

The Influence of Temporal and Spatial Features on the Performance of Next-place Prediction Algorithms

Paul Baumann
WSN Lab
TU Darmstadt, Germany
paul.baumann@wsn.tu-darmstadt.de

Wilhelm Kleiminger
Distributed Systems Group
ETH Zurich, Switzerland
kleiminger@inf.ethz.ch

Silvia Santini
WSN Lab
TU Darmstadt, Germany
santinis@wsn.tu-darmstadt.de

ABSTRACT

Several algorithms to predict the next place visited by a user have been proposed in the literature. The accuracy of these algorithms – measured as the ratio of the number of correct predictions and the number of all computed predictions – is typically very high. In this paper, we show that this good performance is due to the high predictability intrinsic in human mobility. We also show that most algorithms fail to correctly predict *transitions*, i.e. situations in which users move between different places. To this end, we analyze the performance of 18 prediction algorithms focusing on their ability to predict transitions. We run our analysis on a data set of mobility traces of 37 users collected over a period of 1.5 years. Our results show that even algorithms achieving an overall high accuracy are unable to reliably predict the next location of the user if this is different from the current one. Building upon our analysis we then present a novel next-place prediction algorithm that can both achieve high overall accuracy and reliably predict transitions. Our approach combines all the 18 algorithms considered in our analysis and achieves its good performance at the cost of a higher computational and memory overhead.

Author Keywords

Next-place prediction; Predictability; Human mobility.

ACM Classification Keywords

H.m. Information systems: Miscellaneous.

INTRODUCTION

Predicting human mobility has long been a topic of interest for both researchers and practitioners. Models of human mobility have been used for planning urban development, to study the spread of diseases, or to predict the number of calls a telephone station must be able to carry [21]. Recently, human mobility models have also been considered to support a plethora of new applications, like urban navigation or home automation. The ability to predict when a resident will

arrive at home allows for instance to automatically control the home's heating system [10]. Similarly, reliably predicting the next place a person will visit (e.g., the grocery or the gym) allows to provide her with convenient information (e.g., bus timetables or advertisements) [19].

Several algorithms to predict the *next place* that will be visited by a user have been presented in the literature [10, 23, 1, 2, 4, 9, 11, 19, 20]. Many of these next place prediction algorithms estimate users' future whereabouts using as input the sequence of locations visited by the users in the past. These sequences – to which we also refer to as *mobility traces* or *historical data* – can for instance be captured leveraging users' mobile phone as location sensors [12]. Coarse-grained mobility traces can also be obtained from phone call records, whereby the position of the base station to which the phone was associated during the call represents an estimate of the position of the user [21].

While different next place prediction algorithms may use different features of the historical data to compute their predictions, their performance is typically measured in terms of *prediction accuracy*. The prediction accuracy is computed as the ratio of the number of correct predictions and the number of all computed predictions. An accuracy of 80% thus indicates that in 80% of the cases the algorithm predicted the future whereabouts of the user correctly. Recent studies have shown that humans tend to spend most of their time in few important places. For instance, humans spend on average 60% to 65% of their time at home and between 20% and 25% at work (or school) [15]. Thanks to this regularity of human behavior, a naïve algorithm that predicts the next location to be equal to the last observed one will thus easily achieve average overall accuracies equal to or higher than 80%. However, the same algorithm will always make incorrect predictions when the user moves between two different locations (e.g., from home to the office or vice versa). The applications mentioned above, however, must be able to know when transitions between places will most likely take place. The behavior and performance of the application is indeed strongly influenced by the occurrence of such transitions. For instance, an automatic heating system will perform poorly if transitions of the user from and to the home are not predicted correctly most of the time.

In this paper, we show that even algorithms that achieve high prediction accuracy fail to reliably detect transitions. To sup-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UbiComp'13, September 8–12, 2013, Zurich, Switzerland.
Copyright © 2013 ACM 978-1-4503-1770-2/13/09...\$15.00.
<http://dx.doi.org/10.1145/2493432.2493467>

port this claim, we analyze the performance of 18 prediction algorithms. We run our analysis on a large data set of human mobility traces that have been recently made available to the public [12]. We thus argue that the performance evaluation of next place prediction algorithms should explicitly consider, along with the prediction accuracy, also the ability of the algorithms to predict transitions between places. Building upon the results of our analysis, we introduce a novel next place prediction algorithm that combines the predictive power of all the considered 18 approaches. We let these 18 approach run in parallel so that each computes its own prediction. Our algorithm, called MAJOR, predicts the user's next whereabouts from the 18 individual estimates using an heuristic method based on majority voting. The prediction accuracy of our algorithm is comparable to that of state-of-the-art approaches. MAJOR can however also reliably predict transitions. Furthermore, we can estimate the confidence of MAJOR's predictions.

The remainder of this paper is organized as follows. We first introduce the terminology and mathematical notation used throughout the paper and review related work. We then describe the setup of our analysis of the accuracy of next place prediction algorithms and discuss the obtained results. We then present and evaluate the MAJOR algorithm and provide some concluding remarks.

TERMINOLOGY AND NOTATION

Following the notation introduced in [8], we define *relevant places* of a user as the set of places “where the user spends a substantial amount of time and/or visits frequently” [8]. The home, the office, or the gym are typical examples of users' relevant places. We indicate with L_j , $j = 1 : N_L$, the j -th relevant place and define the set $\mathcal{L} = \{L_1, L_2, \dots, L_{N_L}\}$ as the set of N_L places relevant to user i .¹ Physical locations (coordinates) that cannot be associated to one of the N_L relevant places of the user are assigned to a symbolic “irrelevant” place L_x . For each relevant place L_j we further define the probability p_j of the user of being at place L_j at any time. This probability can either be known *a priori* or be estimated *a posteriori* from the actual mobility traces of the user. If L_1 corresponds for instance to the home of the user, $p_1 = 60\%$ implies that the user spends 60% of her time at home. Without loss of generality we assume that $p_1 > p_2 > \dots > p_{N_L}$, i.e., L_j is the (a priori) j -th most visited place of the user. Please note that a “naïve” prediction algorithm that always returns L_1 as the next location of the user achieves a prediction accuracy equal to p_1 .

