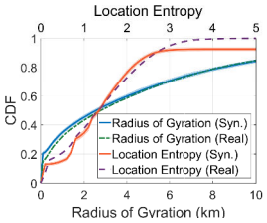
In order to utilize app usage traces in applications such as network optimization and context-aware recommendations [25, 55] without leaking users' privacy, we seek to synthesize plausible app usage traces. Specifically, compared with approaches of adding random noise to user data [1, 2, 18], the synthesized trace is able to preserve the utility of the dataset. We characterize the utility of the dataset by using the similarity of statistical characteristics of synthesized traces to real traces, including time of visits to different locations, location entropy, radius of gyration, probability of using different apps, and distribution of degree of users' contact graph. At the same time, the synthesized trace should protect users' privacy. We characterize the privacy-preserving ability of our model by the spatio-temporal distance between real traces and synthesized traces. If they are far away from each other enough, adversaries are hard to extract users' true location, *i.e.*, users' privacy is preserved. Thus, our model is very useful to these applications based on app usage traces [25, 55], since based on our model they can be implemented without leaking users' privacy.

Parameter Settings: In this application, differently with location and app prediction, we do not split training and testing set. Instead, we use all app usage records as input. Parameters are also similar with those used in location and app prediction. Then, based on Algorithm 3, we synthesize plausible app usage traces of 10,000 users with the time span of one week.

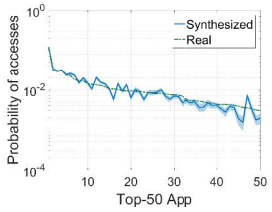
Evaluation Results: We first evaluate the utility of the dataset in terms of statistical characteristics. The synthesized traces should statistically resemble real traces. In order to measure the statistical characteristics of synthesized traces in terms of all dimensions, we elaborately select four metrics, including time of visits to different locations, location entropy, radius of gyration, and probability of using different apps. We show their distribution in Fig. 7(a) to (d), respectively. From Fig. 7(a), we show the temporal distribution of visits to 6



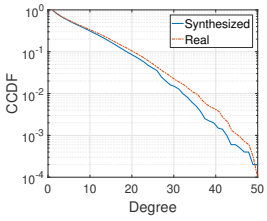
(a) Temporal distribution of visits to 6 locations for example



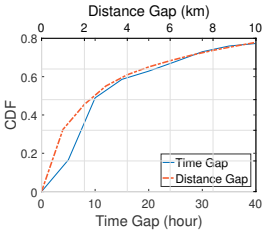
(b) CDF of radius of gyration and location entropy



(c) Probability of accesses of top-50 apps



(d) CCDF of degree of users' contact graph



(e) CDF of time and distance gap

Fig. 7. Synthesized app usage traces based on our model.

locations (cellular base stations) for example, which are referred to as BS1 to BS6, respectively. Further, we use Bhattacharyya coefficient to characterize the performance of synthesized traces compared with real traces in terms of temporal distribution of visits to locations, which can be calculated as following:

$$BC(\hat{\boldsymbol{v}}_l, \boldsymbol{v}_l) = \sum_{t=1}^{24} \sqrt{\frac{\hat{v}_l(t)}{\|\hat{\boldsymbol{v}}_l\|_1} \cdot \frac{v_l(t)}{\|\boldsymbol{v}_l\|_1}}, \tag{14}$$

where $\boldsymbol{v}_l(t)$ and $\hat{\boldsymbol{v}}_l(t)$ are the number of visits to location l at time t of real traces and synthesized traces, respectively. In addition, visits of one week is folded into one day by averaging. Results show that the obtained average Bhattacharyya coefficient of the 6 locations shown in Fig. 7(a) is larger than 0.973, indicating the effectiveness of our proposed model. However, as we can observe from Fig. 7(a), peak of temporal distribution of some locations shows an obvious difference between real traces and synthesized traces. For example, temporal distribution of visits to BS4 of real traces has two peaks at around 8AM and 8PM respectively, while for synthesized traces the temporal distribution has only one peak, indicating some level of limitations of synthesized traces by our proposed AUM in terms of temporal distribution of visits to locations.

