Contextual Conditional Models for Smartphone-based Human Mobility Prediction

Trinh Minh Tri Do
Idiap Research Institute, Switzerland
do@idiap.ch

Daniel Gatica-Perez
Idiap and EPFL, Switzerland
gatica@idiap.ch

ABSTRACT
Human behavior is often complex and context-dependent. This paper presents a general technique to exploit this "multidimensional" contextual variable for human mobility prediction. We use an ensemble method, in which we extract different mobility patterns with multiple models and then combine these models under a probabilistic framework. The key idea lies in the assumption that human mobility can be explained by several mobility patterns that depend on a subset of the contextual variables and these can be learned by a simple model. We showed how this idea can be applied to two specific online prediction tasks: *what is the next place a user will visit?* and *how long will he stay in the current place?*. Using smartphone data collected from 153 users during 17 months, we show the potential of our method in predicting human mobility in real life.

Author Keywords
prediction,user mobility,smartphone,mobile context.

INTRODUCTION

Smartphones have become attractive option for sensing human and social behavior [6, 20, 2]. As phones are usually kept in relatively close proximity [18] and contain many useful sensors that can record contextual and user activity cues (location, application usage and calling behavior), they can be effectively used to capture and mine user behavior in everyday life [6, 5]. The availability of multiple data sources from smartphones also enables the possibility of predicting future user behavior, based on sensed past user activities.

In this paper, we study the prediction of user mobility from smartphone data under a general predictive framework. We propose new algorithms for exploiting multiple contextual variables using a probabilistic approach, that infers the conditional dependencies between contextual input variables and output variables that correspond to predictions. For example, our approach estimates the conditional probabilities over the set of destinations that a user can go to, given his current context such as location and time. In particular, we address two fundamental tasks for mobility prediction: what is the next place a user will visit?; how long will he stay in the current place?.

We formulate the mobility prediction problem as one of learning the conditional distribution of the output variables given multiple contextual input variables. Instead of defining a complex model that exploits all useful context sources, we develop a principled way to combine multiple mobility-related patterns. For a specific task (next place prediction, duration prediction), the framework focuses on finding relevant contextual features, and on building single models that capture specific mobility patterns. The set of single models is then combined together under a probabilistic approach. Another contribution of this work relates to the integration of general and personalized prediction models, where we model the fact that, while human mobility is highly personal, user-independent patterns could be extracted from the data and used as prior knowledge about mobility when making predictions. We show how this approach significantly improves the performance of a personalized predictive model. To our knowledge, this is a novel approach to exploit population-based patterns for predicting mobility of specific persons.

We validate our approach using a large-scale data set involving 153 users in a 17-months data collection campaign. This longitudinal data allows us to study the predictability of user mobility from smartphone data collected in real-life conditions, in which users' locations are not always available. For the two prediction tasks, the proposed ensemble approach is showed to be effective in exploiting multiple dependencies between user context and mobility, leading to improved performance over single basic predictors. The experimental results also showed that the predictability of user mobility depends on many factors. On one hand, there are situations in which the human mobility cannot be predicted reliably due to low number of observations of the same context or the complexity of human behavior (e.g., the large number of options for lunch leads to low predictability of the restaurant the user goes). On the other hand, repetitive mobility pattern can be learned and predicted reliably as long as the smartphone collected enough observations.

The paper is organized as follows. The next section is dedicated to related work, followed by a description of the raw data and place extraction process. Then, we present the general formulation of prediction tasks and detail the models for the two specific prediction tasks. Finally we provide experimental results and present some concluding remarks.

UbiComp '12, Sep 5-Sep 8, 2012, Pittsburgh, USA.

RELATED WORK

Predicting human mobility has been an increasingly relevant topic in pervasive computing thanks to the improved ability to track people location [15, 1]. While our work considers location data with state-of-the-art accuracy extracted from smartphones, it is also possible to get location data of lower resolution using other sensing methods. Song et al. [22] studied the predictability of human mobility from very coarse location data of GSM tower IDs. WiFi access points are another popular data source for localizing and predicting user mobility [23]. Recently, Vu et al. proposed a framework to extract places from WiFi records, and predict human movement using joint WiFi and Bluetooth traces [24].

The prediction of human mobility has also been considered in various settings, such as inferring a destination based on partial paths [13], predicting the location and the stay duration at a given time in the future [21], or estimating the probability that a person is present at a specific place at a given time [12]. All these example can be viewed as learning an input-output function from past, available observations, where the input is the user context (partial path, current location, time), and the output is the variable we want to predict (destination, arrival time).

Previous work on human mobility prediction often considers multiple contextual data sources. For the combination of location and time, a simple, yet effective method is to build a model for each location separately [12]. More sophisticated methods typically require a carefully designed model, such as the spatio-temporal decision tree for mining trajectory pattern proposed in [16]. Recently, Cho et al. [4] proposed a model that combine spatial, temporal, and social relation information for predicting human movement. In these works, the combination method is task-specific and hard to be extended to a large number of contextual variables. As an alternative, ensemble methods represent a promising direction, as recently addressed in [19]. This work is, however, limited to choosing a single best model rather than combining the (typically available) prior information from multiple models. The existing literature on mobility prediction points out to a need for improved, principled ways to exploit and integrate the potentially large number of contextual variables available on mobile devices like smartphones.

RAW MOBILE DATA AND PLACE EXTRACTION

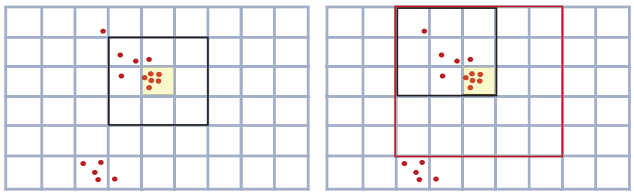
In this paper, we use data collected with Nokia N95 smartphones for a period of 17 months. The data collection campaign started from October 2009 in a European country. There are 153 participants consisting mainly of students, professionals and few others (retired people, housewives). Each participant carried the smartphone as their main and unique phone, and thus recorded data in real-world condition.

The data collection framework was based on a server-client architecture in which a client software was installed in the smartphone to record data and upload it automatically to a server via WiFi network. The client was programmed to record various data types (e.g., GPS, WiFi APs, calling logs, etc.) with dynamic sampling rates depending on the inferred state of the user (e.g., indoor-staying, outdoor-moving, etc) in order to preserve battery life. This enables the phone to record data continuously with the only restriction of charging the phone once a day. The location data comes from two sources: GPS sensor and WiFi data. The estimation location WiFi APs location was integrated in the sensing software, which looked for co-presence of APs and GPS data.

The raw location data was first transformed into a symbolic space which captures most of the mobility information and excludes actual geographic coordinates. This was done by first detecting visited *places* and then mapping the sequence of GPS coordinates into the sequence of visits of checked-in places (represented by a place ID). This procedure has two main advantages: i) the resulted high-level representation is simple for modeling and captures most important information of user mobility, and ii) the model can predict user mobility without knowing the actual coordinates, thus avoiding the privacy issues related to location sharing.

