Where and What: Using Smartphones to Predict Next Locations and Applications in Daily Life

Trinh Minh Tri Do[a], Daniel Gatica-Perez[a,b]

[a]*Idiap Research Institute, Switzerland*
[b]*École Polytechnique Fédérale de Lausanne, Switzerland*
Email: do@idiap.ch, gatica@idiap.ch

Abstract

This paper investigates the prediction of two aspects of human behavior using smartphones as sensing devices. We present a framework for predicting where users will go and which app they will use in the next ten minutes by exploiting the rich contextual information from smartphone sensors. Our first goal is to understand which smartphone sensor data types are important for the two prediction tasks. Secondly, we aim at extracting generic (i.e., user-independent) behavioral patterns and study how generic behavior models can improve the predictive performance of personalized models. Experimental validation was conducted on the Lausanne Data Collection Campaign (LDCC) dataset, with longitudinal smartphone data collected over a period of 17 months from 71 users.

Keywords: smartphone data, human behavior, mobility prediction, app usage prediction, Lausanne Data Collection Campaign

---

1. Introduction

The ability to foresee what mobile users want to do has many applications such as improving user interfaces or providing relevant recommendations. For example, users can have quick access to the map application if the system knows that they are going to a new place and thus positioning assistance is needed. In another context, if the system knows that the user will go for lunch, then it can recommend a list of restaurants together with useful information like today's menu, availabilities, traffic conditions, etc. Mobile phones are convenient options for tracking and mining user behavior in daily life [1, 2] as they are usually placed in close proximity to the users [3]. Smartphones contain many sensors that can record contextual and activity cues, including location, application usage, and calling behavior. This information can be considered as both input and output of a prediction framework, in which the future values of some variables (e.g., next place) are predicted based on the current context (current place, time, etc.).

Constructing predictive models of human behavior has been a topic of interest in the area of recommendation systems [4, 5, 6], context-aware services [7, 8], and personalized and adaptive interfaces [9]. While these studies support the claim that human behavior can be inferred through mobile phones, none of these studies, to our knowledge, exploit the information made available through multiple sensors such as location, call logs, and proximity to others in order to build or enhance predictive models capable of determining aspects of user behavior in the future.

Hence, in this paper, we consider the task of predicting human behavior based on multiple smartphone sensors using statistical methods. Our framework is inspired from prediction algorithms commonly used in signal processing for forecasting time series. More precisely, we predict the next location of a user and which application he/she will use based on the current context consisting of location, time, app usage, Bluetooth proximity, and communication logs. This approach allows modeling the interplay between the predicted variables to study relationships between the place where a user stays and the possibility that he would make a phone call, use the cameras and so on. Furthermore, other sources of information, such as the list of nearby Bluetooth devices or system information, are also exploited in order to enrich the user context and improve the predictive models. In summary, our paper makes the following four contributions. First, we present a general framework for predicting jointly various dimensions of human behavior using multimodal data. To our knowledge, this is the first work attempt to study the application usage prediction task jointly with location. Second, we study the impact of each data type to the predictive performance of the two tasks. Third, we investigate the use of generic behavior patterns for improving prediction performance of personalized models. And finally, we conduct our analysis on a longitudinal dataset involving 71 volunteer users carrying a smartphone over 17 months, from the Lausanne Data Collection Campaign [10]. This dataset allows us to extensively study the predictive performance conditioned on various aspects such as the evolution of predictive performance over a long period of time.

The paper is organized as follows. The next section describes recent work pertaining to human behavior prediction in the context of mobile computing. Next, we describe our prediction framework. Subsequently, we present our results on predicting location and phone usage and summarize our findings.

2. Related work

Analyzing human behavior with mobile phones has received considerable interest in the recent past. For example, the Reality Mining dataset [1, 11] has been extensively used for this purpose. This dataset was collected over a course of 9 months from 94 students and staff at MIT using Nokia 6600 phones for recording call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status. Location that is inferred from cell tower data [12, 13] can be recorded by mobile phone operators for millions of people, offering the possibility of large-scale analysis of human mobility such as predictability of human mobility [14] or identifying human daily activity patterns [15]. Note that the resolution of location data inferred by cell tower IDs is relatively low compared to GPS data, which is used in this paper, resulting in uncertainty on the user position. Besides cell tower and GPS, short distance wireless network data (e.g., Bluetooth and WiFi) can also be used for positioning, with some advantages such as high spatial resolution and the ability to work in indoor environments [16, 17].

Previous works on human behavior prediction focused mainly on predicting human mobility in different settings, such as online inference of a destination based on the location trace [18], location prediction and visit duration estimation for a given time in the future [19], or estimating the present probability at a specific place at a given time [20]. In addition, many modern mobile sensing systems have integrated inference modules for reasoning about human behavior and context to achieve energy-efficient sensing and high-level representation of context [21, 22]. Interestingly, not much work has been done to extend predictions to other facets of users' most common activities such as phone usage or physical interaction with other people. Although phone usage and its relationship to human behavior has been extensively studied [23, 24, 25], little progress has been made in going beyond descriptive statistical analysis to actually perform prediction. In the context of Human Computer Interaction, utilizing information from a user's behavior to modify mobile interfaces has been investigated in the past. Vetek et al. [26] for example, proposed a method to dynamically improve the recommendation of shortcuts on the home-screen of a smartphone using user context. These authors used an unsupervised clustering based method that uses location data, in the form of GSM Cell IDs to learn user contexts to improve recommendations of these shortcuts. Finally, Bridle and McCreath [9] have investigated the use of a richer context, using phone profiles and statistics pertaining to calls/SMS usage in addition to the cell tower IDs for this purpose.

Pattern discovery and prediction are closely related tasks since activity cues are usually periodic and/or repetitive [27, 28]. Human behavior is to some degree predictable by a model that captures some emergent patterns from the behavioral data. For example, Eagle et al. [29] characterized the emergence of mobility behaviors using factor analysis. Similar analysis was also done using topic models [30], which were in the past used in text mining and natural language processing. While not directly considering the prediction task, some work focused on the understanding of how people use their phone [31, 32], which might be used as prior knowledge for prediction methods. In a preliminary work, we presented an analysis on how people use their phone conditioned on various contexts [33], where the context is represented by semantic location (i.e., place categories) and Bluetooth density. In this paper, we go beyond the dependencies between context and phone usage to a prediction framework that infer future user activities. Furthermore, this work considers an enriched context representation with more data types such as app usage and individual Bluetooth proximities.

As discussed earlier, we do not restrict ourselves to phone usage data or location data for the two prediction tasks. The main challenge is how to use efficiently the data from a large number of sensors in a unified framework. As simple method is use discrete representation of context then build a model for each state separately. This is an effective method if the number of data types is limited, such as for the combination of two data types: time and location [20]. More sophisticated methods require redesigning the predictive model to integrate multiple types of observation, such as the spatio-temporal decision tree [34] or the probabilistic latent variable models which combine spatial, temporal, and social relation information for predicting human movements [35]. This approach is, however, task-specific and potentially hard to be extended to a large number of contextual variables. As an alternative, ensemble methods represent a promising direction, as recently addressed in [36]. This work is limited to choosing a single best model rather than combining the (typically available) prior information from multiple models. In this paper, we use multiple data streams for dealing with multimodal data, for which each data type is represented by one or several data streams. Based on this representation, we were able to use statistical predictive methods such as random forest or least-square regression, and also an ensemble approach for combining multiple models. Compared to our recent work considering long range predictions of human mobility such as how long a user will stay in the current place [37], this paper focuses on very short future predictions. We also develop a more complete representation of context which allows us to learn useful generic models for predicting several dimensions of human mobile behavior.