We model the place at which the user is at time t as a variable $X(t)$ that can take values in the set \mathcal{L} . We refer to a *continuous-time next-place prediction algorithm* as an algorithm that can compute an estimate of the place at which the user will be at time $t + \delta$, $\forall \delta \in \mathbb{R}^+$. This estimate is indicated as $\hat{X}(t + \delta)$, whereas δ is referred to as the *prediction horizon*. Several algorithms also allow to estimate the *residence time* at a specific place [23, 2]. In this case, the estimation problem is formulated as finding the mini-

mal value δ^* of δ such that $\hat{X}(t + \delta^*) \neq X(t)$. If time is divided into *slots* (e.g., 15-minutes slots, 96 slots per day), then $X[k]$ represents the place the user is at time slot (or *time step*) k . Accordingly, we refer to a *slot-based next-place prediction algorithm* as an algorithm that computes the estimate $\hat{X}[k + n]$ of the place the user will be at n time steps ahead of k . An algorithm is then said to compute a correct *n -step ahead prediction* if $\hat{X}[k + n] = X[k + n]$. On the contrary, a wrong prediction occurs if $\hat{X}[k + n] \neq X[k + n]$. The *n -step ahead prediction accuracy* A_n of a prediction algorithm over an interval of N_s steps is then defined as the ratio of the number of correct predictions computed in the interval and N_s . We further say that a *place transition* T occurs at time step $k + 1$ when $X[k + 1] \neq X[k]$. Accordingly, a *self-transition* occurs when $X[k + 1] = X[k]$. A *departure event* from location L_j occurs when $X[k] = L_j$ and $X[k + 1] \neq L_j$. Similarly, an *arrival event* to location L_j occurs when $X[k] \neq L_j$ and $X[k + 1] = L_j$. The time instants at which an arrival or departure event occurs are indicated as *arrival time* and *departure time*, respectively. The difference between arrival and departure time is indicated as *residence time*.

To characterize the ability of a prediction algorithm to capture place transitions, we propose a set of metrics that rely upon the definitions of true positive, false positive, true negative, and false negative events typically used in classification problems [25]. We introduce these metrics later in the paper and use them to analyze the performance of a plethora of next place prediction algorithms. Before going into the details of our analysis, though, we summarize related work and outline the novelty of our approach in the following section.

RELATED WORK

Several algorithms to sense and predict human mobility have been presented in the literature. A number of approaches focus on identifying users' relevant places [1, 6, 8]. These algorithms can be broadly categorized into *fingerprint-based* and *geometry-based* algorithms [15]. Fingerprint-based algorithms use radio-frequency (RF) signals from stationary sources such as base station of mobile phone operators or Wi-Fi access points to identify relevant places [5, 8, 15]. When a “*stable radio environment*” [15] is detected – e.g., when a certain number of Wi-Fi access points is visible over a prolonged period of time – a relevant place is accordingly identified. Fingerprint-based algorithms typically do not provide positioning – i.e., they are agnostic of the actual geographical position of relevant places. Geometry-based algorithms, on the other hand, build upon the knowledge of the actual positions of the user (e.g., latitude/longitude coordinates) and cluster them into relevant places [1, 7, 26]. In this paper, we leverage the (fingerprint-based) PlaceSense algorithm by Kim *et al.* to identify relevant places [8].

Several authors have also investigated the next place prediction problem. Performing next place prediction means estimating, given some input data like the current location of the user and/or the time of the day, the next place that will be visited by the user [1, 2, 4, 9, 11, 19, 20]. Early work in this area has for example targeted the facilitation of call handovers in

¹For simplicity, we do not explicitly indicate the index i in the parameters N_L , L_j , \mathcal{L} , etc..

cellular networks by predicting the next cell in which a user will be located [23, 24]. More recent work has focused on predicting when users will leave or arrive at home in order to automatically control the heating system [20]. Next place prediction algorithms can be categorized according to the type of input information they use to compute the prediction. In particular, we differentiate between *spatial* and *temporal* features. The former include information about, e.g., which place(s) have been visited by the user before the current one. The latter correspond to information about, e.g., at which time of the day the user visits certain places. Ashbrook *et al.* for instance leverage only spatial features as input data to a second-order Markov model [1]. A performance comparison of a number of different predictors based on spatial features is provided in [24]. Approaches based on spatial features can forecast the next place visited by the user but not the corresponding arrival, departure, or residence time. The NextPlace algorithm by Scellato *et al.* uses non-linear time series analysis to predict both the arrival and residence time at users' relevant places. It does not use information about previously visited places and is thus agnostic of the spatial information inherent in users' mobility traces. Scott *et al.* make instead use of both temporal and spatial features [20].

In a seminal paper published in 2010, Song *et al.* explore the fundamental limits that characterize human mobility [21]. To support their findings, they leverage a large data set of mobile phone call detail records (CDR) (50,000 users, three months observation period). They identify relevant places visited by the users as the area covered by the base station to which phones connect. For each user, the sequence of places visited during the whole observation period is computed. The considered temporal granularity is one hour, i.e., only the location at which the user stayed for the longest time during a one-hour interval is stored. By adapting the information-theoretic concept of entropy to the realm of human mobility, Song *et al.* provide an answer to the question of with which probability “an appropriate predictive algorithm can predict correctly the users future whereabouts” [21] given the entropy of the sequence of previously visited locations. This probability is called the *predictability* Π of the user and represents an “upper bound that fundamentally limits any mobility prediction algorithm in predicting the next location based on historical records” [13]. Furthermore, Song *et al.* show that Π is bounded from below by the probability Π^{max} , which is achieved by a maximum likelihood algorithm that always predicts the next location of the user to be the most probable one given the previous history. While the work by Song *et al.* provides fundamental insights into the limits of predictability in human mobility, we focus on the specific role of transitions and analyze the performance of existing algorithms under this perspective. Building upon the work by Song *et al.*, McInerney *et al.* [14] address the problem of measuring the *momentary* predictability. To this end, they introduce a metric called Instantaneous Entropy (IE). While Song *et al.* use the entropy of a mobility trace over a given time interval to compute the overall “a posteriori” predictability of a user, instantaneous entropy (IE) aims at measuring the *momentary* predictability at each time instant. Thus, while the analysis by Song *et al.* is done offline

and defines a theoretical upper limit in predictability, the IE can be determined while the user is moving. In our previous work [17] we have analyzed the performance of IE and demonstrated its shortcomings. Building upon these results we propose to estimate the uncertainty related to a next place prediction as the “degree of disagreement” among a set of predictors.

PERFORMANCE ANALYSIS OF NEXT-PLACE PREDICTION ALGORITHMS

The performance of next place prediction algorithms are typically evaluated in terms 1-step ahead prediction accuracy A_1 . As discussed above, high values of A_1 – which is defined as the ratio of the number of correct predictions and the number of all prediction attempts – are easy to achieve even by “naïve” predictors. This is mainly due to the fact that humans spend most of their time in few, well-defined places. For instance, Montoliu *et al.* have shown that humans spend between 60% and 65% of their time at home [15]. An algorithm that predicts the next place of the user to be always the home can thus achieve accuracy values between 60% and 65% for most users. Montoliu *et al.* have also shown that for most users the second most relevant places is either the workplace of (or the school attended by) the user. They also found that people spend between 20% and 25% of their time at this location. A typical user thus spends, on average, at most 20% of her time at locations that are neither her home nor her workplace (or school). A “naïve” next place prediction algorithm can thus easily achieve accuracies of about 80% by predicting the next place visited by the user to be the home if the current place is the home or the workplace (or school) if the current place is the workplace (or school).² Such an algorithm will however always fail to correctly predict a transition of the user from home to the workplace (or school) and vice versa. The ability to reliably predict transitions is however crucial to support many applications, like urban navigation or smart heating control. For instance, the more accurately it is possible to predict transitions from and to the home, the higher the energy savings that can be achieved through an automatic heating control system. These observations allow us to point out that evaluating the performance of next-place prediction algorithms only in terms of A_1 might bring to misleading conclusions. In particular, optimizing the algorithms to make them achieve high values of A_1 might lead to approaches that are useless in practical settings.