Place discovery. The automatic extraction of places that people visit has been addressed in previous works with different ways of defining places [10, 9, 11, 17]. We used a modified version of a recently proposed approach [25]. The raw trajectory data is first used as an input to detect *stay points*, which are defined by circular geographic regions in which users stay for a significant amount of time (in our work, at least 20 minutes). All discovered stay points are then clustered into meaningful locations called *stay regions* using a grid clustering algorithm. This method divides the space with a uniform grid where each cell is a square region of side length equal to D meters. Then the algorithm finds places of 3×3 cells by looking for the highest density cell and sets any unassigned stay point of 3×3 region around the cell to the new stay region (see Fig. 1(a)). This process repeats until all non-empty cells are assigned. A drawback of this method is that the discovered regions are not highly optimized to cover most stay points, see Fig. 1(a), where one point stays out the place. We improved the place discovery algorithm by adding an additional step of maximizing the number of covered stay points with respect to the set of regions that cover the highest density unassigned cell. Figure 1(b) illustrates a simple example of our variant, where we chose the best 3×3 region among 9 possible regions. In our implementation, we also increased the resolution of the grid and consider 5×5-cells region instead of 3×3-cells region. Cell size was set to $D^2 = 50 \times 50$ meters and the final dimension of stay regions is 250×250 meters, which is large enough to handle WiFi-location noise occurred in some places. Compared to the original grid clustering algorithm, our variant generates less but more accurate stay regions by finding the best fit of stay regions for the set of stay points.

High-level representation. Once the visited *places* are detected, we can remap the sequence of raw coordinates into sequence of place IDs. Note that not all coordinates belong to a place as some might correspond to motion of the user, and the place ID is set to NULL for these transition points. The sequence of visits is then derived from the sequence of place IDs, where a visit corresponds to a time interval with a single non-null place ID which last at least 20 minutes. (The duration of a visit is defined as the time difference between the last point and first point of the interval). For a given



(a) Stay points and grid clustering (b) Modified grid clustering

Figure 1. Place discovery from stay points. The grid clustering can be viewed as the process of finding the set of square regions that cover all detected stay points.

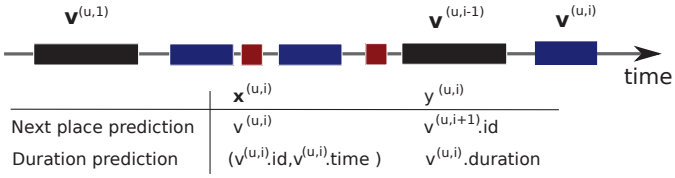


Figure 2. The sequence of visits (colored by place ID) and the two prediction tasks.

user u, the high-level representation of location data is then $v^{(u,1)}, ..., v^{(u,N_u)}$ where $v^{(u,i)}$ denote the i^{th} visit for user u, N_u is the total number of visits for that user, and each visit is characterized by the ID of the place $v^{(u,i)}.id$, the check-in time $v^{(u,i)}.time$, and the duration $v^{(u,i)}.duration$.

CONDITIONAL MODEL FOR MOBILITY PREDICTION

In this paper, we consider the issue of human mobility prediction in an online setting, in which the user model updates its parameters after each visit to predict future visits. In other words, we simulate a smartphone personalized application that learns user mobility patterns and make predictions on the fly. As the available data increases continuously (i.e., the model knows more about user habits), we hypothesize that the prediction accuracy could be improved over time.

Formulation of the prediction tasks

The problem of predicting human behavior can be formulated as finding a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ where $\mathcal{X}$ is the contextual space and $\mathcal{Y}$ is the behavioral output space. Let $\{\mathbf{x}^{(u,i)}, y^{(u,i)}\}_{u=1..U,i=1..N_u}$ be a training set of U users, in which user u has N_u context-output pairs. In our online setting, to predict the output $y^{(u,i)}$ of the context $\mathbf{x}^{(u,i)}$, we can only use data $\{\mathbf{x}^{(u,j)}, y^{(u,j)}\}_{j=1..(i-1)}$ to learn the function f. Note that the both the context space $\mathcal{X}$ and the output space $\mathcal{Y}$ depends on the task. Also, to improve the predictive performance, in addition to location and time, we can also enrich the context by considering some additional contextual information available on the phone, such as the density of nearby Bluetooth devices, or the various phone application logs that might be indicative or the user's states. In this setting, we investigate two predictive tasks which are illustrated in Figure 2.

Task 1 (next place prediction): We want to predict what could be the place ID of the next visit, $v^{(u,i+1)}.id$, given that we have knowledge of $v^{(u,i)}$.

Task 2 (duration prediction): We want to predict the stay duration of the i^{th} visit to a given the place $v^{(u,i)}.id$ with an arrival time $v^{(u,i)}.time$.

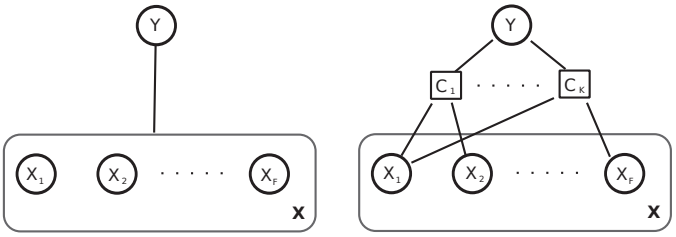


Figure 3. Conditional models

Conditional model

We consider a probabilistic approach in which the context and the outcome are modeled as random variables. Formally, we learn the distribution $P(Y|\mathbf{X})$ where $\mathbf{X} = \{X_1, X_2, ..., X_F\}$ denotes the set of F contextual variables and Y denotes the outcome variable. For example, if both contextual variables and outcome variable are discrete, then we can estimate the conditional probability from observation as follows: $p(Y = y|\mathbf{X} = \mathbf{x}) = \frac{n_{\mathbf{x},y}+\alpha}{\sum_{y'}(n_{\mathbf{x},y'}+\alpha)}$, where $n_{\mathbf{x},y}$ is the count of observing the context-outcome pair $(\mathbf{x}, y)$ in the learning data, and α is a (small) regularization factor. To simplify the presentation, we also use the notation $p(y|\mathbf{x})$ to represent the conditional probability $p(Y = y|\mathbf{X} = \mathbf{x})$ hereafter.

Clearly, the more accurate the context is, the less uncertain the outcome might be. In other words, the model is more accurate if we consider more relevant contextual variables. For example, the combination of current location and time could be more effective than using location or time alone. However, if the contextual space $\mathcal{X}$ is large then we need more data to estimate robustly the distribution $P(Y|\mathbf{X})$. The main challenge then to build a model that can exploit efficiently the available contextual variables from limited data samples. An extreme case is to consider the full conditional model (Fig 3 (a)), in which the outcome variable Y depends on all contextual variables X_f jointly. This could be the ideal model if we had enough samples from the true distribution and the computational power was not a problem. On the extreme, one could consider a Naive Bayes model in which model characterizes dependencies between Y and each contextual variable X_f separately. However, the Naive Bayes model is based on the (strong) hypothesis that contextual variables are independent given the outcome.