3. Framework

In this section, we first describe the dataset used in our study. Next, we describe the task that we address along with various data representations, which will be used throughout this paper. Finally, we describe various statistical models that can be integrated in our prediction framework based on the information we get from smartphone sensors.

3.1. The dataset

We use data collected with Nokia N95 phones and a continuous sensing software capable of periodically saving sensor data from volunteers. This enabled us to collect long-term data from several modalities such as GPS, Bluetooth (BT), WiFi access points, accelerometer, and applications and phone usage logs. The software contained a state-machine approach similar to [38], to define an adaptive sensing procedure. Operating states (like “indoors”, “outdoors”, “travel”, etc.) were determined heuristically to maximize battery life by sampling from the sensors at different rates depending on the operating states. This data set is a subset of the Lausanne Data Collection Campaign [10] which is collected from 71 volunteer users in Switzerland between October 2009 and February 2011: the average duration of a participant’s data is 13.5 months since not all users were active for the entire duration. The population is composed of university students and professionals. Users carried their smartphone as their actual mobile phone, and were asked to charge the phone at least once a day. In summary, the data contains roughly 10 million location points, 1.4 million application usage events (such as opening, closing, minimizing, etc. without including system applications) and 8.8 million non-empty Bluetooth scans.

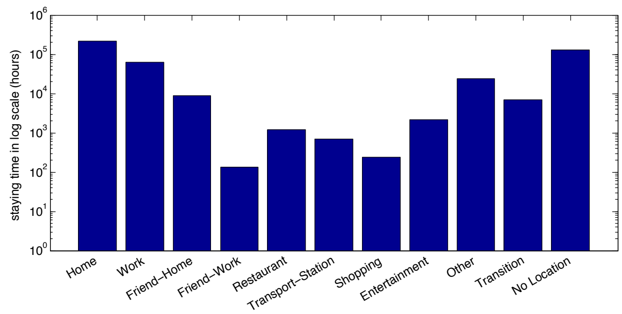


Figure 1: The 10 semantic location labels and their total staying time (in hours) plotted in log scale. The 9 first locations correspond to places, *Transition* corresponds to the case when the user is traveling. No location corresponds to the case when the location of user is unknown.

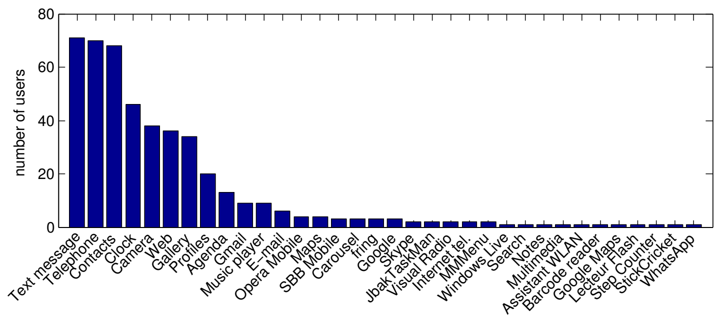


Figure 2: Histogram of frequently used applications across all users.

Location data preprocessing. From the raw location coordinates extracted by the sensing software through GPS and known WiFi-access points, semantic labels were assigned to the most common places using a two-step procedure described in [39]. First, we discovered places that a user visited repeatedly or for a long duration of time, and then self-reported annotations of these places from the users themselves were obtained. Places are defined as small circular regions in which users stay for at least 10 minutes. The radius of place is set to 100 metters in our experiment to deal with the existence of noisy data at some location.

In the second step, the users were asked to annotate these *stay regions* into a set of semantically meaningful labels. Each user annotated eight regions on a map through a web interface. As discussed in [39], five of the eight regions corresponded to the most frequently visited places by the users, and the three remaining regions were randomly chosen from the lowest tenth percentile. There were 22 mutually exclusive predefined labels for annotators, including a special category named "Other" which should be chosen if the 21 remaining labels are not appropriate. Many labels are lengthy and some of them were not frequently used in the annotation, we therefore mapped these labels to the following 8 labels: *Home,Work,Friend-Home,Friend-Work,Restaurant,Transport-Station,Shopping,* and *Entertainment*. In this shorten list, *Friend-Home* stands for home of a friend or relative and Friend-Work stands for working place or school of a friend or relative. The *Transport-Station* category corresponds to static places such as metro station or bus stop. Unlabeled regions or labeled regions that do not belong to any of the 8 labels were finally mapped to an "*Other*" label. In addition, we have two special categories: *Transition* for traveling context and *No Location* for unknown location states due sensing failure (corresponding to 28% of the recording period). While *Transition* belongs to the list of predicted label, we do not consider *No Location* label for the prediction. The total stay time for each *semantic locations* is showed in Figure 1.

In our work, we also use application usage, Bluetooth proximity, communication statistics, and operating state of the sensing software. The user context is then represented by several variables of multiple modalities.

Since people often used a limited number of applications on the N95, we selected only the most frequently used applications for building the user context. The application logs consist of usage events of all applications, including system apps and pre-installed apps like *Camera* or *Clock*, and user downloaded applications like *Gmail*. A histogram of frequently used applications (opened at least once a week) across different users is given in Figure 2. Not surprisingly, the *text message*, *telephone*, and *contacts* applications are the most frequently used.

3.2. Data representation

We use data streams to encode the human behavior over time and use statistical methods to learn a predictive human behavior model. We first group together location labels, applications usage, and other contextual information into 10 minute *time slots* (the length of the time slots is a hyper-parameter and can be changed depending on the application). This enables us to view smartphone data in the form of a set of temporally arranged *data streams* ($\boldsymbol{\Delta}$), with each stream corresponding to one modality such as GPS or BT. This is conceptualized in Figure 3. Several of these streams can now be grouped into categories. In our work, of the many possibilities, we concern ourselves with six categories, that are explained below.

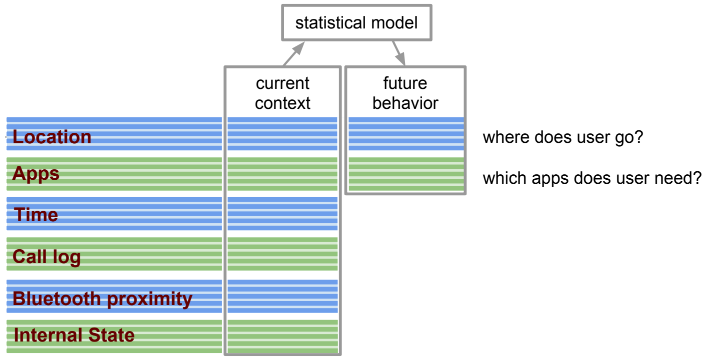


Figure 3: Overview of our prediction framework.

1. Location (Δ^{loc}): Consisting of 10 binary data streams where 10 is the number of semantic location labels. Each stream corresponds to a significant labeled place (stay regions) that a user has been found to visit (see category list in Fig. 1). In cases where a user had less than 10 different visited place labels, extra streams were zero-valued. For example, if a user has visited 5 different semantic places in the past, this category will have 5 data streams corresponding to the 5 locations, in addition to a "transition" stream, occurring when a single location of the user is not determined for the majority of time in a time slot, and 4 other streams that are zero valued.
2. Apps (Δ^{app}): Consisting of the set of app streams, where each stream corresponds to the count of app-view-events of an app that is used least once a week by the user (see Fig 2 for the list of frequently used apps). Since each user used a different set of apps, the number of app streams varies for different users. System apps (e.g., screen saver) are included in the representation of context, but they do not belong to the list of apps of interest to be predicted. The median of number of predicted apps across users is 7.
3. Time (Δ^{time}): We use time-of-day and day-of-week to represent the temporal context. The temporal information is encoded with 48 streams for time-of-day, 7 binary streams for day-of-week, and one final stream indicating if the day is a weekday or weekend.
4. Call log ($\Delta^{call\ log}$): Consisting of data 5 binary streams representing the number of communication events for each time slot. The 5 data streams corresponds to 5 categories of communication event: SMS received, SMS sent, outgoing call, incoming call, and missed call.
5. Bluetooth proximity (Δ^{bt}): From BT scans, we extract the number of nearby BT devices and the presence of frequently observed BT devices.