To illustrate this issue in a quantitative manner we analyze the performance of 18 next-place prediction algorithms. We run our analysis using actual human mobility traces from a large, publicly available data set. To measure the performance of the considered algorithms we use a set of metrics that allow to evaluate both the average prediction accuracy as well as the ability of the algorithms to predict transitions. In the following, we describe the data set and the method we used to extract users' relevant places, the 18 predictors and

²This is consistent with the result obtained by Song *et al.* that the lower limit of the predictability of human mobility is about 80%, on average [21].

the metrics. The results of our analysis are then discussed in the following section.

Data set and identification of relevant places

We perform our analysis of the performance of next-place prediction algorithms on a set of mobility traces derived from data collected during the Lausanne Data Collection Campaign [12]. The data has been made available to the public in 2012 in the context of the Nokia Mobile Data Challenge (MDC) [12]. The data set contains data collected on the mobile phones of 37 users over about 1.5 years. The data includes GPS traces, records of Wi-Fi scans, Bluetooth scans and phone calls.

We extract Wi-Fi scan traces from this data set and apply the PlaceSense algorithm [8] to derive the most relevant places for each user. We use the default settings as described in [8] apart from the value of the *similarity* parameter. Instead of using the 68% value proposed in [8] we have adopted a similarity value of 30%. The reason for this choice is that the default value of 68% causes a high number of close-by relevant places to be identified. In particular, over 75% of the detected places have a median distance to their closest relevant place equal to or less than 10 meters. As for our analysis this fine-grained place detection is not needed, we have reduced the similarity value to 30%. This way, 54% of the detected relevant places have a median distances of less than 100 meters to the closest relevant place.

We build the sequence of places visited by each user over the whole observation period considering 15-minutes slots and determining the place visited by each user in each slot. We then represent the obtained sequences using the notation introduced above. For example, if a user is detected to be at her most visited location (L_1) for two slots, then at an unknown place (L_x) for one slot, and then at her third most visited location (L_3) for one slot, the corresponding mobility trace will be indicated as: $L_1 L_1 L_x L_3$. Figure 1 shows the percentage of time users in our MDC data set spend at their three most visited locations. The x-axis indicates the identifier of the user while the y-axis reports time in percentage with respect to the whole observation period. On average, users spend 56% of their time at location L_1 , 14% at location L_2 and 7% at location L_3 . 8% of the time is spent at other locations while for 15% of the time the algorithm does not recognize the visited place as a relevant one. The mobility traces derived from the MDC data set thus show very similar characteristics to those analyzed by Montoliu *et al.* [15]. Like in [15], we find that users spend most of their time at their two most visited locations.³ This shows that the mobility traces we consider in this paper have characteristics similar to those used by other authors and derived from other data sets. We thus expect our results to be representative also beyond the specific data set considered here. Verifying this

³The fact users in our data set show lower values of p_1 and p_2 with respect to those reported in [15] might be traced back to missing data, partially due to users' mobile phones being turned off. (According to a number of studies [18, 3] users turn off their mobile phones about ~22% of time.)

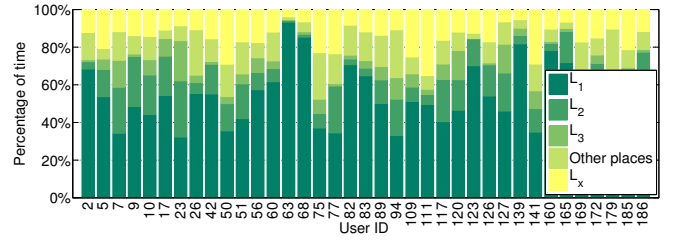


Figure 1: Percentage of time spent by the 37 users in their three most visited places. The plot also indicates the percentage of time spent in other places rather than the three most visited ones and the percentage of time for which the actual place is unknown (L_x).

assertion by repeating our analysis on different data sets is left to future work.

Prediction algorithms

Let us indicate with $\underline{p}^*[k] = \{p_1^*[k], p_2^*[k], \dots, p_{N_L}^*[k]\}$ the probabilities, at time step k , of the next location of the user being L_1, L_2, \dots, L_{N_L} , respectively. For simplicity, we use the notation \underline{p}^* to indicate $\underline{p}^*[k]$. In their seminal work Song *et al.* [21, 22] have shown that given the probabilities \underline{p}^* the highest prediction accuracy A_1^* is obtained by a predictor that chooses, at each time step k , the next location as the location of the user L_j such that $p_j^*[k] = \max(\underline{p}^*[k])$. This *Maximum Likelihood* (ML) predictor is thus the one that can achieve the highest accuracy among all predictors that use the same strategy to compute the probabilities \underline{p}^* .

For our analysis, we consider a set of predictors that use the ML strategy to determine the next location but differ from each other for how they compute the probabilities \underline{p}^* . The simplest predictor that we consider computes the value of $p_j^*[k], \forall k$ as the probability p_j of the user being at place L_j at any time. This algorithm – which we dub the *Prior* predictor – will thus always choose the most visited place L_1 as the next place. Most of the approaches existing in the literature use more complex methods to compute \underline{p}^* . In particular, they involve other *features* in the computation of \underline{p}^* , including the current location of the user (spatial features) or the time of the day (temporal features). In other words, the probabilities \underline{p}^* become *conditional* probabilities.

Several approaches make the values of \underline{p}^* vary depending on the current location of the user $X[k]$ [1]. Other approaches, like, e.g., [23, 16], consider also the previous location of the user $X[k-1]$. These two approaches correspond to the use of a 1st or 2nd order Markov model, respectively. Comparing several different prediction algorithms Song *et al.* have indeed shown that 2nd order Markov models provide for the best overall performance in terms of A_1 , on average [23].⁴ Besides making the probabilities \underline{p}^* depend on the current or previous locations of the user, several approaches also consider temporal features. For instance, Scott *et al.* consider

⁴More specifically, the best performance is provided by a 2nd order Markov model with fallback option to 1st order. This means that a 1st order model is used whenever no sufficient data is available to compute the probabilities necessary to run the 2nd order model.

whether the prediction needs to be computed for a weekday or weekend while Song *et al.* also use the time of the day and the day of the week [22, 20].