Ensemble method. We propose to use an intermediate solution by factorizing the distribution into multiple conditional distributions which take subsets of $\mathbf{X}$ as context as in Figure 3 (b). Formally

$$P(Y|\mathbf{X}) = \frac{\prod_{k=1}^{K} P_k(Y|\mathbf{C}_k)^{w_k}}{Z(\mathbf{X})} \quad (1)$$

where $\mathbf{C}_k \subset \mathbf{X}$ denotes a subset of contextual variables, w_k is a weighting factor for the distribution P_k, and $Z(\mathbf{X})$ is a normalization constant. This is analogous to considering a logarithmic opinion pool [8] of K conditional distribution given by K models, each of which uses a specific subset of contextual variables and a distribution over the output space. Note that we do not restrict that the sets of contextual vari-

ables to be unique. Instead, we could use various models for the same set of contextual variables to improve the predictive performance. The weight vector $\mathbf{w} = (w_1, ..., w_k)$ is a parameter of the combination, and there are several ways to learn it from the given observations. This should in effect give more weight to models that are more accurate. Let $\mathbf{x}_k$ be the shortened context of $\mathbf{x}$ which uses contextual variables in C_k. The model performs the following steps for predicting $y^{(u,i)}$:

Step 1: For each model k, update model parameters based on the mobility history of user u: $\{\mathbf{x}_k^{(u,j)}, y^{(u,j)}\}_{j=1..(i-1)}$

Step 2: Compute conditional output distributions: $P_k(Y|C_k = \mathbf{x}_k^{(u,i)})$.

Step 3: Compute output of Combined model :

$$(y^{(u,i)})^* = \text{argmax}_y \prod_{k=1}^{K} p_k \left(y|\mathbf{x}_k^{(u,i)}\right)^{w_k}.$$

In the above prediction process, there are two types of model parameters: the parameters of each individual conditional distribution $P_k()$ and the combination weight vector $\mathbf{w}$. While conditional distributions $P_k()$ is estimated from mobility history of a single user u (i.e., personalized model), the combination weight can be shared between multiple users. In practice, we can learn $\mathbf{w}$ on a training dataset to obtain a user-independent solution for $\mathbf{w}$. The learning of $\mathbf{w}$ depends on the output space, and it will be detailed in the following sections.

Up to now, we present our general predictive model. In the next sections, we will present in detail models for the two basic prediction tasks, which correspond to two settings: discrete output space and continuous output space.

PREDICTING NEXT PLACE

For the prediction of next place, the outcome could be one of the region IDs that have been previously visited or a new place. Formally, at time t, the output space is:

$$\mathcal{Y} = \{\, v^{(i)}.id \mid v^{(i)}.time \leq t\} \cup \{NewPlace\}$$

where $NewPlace$ corresponds to any previously non-visited place. Since the output space is discrete, we consider a multinomial distribution over the set of possible destinations including the special category, $NewPlace$. For this task, we build the context at visit i based on location and time, as given in Table 1. Although the previous location is also relevant for the prediction task, we did not used it since this information is usually unavailable due to missing data. While most of these variables involve the current visit only, the two variables X_5 and X_6 accumulate information from the past mobility to build contextual information. These variables group places into various categories, depending on frequency and duration of visits (user behavior is expected to be similar for places of the same category). While the place category is less specific than the actual place, its use is helpful for infrequently visited places for which the number of observations per place is limited.

Parameter estimation

For predicting $y^{(u,i)}$ given $\mathbf{x}_k^{(u,i)} = \mathbf{x}_k$, we could use the direct estimate of the conditional distribution (called 'flat' in the discussions to follow) as follows:

	Name	Description	Task
X_1	*LOC*	*ID of the current place*	1 & 2
X_2	*HOUR*	*hour of the day.*	1 & 2
X_3	*DOW*	*day of the week (from Monday to Sunday).*	1 & 2
X_4	*WE*	*workday/weekend indicator.*	1 & 2
X_5	*FREQ*	*frequency of visits to the current place. It is discretized into 5 levels based on monthly frequency. These levels are separated by 4 values of 1, 4, 10 and 30 visits in a month.*	1 & 2
X_6	*DUR*	*The average visit duration of the current place. It is discretized into 4 levels, separated by 1 hour, 2 hours and 4 hours.*	1
X_7	*BT*	*Number of nearby BT devices during the first 10 minutes of the visit. It is discretized into 5 levels that are separated by 4 values 1, 2, 4, and 8 nearby devices.*	2
X_8	*PC*	*Binary variable which indicates if user call or sending SMS to someone during the first 10 minutes of the visit.*	2

Table 1. Contextual variables for the two prediction task. Temporal variables (X_2, X_3, X_4) are computed with leaving time for task 1, and they are computed with arrival time for task 2.

$$p_k^{flat}(y|\mathbf{x}_k^{(u,i)}) = \frac{n_{\mathbf{x}_k,y} + \alpha}{n_{\mathbf{x}_k} + \alpha \cdot |\mathcal{Y}|}, \quad (2)$$

where $n_{\mathbf{x}_k,y} = \sum_{j=1}^{i-1} 1[\mathbf{x}_k^{(u,j)} = \mathbf{x}_k \wedge y^{(u,i)} = y]$ denotes the number of times the user went to place y given the current context $\mathbf{x}_k$, and $n_{\mathbf{x}_k} = \sum_{j=1}^{i-1} 1[\mathbf{x}_k^{(u,j)} = \mathbf{x}_k]$ is the number of occurrences of $\mathbf{x}_k$. In the above formulation, the factor α adds a regularization effect towards the uniform distribution, especially when the counts are small. Note that this effect becomes less important when the context $\mathbf{x}_k$ is popular (i.e., the denominator is large), for which the estimation of conditional probability is more accurate.

The flat estimation of conditional probability in Eq. 2 considers each observation equally, regardless of time. If the user changes his behavior over time (e.g., changing their home address or changing job), then the flat estimation is biased strongly by the previous mobility patterns. This can be solved by introducing the time variable as part of the estimation. Let $(q_1, .., q_{n_{\mathbf{x}_k}})$ be the index sequence of occurrences of the context $\mathbf{x}_k$ in decreasing order of timestamp. The *weighted estimation*, which focuses on recent observations computes the conditional probability as:

$$p_k^{weighted}(y|\mathbf{x}_k) = \frac{\sum_{j=1}^{n_{\mathbf{x}_k}} \frac{1[y^{(u,q_j)}=y]}{j} + \alpha}{\sum_{j=1}^{n_{\mathbf{x}_k}} \frac{1}{j} + \alpha|\mathcal{Y}|}, \quad (3)$$

where each observation is weighted by $\frac{1}{j}$ where j is the number of occurrences of the context $\mathbf{x}_k$ since the visit $v^{(u,q_j)}$. In our experiments, we used both the *flat estimation* and *weighted estimation* models (Cf. Table 2).

Learning the combination weight

In this section, we describe how to learn the combination weight for the next place prediction task. As discussed in the general condition model, the combination weights could be shared across users, although basic models are personalized and are trained for each user separately.