For each BT device that has been observed at least once a week, we dedicate one binary data stream with a "1" indicating the presence of the said BT device within the given time slot. The number of BT data streams varies depending on the user, and the median value is 9.

6. Internal state ($\mathbf{\Delta}^{state}$): Internal operating states of the recording software are also used as context. These states are inferred with several heuristic rules on accelerometer data, WiFi readings, GPS, and system information. More details can be found in [10].

These categories together describe the overall usage of smartphone by a user. At this point, we define the contextual vector $\mathbf{x}_t$ of a user in a time slot t to be a concatenation of all data streams into a single vector, given by:

$$\mathbf{x}_t = [\mathbf{\Delta}_t^{loc}, \mathbf{\Delta}_t^{app}, \mathbf{\Delta}_t^{time}, \mathbf{\Delta}_t^{call\ log}, \mathbf{\Delta}_t^{bt}, \mathbf{\Delta}_t^{state}], \tag{1}$$

where Δ_t indicates the value of the data streams at time t. We can therefore define our task to be the prediction of the next context given the current user activity, in other words, to predict $\mathbf{y}_t$ given $\mathbf{x}_t$,

$$\mathbf{y}_t = \left[\mathbf{\Delta}_{t+1}^{loc}, \mathbb{1}(\mathbf{\Delta}_{t+1}^{app} > 0)\right], \tag{2}$$

corresponding to location and app usage information at time $t+1$. In the formulation above, $\mathbb{1}(.)$ stands for the indicator function so that all elements of $\mathbf{y}_t$ are binary. Since $\mathbf{y}_t$ is also represented by data streams, it is straight forward to build models that output activation scores $\hat{\mathbf{y}}_t$, from which we can derive the final outputs of interest such as the most likely next location, and the top apps that are likely to be used in the next 10 minutes. The prediction tasks of our study and possible application scenarios are discussed in the following subsection.

3.3. Prediction tasks and application scenarios

Task 1. Where can a user be expected to go? This task can be viewed as a multi-class classification task where the ground-truth $\mathbf{\Delta}_{t+1}^{loc}$ has exactly one non-zero element corresponding to the next location. In practice, one can compute the activation score (e.g. probability) for each of the 10 semantic places, where a high score implies that the place is likely to be visited in the next 10 minutes. Then, the place with the highest score is predicted as the next destination. We use the average classification accuracy as evaluation measure.

Task 2. Which applications is a user more likely to use? In the case of app usage prediction, the number of app data streams in $\mathbf{\Delta}^{app}$ depends on the user. Furthermore, users can open multiple applications during a 10-minute time slot, we do not have any constraint on the number of non-zero elements in $\mathbf{\Delta}^{app}_{t+1}$. We formulate this problem as an information retrieval task, where the query is the context and the system is asked to return most likely apps. Among standard IR evaluation measures, we found that *recall values* are particularly appropriate in our application scenario.

Our prediction setting is inspired by the bursty nature of human behavior: there are brief periods of intensive activity followed by long periods of inactivity [40]. While the proposed framework is quite general, **we focus on anticipating human behavior for the very short future (the next ten minutes) and on active time slots of location changes or app usage**. In other words, we predict 1) where a user will be in the next ten minutes if he/she is about to move; and 2) which are the most likely apps given that the user is going to use some apps. Technically speaking, we only consider time slots in which the user is about to move for the mobility prediction analysis, and the time slots for which the user opens at least one app in the next ten minutes for app usage prediction. This selection affects both the training and the evaluation of predictive models.

Application scenarios. Mobility prediction can be used to infer user needs or to define more specific context. One could think of application scenarios related to context-aware recommendation or personal assistance. By including the destination into the context, mobile personal assistance can allow users to ask for services like "Remind me to go to the post office before I arrive home", which is not possible if the destination of the trip is not inferred. Knowing that the user will go to a restaurant soon, the recommendation system can show a list of nearby restaurants together with their daily specials. The use of app usage prediction is also straightforward as mentioned earlier: A context-aware UI can introduce several short-cuts to most likely apps in the home screen, based on the current context.

Note that the conditions that "the user is about to move" or "is going to use some apps" are generally true for context-aware services since people often do not need very long range predictions, especially when they are in the long period of inactivity. For example, if the user is not going to use any app, then the app prediction is useless for context-aware UI. Our setting fits the usual case when users need some context-aware services only for a short time before the activity of interest. Finally, long range prediction problems such as "when will the user leave the current place?" are addressed in our previous works [37] but not considered in this analysis. In principle, our prediction framework can be applied to long range prediction (by integrating the long range outputs) and to make prediction in inactive time slots (by including inactive time slots in the training data). However, we keep the focus on one single prediction setting to highlight the impact of other aspects on predictability, such as data types or generic behavioral patterns.

3.4. Human behavior models

Online prediction simulation. Our model has its theoretical basis in causal predictive models commonly used in signal processing. In general, these models predict the value at time $(t+1)$ when the past values up to time t are given. To simulate online prediction, we divide each user data into K chunks, where the first chunk is not used for evaluation. For data in chunk $k \geq 2$, we train statistical models on chunks $1, ..., (k-1)$, that is, we never use future data as training data. We set $K = 51$ in our experiment, that is, for user who has about one year of data, one chunk corresponds to approximately one week, this setting corresponds to an online prediction system which is retrained once a week to update its parameters. Our setting is more challenging but also more realistic than a hypothetical cross-validation setting in which future data can be used in the prediction of past data.

Since our prediction tasks have relatively large, labeled training data, we use a supervised approach to train the predictive models. For each output stream s where s can denote either location or app usage, we train a regression model which outputs the activation score $y_{t,s}$ for that stream in the next time ten minutes. An exception is the Markov models for location prediction, for which we define the activation scores to be the transition probabilities from the current context to the set of possible destinations.

Least-Square Linear Regression. Linear regression is a simple yet efficient prediction model in which the parameters of the regression model θ are estimated using the least squares solution, given by: $\min_\theta \sum_{t \in \mathcal{T}} \|y_{t,s} - \langle \mathbf{x}_t, \theta \rangle\|^2$ where $\mathcal{T}$ is the set of time slots in training data, and $\langle . \, , . \rangle$ represents a scalar product.

Logistic Regression. Logistic Regression differs from a Linear Regression in that it transforms the scalar product scores estimated by standard linear regression to a probability, using the equation: $P(y_{t,s} = 1 | \mathbf{x}_t; \theta) = \frac{1}{1+e^{-\langle \mathbf{x}_t, \theta \rangle}}$.The model parameters in this case can be estimated by maximizing the conditional likelihood on the training data: $\max_\theta \sum_{t \in \mathcal{T}} log \; P(y_{t,s} | \mathbf{x}_t; \theta)$.

Random Forest. Random Forest is a combination of many decision trees (non-linear models) that predict outputs by sequentially studying the value of each attribute in the input [41]. Being non-linear in nature, decision trees can potentially fit data better than linear models such as Linear Regression. On the flip side, they have been found to overfit the training data. Random Forest overcomes this problem in many cases, by selecting randomly a subset of features in each decision tree.