From the behavior of next-place prediction algorithms existing in the literature we thus define a set of predictors that takes into account also the above mentioned spatial or temporal features. In particular, we identify five main features that can be considered when computing the probabilities p^* : the current location of the user, the current and previous location of the user, the time of the day, the day of the week, and the day being a weekday or a day of the weekend. We refer to these five features with the symbols P_1 , P_2 , H , D , and W , respectively. Features P_1 and P_2 indicate a dependency of p^* from the location of the user while features H , D , and W account for time-dependencies.

The basic idea behind each combination is described as follows. First, we obtain the users' mobility traces until time instant k . Then we derive the values of all five temporal and spatial features at time instant k . For instance, if the instant k corresponds to 6 p.m. on a Monday, the user returns home at 6 p.m. but was still at work at 5:45 p.m., then the values of the features P_1 , P_2 , H , D , and W are: *Home*, $\{Work, Home\}$, 6p.m., *Monday*, and *Weekday*, respectively. We define different combinations of these features to derive a set of 18 different prediction algorithms, listed in Table 1. The first column of the table specifies the name of each algorithm. A symbol is included in the name of a predictor if the predictor uses the corresponding feature to compute p^* . For instance, the P_1 approach corresponds to an algorithm that makes the probabilities p^* depend on the current place of the user. If L_j is the current place, the probability for a place L_k to be visited next is computed as the ratio of the number of times the user has visited L_j and the number of times the user has visited L_k after having been in L_j . The place with the highest probability of "being next" is then taken as the prediction. This corresponds to the use of a 1st order Markov model. The approach named WP_1 also relies on a 1st order Markov model as described above but it computes different sets of probabilities for weekdays or days of the weekend. DP_2 considers the day of the week as well as both the current and previous location of the user. And so on.

The second and third column of Table 1 specify the features used by each of the algorithms. We recall that all the algorithms listed in Table 1 choose, at time step k , the next location using a maximum likelihood strategy [22]. The place with the highest value of $p^*[k]$ is thus the predicted next place. This ensures that performance differences between the algorithms are only due to the different strategies used to compute the probabilities p^* from the historical mobility traces.

Performance metrics

We characterize the ability of a prediction algorithm to capture transitions using a set of metrics that draw upon the definitions of true positive, false positive, true negative, and false negative events typically used in classification prob-

Table 1: Next place prediction algorithms considered in our analysis.

Name	Spatial features	Temporal features
<i>Prio</i>	none	none
<i>W</i>	none	weekday/weekend
WP_1	current location	weekday/weekend
<i>WH</i>	none	weekday/weekend and time of the day
WHP_1	current location	weekday/weekend and time of the day
<i>D</i>	none	day of the week
DP_1	current location	day of the week
<i>DH</i>	none	day of the week and time of the day
DHP_1	current location	day of the week and time of the day
P_1	current location	none
<i>H</i>	none	time of the day
HP_1	current location	time of the day
P_2	current and previous location	none
WP_2	current and previous location	weekday/weekend
WHP_2	current and previous location	weekday/weekend and time of the day
DP_2	current and previous location	day of the week
DHP_2	current and previous location	day of the week and time of the day
HP_2	current and previous location	and time of the day

lems [25]. The mathematical definitions of all the metrics introduced below are reported in Table 2. We define a *true positive transition event* as the event that occurs when a transition is predicted *and* it actually occurs. In mathematical terms this corresponds to the condition $[\hat{X}[k+1] \neq X[k] \& (X[k+1] \neq X[k])]$ being fulfilled. The number of these events over an interval of N_s slots is indicated as TTP . The corresponding *true positive transition rate* $TTPR$ is defined as the ratio of TTP over the sum of all transition events. A *false positive transition event* occurs when a transition is erroneously predicted to occur. We indicate with TFP the number of false positive transition events over an interval of N_s slots and define the corresponding *false positive transition rate* $TFPR$ as the ratio of TFP over the sum of all self-transitions. Further, a *true negative transition event* occurs when the next location is correctly predicted as being the same as the current one. The number of such transitions over an interval of N_s slots is indicated as TTN and the corresponding rate as $TTNR$. Finally, a *false negative transition event* occurs when a transition occurs but is not correctly predicted. Accordingly, TFN indicates the number of such events occurring over an interval of N_s slots and the corresponding rate $TFNR$ is defined as the ratio of TFN and the actual number of transitions in the same interval.

We further define the *transition precision* $TPre$ as the ratio of the number of correctly predicted transitions (TTP) and the total number of predicted transitions ($TTP + TFP$). Similarly, we define the *transition recall* $TRec$ as the ratio of the number of correctly predicted transitions (TTP) and the sum of all actually occurred transitions. Finally, to capture the dependency between precision and recall, we define the *transition harmonic mean* TF_1 as indicated in Equation 8.

Besides looking at transitions in general, we also analyze the performance of the considered algorithms in terms of their ability to predict arrival and departure events to a specific place L_j . We accordingly define the metric TF_1ArrL_j as the TF_1 computed over those time slots in which the condition $[(X[k] \neq L_j) \& (X[k+1] = L_j)]$ is fulfilled. Sim-

Table 2: Performance metrics.

$TTP = \sum_{k=1}^{N_s} [(\hat{X}[k+1] \neq X[k]) \& (X[k+1] \neq X[k])]$	(1)
$TTPR = \frac{TTP}{\sum_{k=1}^{N_s} (X[k+1] \neq X[k])}$	(2)
$TFP = \sum_{k=1}^{N_s} [(\hat{X}[k+1] \neq X[k]) \& (X[k+1] = X[k])]$	(3)
$TFPR = \frac{TFP}{\sum_{k=1}^{N_s} (X[k+1] = X[k])}$	(4)
$TTN = \sum_{k=1}^{N_s} [(\hat{X}[k+1] = X[k]) \& (X[k+1] = X[k])]$	(5)
$TTNR = \frac{TTN}{\sum_{k=1}^{N_s} (X[k+1] = X[k])}$	(6)
$TFN = \sum_{k=1}^{N_s} [(\hat{X}[k+1] = X[k]) \& (X[k+1] \neq X[k])]$	(7)
$TFNR = \frac{TFN}{\sum_{k=1}^{N_s} (X[k+1] \neq X[k])}$	(8)
$TPre = \frac{TTP}{TTP + TFP}$	(9)
$TRec = \frac{TTP}{TTP + TFN}$	(10)
$TF_1 = 2 * \frac{TPre * TRec}{TPre + TRec}$	(11)
$DKL = \sum_{i=1} R(i) * \log \frac{R(i)}{Q(i)}$	(12)
$CP(N_p^{max}) = \frac{\sum_{k=1}^{N_s} [(\hat{X}[k] = X[k]) \& (N_p^{max} = N_p^{max}[k])]}{\sum_{k=1}^{N_s} N_p^{max} = N_p^{max}[k]}$	(13)
$TrP[k] = \sum_{alg=1}^{N_{ALG}} \hat{X}[k+1] \neq X[k]$	(14)

ilarly, we define TF_1DepL_j as the metric TF_1 computed only for the time slots in which $[(X[k] = L_j) \& (X[k+1] \neq L_j)]$ is fulfilled. We focus in particular on transitions to and from L_1 and L_2 and thus consider the metrics TF_1ArrL_1 , TF_1DepL_1 , TF_1ArrL_2 , and TF_1DepL_2 . We further measure the distance between the empirically observed places distribution Q depicted in Figure 1 and the corresponding distribution R computed by each predictor. To this end, we use the Kullback-Leibler Divergence (KL-Divergence) defined as in Equation 12.

