INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2016; **29**:2169–2187
Published online 25 January 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.2909

User location prediction with energy efficiency model in the Long Term-Evolution network

Yuanyuan Qiao[1,*,†], Jie Yang[1], Haiyang He[1], Yihang Cheng[1] and Zhanyu Ma[2]

[1]*Beijing Key Laboratory of Network System Architecture and Convergence, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Pattern Recognition and Intelligent System Lab., School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*

SUMMARY

Predicting users' next location/place allows us to anticipate their future movement. It provides additional time to be ready for that movement and react consequently. Furthermore, many industries, including Internet Service Providers, are still requiring low cost and simple location/place prediction methods that can be implemented on mobile device. This paper studies domain-independent prediction algorithms and spatio-temporal based prediction method using 20-day-long records in Long Term-Evolution(LTE) network, which captures the mobility patterns of 3474 individuals. After examining the prediction accuracy and resource consumption of domain-independent prediction algorithms, we find Markov provides the best tradeoff. Furthermore, Active LeZi outperforms Markov if enough consecutive parsed patterns of users' history movement are captured. In addition, we further group users according to their spatio-temporal entropy profiles in order to predict not only user's future locations but also the place he or she most likely to appear within a specific period. By applying the simple spatio-temporal based method to each group of user, 83.3% accuracy can be achieved for some users. Yet Markov and Active LeZi algorithms perform better for some other users. This implies that we should consider applying different prediction methods to users with distinct spatio-temporal characteristics. Copyright © 2015 John Wiley & Sons, Ltd.

Received 21 July 2014; Revised 11 September 2014; Accepted 7 November 2014

KEY WORDS: location prediction; energy efficiency; LTE network; spatio-temporal based method; domain-independent prediction algorithms

1. INTRODUCTION

Location-based services on mobile device gain mainstream popularity. There is an increasing interest in developing techniques that can predict user's future locations. Location prediction capabilities can be used in many scenarios, such as preheating home environment for the owner's arrival or for preemptive allocation of resources on cellular networks with expected heavy traffic. Furthermore, many entities, such as mobile device, are still in need of techniques that can predict users' locations with high prediction accuracy and do not cause computational overhead. There exists already congested and overloaded networks with a large scale of users moving around simultaneously. Therefore, overly complex prediction methods will be ineffective.

At present, the most detailed information on human mobility across a big city is collected by mobile devices. Mobile devices recorded the closest mobile tower every time the user uses phone.

*Correspondence to: Yuanyuan Qiao, Beijing Key Laboratory of Network System Architecture and Convergence, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China.
†E-mail: yyqiao@bupt.edu.cn

In LTE network, the closest eNodeB is recorded when users use the mobile Internet. The location information is updated every 12 min although user has been logout from LTE network.

In this paper, by using a 20-day-long record in LTE network, which is collected for traffic monitoring and anonymized by the data source and capture the mobility patterns of 3474 individuals, we focus on tracking and predicting user's movement on user's mobile phone because of several reasons, namely:

(i) Protect the user privacy. All the locations that the user have been to would not be send through the network;
(ii) Use all available data on mobile device. We could choose the preferred data collected by mobile device, like Global Positioning System (GPS), WiFi or Global System for Mobile (GSM)/Universal Mobile Telecommunications System (UMTS) data;
(iii) Be able to collect and predict offline. If the Internet is unavailable, the location-based applications can still work in case the map is already downloaded on to mobile device.

However, making any kind of data processing in mobile devices needs to be concerned with the processing speed, limited memory and battery. Thus, from the extensive collection of different options, this work is focusing on some domain-independent prediction algorithms (Markov and Lempel-Ziv (LZ) based algorithms) and simple spatio-temporal based method, which are able to learn mobility patterns and estimate the probable location with low computational complexity and resource needs. This makes it possible to execute them on mobile devices.

In this paper, we perform a comparative analysis of Markov, Lempel–Ziv (LZ), LeZi Update and Active LeZi algorithms, and find that Markov is the best next location prediction algorithm applied on mobile device that provides the best tradeoff between prediction accuracy and resource consumption. Furthermore, our findings show a clear advantage on the adoption of different methods to users with distinct entropy profile for location prediction as we obtain increased prediction accuracy (probability of correctness) at low computing cost. Spatio-temporal based prediction method is more suitable for those who visit very limited locations during the day and follow regular pattern during the week (family person). For those who spend much time on commute and follow fairly regular pattern during the week (post person), the next location prediction is more efficient.

The remaining of the paper is organized as follows. In Section 2, the related works are reviewed. Section 3 discusses the well-known location prediction algorithms such as Markov, LZ, LeZi Update and Active LeZi algorithms that do not require complex computation. Section 4 provides an overview of the collected mobility trajectories and the comparisons of prediction accuracy and resource consumption for each domain-independent algorithm introduced in Section 2. Section 5 further classifies users into three groups according to their entropy profile. Afterwards, the spatio-temporal based prediction method and the next location prediction algorithms are applied to different groups of users to examine prediction accuracy. Finally, conclusions and future work are presented in Section 6.

2. RELATED WORK

Individuals display significant regularity, as they tend to visit a few highly frequented locations, like home or office. Numerous studies show that people's movement trajectory is far from random. By measuring the entropy of each individual's trajectory, [1] achieved 93% potential predictability in user mobility. When considering both the frequencies and temporal correlations of individual movements, the theoretical maximum predictability can reach 88% [2].

Previous studies regarding location prediction include several models and methods. The usage of well-known mobility models was originally applied in the area of location predication [3, 4], like Bayesian approaches [5–7], neural networks [8], Hidden Markov models [9], Markov models [10] and compression algorithms [11–13]. In addition, some recently proposed new algorithms [14, 15] and frames [16–18] all presented very good results.

Recently, regarding the location prediction, scholars start to consider many other factors that could influence the prediction results, like spatial context [4], temporal factors [19, 20], spatio-temporal factors [21, 22] and even demographics (such as gender and age) [23].

DOI: 10.1002/dac

Among them, the authors considered some spatial contexts (user's history trajectories, current position, current direction and user's neighborhood network cells) and proposed a short-term predictor to anticipate the future location of a mobile user in cellular networks [24]. It has been found in [19] that human trajectories showed a high level of temporal and spatial regularity, and human movements are far from random. Each individual has a time independent characteristic length scale and a significant probability to return to a few highly frequented places. In [20], the authors used temporal factors, which significantly impacted randomness, size and probability distribution of people's movements to make simple prediction models for users' visited places. The work conducted in [21] explored the influence of the temporal and spatial dimension for the analysis of complex networks extracted from mobility data. A spatio-temporal mobility model has been proposed in [22], which extended a purely spatial Markov mobility model to effectively tackle the identification problem.

In addition, the mobility prediction with low resource consumption is gaining more and more attentions. Rodriguez-Carrion [25, 26] assessed three LZ-based algorithms by separating each algorithm into two independent phases (tree building and probability calculation) and further discussing hit rate and power consumption. Results demonstrated that LZ-based algorithms could learn mobility patterns and estimate the next place with low resource needs, which makes it possible to apply them on mobile devices. In [27], the authors proposed adaptive duty cycling scheme to provide contextual information about a user's mobility: the mobility prediction-based time-resolved places and paths. This efficient technique can maximize the accuracy of predicting meaningful locations with a given energy constraint.

However, having widespread wireless localization technology, such as pervasive GPS, WiFi or GSM/UMTS location estimation available for only the last few years, many factors that affect human mobility patterns remain under researched. And mobility-based methods are urgently needed by future technology [28, 29]. Apart from the previous work, we consider the effective prediction algorithms (with low resource consumption and relatively high prediction accuracy) for distinct user groups with different mobility pattern instead of using one prediction algorithm to handle all kinds of users.

3. LOCATION PREDICTION ALGORITHMS

In this section, we discuss two domain-independent prediction algorithms, the Markov family and the LZ family. These algorithms are used for predicting the next location. Once the next location is known, the history trajectory is now one symbol longer, and the predictor updates its internal tables in preparation for the next prediction.