Let $\{\mathbf{x}^{(u,i)}, y^{(u,i)}\}_{u=1..U, i=1..N_u}$ be a training set for learning $\mathbf{w}$. For each example (u,i), we always has K output distributions coming from K personalized models that were trained on a subset of data $\{\mathbf{x}^{(u,j)}, y^{(u,j)}\}_{j=1..(i-1)}$. Let $p_{k,y}^{(u,i)} = p_k\left(y|\mathbf{x}_k^{(u,i)}\right)$ be the conditional probability provided by the k^{th} model, then the combined conditional probability can be written as:

$$p(y|\mathbf{x}^{(u,i)}) = \frac{\prod_{k=1}^{K}\left(p_{k,y}^{(u,i)}\right)^{w_k}}{Z(\mathbf{x}^{(u,i)})}, \quad (4)$$

where $Z(\mathbf{x}^{(u,i)}) = \sum_{y'}\prod_{k=1}^{K}\left(p_{k,y'}^{(u,i)}\right)^{w_k}$ is the normalization term. Ideally, we want to find the weight vector $\mathbf{w}$ so that the probability of the true destination, $p(y^{(u,i)}|\mathbf{x}^{(u,i)})$ is higher than other possible destinations. Formally, we want that

$$\begin{aligned} & \prod_{k=1}^{K}\left(p_{k,y^{(u,i)}}^{(u,i)}\right)^{w_k} > \prod_{k=1}^{K}\left(p_{k,y}^{(u,i)}\right)^{w_k} \quad \forall y \neq y^{(u,i)} \\ \iff & \langle \ln \mathbf{p}_{.,y^{(u,i)}}^{(u,i)}, \mathbf{w}\rangle > \langle \ln \mathbf{p}_{.,y}^{(u,i)}, \mathbf{w}\rangle \quad \forall y \neq y^{(u,i)} \end{aligned}$$

where $\mathbf{p}_{.,y}^{(u,i)}$ denotes the K-dimensional vector of probabilities of output y given by all K models. The above inequalities is very similar to ranking problems [3] which could be learned by minimizing the following objective function:

$$\frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{u,i}\sum_{y}\underbrace{\max(0, 1 - \langle \ln \mathbf{p}_{.,y^{(u,i)}}^{(u,i)} - \ln \mathbf{p}_{.,y}^{(u,i)}, \mathbf{w}\rangle)}_{hinge\ loss} \quad (5)$$

where λ is a regularization parameter and $\ln \mathbf{p}_{.,y}^{(u,i)}$ can be viewed as input feature vector to a linear model. We can solve this optimization problem by using Stochastic Gradient Descent (SGD) [14], which is simple to implement and efficient in many machine learning applications.

PREDICTING STAY DURATION

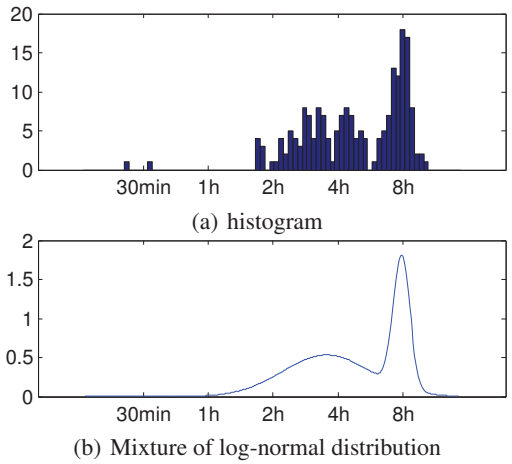
A fundamental modeling difference between predicting the next place and predicting duration lies in the nature of the output variable. The duration is represented by a positive continuous variable: $\mathcal{Y} = \mathbb{R}^+$. For this task, we use all contextual variables for next place prediction except the average stay duration variable since the distribution over duration included already this information. Note that we use arrival time as temporal context since the leaving time is unknown (and the variable to be inferred). In addition to location and time, we also study the use of Bluetooth and phone usage information as contextual variables (see Table 1). Given that the visit duration is unknown, we build the context based on the first 10 minutes of the visit. In other words, if we use Bluetooth and phone usage information, then time of prediction is delayed by 10 minutes after arrival time.

Parameter estimation and prediction

Duration model. The distribution over duration is based on a log-normal distribution which can handle robustly multiple scales (e.g., minute,hour). We used a mixture of log-normal distributions for modeling the conditional distribution of duration y given context $\mathbf{x}_k$:

$$p_k(y|\mathbf{x}_k) = \sum_{m=1}^{M}\frac{p_m}{y\sigma_m\sqrt{2\pi}}\exp\left(\frac{(\ln y - \mu_m)^2}{2\sigma_m^2}\right) \quad (6)$$

where M is the number of components, p_m is the mixture weight of component m, and μ and σ are the mean and standard deviation, respectively, in log-space. This is equivalent to have a mixture of normal distributions on the log-space of the duration. Figure 4 illustrates the distribution of duration at work of one user in our data. As can be seen, there are two peaks at 4 and 8 hours which could be due to the higher likelihood of having lunch outside and or staying at the working place the entire day.



(a) histogram

(b) Mixture of log-normal distribution

Figure 4. Distribution of visit duration at a given place.

For predicting $y^{(u,i)}$ given $\mathbf{x}_k^{(u,j)} = \mathbf{x}_k$, we first select the set of visits that has the previous context: $J = \{j | j < i, \mathbf{x}_k^{(u,j)} = \mathbf{x}_k\}$. The parameters of the mixture model are then estimated based on the set of observed durations $\{v^{(u,j)}.duration\}$ for all $j \in J$. In our implementation, we use the EM algorithm for learning the mixture model.

Leaving time model. We could also predict the duration indirectly by considering the leaving time. The idea is that the arrival time at a place sometimes varies but the the leaving time does not change much (e.g., going to work at 8am whatever the time coming back home the night before). Assume that the leaving time (from 12am to 11:59pm) will not change, we can predict the leaving time then infer indirectly the stay duration. For example, if the user was leaving his office at 6:30pm last time, and he has just arrived at office at 2pm then the potential duration is 4.5 hours. If the leaving time is earlier than the current arrival time then the leaving time is considered to be in the next day. Since we have multiple occurrences of the same context $\mathbf{x}_k$, each occurrence suggests a "potential duration" . Finally, the set of "potential durations" is used to estimate the mixture model.

General vs personalized model. Although human behaviors are personal, people share some common behaviors in real life [7] (e.g., staying long time at home during the night, having lunch at restaurant for about 1 hour, etc.). By exploiting general contextual information such as the time of the day or the frequency of visit, we could predict to some degree the duration of visit. While the global model is not as accurate as the personalized model, it could be useful when the knowledge about user is very limited.