In Fig. 7(b), we show the cumulative distribution function (CDF) of location entropy and radius of gyration. Radius of gyration is defined as the mean square root of the distance of each point in the trace to its center of mass [21]. It reflects the range of a user' activity area. We can observe that the CDFs of real traces and synthesized traces are similar. In Fig. 7(c), we show the probability of using top-50 apps of all users. We can observe that probability of using top-50 apps of synthesized traces waves around that of real traces. In order to investigate the performance of synthesized traces in terms of protecting hidden structures of users' app usage trace, we further construct users' contact graph, where the edge represents two users once visited the same location within the same time period. Then, the complementary cumulative distribution functions (CCDF) of users' degree in the contact graph based on real traces and synthesized traces are shown in Fig. 7(d). We can observe that the distribution of degree of synthesized traces approximates well to real traces, indicating the strong ability of our proposed model in protect users' encountering relationship, which is useful to applications such as location

recommendation [58, 65, 71] and friend recommendation [35, 63] in location-based social networks. Overall, results show that app usage traces synthesized by our model resemble real traces statistically.

Then, we evaluate the privacy-preserving performance of synthesized traces. We clarify this issue in terms of two metrics, *i.e.*, time gap and distance gap. The synthesized traces should have large time gap and distance gap with real traces to hide users' true locations. Time gap is defined as the minimum time interval between each real record and its closest synthesized record at the same location, while the distance gap is defined as the distance of records in the same time bins between real traces and synthesized traces. Their CDFs are shown in Fig. 7(e). The average time gap and distance gap for all records are 15.8 hour and 6.22 km, respectively. Such large time and distance gap indicate the good performance of the synthesized traces in hiding users' true location and preserving user privacy.

Summary: The evaluation results show that our proposed app usage model has strong ability to interpret users' behavior. Specifically, all parameters do have physical meaning, and from these parameters we can tell when, where, and what apps are used. In addition, it outperforms the state-of-the-art algorithms in both location and app prediction by over 11.7% and 11.1% on average, respectively. Compared with existing approaches, our proposed method models users' location and app usage simultaneously, and parameters are shared between different users to overcome the data sparsity challenge, indicating the flexibility and robustness of our proposed method. In addition, due to the Bayesian framework adopted in our proposed model, it is able to synthesize app usage traces with high utility and strong privacy preserving ability, further demonstrating its usefulness. Overall, observations show that by correlatedly modeling location and app usage, the performance of predicting both app usage and mobility is improved, and app usage traces synthesized by our model resemble real traces in terms of both the location-related statistical metric and the app-related statistical metric. All these results demonstrate the correlation between users' visited locations and used apps extracted by our model.

6 RELATED WORK

Mobile App Usage. Researchers have studied how users use smartphones and mobile apps [7, 19, 22, 54, 69, 70], with a focus on user interactions, application usage, network traffic, and energy drain, etc. Falaki et al. [19] show immense diversity in smartphone activities among users. Blaszkiewicz et al. [7] further reveal that smartphone users can be identified through the sets of apps they use. Tu et al. [49] find that 88% of users can be uniquely re-identified by 4 random apps used by them. Other researchers propose to cluster smartphone users according to their app usage behavior in order to provide customized services and recommendations [69]. In particular, users' mobility patterns influence the way an app is used [70]. Contextual features are shown to have a strong influence on personalized app predictions, such as predicting sequentially used apps, and the location and time of app usage [22]. Yu et al. [62] show that app usage of regions can be better predicted by using the Point of Interest (POI) information of regions. Cao et al. [11] compare the difference between revisitation patterns of POIs, websites, and smartphone apps. Xia et al. [52] utilize aggregated app usage in different regions to reveal urban dynamics and infer urban functions. A contextual collaborative forecasting model is proposed in [51] to consider similarity between users and apps, and correlation between uses' movements and physical environments. A multi-faceted approach of app usage predictions is developed in [54]. Most studies focus on small-scale datasets, posing a key challenges to understand the app usage behavior of a large user population.