3.1. The Markov family

The order-k Markov predictor is independent of time, and it assumes that the current location depends only on the previous k movements. If the user's history consists of $L = a_1 a_2 \dots a_n$, let the substring $L(i,j) = a_i a_{i+1} \dots a_j$ for any $1 \leqslant i \leqslant j \leqslant n$. Then, consider the user's location as a random variable X. Let $X(i,j)$ represent the sequence of random variables $X_i, X_{i+1}, \dots X_j$ for any $1 \leqslant i \leqslant j \leqslant n$. Then, for all $a \in A$ (A is the set of all possible locations) and $i \in 1, 2, \dots, n$, we have

$$\begin{aligned} P(X_{n+1} = a | X(1,n) = L) &= P(X_{n+1} = a | X_{n-k+1} = a_{n-k+1}, \dots, X_n = a_n) \\ &= P(X_{i+k+1} = a | X_{i+1} = a_{n-k+1}, \dots, X_{i+k} = a_n), \end{aligned} \tag{1}$$

where the notation $P(X_i = a_i | \dots)$ denotes the probability that X_i takes the value a_i given the previous k observations. We can obtain these probability values by a transition probability matrix M. Both the rows and columns of M are indexed by length-k strings from A^k so that $P(X_{n+1} = a | X(1,n) = L) = M(s, s')$ where s is the string $a_{n-k+1} a_{n-k+2} \dots a_n$, the current k locations, and s' is the string $a_{n-k+2} a_{n-k+3} \dots a_n a$, the next k locations. As a result, M could provide the probability for each possible next symbol of L.

DOI: 10.1002/dac

Because we do not know M, let $N(s', s)$ donate the number of times that substring s' occurs in the string s. For each $a \in A$, we can estimate the M by the following equation

$$P(X_{n+1} = a|L) = \frac{N(a_{n-k+1} \dots a_n a, L)}{N(a_{n-k+1} a_n, L)}. \tag{2}$$

Here, the order-k Markov predictor predicts the symbol $a \in A$ with the maximum probability $P^{(}X_{n+1} = a|L)$. Note that if the next location q has never occurred in the history, the order-k Markov predictor makes no predictions.

3.2. *The LZ family*

The algorithms of LZ family are often used for text compression; they are able to make real time predictions and do not need many resources. We focus on three algorithms of LZ family: LZ, LeZi Update and Active LeZi. These three algorithms are domain independent. Like order-k Markov predictor, the algorithms of LZ family consider each location as an independent symbol. The history locations that the user visited are made of a symbol string. In order to predict the next location, the algorithms of LZ family build a tree to record the frequency of occurrence for each kind of user's mobility patterns. Different algorithms of LZ family build non-identical trees based on distinct rules. The prediction results vary with different trees, so does the total consumption of resources during the prediction.

3.2.1. LZ algorithm. Let r be the empty string and L the input movement history. The LZ algorithm partitions the L into substrings $s_0 s_1 \dots s_m$ such that $s_0 = r$, for all $j \geqslant 1$ the substring s_j without its last character is equal to some previous s_i, for all $0 \leqslant i < j$. The partitioning is performed sequentially, so that when each s_i is determined, the algorithm considers only the remainder of the input string. As for the movement history $L = abcdacbdacadef$, L is divided as follows: $a, b, c, d, ac, bd, aca, de, f$. Then we build an LZ tree to describe the algorithm. LZ tree is dynamically growing during the partitioning process. Each node of the LZ tree represents a substring s_i, and the statistics are stored at each node to mark the number of times the substring appeared, as shown in Figure 1.

In order to predict user's next location, we calculate the probability for each known substring. Vitter and Krishnan [30] considered the generator L as a finite-state Markov source, and the next symbol is only dependent on its current state. The approach for estimating the probability can be expressed as

$$P(X_{n+1} = a|L) = \frac{N^{LZ}(s_m a, L)}{N^{LZ}(s_m, L)}, \tag{3}$$

where $N^{LZ}(l, L)$ denotes the frequency of the substring l occurs in the LZ tree, and the LZ algorithm chooses the symbol with the highest probability as the prediction results of next location. The

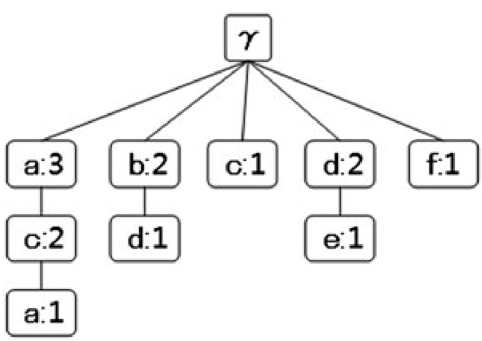


Figure 1. LZ tree for the example movement history $L = abcdacbdacadef$.

same as order-k Markov predictor, LZ algorithm is not able to make any prediction when s_{m+1} did not occur before. What is more, when we build an LZ tree, the patterns contained within substrings and between two partitioned substrings are lost.

3.2.2. LeZi Update algorithm. Bhattacharya and Das [12] improved LZ algorithm by adding the partition substring and all the suffixes of each substring to LeZi Update(LZU) tree. Therefore, the patterns contained within substrings can be captured and used for prediction. As a result, we could divide the movement history L as follows: $a, b, c, d, ac\{c\}, bd\{d\}, aca\{ca, a\}, de\{e\}, f$. Here, the substrings outside the brackets are the partition results of LZ tree and the ones inside the brackets are the new patterns that LZU tree introduced.

LeZi Update algorithm uses Prediction by Partial Matching (PPM) [31] to calculate the probability for each known substring. PPM only considers the last substrings outside the brackets as the prediction context (the longest substring that starts by the last symbol of L). Let k be the length of this prediction context, $s_m(k)$, and we have the following recursive equation:

$$P(X_{n+1} = a) = P_k(a) = P(a|s_{m(k)}) + P(esc|s_{m(k)}) \cdot P_{k-1}(a), \tag{4}$$

where $P(a|s_{m(k)})$ is the probability of a (the next symbol) given $s_{m(k)}$, and $P(esc|s_{m(k)})$ is the escape probability (the probability of $s_{m(k)}$ that is not followed by any symbol). In addition, the probabilities of every lower orders ($s_{m(k-1)}$ up to $s_{m(0)}$) need to be calculated. Here, the lower order of $s_{m(k)}$ is $s_{m(k-1)}$, which is the substring $s_{m(k)}$ removing the last symbol, and the length of $s_{m(k-1)}$ is $k-1$. In addition, LeZi Update algorithm applies the exclusion technique, which means that it only considers the number of times the substring $s_{m(k)}a$ (instead of the substrings start with $s_{m(k)}a$) occurs when counting the pattern.

Comparing with LZ algorithm, LeZi Update algorithm extracts more information from user's history movement and takes into account more patterns during the probability estimations process.

3.2.3. Active LeZi algorithm. Gopalratnam [13] further improved the LeZi Update algorithm by using a variable length window to obtain the consecutive parsed patterns of user's history movement and build the Active LeZi (ALZ) tree. The length of the window will increase if the longest pattern parsed by LZ algorithm at the current step is longer than the last step. And all the suffixes for substring at each step are also added to the ALZ tree. As last, the movement history L can be divided as follows: $a, b, c, d, a, ac\{c\}, cb\{b\}, bd\{d\}, da\{a\}, ac\{c\}, aca\{ca, a\}, cad\{ad, d\}, ade\{de, e\}, def\{ef, f\}$. Here, the ones inside the brackets are the suffixes of substring at each step.

Active LeZi uses PPM algorithm to calculate the probability, but it does not apply the exclusion method. For example, Active LeZi considers the number of times $s_{m(k)}a$ occurs plus the number of times $s_{m(k)}a$ occurs as a prefix among the substrings $s_0 s_1 \ldots s_m$ when counting the pattern.

4. EXPERIMENT

In this section, we evaluate the Markov and LZ-based prediction algorithms. We want to find out which algorithm performs better in prediction of the next location in the LTE network.

Firstly, we introduce the data set by a brief overview of 4G cellular data network architecture and how the data is collected from network. Secondly, the method for extracting mobility trajectories from traffic data is also illustrated. Then, we evaluate Markov and LZ prediction algorithms from the prediction accuracy and resource consumption point of view. Finally, we provide qualitative and quantitative comparative assessments of Markov and LZ-based prediction algorithms and schemes in the literature.