Markov Models. These models have been widely used for location prediction in the past and they can be applied naturally to our location data as well [42]. In this work, we define each state to correspond to one location and the Markov models are trained on the sequence of location changes (i.e., two consecutive elements cannot be the same state).

Blending Predictors. In order to study the advantages of using information from all the data streams and models given above, we train a fusion model that is a linear regression of the output of all other models. To avoid overfitting, we use a "delayed technique" in which the fusion coefficient is estimated on validation sets. Let m_c be a model of family m which was trained on data up to chunk c, let $\hat{\mathbf{y}}_t^{c,m}$ be the output scores of the model m_c on the time slot t, and let $\hat{\mathbf{y}}_t^{c,\cdot}$ be the concatenation of $\hat{\mathbf{y}}_t^{c,m}$ for all families m. For the training data from chunk 1 to chunk K, the fusion coefficients for the output stream s are estimated as follows:

$$\theta_{blend}^{K} = \operatorname*{argmin}_{\theta} \sum_{k=2..K} \sum_{t \in \mathcal{T}_k} w_t \| y_{t,s} - \langle \theta, \hat{\mathbf{y}}_t^{c,\cdot} \rangle \|^2$$

where $\mathcal{T}_k$ stands for time slots in chunk k, and the term w_t was introduced in order to give more weight to recent activities. Formally, we set $w_t = \frac{1}{T-t+1}$ with T being the last timeslot in $\mathcal{T}_K$.

3.5. Exploiting generic patterns of behavior.

Our predictive model can be personalized or generic (i.e., user-independent) depending on the training data. We investigate the use of a generic model to improve the personalized model.

Generic data types from multiple users. The main challenge of building a generic model lies in the fact that the dimensionality of context and output varies depending on the user. To learn a generic model and apply it for a given user, generic features and output are needed. Interestingly, many of our features have a shared meaning across users thus can be integrated into the generic model. As an important example, our location data – which has been mapped to semantic place categories – is ready to be integrated in the generic model. Concerning app usage, we can rely on the UIDs of apps which are globally defined within an app ecosystem. Note, however, that each user used a different set of apps, and many apps are used by only a few users. To ensure the generalization of the generic models, we only include apps which were used by at least 10 users for building contextual app streams (i.e., feature vector), while the output of the generic model corresponds to the apps which are used by at least 2 users. At the end, there are 9 common apps in the input and 23 apps in the output of generic models. The only remaining non-generic data stream is Bluetooth proximity since we have one data stream for each frequently observed BT devices. While the MAC address of BT devices are defined globally, we cannot include these BT streams as part of the generic data due to two reasons. First, each individual BT device is observed by a rather small number of users. Second, the "relationships" to the device's owner are user-specific. For example, if A and B are colleagues, and B and C are family members then A usually see B at work while C usually see B at home. The context of "seeing B" cannot be defined as a common contextual cue for the general population. At the end, we only use one BT stream, which is the count of the number of nearby BT devices. Finally, the two last data stream categories (Call log and Internal State) are integrated into the generic prediction framework without any modification.

Combining generic model with personalized model. Given data from multiple users, we can apply the standard statistical methods described above to learn the generic model and then combining the prediction scores of the generic model with the scores given by the personalized model. We use two methods for the combination. In the first method, we consider the generic model as an elementary predictor and integrate its prediction scores in the Blending model presented earlier. In the second method, we combine the prediction scores of first method and the prediction scores of the generic model directly. Let $\hat{y}_{t,s}^{Gen}$ be the output score of the generic model, and $\hat{y}_{t,s}^{Blend\ Per+Gen}$ be the output of the first combination method. The final combination is computed as follows:

$$\hat{y}_{t,s}^{Per+Gen} = (1-\alpha_t)\hat{y}_{t,s}^{Blend\ Per+Gen} + \alpha_t \hat{y}_{t,s}^{Gen}, \qquad (3)$$

where $\alpha_t = \frac{t_0}{t_0+t}$ decreases as t increases. The constant value t_0 is set to 500 in our implementation.

Real deployment scenario. The algorithmic cost for making prediction with our framework is relatively low, but the learning process can be

Table 1: Correlation between next locations and context variables. Top 8 variables with strongest correlation values are shown for each location data stream.

Home		Work		Friend-Home	
$\mathbf{\Delta}^{loc}(Home)$	-0.18	$\mathbf{\Delta}^{time}(IsWeekEnd)$	-0.19	$\mathbf{\Delta}^{loc}(Work)$	0.08
$\mathbf{\Delta}^{time}(3:30AM)$	0.17	$\mathbf{\Delta}^{time}(Sat)$	-0.13	$\mathbf{\Delta}^{state}(pluggedin)$	0.07
$\mathbf{\Delta}^{loc}(Transition)$	0.14	$\mathbf{\Delta}^{loc}(Transition)$	0.13	$\mathbf{\Delta}^{time}(IsWeekEnd)$	0.06
$\mathbf{\Delta}^{loc}(Work)$	-0.12	$\mathbf{\Delta}^{loc}(Work)$	-0.13	$\mathbf{\Delta}^{bt}(num\ devices)$	-0.06
$\mathbf{\Delta}^{state}(known\ wlan\ lost)$	-0.11	$\mathbf{\Delta}^{time}(8:00AM)$	0.12	$\mathbf{\Delta}^{loc}(Home)$	-0.05
$\mathbf{\Delta}^{bt}(num\ devices)$	-0.10	$\mathbf{\Delta}^{time}(Sun)$	-0.11	$\mathbf{\Delta}^{time}(Sun)$	0.04
$\mathbf{\Delta}^{state}(stationary)$	-0.09	$\mathbf{\Delta}^{loc}(Home)$	-0.11	$\mathbf{\Delta}^{time}(12:30AM)$	0.04
$\mathbf{\Delta}^{time}(8:00AM)$	-0.08	$\mathbf{\Delta}^{time}(8:30AM)$	0.10	$\mathbf{\Delta}^{time}(5:00AM)$	0.04

Friend-Work		Restaurant		Transport-Station	
$\mathbf{\Delta}^{time}(7:30AM)$	0.14	$\mathbf{\Delta}^{bt}(num\ devices)$	0.14	$\mathbf{\Delta}^{state}(indoor\ mobile)$	0.06
$\mathbf{\Delta}^{time}(3:30PM)$	0.12	$\mathbf{\Delta}^{state}(known\ wlan)$	0.07	$\mathbf{\Delta}^{app}(Web)$	0.05
$\mathbf{\Delta}^{time}(5:00PM)$	0.10	$\mathbf{\Delta}^{time}(11:30AM)$	0.06	$\mathbf{\Delta}^{app}(Telephone)$	0.05
$\mathbf{\Delta}^{state}(urban\ stationary)$	0.09	$\mathbf{\Delta}^{state}(outdoormob.2)$	-0.06	$\mathbf{\Delta}^{loc}(Transition)$	0.05
$\mathbf{\Delta}^{time}(IsWeekEnd)$	-0.08	$\mathbf{\Delta}^{state}(known\ wlan\ lost)$	0.06	$\mathbf{\Delta}^{bt}(num\ devices)$	0.05
$\mathbf{\Delta}^{time}(Sat)$	-0.05	$\mathbf{\Delta}^{time}(IsWeekEnd)$	-0.05	$\mathbf{\Delta}^{app}(Standbymode)$	0.04
$\mathbf{\Delta}^{bt}(num\ devices)$	0.05	$\mathbf{\Delta}^{loc}(Other)$	-0.04	$\mathbf{\Delta}^{state}(known\ wlan\ lost)$	0.04
$\mathbf{\Delta}^{time}(Sun)$	-0.05	$\mathbf{\Delta}^{state}(outdoormob.3)$	-0.04	$\mathbf{\Delta}^{app}(Log)$	0.03