Spatio-Temporal Recommendation. Spatio-temporal context information has been playing an important role in location-based services. A number of spatio-temporal recommendation systems have been proposed in existing studies [58, 65, 71]. Ye et al. [58] develop a location recommendation based the social and geographical characteristics of users and locations. Tu et al. [48] find that it is feasible to make personalized location recommendation by learning user interest and location features from app usage data. Zheng et al. [71] recommend

interesting locations and possible activities to users based on their GPS and comments through a collaborative filtering method. Yuan et al. [65] develop a collaborative recommendation model for POI that is able to incorporate temporal information. In addition, a number of context-aware app recommendation systems are developed by using additional information of location, time, and activity [25, 33, 71, 72]. On the other hand, personalized recommendation is usually achieved by using more user profile information [9, 29, 32]. App similarity that is important for recommendation is usually calculated by graph [4] or kernel function [12], which is utilized in ranking [56] and popularity [73] based recommendation. In addition, problems of data sparsity [45] and cold-start [30] have been studied by using specific apps' features of similarity. Moreover, privacy and security awareness [20, 31, 60, 74] have also been considered. These approaches are mainly designed to recommend new items by reconstructing the missing values in user's spatio-temporal records, which is a different problem with what we investigate in this paper, *i.e.*, predict the future location and app usage based on historical records.

Human Mobility Modeling. Various statistical models have been proposed to characterize people's mobility patterns. For example, Lu et al. [34] utilize Markov Chain model to simulate users' transition patterns. As a variant, hidden Markov model (HMM) is also widely used in human mobility modelling. Mathew et al. [36] assume each latent state has a multinomial distribution over the locations. Zhang [68] et al. improve HMM by integrating both user grouping and mobility modelling in the same model, in order to enhance the modelling ability and improve prediction accuracy. In addition, Dirichlet process is also widely adopted in modelling users' mobility. Jeong et al. [24] use Dirichlet process mixture model to cluster users based on their transition kernels. McInerney et al. [37] propose a hierarchical Dirichlet model, LocHDP, which shares temporal parameters within users and keep spatial parameters unique to each user. More recently, researchers consider context-aware mobility modelling, *i.e.*, considering the information from external channels like social media content and social network graphs. For example, Zhang et al. [67] leverage the geo-tagged tweet to build a local event detection system, and Wang et al. [50] combine human mobility modeling with the social network analysis to improve the prediction accuracy. Different with them, we seek to build a new model to characterize and predict the spatio-temporal patterns of *app usage* correlated with mobility.

Trace Synthesizing. Synthesizing location traces has been studied in a number of works [6, 23, 27, 42, 59]. Bindschaedler et al. [6] propose a privacy-preserving generative model to synthesize plausible location traces which protect the semantic features of each trajectory and hide the user' location. Isaacman et al. [23] propose a statistical probability model to synthesize spatio-temporal data in the form of call detail records (CDRs), which takes as input certain spatial and temporal probability distributions (e.g., distribution of commute distances, probability of a call at each time bin) drawn from empirical data. Krumm et al. [27] propose a probability model to synthesize driving trips, which examine a variety of features including road types and through traffic. Ouyang et al. [42] use generative adversarial networks (GAN) to synthesize human trajectories, where each trajectory is transform into a fixed-length sequence of stays in the geographic grid with their start time and durations. Yin et al. [59] develop a input-output hidden Markov model (IO-HMM) to synthesize travel plans of users, which incorporates context information (e.g., the time of day) and define unobserved activity types as latent variables. These studies mainly focus on protecting statistical characteristics [23, 42], or semantic features [6] of synthesized traces. We further consider the performance of protecting the encountering relationship of users in our work. In addition, the synthesized traces based on our model should resemble real traces in terms of both location-related statistical metrics and app-related statistical metrics.

7 CONCLUSIONS

In this paper, we model the spatio-temporal app usage patterns for a large user population. Specifically, by building a Bayesian mixture model to capture when, where and what apps are used. Extensive evaluations show our model achieves a high accuracy in predicting future user locations and app usage. In addition, it shows a great

potential to synthesize app usage traces to protect user privacy while allowing meaningful data mining. We will release parts of the dataset as well as our code. We believe this work paves the way toward understanding spatio-temporal app usage patterns over a large user population. Future work will look into incorporating location contexts and the user interests to build a more comprehensive model for mobile app usage.