4.1. Data collection

In this paper, the traffic data are from a large Chinese 4G service provider collected from October 10, 2013 to October 31, 2013. The high level view of an LTE mobile network with traffic data

DOI: 10.1002/dac

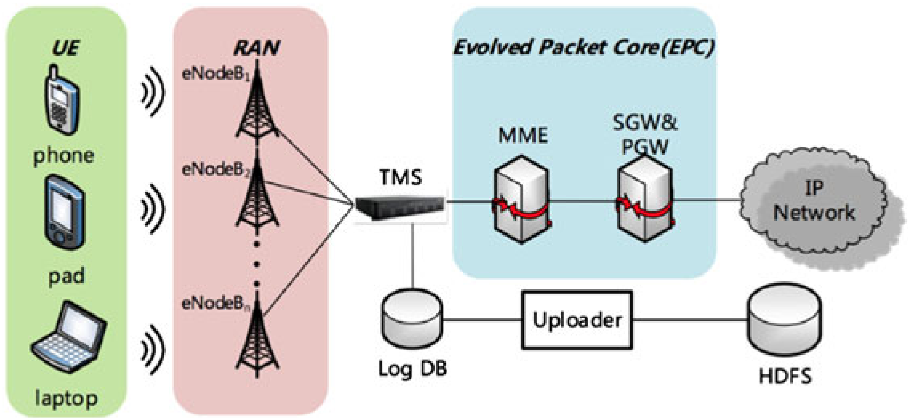


Figure 2. LTE network architecture. User Equipment (UE), Radio-Access Network (RAN), Evolved Packet Core (EPC), Traffic Monitoring System (TMS), Mobility Management Entity (MME), Serving GateWay (SGW & PGW) & Packet Data Network (PDN) GateWay, DataBase (DB), Hadoop Distribute File System (HDFS)

capture device is shown in Figure 2. There are three major components in the LTE mobile network: the User Equipment (UE), Radio-Access Network (RAN) and Evolved Packet Core (EPC).

(i) UE is any device used directly by an end-user to communicate. It can be a smart phone, a tablet, a laptop computer equipped with a mobile broadband adapter or any other device;
(ii) RAN establishes the connection between UE and EPC. It uses a flat architecture with multiple evolved Node Bs, which is like a base transceiver station in GSM networks. The eNodeB is the hardware that is connected to the mobile phone network that communicates directly with mobile handsets (UEs);
(iii) EPC is a packet-only core network. The EPC will serve as the equivalent of LTE Service networks via the Mobility Management Entity (MME), Serving GateWay (SGW) and Packet Data Network (PDN) GateWay subcomponents.

As mentioned earlier, the data sets used in this study are collected by our Traffic Monitoring System (TMS) and this device has been deployed in the production networks by several Internet Service Providers (ISPs) for traffic monitoring purposes, which is deployed between eNodeBs and MME.

LTE control-plane data has been collected from a large Chinese ISP that owns a large city area network in Southern China. The collected data set is composed of a sequence of time-stamped records, each of which contains current service eNodeB Internet Protocol (IP), signaling procedure Code, International Mobile Equipment Identity (IMEI) and so on. After extracting eNodeB IP-based location trajectories of every individual during the Evolved Packet System (EPS) Connection Management (ECM)-CONNECT state, we associate the connection establishment produce, path switch procedure and handover procedure with connection release procedure of a certain user in huge volume of data set. The data is stored in a DataBase (DB) and periodically uploaded by an Uploader to Hadoop Distribute File System (HDFS) [32].

For the security reason, users' privacy information is replaced by a hashed number, which could be used for marking subscribers, without affecting the usefulness of our analysis.

4.2. Mobility trajectories

In this paper, we use the longitude and latitude of eNodeBs to define users' location. The information of the eNodeBs users access can provide rough location data of the users, which, however, is sufficient for capturing people's daily movement patterns in a large city area.

While a UE is in active state, its location is known by the LTE network. However, while the UE is in idle state, its location is known by the LTE network every time that UE send a tracking area update request.

DOI: 10.1002/dac

When a user accesses to LTE network, the locations of eNodeBs the user attach are marked as his or her trajectory. If the user does not access to the LTE network, UE in state Evolved Packet System (EPS) Mobility Management (EMM)-REGISTERED shall initiate the tracking area updating procedure by sending a tracking area update request every 12 min. That is to say, we can obtain the real movement of a user when he or she accesses to the LTE network. However, when he or she does not connect to the LTE network, we can only update the user's current location every 12 min.

We select at random 200 trajectories (users) among 3474 trajectories. The average online time for these 200 users is 91 h. There are 729 eNodeBs in the area. On average, users visited 48 eNodeBs in 20 days and the most active user visited 249 eNodeBs. Note that those who only visited only one eNodeB in 20 days are excluded from our data set, because the position prediction for those users is meaningless.

Here, we consider only the changing of location (the different eNodeB) when extracting the trajectories of users, for example, we record the location of the eNodeB only when the user switches to another eNodeB, and the locations of eNodeB that the user attached from the trajectories. The location of an eNodeB is considered as the next movement of the user only when user switches to that eNodeB, and no new location is added to the trajectories unless the user switches to another eNodeB. In this case, no matter how long the time that the user stays in one eNodeB, the trajectories of this user would not change as long as he or she does not switch the eNodeB.

Note that the time that the user switches to the next eNodeB is also recorded. Therefore, temporal features of users' movement are available.

4.3. Performance evaluation

This section offers methodology and background information regarding our experimental evaluations. In particular, we discuss the method by which we gauge model accuracy, the comparison of prediction results between different kinds of trajectories, and the resource consumption for different algorithms.

The results in this section are obtained with two Intel Xeon E5620 2.40 GHz CPU (eight-core for each CPU) with four 4 GB 1333 MHz memory and 4 TB hard disk computer.

4.3.1. Prediction accuracy. The most common statistical metric for evaluating Markov and LZ algorithms to predict the users future movements is the probability of correct predictions. For each individual, if the size of symbolic locations is m, we use the first n locations to predict the $n+1$ location ($1 < n < m$). Note that, Markov and LZ algorithms will inevitably encounter the situations that they are unable to make a prediction. If the predictor returns 'no prediction', it is counted as an erroneous prediction. In addition, in this section, we consider only the first 1500 moves of each

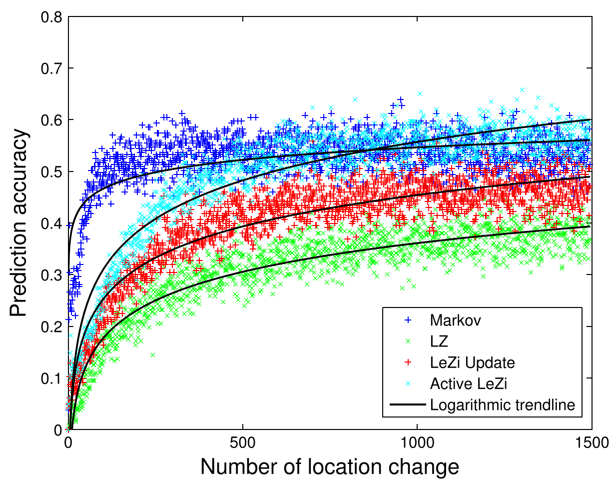


Figure 3. The average accuracy of each algorithm for each location varies over the course of user's history.

DOI: 10.1002/dac

user's history movement because it is enough to capture the difference between different prediction algorithms and posses more details regarding the changing of prediction accuracy.

We run each user's trace independently with Markov, LZ, LeZi Update and Active LeZi algorithms, respectively. For each location in the trace, the algorithm makes a prediction about the next location. Firstly, we define the accuracy of each predictor for each location to be the fraction of users for which the algorithm correctly identified the next location. Then we obtain the average prediction accuracy for each predictor at each location. Figure 3 shows how the prediction accuracy varies over the course of users history for each prediction algorithms.

In Figure 3, we can see clearly that the accuracy of Markov start growing quickly but stop increasing so fast at a lower level, achieving slightly lower accuracy than Active LeZi in the end. And the accuracy of LZ and LeZi Update algorithms increases relatively slowly during the whole prediction process.

In Figure 4, we draw that the prediction accuracy varies over the course of one user's history. We want to study the difference between different prediction algorithms for each location, for example, the fraction of correct prediction from the first move to the current move for a particular user. Thus, for each location, the accuracy of user is the number of actual next location matches for history movement divided by the total number of location changes. It is observed that LZ predictor is the worst, and Active LeZi predictor beats Markov predictor after the number of user's history move is more than 800.

In addition, we further study the distribution of prediction accuracy in each move (the average prediction accuracy from the first move to 1500 moves for each user) for each prediction algorithm, as shown in Figure 5. Obviously, Markov algorithm outperforms others that 85.5% of users achieved 50% accuracy, and in the case of LZ, LeZi Update and Active LeZi algorithms, the percentage of users is 0%, 8.5% and 79%, respectively.