Unlike the personalized model, the general model is estimated only once, using data from multiple users. We also use a mixture model to represent the conditional distribution

of duration given contexts. Given the large number of observations from all users over the whole recording period, one could expect the general model to be a robust prior for improving the performance of the personalized model. Let $\hat{p}(y|\mathbf{x})$ be the conditional probability given by the general model and $p_k(y|\mathbf{x}_k)$ is the conditional probability given by a personalized model. We used a linear combination of General model and Personalized model for predicting:

$$p(y|\mathbf{x}) = \frac{1}{n_{\mathbf{x}_k}+1}\hat{p}(y\,|\mathbf{x}) + 1 - \frac{n_{\mathbf{x}_k}}{n_{\mathbf{x}_k}+1}p_k(y\,|\mathbf{x}_k) \quad (7)$$

where $n_{\mathbf{x}_k}$ is the number of occurrences of $\mathbf{x}_k$ in the mobility history of the considered user, which acts as a reliability measure on the personalized model.

Predicting duration from a conditional distribution. A simple method to predict duration is to output the mode of the estimated conditional distribution (Eq. 6). In our implementation, we use a more sophisticated method, which is based on the expected error on estimated distribution. Since the stay duration can vary significantly, we need an error measure that is meaningful across long stays and short stays. For this reason, we normalize the time difference between predicted and actual duration by the max value of these two. Formally, we define an error function as: $error(y', y) = \frac{|y'-y|}{\max(y',y)}$. This relative error function ranges between 0 and 1, with corresponds to the percentage of error made by the prediction. Given the conditional distribution $p(y|\mathbf{x}_k)$, the expected loss $E(y')$ of a given prediction y' is defined as $E(y') = \int_y error(y', y)p(y|\mathbf{x}_k)\partial y$. The predicted duration is the one that minimizes the expected loss: $y^* = \text{argmin}_{y'}\, E(y')$.

Note that the estimation of $E(y')$ involves an integral which might be intractable. A solution is to approximate the 1-D function with a step function, whom integral might be computed efficiently. The basic idea of approximation by step function is to divide the continous space into segments using a set of boundaries $\{v_j\}$ of increasing order, then the approximated value is constant in each segments. Formally, a function $g(y)$ can be approximated by $\tilde{g}(y)$ where: $\tilde{g}(y) = g\left(\frac{v_j+v_{j+1}}{2}\right) \forall y \in (v_j, v_{j+1})$. Clearly, the approximation is more accurate if we use many steps of small width, but it also increase the algorithmic complexity. In our implementation, we found that using a few hundreds bins is enough to obtain optimum performance. This approximation technique is also used in the next section.

Learning the combination weight

For a given combination weight vector $\mathbf{w}$, the combined probability of stay duration y at the i^{th} visit of user u can be written as follows:

$$p(y|\mathbf{x}^{(u,i)}) \;\; = \frac{\prod_{k=1}^{K}\left(p_{k,y}^{(u,i)}\right)^{w_k}}{Z(\mathbf{x}^{(u,i)})} = \frac{\exp\langle \ln \mathbf{p}_{.,y}^{(u,i)}, \mathbf{w}\rangle}{Z(\mathbf{x}^{(u,i)})}, \quad (8)$$

where $Z(\mathbf{x}^{(u,i)}) = \int_y \exp\langle \ln \mathbf{p}_{.,y}^{(u,i)}, \mathbf{w}\rangle \partial y$ is the normalization factor. Similar to the previous section, the goal is to find $\mathbf{w}$ that gives high probability to the true duration and low probability to other possible durations. Since $\mathcal{Y}$ is continuous, it is not possible to add pairwise loss between the true duration and all possible durations in the learning problem. Instead, we could learn $\mathbf{w}$ to maximize the conditional likelihood : $\prod_{u,i} p(y^{(u,i)}|\mathbf{x}^{(u,i)})$.Taking the log, and substituting $p(y|\mathbf{x}^{(u,i)})$ in Eq. 8, we get the following problem:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\text{argmax}} \sum_{u,i} \langle \ln p_{.,y}^{(u,i)}, \mathbf{w}\rangle - \ln \int_y \exp\langle \ln p_{.,y}^{(u,i)}, \mathbf{w}\rangle \partial y.$$

Since the computation of $Z(\mathbf{x}^{(u,i)})$ is intractable, we use the step function to approximate the integral. The approximated optimization problem is then:

$$\underset{\mathbf{w}}{\text{argmax}} \sum_{u,i} \langle \ln p_{.,y}^{(u,i)}, \mathbf{w}\rangle - \ln \sum_j (v_{j+1} - v_j) e^{\langle \ln p_{.,\bar{v}_j}^{(u,i)}, \mathbf{w}\rangle}$$

where $\bar{v}_j = \frac{v_j+v_{j+1}}{2}$ are representative value of the segment $[v_j, v_{j+1}]$. In our implementation, SGD [14] was used to solve this optimization problem.

EXPERIMENTAL RESULTS

In this section we first present a procedure we implemented to prepare the data for the prediction tasks. We then present results on next place prediction and on visit duration prediction tasks, and finalize with a discussion about computational complexity of our method.

Handling real-life data for predictive tasks

Basic location statistics. Given the real-life recording conditions, the collected data contains time intervals in which the phone did not record any data (either the phone was off or the recording was manually turned off by participants) or when location data is missing. While the set of volunteers participated in the campaign for 1 year of data on average (each user contributed between 3 and 17 months of data), only 74% user-days have at least 1 location records (active days). Considering these days, we found that 62% of all 10-minute slots have at least 1 location record. This sparsity of the data generates difficulties in both the learning of user behavior and the evaluation of prediction methods.

Defining trusted observations. Due to the missing location data during the recording, some actual visits might have not been detected, and some detected visits might be erroneous. For example, if the phone is out of battery 30 minutes after the user arrived to his office, the detected visit would have the correct start time but an erroneous end time. For this reason, we introduced the concept of *trusted transition* and *trusted visit* for learning and evaluation of our predictive model.

Trusted transition: A transition between visit $v^{(u,i)}$ and $v^{(u,i+1)}$ is trusted if there are location data points of user u every 10 minutes between the leaving time of $v^{(u,i)}$ and the starting time of $v^{(u,i+1)}$.

Trusted visit: A visit $v^{(u,i)}$ is trusted if there are location data points in both the period of 15 minutes before the arrival time and the period of 15 minutes after the leaving time.

By construction, during a trusted transition (in which user's location is available every 10 minutes), we are certain that the user did not stay at any places more than 20 minutes, so that the detected next place in the data set is the actual next place. For a trusted visit, we know that the user was outside the visited place right before arrival and after leaving, so both the starting time and leaving time are trusted.

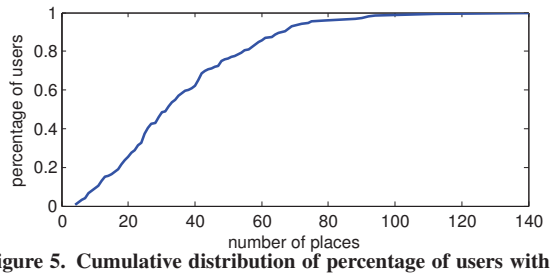


Figure 5. Cumulative distribution of percentage of users with respect to number of places.