ACKNOWLEDGMENTS

This work was supported in part by The National Key Research and Development Program of China under grant 2017YFE0112300, the National Nature Science Foundation of China under 61861136003, 61621091 and 61673237, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

REFERENCES

[1] Gergely Acs and Claude Castelluccia. 2014. A case study: Privacy preserving release of spatio-temporal density in paris. In *Proc. CCS*.
[2] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *Proc. ACM CCS*.
[3] Ricardo Baeza-Yates, Di Jiang, Fabrizio Silvestri, and Beverly Harrison. 2015. Predicting the next app that you are going to use. In *Proc. WSDM*.
[4] Upasna Bhandari, Kazunari Sugiyama, Anindya Datta, and Rajni Jindal. 2013. Serendipitous recommendation for mobile apps using item-item similarity graph. In *Asia Information Retrieval Symposium*. 440–451.
[5] Igor Bilogrevic, Kévin Huguenin, Stefan Mihaila, Reza Shokri, and Jean-Pierre Hubaux. 2015. Predicting users' motivations behind location check-ins and utility implications of privacy protection mechanisms. In *Proc. NDSS*.
[6] Vincent Bindschaedler and Reza Shokri. 2016. Synthesizing Plausible Privacy-Preserving Location Traces. In *Proc. IEEE SP*.
[7] Konrad Blaszkiewicz, Konrad Blaszkiewicz, Konrad Blaszkiewicz, and Alexander Markowetz. 2016. Differentiating smartphone users by app usage. In *Proc. UbiComp*.
[8] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
[9] Matthias Böhmer, Lyubomir Ganev, and Antonio Krüger. 2013. Appfunnel: A framework for usage-centric evaluation of recommender systems that suggest mobile applications. In *Proc. IUI*.
[10] D Brockmann, L Hufnagel, and T Geisel. 2006. The scaling laws of human travel. (2006).
[11] Hancheng Cao, Zhilong Chen, Fengli Xu, Yong Li, and Vassilis Kostakos. 2018. Revisitation in Urban Space vs. Online: A Comparison across POIs, Websites, and Smartphone Apps. *Proc. ACM IMWUT*.
[12] Ning Chen, Steven CH Hoi, Shaohua Li, and Xiaokui Xiao. 2015. SimApp: A framework for detecting similar mobile applications by online kernel learning. In *Proc. WSDM*.
[13] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proc. KDD*.
[14] Yohan Chon, Elmurod Talipov, Hyojeong Shin, and Hojung Cha. 2011. Mobility prediction-based smartphone energy optimization for everyday location monitoring. In *Proc. SenSys*.
[15] Gobinda G Chowdhury. 2010. *Introduction to modern information retrieval*. Facet publishing.
[16] Justin Cranshaw, Eran Toch, Jason Hong, Aniket Kittur, and Norman Sadeh. 2010. Bridging the gap between physical location and online social networks. In *Proc. UbiComp*.
[17] Abdelhamid Djouadi, Oe. Snorrason, and FD Garber. 1990. The quality of training sample estimates of the bhattacharyya coefficient. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 1 (1990), 92–97.
[18] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proc. CCS*.
[19] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. 2010. Diversity in smartphone usage. In *Proc. MobiSys*.
[20] Chen Gao, Chao Huang, Yue Yu, Huandong Wang, Yong Li, and Depeng Jin. 2019. Privacy-preserving Cross-domain Location Recommendation. *Proc. ACM IMWUT*.
[21] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *Nature* 453, 7196 (2008), 779–782.
[22] Ke Huang, Chunhui Zhang, Xiaoxiao Ma, and Guanling Chen. 2012. Predicting mobile application usage using contextual information. In *Proc. UbiComp*.