The size of the user's history movement is directly linked to prediction accuracy, especially when the historical trajectory is very short, which can be observed at the beginning of predication in Figure 3. Limiting the size of the user's movement history leads to loss of information and results in false predictions. As it can be observed in Figures 3, 4 and 5, if user's movement history is relatively short, Markov algorithm shows the best ability to predict. The Active LeZi algorithm outperforms others after a while because enough consecutive parsed patterns could be obtained when the ALZ tree is big enough (built from more than 800 history movements in our case).

4.3.2. Resource consumption. Mobile phone resources can be classified into computation, memory, and input and output (I/O) to storage. A prediction algorithm can run slowly if any of these resources perform badly. High CPU utilization is the result of intensive computation, and it tends to consume

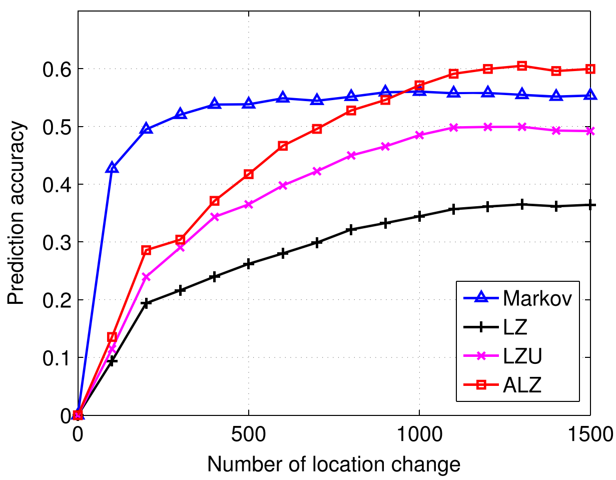


Figure 4. Comparing the accuracy of Markov with that of LZ, LeZi Update and Active LeZi algorithms for a sample user.

DOI: 10.1002/dac

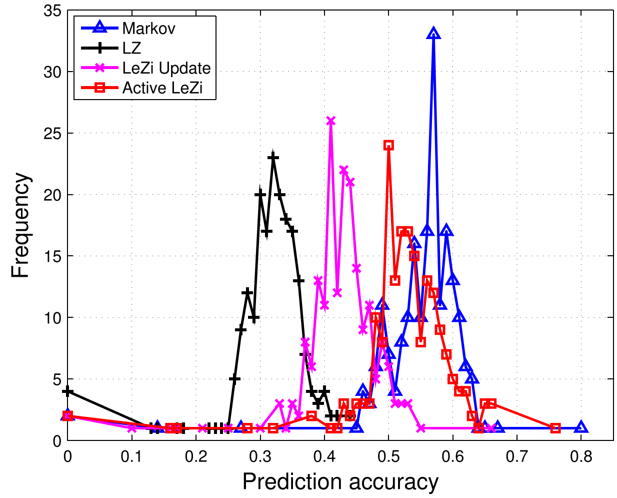


Figure 5. The distribution of prediction accuracy of Markov, LZ, LeZi Update and Active LeZi algorithms for a sample user.

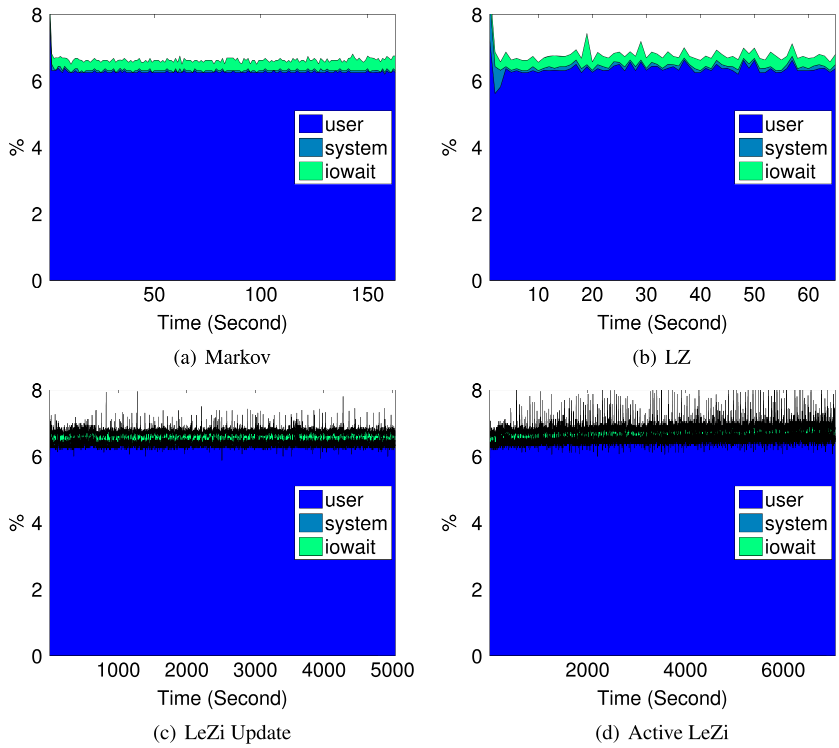


Figure 6. CPU utilization for each prediction algorithms.

more power. The amount of memory available on the prediction is another potential bottleneck that can have significant effects. What is more, disk I/O throughput can become a bottleneck if there are too much data to be analyzed. Our goal is to consume as few resources as possible and spend as little time as possible during the prediction.

Note that the performances of different mobile phones with different brands, different models or different battery capacities are very different. We have not tested the power expenditure of prediction algorithm on all kinds of mobile phone. However, the results in this section can still be a reasonable reference for the service provider.

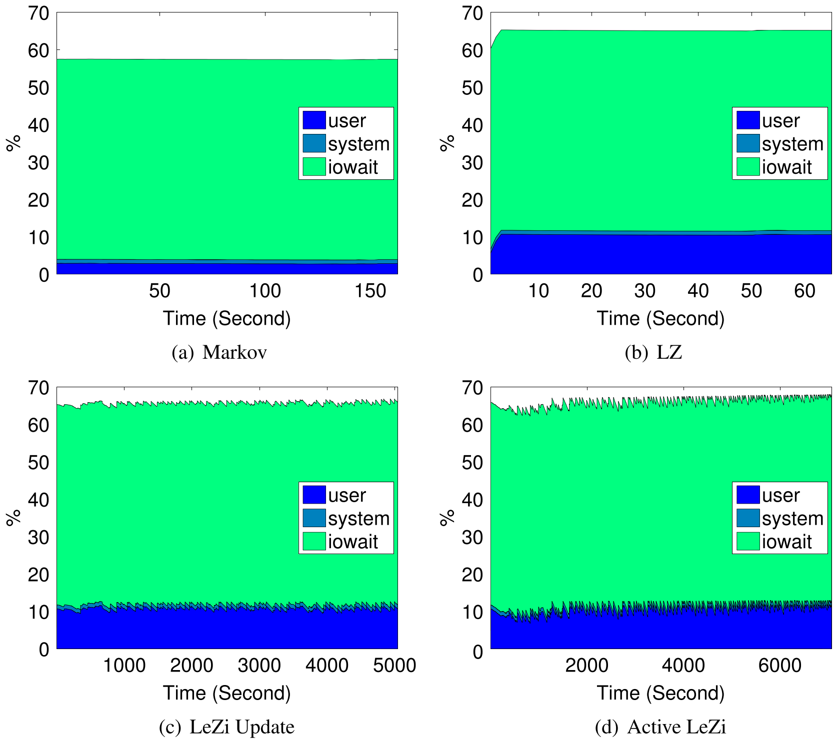


Figure 7. Memory utilization for each prediction algorithms.

Table I. Some metrics for a sample user.

Prediction algorithm	The number of nodes	Time	Quantified CPU	Prediction accuracy
Markov	–	1.39 s	24.55	0.553
LZ	433	1.01 s	9.91	0.364
LeZi Update	1496	16.19 s	766.44	0.492
Active LeZi	3194	32.04 s	1097.18	0.6

In order to examine the power expenditure of the Markov family and the LZ family, we examine the resource utilizations of each prediction by collecting the quantitative results from executing the prediction algorithms. In our case, we consider only two resources, including CPU (user, system and iowait) and memory (used, buffer and cache). Disk I/O utilization would not become a bottleneck in that there would not be many reading and writing operations. Figures 6 and 7 show the timeline-based CPU and memory utilizations of each prediction algorithms during the predicting process for 200 users.

Markov algorithm consumes many CPU resources and very little memory resources during prediction. For LZ-based algorithms, the whole prediction process could be split into two phases–tree building and probability calculation. The first one, tree building process, takes care of updating the pattern tree. Hence, this process is tightly coupled with memory consumption. The second step, which is in charge of calculating probabilities, depends on the complexity of the method used. However, these two phases cannot be clearly distinguished by CPU and memory resources consumption. As we can observe in Figures 6 and 7, CPU resources are the main bottleneck for all prediction algorithms. Note that prediction process only uses one CPU core of 16 CPU logical cores. Therefore, as for a multi-core phone, parallel computing of prediction algorithms may improve the computational efficiency.