RESULTS

Table 3 shows the results obtained running the 18 prediction algorithms described in the previous section over the mobility traces of the 37 users. Each row reports the performance of one of the considered predictors in terms of the metrics introduced above. Each column thus reports the value of one of the metrics for all 18 predictors. The figures in the table are median values of all 37 users. Each algorithm was used to predict the next location 1-time step ahead. The last row of Table 3 reports the performance of a “fictive” optimal algorithm – dubbed *Best*. If at least one of the 18 considered predictors correctly predicts the next location, *Best* will pick this prediction as its own. The three-but-last rows of Table 3 report the performance of the MAJOR approach that will be introduced in the next section.

The results reported in Table 3 allow to make a number of observations. For instance, the first column of the table shows the performance of all predictors in terms of prediction accuracy A_1 . The highest accuracy (87%) is achieved by WP_1 and P_1 immediately followed by DP_1 , P_2 , and WP_2 (86%). This shows that using Markov models of 1st or 2nd order – i.e., including the features P_1 or P_2 in the computation of p^* – guarantees for the best performance in terms of A_1 , on

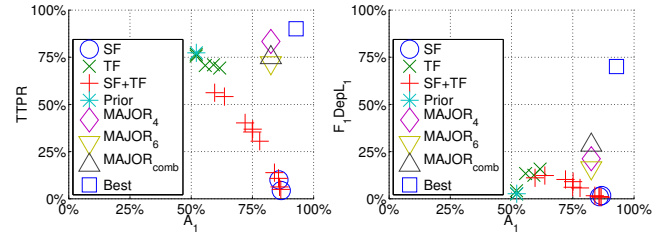


Figure 2: Correlation between A_1 and $TTPR$ (left) and A_1 and TF_1DepL_1 (right).

average. This confirms what also observed by Song *et al.* in [23] as well as by other authors [2]. The same algorithms show different abilities to correctly predict the occurrence of a transition. For instance, the $TTPR$ of P_2 is only 10%. This means that P_2 correctly predicts the occurrence of only 10 over 100 transitions.

To gain a better understanding of the performance of the different algorithms we report a subset of our results in Figure 2. The two plots depict the correlation between A_1 and $TTPR$ (left) and A_1 and TF_1DepL_1 (right). A data point is reported on the plot for each of the 18 predictors. A data point represents the median value, computed over all 37 users, of the corresponding metric. Different markers represent different categories of the considered predictors. *Prior*, *Best*, and MAJOR are as defined above. We then differentiate between algorithms using spatial features (SF), temporal features (TF) or both (SF+TF). The first category includes predictors P_1 , P_2 , which compute probabilities p^* using information about the place of the user only. The second category includes W , WH , D , DH , and H , i.e., predictors that consider temporal features only. The third category finally includes all other approaches – WP_1 , WHP_1 , DP_1 , DHP_1 , HP_1 , WP_2 , WHP_2 , DP_2 , DHP_2 , HP_2 – that combine both spatial and temporal features. Figure 2 (left) shows the existence of a negative correlation between A_1 and $TTPR$. This means that the higher the accuracy of a predictor, the less the same predictor will be able to capture transitions. In other words, being effective at capturing transitions makes predictors more prone to failures, as a higher *risk* is taken when computing the prediction. This observation is supported by the existence of a negative correlation between $TTPR$ and $TTNR$ (not shown for space constraints). Figure 2 also shows that the MAJOR approach achieves a very good trade-off between the values of A_1 and $TTPR$. Figure 2 (right) shows that most algorithms can correctly detect departure events from the most visited place L_1 in less than 20% of the cases. This occurs despite the overall accuracy A_1 being very high. Indeed, high values of A_1 are obtained thanks to the fact that most self-transitions (i.e., situations in which $X[k+1] = X[k]$) are correctly detected. The plot also shows that the MAJOR approach outperforms other predictors in correctly detecting departures from L_1 . The performance achieved by the *Best* approach implies that in almost 75% of the situations there exists at least one of the 18 algorithms that correctly predicts the occurrence of a departure event from L_1 . This allows to infer that depending on the specific situation

Table 3: Performance of all 18 predictors. Each performance figure is the median value over all 37 users. Each algorithm was used to predict the next place 1-time step ahead.

	A_1	$TTPR$	$TTNR$	$TFPR$	$TFNR$	$TRec$	$TPre$	TF_1	TF_1ArrL_1	TF_1DepL_1	TF_1ArrL_2	TF_1DepL_2	D_{KL}
<i>Prior</i>	52	77	56	23	44	77	22	34	13	3	7	20	0.61
<i>W</i>	52	77	56	23	44	77	22	34	14	3	7	19	0.44
<i>WP₁</i>	87	5	99	95	1	5	35	8	5	1	2	1	0.004
<i>WH</i>	62	69	65	31	35	69	22	34	16	16	7	24	0.03
<i>WHP₁</i>	75	37	84	63	16	37	25	28	15	9	3	15	0.03
<i>D</i>	52	76	56	24	44	76	22	34	14	4	7	17	0.38
<i>DP₁</i>	86	6	98	40	2	6	32	10	6	1	3	2	0.005
<i>DH</i>	56	71	58	29	42	71	20	31	15	13	5	24	0.01
<i>DHP₁</i>	64	54	73	46	27	54	22	30	14	12	5	21	0.10
<i>P₁</i>	87	4	99	96	1	4	35	8	3	2	2	2	0.004
<i>H</i>	59	71	62	29	38	71	22	33	16	12	7	24	0.05
<i>HP₁</i>	78	31	89	69	11	31	27	27	15	6	4	15	0.02
<i>P₂</i>	86	10	97	90	3	10	31	15	12	1	5	1	0.01
<i>WP₂</i>	86	11	97	89	3	10	30	16	11	1	5	2	0.01
<i>WHP₂</i>	72	40	81	60	19	40	23	28	16	10	4	16	0.05
<i>DP₂</i>	84	14	95	86	5	14	29	18	15	2	6	3	0.01
<i>DHP₂</i>	60	56	67	44	33	56	19	28	14	11	4	18	0.15
<i>HP₂</i>	75	35	84	65	16	31	24	27	16	6	5	15	0.03
<i>MAJOR</i>	82	21	93	79	7	21	32	25	19	3	5	21	0.006
<i>MAJOR₄</i>	82	84	64	16	36	84	25	39	17	22	8	29	0.006
<i>MAJOR₆</i>	82	72	77	28	23	72	32	44	20	17	11	34	0.006
<i>MAJOR_{comb}</i>	82	75	77	25	23	75	34	46	20	29	11	34	0.006
<i>Best</i>	93	90	99	10	1	98	90	93	87	70	77	99	0.013

the use of spatial, temporal, or both types of features might allow to correctly predict the occurrence of departure events. Which features allow to make a correct prediction might and do however change over time. Further results about the ability of the single algorithm to correctly predict arrival and departure events (from L_1 and L_2) are reported in Table 3 but are not discussed in detail due to space constraints.