[23] Sibren Isaacman, Richard Becker, Margaret Martonosi, James Rowland, Alexander Varshavsky, and Walter Willinger. 2012. Human mobility modeling at metropolitan scales. In *Proc. MobiSys*.
[24] Jaeseong Jeong, Mathieu Leconte, and Alexandre Proutiere. 2016. Cluster-aided mobility predictions. In *Proc. INFOCOM*.
[25] Alexandros Karatzoglou, Linas Baltrunas, Karen Church, and Matthias Böhmer. 2012. Climbing the app wall: enabling mobile app discovery through context-aware recommendations. In *Proc. CIKM*.
[26] Ron Kohavi et al. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. IJCAI*.
[27] John Krumm. 2009. Realistic Driving Trips For Location Privacy. In *Proc. Pervasive*.
[28] John Krumm and Eric Horvitz. 2006. Predestination: Inferring destinations from partial trajectories. In *Proc. UbiComp*.
[29] Chen Lin, Runquan Xie, Xinjun Guan, Lei Li, and Tao Li. 2014. Personalized news recommendation via implicit social experts. *Information Sciences* 254 (2014), 1–18.
[30] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. 2013. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *Proc. SIGIR*.
[31] Bin Liu, Deguang Kong, Lei Cen, Neil Zhenqiang Gong, Hongxia Jin, and Hui Xiong. 2015. Personalized mobile app recommendation: Reconciling app functionality and user privacy preference. In *Proc. WSDM*.
[32] Duen-Ren Liu, Pei-Yun Tsai, and Po-Huan Chiu. 2011. Personalized recommendation of popular blog articles for mobile applications. *Information Sciences* 181, 9 (2011), 1552–1572.
[33] Qi Liu, Haiping Ma, Enhong Chen, and Hui Xiong. 2013. A survey of context-aware mobile recommendations. *International Journal of Information Technology & Decision Making* 12, 01 (2013), 139–172.
[34] Xin Lu, Erik Wetter, Nita Bharti, Andrew J Tatem, and Linus Bengtsson. 2013. Approaching the limit of predictability in human mobility. *Scientific reports* 3 (2013).
[35] Yao Lu, Zhi Qiao, Chuan Zhou, Yue Hu, and Li Guo. 2016. Location-aware friend recommendation in event-based social networks: A bayesian latent factor approach. In *Proc. CIKM*.
[36] Wesley Mathew, Ruben Raposo, and Bruno Martins. 2012. Predicting future locations with hidden Markov models. In *Proc. UbiComp*.
[37] James McInerney, Jiangchuan Zheng, Alex Rogers, and Nicholas R Jennings. 2013. Modelling heterogeneous location habits in human populations for location prediction under data sparsity. In *Proc. UbiComp*.
[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*.
[39] David Mimno and Andrew McCallum. 2008. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *Proc. UAI*.
[40] Kevin P Murphy. 2007. Conjugate Bayesian analysis of the Gaussian distribution. *Technical report* (2007).
[41] Hitoshi Niigaki, Jun Shimamura, and Masashi Morimoto. 2012. Circular object detection based on separability and uniformity of feature distributions using Bhattacharyya coefficient. In *Proc. ICPR*.
[42] Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. 2018. A Non-Parametric Generative Model for Human Trajectories. In *Proc. IJCAI*.
[43] Brian D Ripley. 2007. *Pattern recognition and neural networks*. Cambridge university press.
[44] Jonathan Richard Shewchuk et al. 1994. An introduction to the conjugate gradient method without the agonizing pain.
[45] Kent Shi and Kamal Ali. 2012. Getjar mobile application recommendations with very sparse datasets. In *Proc. KDD*.
[46] Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2005. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Proc. NIPS*.
[47] Ionut Trestian, Supranamaya Ranjan, Aleksandar Kuzmanovic, and Antonio Nucci. 2009. Measuring serendipity: connecting people, locations and interests in a mobile 3G network. *Proc. SIGCOMM*.
[48] Zhen Tu, Yali Fan, Yong Li, Xiang Cheng, Li Su, and Depeng Jin. 2019. From Fingerprint to Footprint: Cold-start Location Recommendation by Learning User Interest from App Data. *Proc. ACM IMWUT*.
[49] Zhen Tu, Runtong Li, Yong Li, Gang Wang, Di Wu, Pan Hui, Li Su, and Depeng Jin. 2018. Your Apps Give You Away: Distinguishing Mobile Users by Their App Usage Fingerprints. *Proc. UbiComp*.
[50] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert Laszlo Barabasi. 2011. Human mobility, social ties, and link prediction. In *Proc. KDD*.
[51] Yingzi Wang, Nicholas Jing Yuan, Yu Sun, Fuzheng Zhang, Xing Xie, Qi Liu, and Enhong Chen. 2016. A contextual collaborative approach for app usage forecasting. In *Proc. UbiComp*.
[52] Tong Xia and Yong Li. 2019. Revealing Urban Dynamics by Learning Online and Offline Behaviours Together. *Proc. ACM IMWUT*.
[53] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Mao, Jeffrey Pang, and Shobha Venkataraman. 2011. Identifying diverse usage behaviors of smartphone apps. In *Proc. IMC*.
[54] Ye Xu, Mu Lin, Hong Lu, Giuseppe Cardone, Nicholas Lane, Zhenyu Chen, Andrew Campbell, and Tanzeem Choudhury. 2013. Preference, context and communities:a multi-faceted approach to predicting smartphone app usage patterns. In *Proc. ISWC*.