DOI: 10.1002/dac

In addition, we further quantify the resource consumption of CPU during the prediction by sampling the CPU usage in clock cycles of a prediction job. The total CPU usage in clock cycles of the prediction algorithm is calculated as

$$cpu = f_{clock} \times \sum_{i=1}^{N} C_i, \tag{5}$$

where f_{clock} is the CPU clock frequency, N is the number of seconds that the prediction algorithm spent and C_i is the value of CPU utilization at ith second during the running of prediction algorithm. The total CPU usage among with some other metrics of each prediction algorithm for a sample user is shown in Table I.

Here, all the metrics in Table I are calculated when the predictions of 1500 history locations are completed. Overall, LZ algorithm is the most energy saving predictor, and the Markov algorithm achieves the best prediction accuracy when we consider the whole 1500 history locations. The LZ-based algorithms build the LZ, LZU or ALZ tree during the prediction. That is no doubt that the ALZ tree has the largest number of nodes because the patterns captured by Active LeZi algorithm is much more than that of LZ and LeZi Update algorithms. Therefore, Active LeZi algorithm spends the longest time on prediction and consumes the most CPU resources among others.

As a result, Markov algorithm seems to be a better choice when predicting the next location in LTE network. When the ALZ tree is big enough (built from more than 800 locations in our data set), Active LeZi algorithm can achieve the best prediction accuracy among others, yet the resource consumption is too much.

In conclusion, we can obtain the best trade off in the following way.

(1) Obtain the encounter value, for example, the value of y-axis (number of location change) when the logarithmic trendlines for Markov algorithm and Active LeZi algorithm encounter with each other.
(2) When the number of history locations is smaller than the encounter value, Markov outperforms other prediction algorithms in resource consumption and prediction accuracy.
(3) When the number of history locations is smaller than the encounter value, Markov is still the better choice because of the low resource consumption. However, user can also apply the Active LeZi algorithm to obtain the better results on the cost of high resource consumption.

5. PREDICTION METHOD FOR INDIVIDUAL

In the previous section, Markov prediction algorithm outperforms the algorithms in LZ family. In order to improve the prediction accuracy without increasing too much computational complexity, we further take the user mobility characteristics and temporal factors into account during the prediction. In this section, we group users according to their temporal and spatial characteristics. Our goal is to study the difference between users' movement pattern and predict not only the user's next location but also the place he or she most likely visits within a specific period of time with relatively low resource consumption.

5.1. User groups

Intuitively, different users tend to have different routine pattern and a user's current location should correlate with time. For example, workers usually follow daily pattern in the weekdays and move more randomly in the weekends. For travelers, their visiting locations are more random. Hence, some users' movement patterns are more predictable while others are difficult to predict. In order to identify user groups according to their temporal and spatial characteristics, we discretize day into 24 time segments, each segment lasts 1 h long, as shown in Table II.

Here, we consider only weekdays and weekends instead of considering each day of the week (Monday, Tuesday, ... , Sunday) separately. More specific, segment 0 in weekdays contains the time segment between 0:00–0:59 a.m. of Monday to Friday and segment 23 in weekends is the time segment between 11:00–11:59 p.m. of Saturday and Sunday.

DOI: 10.1002/dac

Table II. Time segment of the corresponding time interval.

Time segment	Time interval
0	0:00 ∼ 0:59
1	1:00∼1:59
2	2:00∼2:59
...	...
22	22:00∼22:59
23	23:00∼23:59

Then we record the total duration each user spends at a certain location for each time segments in weekdays and weekends, respectively. If a user follows the similar movement pattern (go to work in the morning and go home at night) in weekdays, we can identify the user's most significant locations for each time segment, yet there have been no long time staying location for users with random pattern. In order to describe the diversity of users' movement patterns, we use entropy as the metric. Entropy is probably the most fundamental quantity capturing the degree of predictability characterizing a time series, which is defined as follow:

$$H(X) = \sum_{i=1}^{n} p(x_i) I(x_i) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i), \tag{6}$$

where n is the number of different locations user visited in one time segment, each different i represents different locations the user visited in one time segment, and b is a constant value. $p(x_i)$ is the probability for user staying in a certain place in one time segment, and

$$p(x_i) = \frac{total\ time\ duration\ user\ staying\ in\ location\ i\ in\ current\ time\ segment}{total\ time\ duration\ user\ staying\ in\ current\ time\ segment}. \tag{7}$$

The bigger the entropy value is, the more locations the user visits in current time segment, for example, the more the uncertainty for this user's being at a specific location at a certain time. In contrary, a small entropy value implies that the user visits very limited number of locations in current time segment. For each user, we build two entropy vectors: the entropy value for each of 24 segments in weekdays and weekends, respectively.

$$E_{weekday} = [e_{weekday}/segment_0, e_{weekday}/segment_1, \ldots, e_{weekday}/segment_{23}] \tag{8}$$

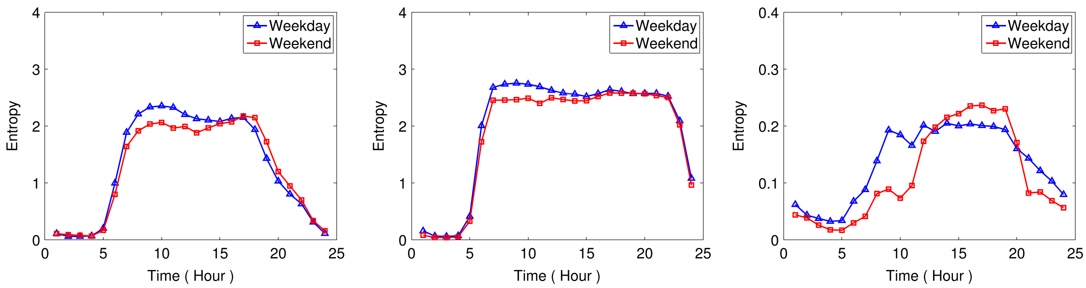
$$E_{weekend} = [e_{weekend}/segment_0, e_{weekend}/segment_1, \ldots, e_{weekend}/segment_{23}] \tag{9}$$

Considering the correlation between location and time, we group users by clustering them with k-means clustering. In our case, k is empirically set to be equal to 3 to distinguish main group with distinct mobility pattern. We give each cluster a label, which intuitively describes the properties users' mobility pattern as shown in Figure 8. Here, we use the entropy profiles to describe the diversity of users movement patterns varying with time.

(i) Normal post person: during the day he or she spends time at various locations, and after 6 p.m., the entropy obtains lower with the decreasing of user's activity;
(ii) Hard post person: during the day he or she spends time at various locations, and entropy is relatively higher than other types of person, yet he or she would not rest until 10 p.m. at night;
(iii) Family person: his or her entropy is low in that he or she visits limited location during the day.

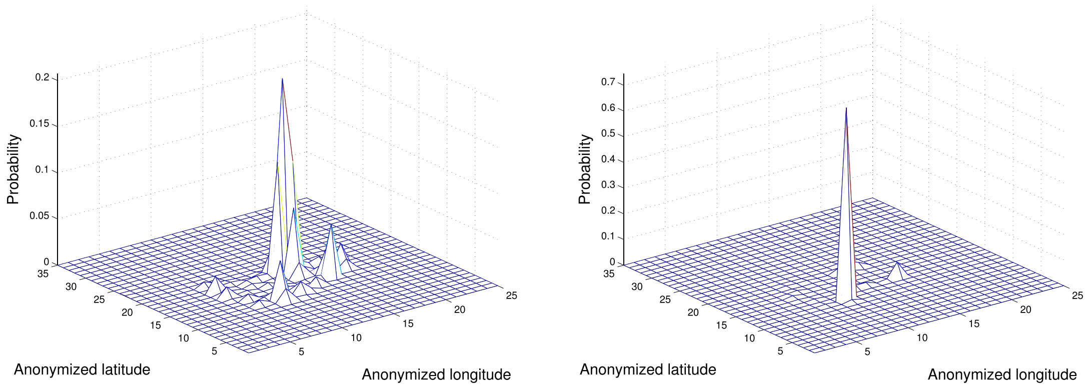
In Figure 8, we can clearly see that the difference of entropy profiles for each type of user in weekdays and weekends is not quite significant.