Shopping		Entertainment		Other		Transition	
$\mathbf{\Delta}^{state}(urban\ stationary)$	0.07	$\mathbf{\Delta}^{loc}(Transition)$	0.05	$\mathbf{\Delta}^{loc}(Other)$	-0.14	$\mathbf{\Delta}^{loc}(Transition)$	-0.37
$\mathbf{\Delta}^{state}(outdoormob.1)$	0.05	$\mathbf{\Delta}^{state}(indoor\ mobile)$	0.04	$\mathbf{\Delta}^{loc}(Transition)$	0.13	$\mathbf{\Delta}^{loc}(Home)$	0.33
$\mathbf{\Delta}^{loc}(Transition)$	0.04	$\mathbf{\Delta}^{time}(6:30PM)$	0.04	$\mathbf{\Delta}^{state}(outdoormob.2)$	0.12	$\mathbf{\Delta}^{loc}(Other)$	0.24
$\mathbf{\Delta}^{time}(Sat)$	0.04	$\mathbf{\Delta}^{state}(outdoormob.2)$	0.03	$\mathbf{\Delta}^{state}(known\ wlan)$	-0.11	$\mathbf{\Delta}^{loc}(Work)$	0.22
$\mathbf{\Delta}^{state}(outdoormob.2)$	0.04	$\mathbf{\Delta}^{time}(7:00PM)$	0.03	$\mathbf{\Delta}^{time}(IsWeekEnd)$	0.10	$\mathbf{\Delta}^{state}(outdoormob.2)$	-0.12
$\mathbf{\Delta}^{app}(Profiles)$	0.03	$\mathbf{\Delta}^{state}(pluggedin)$	-0.02	$\mathbf{\Delta}^{state}(outdoormob.1)$	0.10	$\mathbf{\Delta}^{state}(outdoormob.1)$	-0.11
$\mathbf{\Delta}^{time}(Sun)$	-0.03	$\mathbf{\Delta}^{time}(6:00PM)$	0.02	$\mathbf{\Delta}^{time}(Sat)$	0.09	$\mathbf{\Delta}^{state}(known\ wlan\ lost)$	0.11
$\mathbf{\Delta}^{state}(stationary)$	-0.03	$\mathbf{\Delta}^{time}(3:30PM)$	0.02	$\mathbf{\Delta}^{loc}(Home)$	-0.07	$\mathbf{\Delta}^{time}(3:30AM)$	-0.08

expensive depending on the model (e.g., Random Forest is much more costly than Linear Regression) and the volume of data. While the generic model can be learned offline, the personalized models need to be updated on the fly to exploit the latest recorded data. The learning process of all presented models require memory and computational resources that scale linearly or log-linearly with the volume of data. In our experiments, one year of user data corresponds to few thousands data points, for which learning a statistical model should not be a problem for a typical smartphone. In practice, we can limit the system to store and learn on recent data (e.g., data up to 6 previous months) by discarding old data. While our framework does not require a client-server architecture, cloud computing can be applied if the memory and computation resources of the mobile device are very limited.

4. Mobility prediction: results and discussion

Our analysis starts with a basic analysis of generic behavior patterns and initial results of the prediction task. Then we present experimental results for personalized prediction, and investigate the combination of generic and personalized prediction.

4.1. Generic human mobility patterns

Correlation Analysis. We use correlation analysis to characterize the dependency between contextual variables and the next location. A strong correlation indicates a high predictability degree of the output using a linear model on the considered contextual variable. Table 1 reports the top 8 variables for each location data stream. The strong negative correlation between a destination and the contextual data stream of the same place category (e.g. Home vs. $\mathbf{\Delta}^{loc}(Home)$) reflects the fact that we consider the sequence of location changes in this analysis. That is, if the current location is Home, then the next location cannot be Home by construction. Note that the correlation is not perfect ($|\rho| < 1$) since if the current location is not Home ($\mathbf{\Delta}_t^{loc}(Home) = 0$) then the next location can be Home or another label (i.e., the value of $\mathbf{\Delta}_{t+1}^{loc}(Home)$ is not determinable).

Among the 6 categories of data streams, we observe that $\mathbf{\Delta}^{loc}$ and $\mathbf{\Delta}^{time}$ dominate the set of top variables. For example, Transition is positively correlated with the three dominant locations Home, Other and Work. Work is highly correlated with morning time slots and negatively correlated with weekends. Restaurant is positively correlated with the time slots around noon. (We note that the correlation between Home and the time slot 3:30AM-4:00AM comes a daily reset of the phone client software which may generate unknown location states.) Interestingly, the inferred state data streams $\mathbf{\Delta}^{state}$ provided by the recording software also appear frequently in the table and become important for predicting infrequent place categories such as Transport-Station, Shopping, Entertainment, or Other. Finally BT and Call-log data streams appear to have a weaker correlation to the next destinations but all correlation values are statically significant.

Generic human mobility prediction model. The above analysis suggests that generic mobility patterns exist, so that we can learn generic models to predict human mobility. Our preliminary experiments show that Random Forest is slightly better than Linear Regression models, but the Linear Regression model is much faster. To this end, we use Linear Regression to train our generic mobility prediction model.

Table 2 reports the accuracies of generic Linear Regression models using different sets of features. All results are computed with leave-one-user-out cross-validation. The baseline performance corresponds to the static model that always predict the most frequent destination as output, which is Transition in our data. Looking at the accuracies obtained with single data type, we found similar findings which are highlighted in the correlation analysis:

Table 2: Prediction accuracy of generic linear model for predicting next place.

Feature set	Accuracy	Feature set	Accuracy
Location	0.427	L	0.427
Time	0.386	L+T	0.518
State	0.356	L+T+S	0.539
App	0.322	L+T+S+A	0.541
Call log	0.321	L+T+S+A+C	0.541
BT	0.321	L+T+S+A+C+B	0.543
Baseline performances (MostFrequent)			0.321

Location and Time are the two most important contexts, followed by State and App. Furthermore, the accuracy can be significantly improved by combining multiple data types, from 0.427 with only Location data streams to 0.539 with Location, Time and State data streams. Adding Call log and BT data stream do not improve significantly the accuracy of the generic model.

4.2. Personalized mobility prediction

Accuracy. While generic human mobility can be captured, people have their "own" routines in real life. Therefore, personalized predictive models are expected to be more accurate than generic model. Table 3 reports prediction accuracies with different models. Note that the baseline result of MostFrequent model (accuracy=0.368) is slightly better than the result of generic MostFrequent model (accuracy=0.321) since the most frequent location stream changes depending on the user. Note that Markov models only consider location data, while other personalized models use all data types. In contrast to the results for generic models, we found that Random Forest is significantly better than Linear Regression on learning personalized model. We believe that complex patterns exist in both generic and personalized data, but generic complex patterns cannot be extracted efficiently from a data collected by a rather low number of users (N=71). A more advanced generic human mobility model calls for data from a large number of users and the ability to exploit group people with similar patterns.

By combining single personalized predictors from the family of Markov models, Linear Regression, and the Random Forest, we obtain the best personalized model (Blend Personalized) which reaches an accuracy of 0.641 using only personal behavior data. Finally, combining the generic model with the personalized model increases the overall accuracy to 0.657, which shows the usefulness of generic model on the prediction problem. We will revisit this aspect later when discussing the analysis of prediction accuracies over time.

Table 3: Mobility prediction accuracies with various personalized and blending models. For Markov models, the number in parentheses indicates the order of the model.