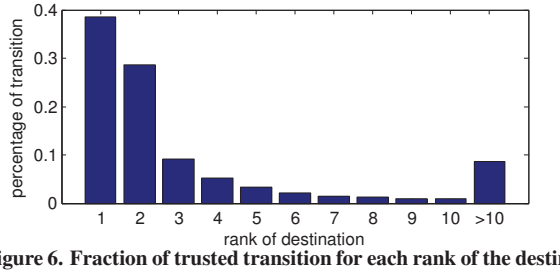


Figure 6. Fraction of trusted transition for each rank of the destination, ordered by number of visit.

Note, however, that these filtering methods work under the assumption that the raw location data is always accurate (e.g., 50 meters or less). This is in general true given the location estimation method used in our framework, but in practice we detected a small amount of non-accurate data after manually checking some abnormal (most likely impossible) behaviors (e.g., multiple visits at a same place during nights). Unfortunately, we do not have a fully reliable way to detect and filter out these anomalies which come mainly from the data sources (e.g., instabilities in WiFi).

Final data for prediction tasks. The processed data consists of 98.000 visits from 153 users, corresponding to an average of 2.5 visits per user per active day. On average, the place discovery algorithms outputs 37 distinct places per user, but this number varies significantly depending on the user. As can be seen in Figure 5, the largest fraction of users visited 20-80 distinct places in which they stayed for at least 20 minutes. After filtering, we get 30,000 trusted transitions for next place prediction and 41,000 trusted visit for duration prediction. These numbers correspond to 0.6 trusted transitions per day per user and 0.8 trusted visits per day during the recording periods for each user. As user behavior models are trained from trusted transitions and trusted visits, the relative low rate of trusted observations is expect to have a direct effect on the predictive performance, especially for infrequent places for which the number of observations is already low. For frequently visited places, one could expect that the model can make accurate predictions in the long term. To study the distribution of visits over places, we sort the set of places of each user by the total number of visits. Then we report the percentage of trusted transitions for each place-rank of destination (e.g., most visited, second most visited, etc) in Figure 6. Note that, to reduce the influence of missing data, we only consider trusted transitions. On one hand, a few top places have very large fraction of visits compared to the rest of place. On the other hand, despite the low frequency of visit, the total number of occasional visits are not ignorable (e.g., 10% of trusted transitions' destinations are places outside the top-10 places). While only a few top places are involved in daily routines, we also found that the mobility patterns are quite complex in our data with 18% of loop-transitions (i.e., $v^{(u,i)}.id = v^{(u,i+1)}.id$) in the set of trusted transitions (which could be affected by erroneous raw location data). In the leave-one-user-out cross-validation, we train the combination weight vector on data from 152 users and then making prediction for every trusted observation of the remaining user. Recall that to make prediction at time t, all model parameters are estimated with user's data up to time t, in other words, the model always predict the future.

Table 2. Average accuracy of conditional models for next place prediction. The higher the better. The baseline performance is 0.411.

Feature set	Flat	Wei.	Feature set	Flat	Wei.
LOC	0.577	0.590	FREQ+HOUR	0.568	0.573
HOUR	0.559	0.561	FREQ+HOUR+WE	0.572	0.576
DAY	0.411	0.392	DUR+HOUR	0.573	0.577
WE	0.432	0.424	DUR+HOUR+WE	0.580	0.583
FREQ	0.483	0.497	FREQ+DUR	0.514	0.523
DUR	0.481	0.486	FREQ+DUR+HOUR	0.570	0.573
LOC+HOUR	0.602	0.604	FREQ+DUR+HOUR+WE	0.572	0.575
LOC+HOUR+DAY	0.544	0.545	ALL	0.520	0.520
LOC+HOUR+WE	0.599	0.601			
Ensemble method	0.64				

Wei.=Weighted

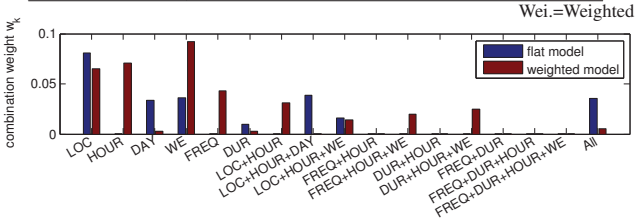


Figure 7. Learned w of the ensemble model for next place prediction.

Next place prediction results

Accuracy. Table 2 reports the all-time average prediction accuracy over the set of trusted transitions. The baseline accuracy is 0.411, which corresponds to the model that outputs the most visited place up to prediction time with an exception that if the user was in the most visited place then the next destination is predicted to be the second most visited place. As described earlier, we consider various sets of contextual variables for building single models, then we combined them together to get a ensemble model that exploits all the contextual information. For each set of variables, we considered two models corresponding to two ways of estimating the probability. These sets were obtained by a heuristic greedy process which starts with individual features, then adding more features to the existing sets of features until the performance is not improved. Note that *LOC* is more specific than the place categories *FREQ* or *DUR*, combining the actual location and its categories does not enrich the context. Finally, we also consider the set of all contextual variables.

Among the set of conditional variables, we see that *LOC* is the most important for predicting next place with accuracy 0.59 using weighted estimation. In general, weighted estimation is slightly better that flat estimation, which can be explained by the fact that people change their mobility patterns over time. Note that the model using only location with flat estimation is equivalent to a 1st-order Markov chain defined on the set of destinations. *HOUR* is the second most important contextual variable and the combination of location and hour also result the best single model with 0.604

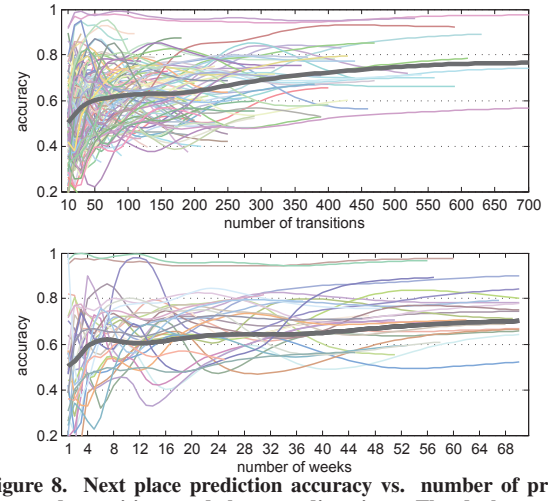
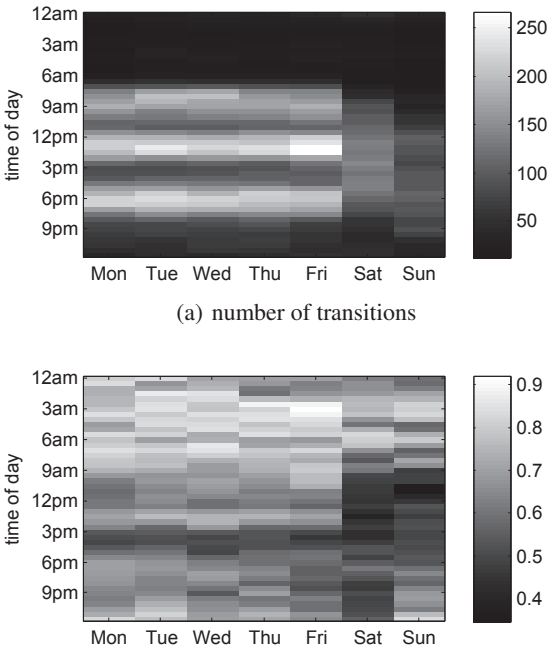


Figure 8. Next place prediction accuracy vs. number of previously observed transitions and the recording time. The dark curve is the average estimated on all users. Each fine curve corresponds to one user.

accuracy. Note that using richer context does not improve the accuracy, which seems to come from the sparsity of observed data in the contextual space. For example, the models that consider the all contextual variables have accuracy around 0.53, even worse than using *LOC* or *HOUR* only.