The last column of Table 3 shows the values of KL-Divergence (D_{KL}) for each of the considered algorithms. Most predictors, including MAJOR, exhibit very low values of D_{KL} . This indicates that the empirically observed places distribution Q (see Figure 1) and the corresponding distribution R computed by each predictor are very similar. The results in Table 3 also show that the higher the value of A_1 the better the match between the observed and the predicted distributions. To our surprise, the predictor *Best* does not exhibit the better performance in terms of D_{KL} .

The main conclusions we can draw from the results reported and discussed above is that none of the considered predictors is able to provide good performance both in terms of A_1 and in terms of ability to reliably predict transitions. To overcome this drawback, we propose a novel next-place prediction algorithm that combines several predictors instead of relying on a single one. We describe and discuss this approach in the following section.

A NOVEL NEXT-PLACE PREDICTION ALGORITHM

The results discussed in the previous section show that the approach named *Best* achieves an average accuracy A_1 of 93% and at the same time a $TTPR$ of 90% and $TTNR$ of 99%. *Best* picks at each time step the correct next place prediction, provided that at least one of the 18 considered algorithms actually computed the correct estimate. The fact that *Best* has an accuracy of 93% thus means that in 93% of the time steps at least one of the 18 predictors is able to correctly predict which place the user will visit next. This

in turn implies that only in 7% of the situations none of the 18 predictors was able to correctly estimate the next place of the user. A $TTPR$ of 90% further indicates that in 90% of the cases at least one of the 18 predictors correctly predicted the occurrence of a transition. And a $TTNR$ of 99% indicates that in 99% of the cases at least one of the 18 predictors correctly predicted the occurrence of a self-transition. Interestingly, the 93% accuracy value achieved by *Best* coincides with the upper limit of the predictability of human mobility reported by Song *et al.* in [21]. This indicates that by combining the predictive power of several predictors the performance of the *Best* approach achieves values that are very close to the theoretical predictability limits intrinsic in human mobility. This also allows us to argue that the spatial and temporal features considered in this paper allow to exhaustively capture this predictability. Building upon these considerations we thus propose a novel next-place prediction algorithm that combines the predictive power of several algorithms.

Our approach, called MAJOR, works as follows. MAJOR lets a number N_{ALG} of next-place prediction algorithms run in parallel. In particular, we set $N_{ALG} = 18$ and use the 18 predictors described in Table 1. At each time step k , MAJOR counts the number of predictors $N_{pL_j}[k+1]$ that indicate the place L_j as the place the user will visit at time step $k+1$. We collect these counters in the vector $\underline{N}_p[k+1] = [N_{pL_1}[k+1], N_{pL_2}[k+1], \dots, N_{pL_{N_L}}[k+1]]$. Please note that $\underline{N}_p[k+1]$ is computed at time step k . The value of each $N_{pL_j}[k+1]$ can vary between 0 and N_{ALG} and $\sum_{j=1}^{N_L} N_{pL_j}[k+1] = N_{ALG}, \forall k$. At time step k , the estimate of the location visited by the user at time step $k+1$ is then chosen as the place L_j for which $N_{pL_j}[k+1] = \max \underline{N}_p[k+1]$. If two or more of the $N_{pL_j}[k+1]$ counters reach the maximal value, MAJOR chooses the next place randomly among those receiving the maximal number of “votes”. The name of the algorithm, MAJOR, comes from

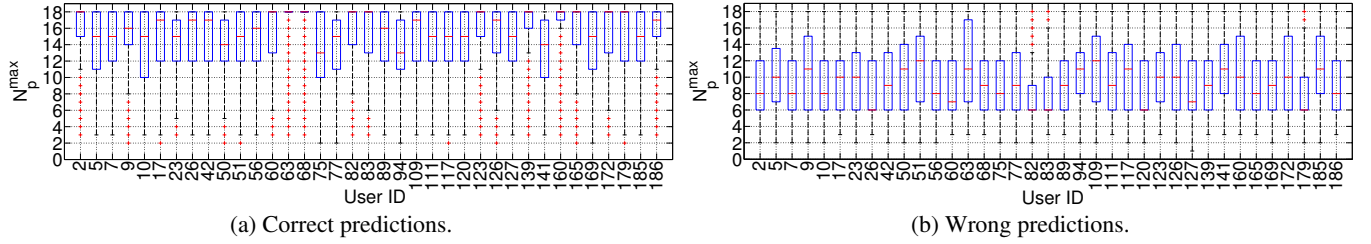


Figure 3: Number of approaches that agree on the next-place prediction for situations in which the prediction is correct (a) or wrong (b).

the fact that it relies on a majority vote approach to choose the next place of the user.

The row labelled as “MAJOR” in Table 3 shows the performance achieved by this simple version of the MAJOR algorithm. The data in the table shows that the 1-step ahead accuracy A_1 of MAJOR is 82%. This is 5% lower than the two best performing “individual” predictors WP_1 and P_1 but 12% higher than the average accuracy of all 18 individual predictors, which is 70%. MAJOR also shows very good performance in terms of $TNRR$ and $TFPR$ but is able to correctly detect the occurrence of a transition only in 21% of the cases ($TTPR$). This is due to the fact that many of the individual predictors are indeed unable to detect such transitions. Relying on simple majority voting as described above thus makes MAJOR also unable to reliably detect transitions. In order to cope with this problem, we refine the design of MAJOR as described below.

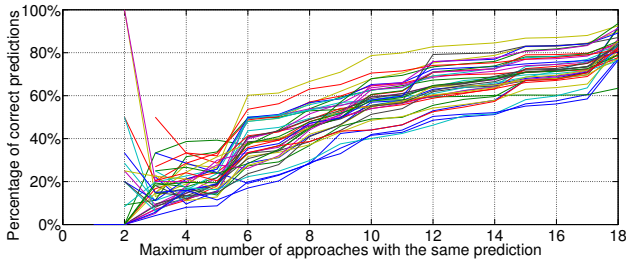


Figure 4: The x-axis shows the maximum number of algorithms agreeing on the same next-place prediction. The corresponding y-values represent the number of times the prediction is correct (shown as a percentage over the whole number of computed predictions). Different lines correspond to different users in the data set.