Model	Accuracy	Model	Accuracy
MostFrequent	0.368	LinReg	0.598
Markov(1)	0.482	LogReg	0.582
Markov(2)	0.516	Random Forest	0.635
Markov(3)	0.518	Blend Personalized	0.641
Markov(4)	0.514	Personalized+Generic	0.657
Markov(5)	0.507		

Table 4: Mobility prediction accuracies of personalized RF models using different sets of features.

Feature set	Accuracy	Feature set	Accuracy
Location	0.481	L	0.481
Time	0.444	L+T	0.563
State	0.421	L+T+S	0.589
App	0.371	L+T+S+A	0.590
Call log	0.367	L+T+S+A+C	0.588
BT	0.474	L+T+S+A+C+B	0.635
Baseline performances (MostFrequent)			0.368

Which are important data types for personalized location prediction? Again, we study the impact of each data types on the prediction accuracy. The results for Random Forest (the best personalized single predictor) are reported in Table 4. Location remains the most important contextual variable for predicting mobility, yielding a prediction accuracy of 0.481. Surprisingly, BT becomes key contextual information for predicting next location. Using BT data streams alone, an accuracy of 0.474 could be reached, better than the one using Time (0.444) and only lower than the one with Location. Furthermore, adding BT data streams to the model significantly improves the accuracy from 0.588 to 0.635. As BT data streams are proxies of face-to-face interactions, these results suggest that information about whom the user is with has a significant impact on where will he go.

Predictability over time. User predictability also depends on what we know about users. Figure 4 plots the performances of two predictive models as a function of time. Recall that the Personalized model only uses personal data of the user, while the Personalized+Generic model exploits generic models trained on data from other users as well. As shown in the figure, the prediction performances improve over time and start to stabilize after about 20 weeks. Furthermore, the more the personal data is available, the closer the Personalized model approaches the combined model. Notably, the improvement by exploiting generic model is important at the beginning

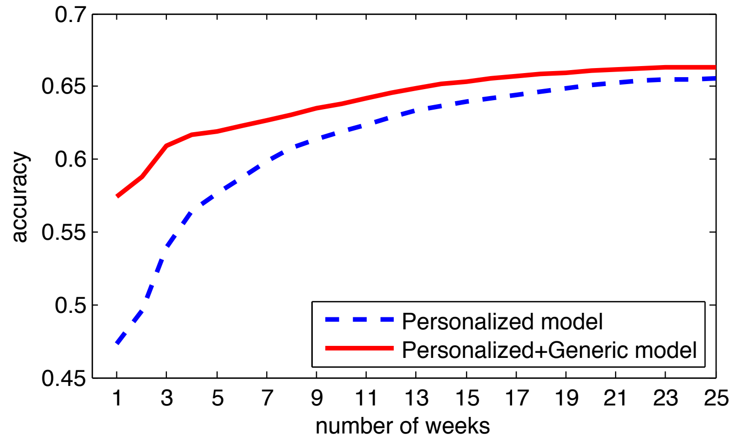


Figure 4: Mobility prediction accuracy increases over time and starts stabilizing after 20 weeks of data collection. Generic model appears to be significantly helpful during the initial weeks.

as we observe significant improvements in the two first months.

5. App usage prediction: results and discussion

5.1. Generic app usage patterns

Correlation between contextual variables and apps that will be opened. Table 5 reports the top 8 contextual variables for each of the 9 common apps that are used by at least 10 users. One common observation is that the use of an app at a given time is highly correlated with the use of that app in the next 10 minutes. For example, in the case of Camera ($\rho = 0.38$), if the user is using camera then there is a high probability that the user continues to take more photos/videos in the next ten minutes. Similar to the case of mobility correlation analysis, most of correlation values in Table 5 are small but statically significant (p-value<0.01). We can find many meaningful results such as Text message is likely to be opened in the next 10 minutes (to send or to receive messages) if the user recently sent/received SMS. Another interesting example is the case of Profiles app which is used for changing ringing types of the phone. This application is highly correlated with the "unstable state" inferred by the recording software such as $\mathbf{\Delta}^{state}(known\ wlan\ lost)$ or $\mathbf{\Delta}^{state}(known\ wlan)$ which correspond to the arrival/departure to/from a known place.

Generic app usage prediction model. We use Linear Regression to train a generic app usage prediction model which outputs activation scores

Table 5: Correlation between contextual variables and the fact that a given app will be used in the next ten minutes.

Text message		Telephone		Contacts	
$\Delta^{app}(Text\ message)$	0.29	$\Delta^{app}(Telephone)$	0.19	$\Delta^{app}(Contacts)$	0.21
$\Delta^{app}(Telephone)$	-0.07	$\Delta^{app}(Text\ message)$	-0.14	$\Delta^{app}(Standby\ mode)$	0.04
$\Delta^{call\ log}(SMS-in)$	0.06	$\Delta^{app}(Log)$	0.14	$\Delta^{state}(urban\ stationary)$	0.04
$\Delta^{app}(Camera)$	-0.06	$\Delta^{app}(Web)$	-0.09	$\Delta^{app}(Web)$	-0.03
$\Delta^{app}(Log)$	-0.05	$\Delta^{loc}(Work)$	0.08	$\Delta^{loc}(Work)$	0.03
$\Delta^{app}(Web)$	-0.05	$\Delta^{app}(Camera)$	-0.08	$\Delta^{app}(Camera)$	-0.03
$\Delta^{call\ log}(SMS-out)$	0.05	$\Delta^{app}(Clock)$	-0.07	$\Delta^{time}(6:30AM)$	-0.03
$\Delta^{app}(Clock)$	-0.04	$\Delta^{app}(sys)$	-0.07	$\Delta^{state}(pluggedin)$	-0.02

Clock		Camera		Web	
$\Delta^{loc}(Home)$	0.23	$\Delta^{app}(Camera)$	0.38	$\Delta^{app}(Web)$	0.36
$\Delta^{app}(Clock)$	0.23	$\Delta^{time}(IsWeekEnd)$	0.08	$\Delta^{app}(sys)$	0.11
$\Delta^{time}(11:30PM)$	0.16	$\Delta^{state}(known\ wlan)$	-0.08	$\Delta^{app}(Standbymode)$	0.11
$\Delta^{time}(12:00AM)$	0.15	$\Delta^{loc}(Work)$	-0.07	$\Delta^{state}(known\ wlan)$	-0.06
$\Delta^{time}(11:00PM)$	0.14	$\Delta^{app}(Text\ message)$	-0.06	$\Delta^{app}(screensaver)$	0.06
$\Delta^{bt}(num\ devices)$	-0.13	$\Delta^{time}(Sun)$	0.06	$\Delta^{time}(7:00AM)$	0.06
$\Delta^{time}(6:00AM)$	0.13	$\Delta^{state}(known\ wlan\ lost)$	-0.06	$\Delta^{app}(EPFLscope)$	0.05
$\Delta^{loc}(Work)$	-0.13	$\Delta^{loc}(Home)$	-0.05	$\Delta^{time}(6:30AM)$	0.05

Gallery		Profiles		Agenda	
$\Delta^{app}(Gallery)$	0.28	$\Delta^{app}(Profiles)$	0.08	$\Delta^{app}(Agenda)$	0.20
$\Delta^{app}(sys)$	0.07	$\Delta^{state}(known\ wlan\ lost)$	0.06	$\Delta^{loc}(Work)$	0.06
$\Delta^{app}(Camera)$	0.07	$\Delta^{state}(known\ wlan)$	0.04	$\Delta^{state}(urban\ stationary)$	0.05
$\Delta^{app}(Text\ message)$	-0.04	$\Delta^{app}(screensaver)$	-0.04	$\Delta^{app}(Text\ message)$	-0.04
$\Delta^{app}(EPFLscope)$	0.03	$\Delta^{state}(pluggedin)$	0.04	$\Delta^{app}(sys)$	0.04
$\Delta^{loc}(Home)$	-0.02	$\Delta^{bt}(num\ devices)$	-0.04	$\Delta^{bt}(num\ devices)$	0.03
$\Delta^{app}(Standbymode)$	0.02	$\Delta^{app}(Standbymode)$	-0.04	$\Delta^{loc}(Home)$	-0.03
$\Delta^{state}(known\ wlan)$	-0.02	$\Delta^{time}(8:00AM)$	0.04	$\Delta^{app}(Web)$	-0.03

Table 6: Recall values generic linear model for predicting app usage.