Although the use of rich contextual set does not improve the performance of single predictor, it can contribute to the final solution of the ensemble method. Our ensemble method successfully exploits the large number of contextual variables by combining the output probability from multiple models. Using prior information from these models, we reach the all-time average accuracy of 0.64. Figure 7 illustrates the weights learned for each single model. As can be seen, almost all selected subsets of contextual variables contribute to the final solution.

Predictability over time. Since our personal model parameters are updated after each observation, we could expect that the prediction accuracy would improve over time. Figure 8(a) illustrates the accuracy of the ensemble model as a function of number of previously observed transitions. Kernel density estimation was used to estimate the accuracy at a given number of trusted transitions (we used normal kernel with $\sigma = 4 + 0.2n$ where n is the number of transitions). The dark curve is estimated on the whole set of trusted transitions for all users and the fine curve is estimated on data for each user separately. Note that the number of transitions varies depending on the user, which results in curves of different length. As expected, the accuracy has the tendency to increase as the number of observed transitions increased. However, we also observed a large variance of accuracy at the beginning (10-50 transitions), which reduces as the number of transitions increases. Also, many user curves are not monotone, which suggests again that user mobility patterns might change over time. Figure 8(b) shows the evolution of accuracy as a function of time since the first trusted transition had occurred. To improve the readability of the figure, we only show curves for people who contribute at least 6 months of data and have at least 300 trusted transitions. Again, we observe that the accuracy generally improved over time. However, its seems that the correlation between the accuracy and the recording time is weaker than the



(a) number of transitions

(b) accuracy

Figure 9. Human mobility in weekly calendar.

correlation between the accuracy and the number of trusted transitions. This justifies that the number of observations plays a central role in prediction.

Finally, we study the predictability of human mobility with respect to the weekly calendar. Figure 9(a) and (b) show the number of trusted transitions and the accuracy of the ensemble model, respectively, in each 30-minute time slot in the weekly calendar. As can be seen, the number of transitions reflects the daily movement of people in real life, such as going to work in the morning or having lunch at noon in weekdays. While there are few transitions during the nights, the destinations of these transitions are quite easy to predict (probably going home). On the contrary, 3-4pm is the most inactive period in office hours, but the predictability is low, illustrated by the dark colors in Figure 9(b). Transitions in weekends are hardest to predict, especially from 9am-4pm. It is also interesting to note that the accuracy on Sunday evening is higher than on Saturday evening, which reflects the pattern of going home on Sunday evening and getting ready for work on Monday.

Visit duration prediction

In this section, we study the performance of our models on the data set consiting of 41,000 trusted visits. Recall that we developed a user-independent model (called general model) which exploits the general dependencies between duration and general contextual variables such *HOUR* or *DOW*. For all mixture models, we set the maximum number of components to be 2 (larger number does not help improving the performance). The baseline results correspond to a model that always outputs the median of trusted visit duration, estimated to be 2.3 hours. Table 3 reports relative error of duration prediction of general model with various sets of contextual variables. These sets were obtained by heuristic greedy process similar to the case of next place prediction. The best contextual variable was found to be *HOUR* while the best

Table 3. All-time averaged error of general conditional models. The lower the better.

Feature	Error	Feature set	Error
FREQ	0.519	FREQ + HOUR	0.442
HOUR	0.505	FREQ + HOUR + DOW	0.445
DOW	0.592	FREQ + HOUR + WE	0.443
WE	0.587	FREQ + HOUR + WE + BT	0.442
BT	0.586	FREQ + HOUR + WE + BT + PC	0.443
PC	0.596	FREQ + HOUR + BT	0.441
		FREQ + HOUR + PC	0.442
		FREQ + HOUR + BT + PC	0.445
baseline	0.590		

Table 4. Duration error of Personalized model and General+Personalized model.

context	Personalized		General+Personalized	
	Dur. mod.	Lea. mod.	Dur. mod.	Lea. mod.
WE	0.536	0.672	0.531	0.669
DOW	0.543	0.668	0.529	0.658
HOUR	0.470	0.477	0.447	0.456
LOC	0.423	0.562	0.407	0.543
PC	0.539	0.672	0.535	0.669
BT	0.525	0.654	0.517	0.648
FREQ	0.481	0.661	0.477	0.655
LOC + WE	0.427	0.548	0.407	0.524
LOC + DOW	0.461	0.545	0.420	0.503
LOC + HOUR	0.420	0.428	0.376	0.384
LOC + HOUR + WE	0.429	0.437	0.377	0.385
LOC + HOUR + DOW	0.477	0.481	0.398	0.403
BT + PC + HOUR + WE	0.490	0.497	0.436	0.443
FREQ + WE	0.481	0.652	0.474	0.643
FREQ + HOUR + WE	0.444	0.455	0.404	0.413
FREQ + BT	0.480	0.627	0.468	0.613
FREQ + BT + WE	0.483	0.616	0.465	0.596
FREQ + BT + HOUR + WE	0.474	0.480	0.411	0.417
Ensemble distribution	0.355			
Ensemble output	0.375			

Dur. mod.=Duration model ; Lev. mod.=Leaving time model

set of contextual variables was $\{FREQ,HOUR,BT\}$, which gives a relative error of 0.441.

Thanks to its user-independent nature, the general model can be used for new users without retraining. However, we could not expected that this model could have optimal performance since each user has a different mobility pattern. The results of the personalized models are reported in Table 4. We found that the best single personalized model is based on *LOC* and *HOUR*, giving a relative error of 0.42, which is slightly better than the performance of the general model. By combining the best general model with the personalized model, the relative error drops significantly to 0.376 with the best single model. Finally, the ensemble method over general+personalized models also improve the performance: our ensemble method (called ensemble distribution) reduces the relative error to 0.355. We also implemented a linear combination approach (called ensemble output) which is a popular combination method that used the output value of basic models only. As can be seen, this baseline approach for combining multiple models improves very slightly the best general+personalized model (0.375 vs 0.376). This result emphasizes the strength of our approach which can combine efficiently multiple models.