Confidence of MAJOR’s estimates

In order to improve MAJOR’s ability to detect transitions we first measure the number of predictors that return correct or wrong predictions. We define $N_p^{max}[k+1]$ as the maximum number of predictors that provide, at time step k , the same estimate for time step $k+1$ (i.e., $N_p^{max}[k+1] = \max(\underline{N}_p[k+1])$). We then calculate, for each user in our data set, the values assumed by $N_p^{max}[k+1]$ for the cases in which the $N_p^{max}[k+1]$ predictors estimate the correct next place of the user. Figure 3a shows the corresponding results. The segment in each box shows the median,

the edges of the box indicate the 25th and 75th percentiles, and the whiskers cover 99.3% of the data (assuming it is normally distributed). We observe that for most users the value of $N_p^{max}[k+1]$ for correct 1-step ahead predictions is higher than 12 in more than 75% of the cases. I.e., when 12 or more predictors signal the same place as the next one, then in 75% of the cases the corresponding prediction is correct. We then run the same analysis for situations in which wrong predictions are computed. The corresponding results are depicted in Figure 3b and show that the value of N_p^{max} in this case varies roughly between 6 and 12. This means that when the maximum number of predictors agree on an incorrect prediction then their number is between 6 and 12 in most cases. The fact so many algorithms might compute a wrong estimate is due to the inability of most predictors to correctly detect transitions. In particular, algorithms that rely on spatial features have a strong bias in predicting self-transitions and thus tend to “miss” the occurrence of actual transitions in most cases. On the other side, algorithms relying on temporal features incur in high false positive rates because they tend to predict transitions more often than approaches based on spatial features. Combining the capabilities of both spatial and temporal features is MAJOR’s strength. In order to leverage the full potential of this strength we slightly adapt MAJOR’s majority voting approach and differentiate between situations in which self-transitions or transitions are predicted, as described below.

Before going further, we would like to outline the considerations above also allow to use the value of $N_p^{max}[k+1]$ as an empirical measure of the *confidence* of the next place prediction $\hat{X}[k+1]$ computed by MAJOR. To illustrate this point, we analyze the ratio of the number of correct 1-step ahead predictions and the total number of predictions for each possible value of N_p^{max} . The ratio is indicated as $CP(N_p^{max})$ and defined by Equation (13) in Table 2. We compute $CP(N_p^{max})$ for each user and for N_p^{max} values between 2 and 18. Figure 4 shows the corresponding results whereas each line corresponds to one of the 37 users. The plot shows the existence of a positive correlation between the value of N_p^{max} and the percentage of correct predictions. Hence, as expected, higher values of N_p^{max} indicate a higher probability of MAJOR computing correct 1-step ahead predictions. The value of $CP(N_p^{max})$ thus indicates, even though only empirically, the *confidence* of the prediction $\hat{X}[k+1]$.

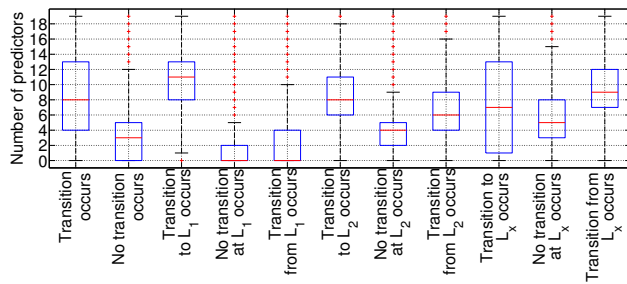


Figure 5: Number of algorithms that predicts the occurrence of a transition in the cases indicated on the x-axis.

MAJOR's ability to predict transitions

The considerations reported above hint at the fact that the ability of MAJOR to correctly predict transitions can be improved by considering the actual number of predictors “pointing at” or not at a transition. To support this claim we discuss the results reported in Figure 5. We first compute the number of predictors that, at each time step k , correctly predict a transition to occur (irrespective of from or to the transition occurs). The leftmost boxplot in Figure 5 shows the corresponding results, averaged over all time steps. The plot shows that the median number of approaches correctly predicting a transition to occur is 8. The second boxplot from the left in Figure 5 shows instead that the median number of approaches predicting a transition when actually none will occur is 3. We thus suggest to introduce a threshold α that allows to differentiate between these two cases. In particular, along with the majority vote policy described above MAJOR will predict a transition to occur when the number of approaches predicting a transition to occur is higher than α . I.e., even if the majority of the approaches predicts a self-transition, as soon as at least α approaches predict a transition to occur then MAJOR will also predict the transition to occur. If training data is available the actual value of α can be set offline. For instance, we have found that a value of α equal to 6 allows MAJOR to achieve the highest TF_1 value for our data set. We indicate this version of MAJOR as $MAJOR_6$ and its performance metrics are reported in the corresponding row in Table 3. Noticeably, the performance of $MAJOR_6$ exceeds those of individual predictors for most of the considered metrics.

For some applications it might be interesting to tune MAJOR so as to increase its ability to detect transitions from and to specific places like, e.g., L_1 , L_2 , or even L_x . We thus now focus on arrival and departure events as well as self-transitions from and to these locations. The 3rd, 4th, and 5th plot from left in Figure 5 are computed considering only those time steps in which the place at k or $k+1$ is L_1 . The first of these three plots shows the number of predictors that indicate a transition to L_1 to occur (arrival event) when it actually occurs. The second plot shows the number of predictors that indicate a transition from L_1 to occur (departure event) when actually no transition occurs (in this case thus a self-transition to L_1 occurs). The third plot shows the number of predictors that indicate a transition from L_1 to occur when a transition from L_1 actually occurs. The other plots in

Figure 5 (6th to 11th from left) show the same results but relative to L_2 and L_x . The 5th plot in Figure 5 allows to make an interesting observation: when a transition from L_1 occurs, the median number of algorithm predicting it correctly is 0. Also, in 75% of the cases less than 4 approaches will correctly predict a transition from L_1 . Keeping the threshold α equal to 6 will thus cause many of these transitions not to be recognized by MAJOR. We performed an exhaustive search and found that setting α equal to 4 allows to achieve the highest value of the $TF_1 DepL_1$ metric (see also results in Table 3, row $MAJOR_4$). At the same time, α equal to 6 allows to achieve better performance for most of the other metrics. We thus introduce a second threshold, called β to differentiate these cases. We make MAJOR use α as a threshold to decide that a transition will actually take place. For departure events from L_1 , however, MAJOR will predict the transition to occur when at least β algorithms will accordingly predict it to occur. According to the observations summarized above, we set $\alpha = 6$ and $\beta = 4$. The performance achieved by MAJOR through the “combined” use of both thresholds α and β are summarized in Table 3, row $MAJOR_{comb}$. These results show that this version of MAJOR not only achieves very high accuracy A_1 (82%) but also shows a superior overall performance in detecting transitions when compared to the individual predictors. Further investigations concerning the design and performance analysis MAJOR are left to future work.