Feature set	Top-1	Top-3	Top-5	Feature set	Top-1	Top-3	Top-5
App	0.468	0.817	0.922	A	0.468	0.817	0.922
Time	0.395	0.815	0.919	A+T	0.477	0.849	0.926
State	0.378	0.786	0.914	A+T+S	0.476	0.849	0.929
Location	0.378	0.789	0.905	A+T+S+L	0.477	0.849	0.930
BT	0.378	0.780	0.912	A+T+S+L+B	0.476	0.849	0.931
Call log	0.384	0.780	0.908	A+T+S+L+B+C	0.479	0.849	0.931
Baseline performances (MostFrequent)					0.378	0.780	0.908

for the 23 common apps that are used by at least 2 users. Note that a large number of output channels means that the generic model can be used for predicting activation score for a large number of apps. As mentioned earlier, our main evaluation measure is recall value which is meaningful for the application scenario of introducing a few short-cuts for most likely apps in the UI. In particular, we reports recall values top-1, top-3, and top-5 app candidates. As can be seen in Table 6, App and Time are the two most important data types for predicting app usage, followed by State. Unfortunately, the State data streams contributes very little to the final performance, while Location, BT and Call log data streams are almost useless. To verify if these data stream are completely useless for app usage prediction or generic app usage patterns cannot be captured from the small sample of the population, we study the impact of each data type on the personalized model.

Table 7: Recall values of personalized RF models for predicting app usage.

Feature set	Top-1	Top-3	Top-5	Feature set	Top-1	Top-3	Top-5
App	0.508	0.822	0.916	A	0.508	0.822	0.916
Location	0.463	0.811	0.913	A+L	0.511	0.835	0.924
BT	0.460	0.805	0.911	A+L+B	0.510	0.835	0.926
Time	0.454	0.799	0.907	A+L+B+T	0.513	0.839	0.928
State	0.453	0.796	0.903	A+L+B+T+S	0.514	0.842	0.929
Call log	0.457	0.794	0.900	A+L+B+T+S+C	0.514	0.841	0.929
Baseline performances (MostFrequent)					0.458	0.795	0.901

Table 8: App usage prediction performance of different models.

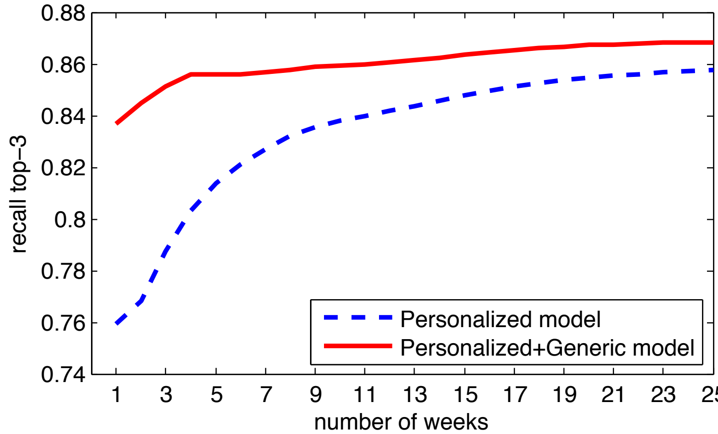
	Recall Top-1	Recall Top-3	Recall Top-5
MostFrequent	0.458	0.795	0.901
LinReg	0.504	0.832	0.921
LogReg	0.505	0.837	0.927
Random Forest	0.514	0.841	0.929
Blend Personalized	0.521	0.845	0.929
Personalized+Generic	0.541	0.862	0.941

5.2. Personalized app usage prediction

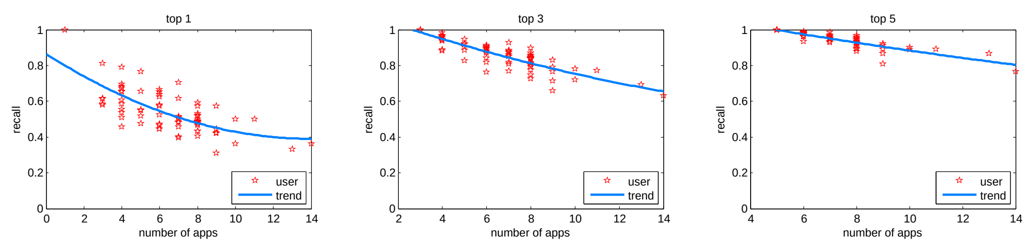
Which data types are important for personalized app usage prediction? Table 7 reports evaluation results of the best single model (Random Forest) using different feature sets. Compared to results of generic model, we find that the importance order of data types has changed significantly. While App remains the most important contextual information, Location and Bluetooth are now found to be useful for predicting app usage. State and Call log remain relatively useless for the prediction.

The detail results of app usage prediction with different models are shown in Table 8, which show that context-aware models outperform significantly the baseline predictive model using only app usage frequency. Furthermore, we observe again that the prediction performance of personalized model can be improved by exploiting generic model. In particular, the improvement is significant in the first few weeks of data collection as illustrated in Figure 5.2.

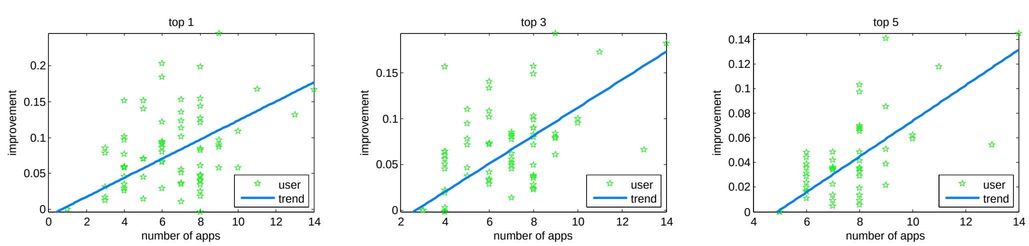
The effect of the number of apps. Finally, we study the effect of number of apps on the performance and on the improvement of context-aware model over the baseline (MostFrequent). Figure 5 plots the recall values and the improvement of the Personalized+Generic method over the baseline, and the number of candidate apps for each user. To track the tendency of performance with respect to the number of apps, we used a second order polynomial to fit the data. The drop in recall values as the number of applications used increases is not linear, as seen in these graphs. Furthermore, we found that the improvement in performance of our models



Recall values increases over time and start stabilizing after 16 weeks of data collection. Generic model appears to be significantly helpful during the initial weeks.



(a) Recall values of the best predictor.



(b) Improvement over baseline performance.

Figure 5: App predictive performance with respect to the number of frequently used apps.

over the baseline was higher for users who had used larger number of apps.

6. Conclusion

In this paper, we have utilized large-scale, long-term longitudinal smartphone data to predicti two aspects of user behavior, i.e., location and app usage. We integrate the rich multimodal information made available through the smartphone sensors to predict location and application usage of the user, based on past and current data, experimenting with data from 71 users and 17 months of time.