Finally, we study the average error conditioned on the stay duration. The set of visits is divided into 5 categories: *less than 1h, 1-2h, 2-4h, 4-8h, more than 8h*. The histogram of stay duration in Figure 10(top) shows that *short visit* (less than 1 hour) and *long visit* (more than 8 hours) are the two most popular categories. However, the average errors for

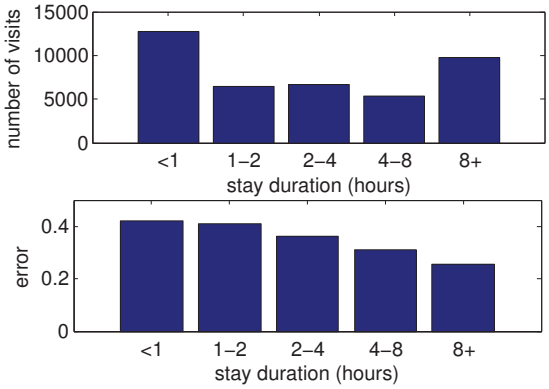


Figure 10. Average relative error with respect to stay duration.

	<1h	1-2h	2-4h	4-8h	>8h	accuracy
<1h	8034	1913	1021	601	1168	0.63
1-2h	2318	1992	981	533	678	0.31
2-4h	953	978	2969	1170	656	0.44
4-8h	402	364	1221	2667	675	0.50
>8h	270	428	492	1087	7490	0.77

Table 5. Confusion matrix of duration prediction and accuracy per category. Rows correspond to actual duration and columns correspond to predicted duration. The overall classification accuracy is 0.56.

these two categories are rather different. Figure 10(bottom) shows that the relative error for long visits are significantly lower than the one for short visits (0.254 vs. 0.420), which means that most prediction errors come from short visits. To provide a more common evaluation measure, we consider a classification task where the predicted durations and the actual durations are mapped into the above 5 categories and report the confusion matrix (see Table 5). Interestingly, while having high relative error, short visits of less than 1 hour have a classification accuracy rate of 63%, only lower than the recognition rate of the longest category (77%).

Algorithmic complexity of our framework

The computational cost of the framework consists of learning the combination weights, updating the user behavior model and making predictions. Learning the combination weights is the most expensive part, which is done on a dedicated data set for calibrating the weights of single predictors (e.g., the objective function in Eq. (5) involves 480,000 hinge loss terms in our experiment). However, the weight $\mathbf{w}$ is estimated just once, before the actual use of the predictive module. This computation would not represent a load to a mobile device running the prediction application.

For next place prediction, the output space is discrete, therefore the cost for updating parameters (basically, updating the counts of destination for a given context) and the cost for making predictions (combining conditional distributions, Eq. 4) are low. In the case of duration prediction, both parameter updating and predicting require more computations. Note that as we use 1-D mixture model to represent the conditional distribution on the continuous output variable, the basic models are to be estimated with EM (in practice, less than 10 iterations). For predicting duration, recall that we relied on a step approximation of the continuous output distribution since an analytic solution is not available. The cost for building the approximation and making predictions is then $O(K \times L)$ where K is the number of predictors

($K = 36$ for duration prediction experiments) and L is the number of steps in the approximation (in practice, we set $L = 500$, corresponding to roughly 1% of relative error on the duration). Hence, the overall on-device computational cost are relative low, and can be handled by most modern mobile devices.

CONCLUSION

In this paper, we developed a general framework for predicting human mobility behavior and applied it to two specific tasks. Our approach can be viewed as an ensemble method that combines a set of models which learn various mobility patterns from past observations. Considering a difficult real-life data set, we demonstrated the potential of our approach on predicting human mobility in real world conditions. While human mobility is not always predictable, repetitive routines can be learned and predicted from contextual cues sensed by smartphones. The set of contextual cues can be efficiently exploited in a principled way using ensemble methods, leading to improved performance.

One issue that we encountered is the low rate of trusted observations, this can be solved by improving the sensing technique and/or exploiting untrusted/missing observation in the learning framework. Among many other directions to explore, we would like to consider more types of contextual variables and other mobility patterns for improving predictive performance. This would increase both the number of contextual variables and the number of basic models for capturing different types of mobility patterns. Since our approach is general, we are also interested in applied it to other prediction tasks including human behavior beyond mobility. This paper considers the prediction of user mobility when they arrive to or leave a place. In practice, it could be relevant to predict user behavior at any time the context changes. Finally, while this work considers only discrete contextual variables to simplify the estimation of basic models, we also plan to study other types of contextual variables or to use non-parametric methods for basic models.

Acknowledgments

This work was funded by Nokia Research Center Lausanne (NRC) through the LS-CONTEXT project.

REFERENCES

1. R. Bajaj, S. L. Ranaweera, and D. P. Agrawal. GPS: location-tracking technology. *Computer*, 35(4):92–94, 2002.
2. A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson. People-centric urban sensing. In *Proc. WICON*, 2006.
3. O. Chapelle and S. S. Keerthi. Efficient algorithms for ranking with svms. *Inf. Retr.*, 13:201–215, June 2010.
4. E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proc. KDD*, pages 1082–1090, 2011.
5. T.-M.-T. Do and D. Gatica-Perez. Groupus: Smartphone proximity data and human interaction type mining. In *Proc. ISWC*, June 2011.
6. N. Eagle and A. (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, 2006.
7. M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.
8. T. Heskes. Selecting weighting factors in logarithmic opinion pools. *Proc. NIPS*, pages 266–272, 1998.
9. J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes. Learning and recognizing the places we go. In *Proc. UbiComp*, pages 159–176. 2005.
10. J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *Proc. WMASH*, 9(3):110–118, 2004.
11. M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *Proc. INFOCOM*, 2006.
12. J. Krumm and A. J. B. Brush. Learning time-based presence probabilities. In *Proc. Pervasive*, 2011.
13. J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. In *Proc. Ubicomp*, pages 243–260, 2006.
14. H. J. Kushner and G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer Verlag, 1997.
15. T. Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks. *Selected Areas in Communications*, 16(6):922–936, 1998.
16. A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proc. KDD*, pages 637–646, 2009.
17. P. Nurmi and S. Bhattacharya. Identifying meaningful places: The non-parametric way. In *Proc .Pervasive Computing*, 2008.
18. S. Patel, J. Kientz, G. Hayes, S. Bhat, and G. Abowd. Farther than you may think: An empirical investigation of the proximity of users to their mobile phones. In *Proc. Ubicomp*, pages 123–140, 2006.
19. A. Peddemors, H. Eertink, and I. Niemegeers. Predicting mobility events on personal devices. *Pervasive Mob. Comput.*, 6:401–423, August 2010.
20. M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. Contextphone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4(2):51–59, 2005.
21. S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Proc. Pervasive*, pages 152–169, 2011.
22. C. Song, Z. Qu, N. Blumm, and A.-L. Barabsi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.
23. L. Song, D. Kotz, R. Jain, and X. He. Evaluating next-cell predictors with extensive wi-fi mobility data. In *IEEE Transactions on Mobile Computing*, pages 1414–1424, 2004.
24. L. Vu, Q. Do, and K. Nahrstedt. Jyotish: A novel framework for constructing predictive model of people movement from joint wifi/bluetooth trace. In *PerCom*, pages 54–62, 2011.
25. V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proc. WWW*. ACM, 2010.