[55] Tingxin Yan, David Chu, Deepak Ganesan, Aman Kansal, and Jie Liu. 2012. Fast app launching for mobile devices using predictive user context. In *Proc. MobiSys*.
[56] Dragomir Yankov, Pavel Berkhin, and Rajen Subba. 2013. Interoperability ranking for mobile applications. In *Proc. SIGIR*.
[57] Hongyi Yao, Gyan Ranjan, Alok Tongaonkar, Yong Liao, and Zhuoqing Morley Mao. 2015. SAMPLES: Self Adaptive Mining of Persistent LExical Snippets for Classifying Mobile Application Traffic. In *Proc. MobiCom*.
[58] Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location recommendation for location-based social networks. In *Proc. SIGSPATIAL*.
[59] Mogeng Yin, Madeleine Sheehan, Sidney Feygin, Jean-François Paiement, and Alexei Pozdnoukhov. 2018. A generative model of urban activities from cellular data. *IEEE Transactions on Intelligent Transportation Systems* 19, 6 (2018), 1682–1696.
[60] Peifeng Yin, Ping Luo, Wang-Chien Lee, and Min Wang. 2013. App recommendation: a contest between satisfaction and temptation. In *Proc. WSDM*.
[61] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wen-Ning Kuo, and Vincent S Tseng. 2012. Urban point-of-interest recommendation by mining user check-in behaviors. In *Proc. KDD*.
[62] Donghan Yu, Yong Li, Fengli Xu, Pengyu Zhang, and Vassilis Kostakos. 2018. Smartphone App Usage Prediction Using Points of Interest. *Proc. ACM IMWUT*.
[63] Fei Yu, Nan Che, Zhijun Li, Kai Li, and Shouxu Jiang. 2017. Friend recommendation considering preference coverage in location-based social networks. In *Proc. PAKDD*.
[64] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *Proc. KDD*.
[65] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proc. SIGIR*.
[66] Quan Yuan, Wei Zhang, Chao Zhang, Xinhe Geng, Gao Cong, and Jiawei Han. 2017. PRED: periodic region detection for mobility modeling of social media users. In *Proc. WSDM*.
[67] Chao Zhang, Liyuan Liu, Dongming Lei, Quan Yuan, Honglei Zhuang, Tim Hanratty, and Jiawei Han. 2017. TrioVecEvent: Embedding-Based Online Local Event Detection in Geo-Tagged Tweet Streams. In *Proc. KDD*.
[68] Chao Zhang, Keyang Zhang, Quan Yuan, Luming Zhang, Tim Hanratty, and Jiawei Han. 2016. Gmove: Group-level mobility modeling using geo-tagged social media. In *Proc. KDD*.
[69] Sha Zhao, Julian Ramos, Jianrong Tao, Ziwen Jiang, Shijian Li, Zhaohui Wu, Gang Pan, and Anind K. Dey. 2016. Discovering different kinds of smartphone users through their application usage behaviors. In *Proc. UbiComp*.
[70] Xiaoxing Zhao, Yuanyuan Qiao, Zhongwei Si, Jie Yang, and Anders Lindgren. 2016. Prediction of user app usage behavior from geo-spatial data. In *Proc. GeoRich*.
[71] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with GPS history data. In *Proc. WWW*.
[72] Hengshu Zhu, Enhong Chen, Kuifei Yu, Huanhuan Cao, Hui Xiong, and Jilei Tian. 2012. Mining personal context-aware preferences for mobile users. In *Proc. ICDM*.
[73] Hengshu Zhu, Chuanren Liu, Yong Ge, Hui Xiong, and Enhong Chen. 2015. Popularity modeling for mobile Apps: A sequential approach. *IEEE transactions on cybernetics* 45, 7 (2015), 1303–1314.
[74] Hengshu Zhu, Hui Xiong, Yong Ge, and Enhong Chen. 2014. Mobile app recommendations with security and privacy awareness. In *Proc. KDD*.

Received August 2018; revised November 2018; accepted January 2019