DOI: 10.1002/dac

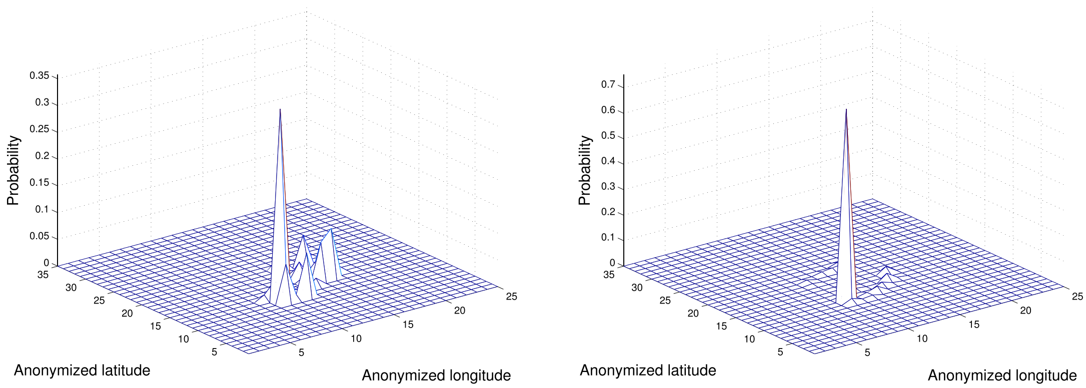


(a) Normal post person (b) Hard post person (c) Family person

Figure 8. The entropy profiles of three different types of users.



(a) Work hours of weekday (b) Non-work hours of weekday

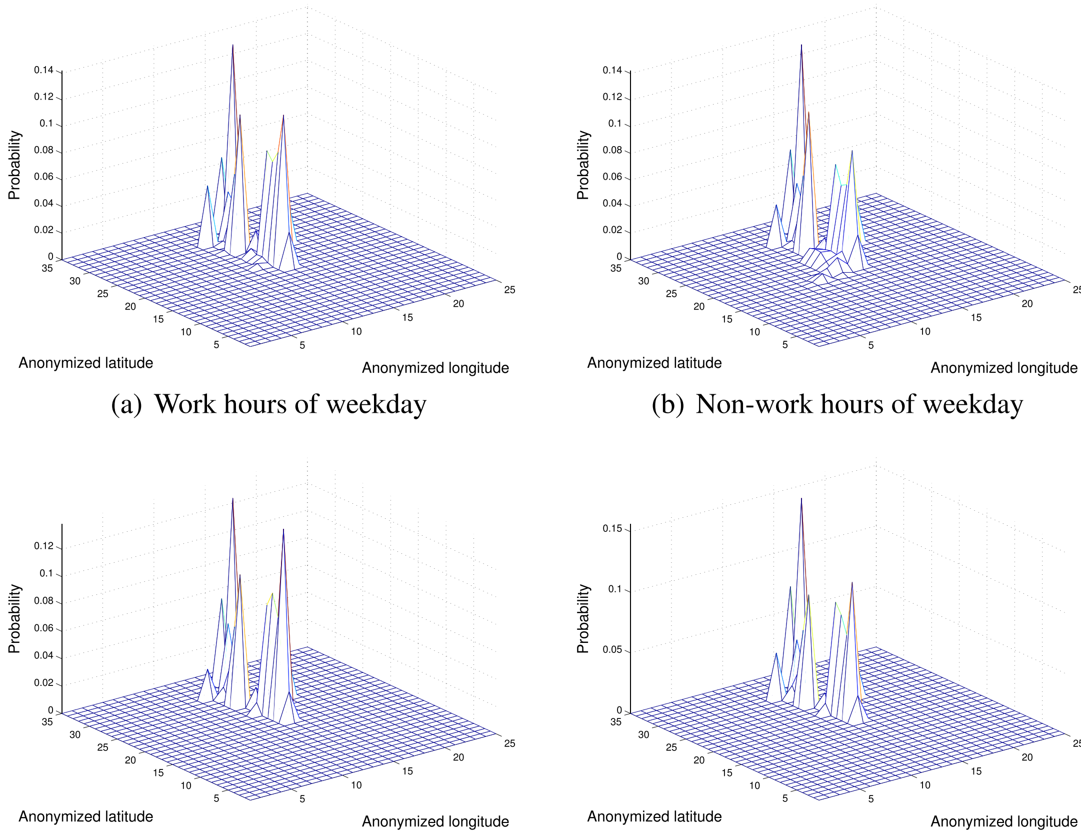


(c) Work hours of weekend (d) Non-work hours of weekend

Figure 9. Spatial probability distribution of a normal post person's locations.

5.2. Spatial probability distribution

In order to further illustrate the mobility pattern for each type of users in different time frame, we divide the data into four sample sets based on their time interval: work hours of weekdays (8 a.m. to 5 p.m., Monday–Friday), non-work hours of weekdays (5 p.m. to 8 a.m., Monday–Friday), work hours of weekends (8 a.m. to 5 p.m., Saturday and Sunday) and non-work hours of weekends (5 p.m. to 8 a.m., Saturday and Sunday). We calculate and compare the two-dimensional probability distribution of three types of users' locations for these four time intervals. The spatial (two-dimensional) probability distribution of three sample users' location is depicted in Figures 9 (normal post person), 10 (hard post person) and 11 (family person).



(a) Work hours of weekday

(b) Non-work hours of weekday

(c) Work hours of weekend

(d) Non-work hours of weekend

Figure 10. Spatial probability distribution of a hard post person's locations.

Individuals display significant regularity, as they return to a few highly frequented locations, as home or work. In Figure 9, a sampled normal post person, during the work hours he or she spends time at various locations, likely starts from places close to office in the morning then moves around and then comes back to office. During the non-work hours, he or she spends much of his or her time at home. He or she stays at home or moves not far from the home on weekends. Yet, there is no clear temporal pattern for the sampled hard post person in Figure 10. In addition, the sampled family person in Figure 11 has the most regular pattern (go to the office during work hours, stay at home during non-work hours and work overtime at office on the weekends).

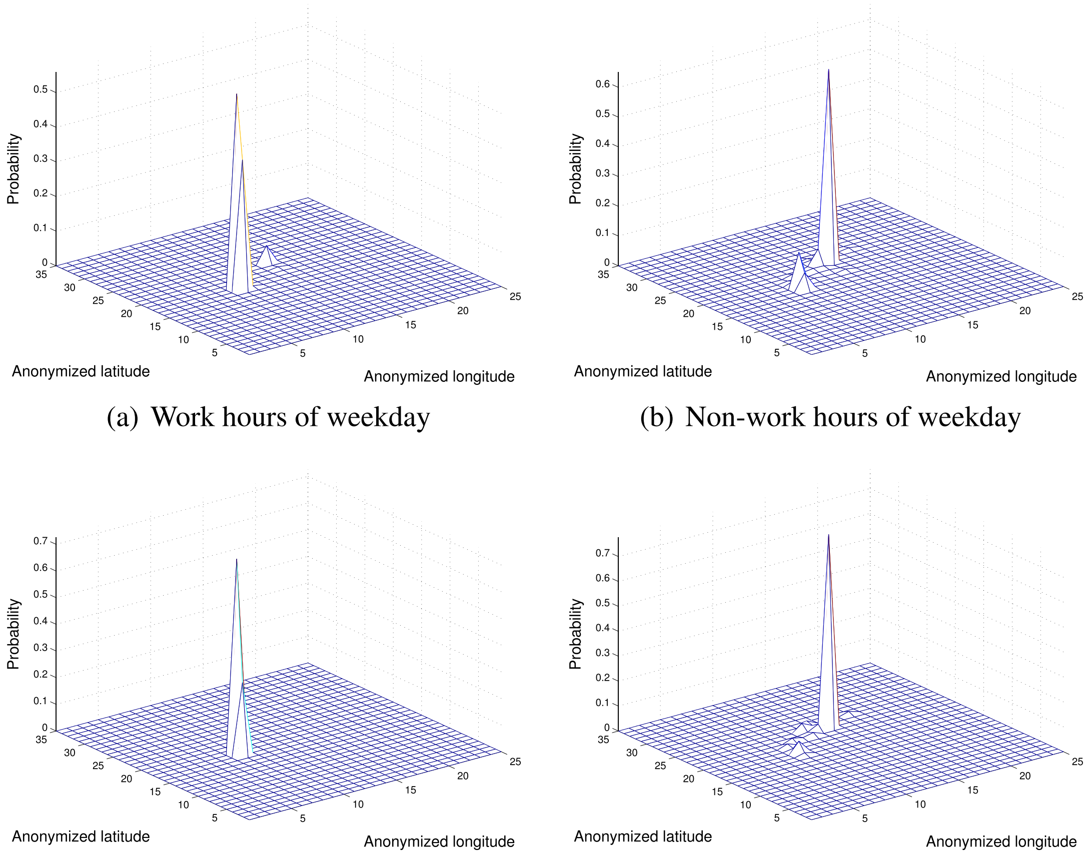
Among distinct types of persons, mobility pattern varies during different periods in a day. This spatial distribution is the characteristic that models a user's mobility and can be used for the location prediction. This feature changes significantly between the two temporal conditions (work hours and non-work hours) and quantifies the probability of user's presence at home, office and some other locations. Furthermore, family person shows a more gather distribution of the user's presence during different time frame.