In the case of predicting location, we found that Bluetooth proximity is an important contextual cue along with Location and Time. This finding confirms again the dependency between human mobility and social interactions. In the second prediction task, we showed the potential of our method to infer application usage in the future. To our knowledge, little work has been done to predict applications usage based on past activities. Therefore, our results show the potential of this research direction.

Our analysis of generic behavior models shows that combining a generic model with a personalized model is a plausible approach for making accurate predictions, especially when the training data for the personalized model is small (a common situation in practice). We demonstrate a principled way to build a generic behavior model by transforming user specific contexts and variables into generic concepts (e.g., from physical location to semantic location). Our experimental results on the two chosen prediction tasks suggest promising applications of our approach to other problems of user behavior prediction in the future.

Acknowledgement

We thank the partial support of Nokia Research (LS-CONTEXT project) and SNSF (SONVB project).

References

[1] N. Eagle, A. (Sandy) Pentland, Reality mining: sensing complex social systems, Personal and Ubiquitous Computing 10 (4) (2006) 255–268.

[2] T. M. T. Do, D. Gatica-Perez, Human interaction discovery in smartphone proximity networks, Personal and Ubiquitous Computing (2012) 1–19.

[3] S. N. Patel, J. A. Kientz, G. R. Hayes, S. Bhat, G. D. Abowd, Farther than you may think: An empirical investigation of the proximity of users to their mobile phones, Ubiquitous Computing 4206 (2006) 123–140.

[4] V. Bellotti, B. Begole, E. H. Chi, N. Ducheneaut, J. Fang, E. Isaacs, T. King, M. W. Newman, K. Partridge, B. Price, P. Rasmussen, M. Roberts, D. J. Schiano, A. Walendowski, Activity-based serendipitous recommendations with the Magitti mobile leisure guide, in: Proc. CHI, 2008, pp. 1157–1166.

[5] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, J. Crowcroft, Recommending social events from mobile phone location data, in: Proc. ICDM, 2010, pp. 971–976.

[6] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, Recommender Systems Handbook (2011) 217–253.

[7] T. Gu, H. Pung, D. Zhang, A service-oriented middleware for building context-aware services, Journal of Network and computer applications 28 (1) (2005) 1–18.

[8] B. D. Ziebart, A. L. Maas, A. K. Dey, J. A. Bagnell, Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior, in: Proc. UbiComp, 2008, pp. 322–331.

[9] R. Bridle, E. McCreath, Inducing shortcuts on a mobile phone interface, in: Proc. IUI, 2006, pp. 327–329.

[10] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, J. Laurila, Towards rich mobile phone datasets: Lausanne data collection campaign, in: Proc. ICPS, 2010.

[11] N. Eagle, A. S. Pentland, D. Lazer, Inferring social network structure using mobile phone data, PNAS 106(36) (2009) 15274–15278.

[12] T. Liu, P. Bahl, I. Chlamtac, Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks, Selected Areas in Communications 16 (6) (1998) 922–936.

[13] R. Bajaj, S. L. Ranaweera, D. P. Agrawal, GPS: location-tracking technology, Computer 35 (4) (2002) 92–94.

[14] C. Song, Z. Qu, N. Blumm, A.-L. Barabsi, Limits of predictability in human mobility, Science 327 (5968) (2010) 1018–1021.

[15] S. Phithakkitnukoon, T. Horanont, G. Di Lorenzo, R. Shibasaki, C. Ratti, Activity-aware map: identifying human daily activity pattern using mobile phone data, in: Proc. HBU, 2010, pp. 14–25.

[16] N. Banerjee, S. Agarwal, V. Bahl, R. Chandra, A. Wolman, M. D. Corner, Virtual Compass: relative positioning to sense mobile social interactions, in: Proc. Pervasive, 2010, pp. 1–21.

[17] L. Vu, Q. Do, K. Nahrstedt, Jyotish: A novel framework for constructing predictive model of people movement from joint WiFi/bluetooth trace., in: Proc. PerCom, 2011, pp. 54–62.

[18] J. Krumm, E. Horvitz, Predestination: Inferring destinations from partial trajectories, in: Proc. UbiComp, 2006, pp. 243–260.

[19] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, A. T. Campbell, Nextplace: a spatio-temporal prediction framework for pervasive systems, in: Proc. Pervasive, 2011, pp. 152–169.

[20] J. Krumm, A. J. B. Brush, Learning time-based presence probabilities., in: Proc. Pervasive, 2011, pp. 79–96.

[21] M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, A. Campbell, Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones, Pervasive Computing (2010) 355–372.

[22] E. Miluzzo, C. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, A. Campbell, Darwin phones: the evolution of sensing and inference on mobile phones, in: Proc. MobiSys, 2010, pp. 5–20.

[23] C. Y. Wei, Capturing mobile phone usage: Research methods for mobile studies, in: Proc. IPCC, 2007, pp. 1–6.

[24] A. Oulasvirta, R. Petit, M. Raento, S. Tiitta, Interpreting and acting on mobile awareness cues, Human–Computer Interaction 22 (1-2) (2007).

[25] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, D. Estrin, Diversity in smartphone usage, in: Proc. MobiSys, 2010.

[26] A. Vetek, J. Flanagan, A. Colley, T. Keränen, Smartactions: Context-aware mobile phone shortcuts, Human-Computer Interaction (2009).

[27] A. Clauset, N. Eagle, Persistence and periodicity in a dynamic proximity network, in: Proc. DIMACS, 2007.

[28] M. C. Gonzalez, C. A. Hidalgo, A.-L. Barabasi, Understanding individual human mobility patterns, Nature 453 (7196) (2008) 779–782.

[29] N. Eagle, A. Pentland, Eigenbehaviors: identifying structure in routine, Behavioral Ecology and Sociobiology 63 (7) (2009) 1057–1066.

[30] K. Farrahi, D. Gatica-Perez, Discovering routines from large-scale human locations using probabilistic topic models, Transactions on Intelligent Systems and Technology 2 (1) (2011) 3.

[31] S. Nylander, T. Lundquist, A. Brännström, At home and with computer access: why and where people use cell phones to access the internet, in: Proc. CHI, 2009, pp. 1639–1642.

[32] T. M. T. Do, D. Gatica-Perez, By their apps you shall understand them: mining large-scale patterns of mobile phone usage, in: Proc. MUM, 2010.

[33] T.-M.-T. Do, J. Blom, D. Gatica-Perez, Smartphone usage in the wild: a large-scale analysis of applications and context, in: Proc. ICMI, 2011.

[34] A. Monreale, F. Pinelli, R. Trasarti, F. Giannotti, Wherenext: a location predictor on trajectory pattern mining, in: Proc. KDD, 2009.

[35] E. Cho, S. A. Myers, J. Leskovec, Friendship and mobility: user movement in location-based social networks, in: Proc. KDD, 2011.

[36] A. Peddemors, H. Eertink, I. Niemegeers, Predicting mobility events on personal devices, Pervasive and Mobile Computing 6 (2010) 401–423.

[37] T. M. T. Do, D. Gatica-Perez, Contextual conditional models for smartphone-based human mobility prediction, in: Proc. UbiComp, 2012.

[38] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, N. Sadeh, A framework of energy efficient mobile sensing for automatic user state recognition, in: Proc. MobiSys, 2009, pp. 179–192.

[39] T. M. T. Do, D. Gatica-Perez, The places of our lives: Visiting patterns and automatic labeling from longitudinal smartphone data, IEEE Transactions on Mobile Computing (2013) 1.

[40] A. Barabasi, The origin of bursts and heavy tails in human dynamics, Nature 435 (7039) (2005) 207–211.

[41] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth and Brooks, 1984.

[42] D. L. Isaacson, R. W. Madsen, Markov chains: theory and applications, Krieger Publishing Company, 1985.