5.3. *Spatio-temporal based prediction*

After investigating the spatial and temporal features of different groups of users with distinct mobility pattern, in this section, we try to take temporal features into account to improve the accuracy of prediction for different groups of users with keeping the task of location prediction simple and avoiding highly computational complexity at the same time. We want to predict not only users' next location but also the place he or she most likely being within a specified period.

We further calculate location entropy and radius of gyration [19] for family person, normal post person and hard post person, as shown in Table III.

DOI: 10.1002/dac



(a) Work hours of weekday
(b) Non-work hours of weekday
(c) Work hours of weekend
(d) Non-work hours of weekend

Figure 11. Spatial probability distribution of a family person's locations.

Table III. Some metrics for measuring the predictability of family person, normal post person and hard post person.

User type	Location entropy	Radius of gyration
Family person	0.85	0.21
Normal post person	2.98	0.36
Hard post person	2.77	0.44

When the location entropy becomes smaller, the uncertainty that user appears at one location at a specific time is also lowered. It is the same for the radius of gyration, which measures the size of the footprint of a user. Here, we employ a spatio-temporal based location prediction method. Firstly, we obtain the probability distribution of visited location for each segment (e.g., discretize day into 24 time segments, each segment lasts 1 h long, as shown in Table II). Then the location with highest probability in a particular segment is the most probable location in that segment for the next day. We use 10 day-long history trajectory to predict the most likely place he or she will be for each hour of next day. Finally, family person achieves 83.3% (4 wrong prediction and 20 right prediction for 24 segments in a day) accuracy.

In addition, we further study the credibility of the spatio-temporal based prediction method. We obtain the most probable location according to the probability distribution of each segment for the first 10 days. Then we calculate the probability distribution of each segment for the next 10 days. If we have a as the most probable location in segment i in the first 10 days, then we can obtain the probability that user visits a in segment i in the next 10 days, as shown in Figure 12.

DOI: 10.1002/dac

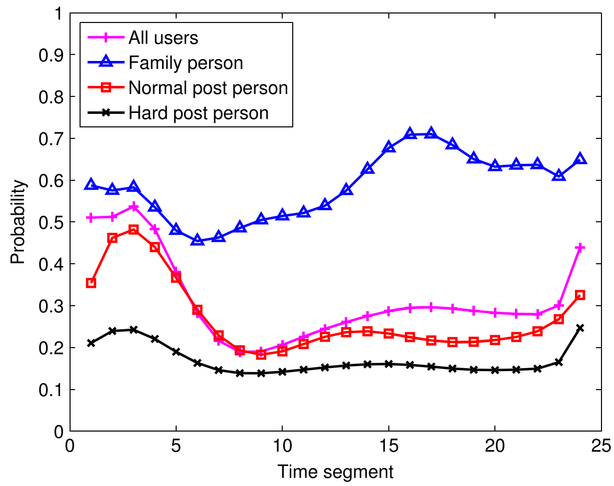


Figure 12. The probability that user visits the prediction location in each segment.

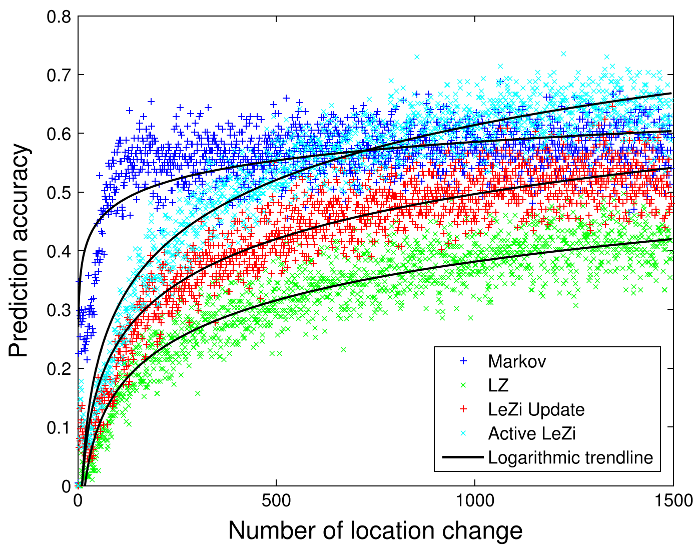


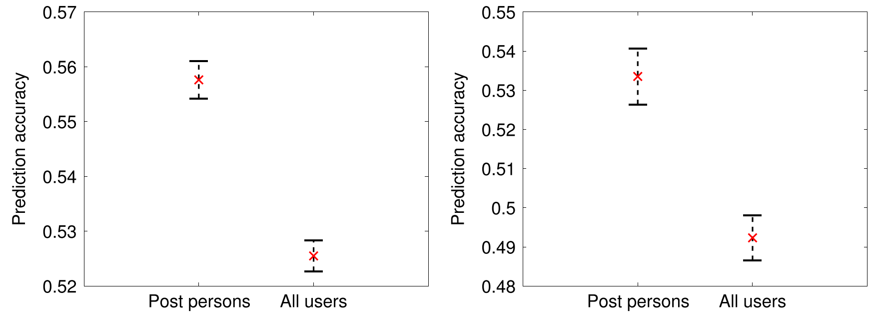
Figure 13. The average accuracy of each algorithm for each location varies over the course of post person's history.

Obviously, for those who are in the two-line life of 'at work' and 'home' (family person), their movements are more predictable than others, and the spatio-temporal based prediction method works well. Family person tends to stay in the same location (office) between 4 p.m. and 5 p.m. in the afternoon. The most probable location for the segments between 5 a.m. and 9 a.m. is very hard to predict when he or she is on the way to the office. However, as for normal post person and hard post person, the spatio-temporal based prediction method goes worse.

In Figure 12 and Table III, we can clearly see that the differences between family persons and post persons (both normal post persons and hard post persons) are bigger than that of normal post persons and hard post persons. As a result, we further applied Markov family and LZ family to 200 randomly selected post persons, in order to examine the prediction accuracy of these algorithms, as shown in Figure 13.

Comparing the logarithmic trendline of Figure 13 with Figure 3, the prediction accuracy at 1500th location for post person is higher than that of all users. We further calculate the statistical characteristics for post persons and all users, as shown in Figure 14.

DOI: 10.1002/dac



(a) Markov (b) Active LeZi

Figure 14. The 95% confidence intervals and the mean values of prediction accuracies for post persons and all users.

Figure 14 illustrates the 95% confidence intervals (the horizontal line) and mean value (the red cross between two horizontal lines) of prediction accuracies. Comparing post persons with the total users, the mean value of prediction accuracy for Markov algorithm is improved from 0.5255 to 0.5576, and in the case of Active LeZi algorithm, the mean value increased from 0.4923 to 0.5335. From these statistical characteristics, we can conclude that the next location algorithms (Markov and Active LeZi) are more suitable for post persons, and users with distinct mobility pattern should be applied to different predicting methods.

In conclusion, spatio-temporal based prediction method is more suitable for family person who visits very limited locations during the day and follows regular pattern during the week. For those who spend much time on commute and follow fairly regular pattern during the week (post person), the next location prediction is more efficient.

6. CONCLUSIONS

In this paper, by using the real users' trajectories collected from LTE network in a city area of south China, we studied the simple location prediction algorithms with low resource consumption that could be applied on mobile device.

Findings

We introduced two families of domain-independent prediction algorithms, the Markov algorithm and the LZ-based algorithms to predict the next location. By applying these prediction algorithms to our data set, we found that the Markov algorithm achieves the best accuracy at the beginning of prediction, achieving a little lower accuracy than Active LeZi in the end. The Active LeZi algorithm outperformed Markov when the length of history trajectory is more than 800. And the accuracy of LZ and LeZi Update algorithm increased relatively slowly during the whole prediction process.

For resource consumption, CPU resource is the main bottleneck for all the prediction algorithms. After quantifying the resource consumption of CPU during the prediction, we found that although Active LeZi achieved the highest prediction accuracy, it spent the longest time on prediction and consumed the most CPU resources among all algorithms. When predicting the next location, Markov algorithm provided the best tradeoff between prediction accuracy and resource consumption.

Then, in order to improve the prediction accuracy, we classified users into three different groups (normal post person, hard post person and family person) according to their distinct entropy profiles and examined the mobility patterns for each type of users in different time frame. We found that spatio-temporal based prediction method achieved 83.3% accuracy for family person. However, as for post person, the next location prediction algorithms are more suitable. Comparing post persons with the total users, the mean value of prediction accuracy for Markov algorithm was improved from 0.5255 to 0.5576, and in the case of Active LeZi algorithm, the mean value increased from 0.4923 to 0.5335.

Implications

Regarding user's location prediction on mobile device, in order to obtain the tradeoff between prediction accuracy and resource consumption, we should consider applying different prediction methods to users with distinct mobility pattern. For those who visit very limited locations during the day and follow regular pattern during the week, spatio-temporal based prediction method is more effective so that it can provide not only user's next location but also the place he or she most likely visiting within a specified period with high accuracy. For those who spend much time on commute and follow fairly regular pattern during the week (post person), the next location prediction is more efficient.

Future work

Different prediction method should be applied to different users with distinct mobility patterns and spatio-temporal characteristics. Most users have the division between work hours, nights, weekends and commute hours in common. However, the hours of the day that define work hours, nights, commute hours or weekend irregularities can be different from one user to another. In the future, we will try to investigate the more effective grouping methods to accurately classify users by extracting the mobility pattern and spatio-temporal characteristics. In addition, more effective and user-based prediction algorithms with high prediction accuracy and lower resource consumption are expected.

ACKNOWLEDGEMENTS

This work was supported in part by National Natural Science Foundation of China (61072061, 61402047), Important National Science and Technology Specific Projects (2012ZX03002008), Director Foundation Project (2014BKL-NSAC-ZJ-01), 111 Project of China (B08004) and EU FP7 IRSES MobileCloud Project (612212).

REFERENCES

1. Song C, Qu Z, Blumm N, Barabási AL. Limits of predictability in human mobility. *Science* 2010; **327**(5968): 1018–1021.
2. Lu X, Wetter E, Bharti N, Tatem AJ, Bengtsson L. Approaching the limit of predictability in human mobility. *Scientific reports* 2013; **3**:2923.
3. Cheng C, Jain R, van den Berg E. *Location Prediction Algorithms for Mobile Wireless Systems: Wireless Internet Handbook*. CRC Press Inc.: Boca Raton, FL, USA, 2003.
4. Song L, Kotz D, Jain R, He X. Evaluating location predictors with extensive Wi-Fi mobility data: INFOCOM 2004. *Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, Hong Kong, China, 2004. IEEE.
5. Petzold J, Bagci F, Trumler W, Ungerer T. Comparison of different methods for next location prediction. In *Euro-Par 2006 Parallel Processing*. Springer: Berlin Heidelberg, 2006. 909–918.
6. Ma Z, Leijon A. Bayesian estimation of beta mixture models with variational inference. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 2011; **33**(11):2160–2173.
7. Ma Z, Rana PK, Taghia J, *et al.* Bayesian estimation of Dirichlet mixture model with variational inference. *Pattern Recognition* 2014; **47**(9):3143–3157.
8. Petzold J, Bagci F, Trumler W, Ungerer T. Next location prediction within a smart office building. *Proceedings of 1st International Workshop on Exploiting Context Histories in Smart Environments (ECHISE'05) at the 3rd International Conference on Pervasive Computing,* Vol. 577, 2005; 69.
9. Si H, Wang Y, Yuan J, Shan X. Mobility prediction in cellular network using hidden Markov model. *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, Las Vegas, NV, USA, 2010. IEEE.
10. Nicholson AJ, Noble BD. Breadcrumbs: forecasting mobile connectivity. *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, San Francisco, CA, USA, 2008. ACM.
11. Ziv J, Lempel A. Compression of individual sequences via variable-rate coding. *Information Theory, IEEE Transactions on* 1978; **24**(5):530–536.
12. Bhattacharya A, Das SK. Lezi-update: an information-theoretic framework for personal mobility tracking in pcs networks. *Wireless Networks* 2002; **8**(2/3):121–135.
13. Gopalratnam K, Cook DJ. Online sequential prediction via incremental parsing: the Active LeZi algorithm. *Intelligent Systems, IEEE* 2007; **22**(1):52–58.
14. Isaacman S, Becker R, Cáceres R, Martonosi M, Rowland J, Varshavsky A, Willinger W. Human mobility modeling at metropolitan scales. *Proceedings of the 10th International Conference on Mobile systems, Applications, and Services*, Ambleside, UK, 2012. ACM.

15. Misra S, Singh S, Khatua M, Obaidat MS. Extracting mobility pattern from target trajectory in wireless sensor networks. *International Journal of Communication Systems* 2013; **28**(2):213–230. DOI: 10.1002/dac.2649.
16. Nguyen L, Cheng H, Wu P, Buthpitiya S, Zhang Y. Pnlum: system for prediction of next location for users with mobility. *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated Challenge*, 2012.
17. Zheng J, Ni LM. An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data. *Proceedings of the 2012 ACM Conference on Ubiquitous Computing,* Pittsburgh, PA, USA, 2012. ACM.
18. Bayir MA, Demirbas M, Eagle N. Mobility profiler: a framework for discovering mobility profiles of cell phone users. *Pervasive and Mobile Computing* 2010; **6**(4):435–454.
19. Gonzalez MC, Hidalgo CA, Barabasi A. Understanding individual human mobility patterns. *Nature* 2008; **453**(7196):779–782.
20. Gatmir-Motahari S, Zang H, Reuther P. Time-clustering-based place prediction for wireless subscribers. *IEEE/ACM Transactions on Networking (TON)* 2013; **21**(5):1436–1446.
21. Coscia M, Rinzivillo S, Giannotti F, Pedreschi D. Spatial and temporal evaluation of network-based analysis of human mobility. In *State of the Art Applications of Social Network Analysis*. Springer International Publishing, 2014; 269–293.
22. Pham N, Cao T. A spatio-temporal profiling model for person identification. In *Knowledge and Systems Engineering*. Springer International Publishing, 2014; 363–373.
23. Lian D, Zhu Y, Xie X, Chen E. Analyzing location predictability on location-based social networks. In *Advances in Knowledge Discovery and Data Mining*. Springer, 2014; 102–113.
24. Anagnostopoulos T, Anagnostopoulos C, Hadjiefthymiades S. Efficient location prediction in mobile cellular networks. *International Journal of Wireless Information Networks* 2012; **19**(2):97–111.
25. Rodriguez-Carrion A, Garcia-Rubio C, Campo C, Cortés Martín A, Garcia-Lozano E, Noriega-Vivas P. Study of LZ-based location prediction and its application to transportation recommender systems. *Sensors* 2012; **12**(6): 7496–7517.
26. Rodriguez-Carrion A, Garcia-Rubio C, Campo C. Performance evaluation of LZ-based location prediction algorithms in cellular networks. *Communications Letters, IEEE* 2010; **14**(8):707–709.
27. Chon Y, Talipov E, Shin H, Cha H. SmartDC: mobility prediction-based adaptive duty cycling for everyday location monitoring. *IEEE Transactions on Mobile Computing* 2014; **13**(3):512–525.
28. Wu Y, Ge J, You J. An operational mobility model over IPv6: design, modeling, and evaluation. *International Journal of Communication Systems* 2014; **27**(2):341–356. DOI: 10.1002/dac.2364.
29. Cui Q, Zhang X. Research analysis of wireless localization with insufficient resources for next-generation mobile communication networks. *International Journal of Communication Systems* 2013; **26**(9):1206–1226. DOI: 10.1002/dac.2370.
30. Vitter JS, Krishnan P. Optimal prefetching via data compression. *Journal of the ACM (JACM)* 1996; **43**(5):771–793.
31. Cleary JG, Teahan WJ. Unbounded length contexts for PPM. *The Computer Journal* 1997; **40**(2 and 3):67–75.
32. Shvachko K, Kuang H, Radia S, Chansler R. The hadoop distributed file system. *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium* IEEE, Incline Village, NV, USA, 2010; 1–10.