CONCLUSIONS

In this paper we have investigated the performance of 18 next-place prediction algorithms. Our analysis shows that high average prediction accuracies can be obtained even by “naïve” algorithms that are largely unable to detect transitions between different places. We have defined a set of metrics that allow to characterize the ability of a predictor to capture such transitions. We advocate that a comprehensive description of the performance of next-place prediction algorithms must include also an analysis of these metrics – or at least of a subset thereof. Building upon the results of our analysis we propose a novel next-place prediction algorithm, called MAJOR, that can both achieve high prediction accuracy and reliably predict transitions. MAJOR's good performance is obtained by combining the 18 approaches considered in our analysis in a single algorithm and using a majority vote approach to compute the final prediction. We also showed that the number of individual algorithms agreeing on the same next place prediction can be used as an indicator for the confidence of MAJOR's prediction.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable comments. This work has been partially supported by the Collaborative Research Center 1053 funded by the German Research Foundation and by the LOEWE Priority Program Cocoon funded by the LOEWE research initiative of the state of Hesse, Germany.

REFERENCES

1. D. Ashbrook and T. Starner. Using GPS to Learn Significant Locations and Predict Movement Across

- Multiple Users. *Personal and Ubiquitous Computing*, 7(5):275–286, Oct. 2003.
2. Y. Chon, H. Shin, E. Talipov, and H. Cha. Evaluating Mobility Models for Temporal Prediction with High-granularity Mobility Data. In *10th Intl. Conf. on Pervasive Computing and Communications (PerCom'12)*. IEEE, Mar. 2012.
 3. A. Dey, K. Wac, and D. Ferreira. Getting Closer: An Empirical Investigation of the Proximity of Users to their Smart Phones. In *13th Intl. Conf. on Ubiquitous Computing (UbiComp'11)*. ACM, Sept. 2011.
 4. M. D. Domenico, A. Lima, and M. Musolesi. Interdependence and Predictability of Human Mobility and Social Interactions. In *Nokia Mobile Data Challenge Workshop in Conjunction with 10th Intl. Conf. on Pervasive Computing (Pervasive'12)*. Springer, June 2012.
 5. O. Dousse, J. Eberle, and M. Mertens. Place Learning via Direct WiFi Fingerprint Clustering. In *13th Intl. Conf. on Mobile Data Management (MDM'12)*. IEEE, July 2012.
 6. J. Hightower, S. Consolvo, and A. LaMarca. Learning and Recognizing the Places We Go. In *7th Intl. Conf. on Ubiquitous Computing (UbiComp'05)*. Springer, Sept. 2005.
 7. J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting Places from Traces of Locations. *Mobile Computing and Communications Review*, 9(3):58, July 2005.
 8. D. H. Kim, J. Hightower, R. Govindan, and D. Estrin. Discovering Semantically Meaningful Places from Pervasive RF-Beacons. In *11th Intl. Conf. on Ubiquitous Computing (UbiComp'09)*. ACM, Sept. 2009.
 9. M. Kim, D. Kotz, and S. Kim. Extracting a Mobility Model from Real User Traces. In *25th Intl. Conf. on Computer Communications (INFOCOM'06)*. IEEE, Apr. 2006.
 10. J. Krumm and A. J. B. Brush. Learning Time-based Presence Probabilities. In *9th Intl. Conf. on Pervasive Computing (Pervasive'11)*. Springer, June 2011.
 11. J. Krumm and E. Horvitz. Predestination: Inferring Destinations from Partial Trajectories. In *8th Intl. Conf. on Ubiquitous Computing (UbiComp'06)*. ACM and Springer, Sept. 2006.
 12. J. Laurila, D. Gatica-Perez, and I. Aad. The Mobile Data Challenge: Big Data for Mobile Computing Research. In *10th Intl. Conf. on Pervasive Computing (Pervasive'12)*. Springer, June 2012.
 13. M. Lin, W.-J. Hsu, and Z. Q. Lee. Predictability of Individuals' Mobility with High-resolution Positioning Data. In *14th Intl. Conf. on Ubiquitous Computing (UbiComp'12)*. ACM, Sept. 2012.
 14. J. McInerney, S. Stein, A. Rogers, and N. Jennings. Exploring Periods of Low Predictability in Daily Life Mobility. In *10th Intl. Conf. on Pervasive Computing (Pervasive'12)*. Springer, June 2012.
 15. R. Montoliu, J. Blom, and D. Gatica-Perez. Discovering Places of Interest in Everyday Life from Smartphone Data. *Multimedia Tools and Applications*, 62:179–207, Jan. 2013.
 16. A. J. Nicholson and B. D. Noble. BreadCrumbs: Forecasting Mobile Connectivity. In *14th Intl. Conf. on Mobile Computing and Networking (MobiCom'08)*. ACM, Sept. 2008.
 17. P. Baumann and S. Santini. On the Use of Instantaneous Entropy to Measure the Momentary Predictability of Human Mobility. In *14th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC'13)*. IEEE, June 2013.
 18. S. Patel, J. Kientz, G. Hayes, S. Bhat, and G. Abowd. Farther Than You May Think: An Empirical Investigation of the Proximity of Users to their Mobile Phones. In *8th Intl. Conf. on Ubiquitous Computing (UbiComp'06)*. ACM and Springer, Sept. 2006.
 19. S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. NextPlace: A Spatio-temporal Prediction Framework for Pervasive Systems. In *9th Intl. Conf. on Pervasive Computing (Pervasive'11)*. Springer, June 2011.
 20. J. Scott, A. Brush, and J. Krumm. PreHeat: Controlling Home Heating Using Occupancy Prediction. In *13th Intl. Conf. on Ubiquitous Computing (UbiComp'11)*. ACM, Sept. 2011.
 21. C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of Predictability in Human Mobility. *Science (New York, N.Y.)*, 327(5968):1018–21, Feb. 2010.
 22. C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Supplementing Online Material on Limits of Predictability in Human Mobility. *Science (New York, N.Y.)*, 327(5968):1–21, Mar. 2010.
 23. L. Song and U. Deshpande. Predictability of WLAN Mobility and its Effects on Bandwidth Provisioning. In *25th Intl. Conf. on Computer Communications (INFOCOM'06)*. IEEE, 2006.
 24. L. Song, D. Kotz, R. Jain, and X. He. Evaluating Location Predictors with Extensive WiFi Mobility Data. *Mobile Computing and Communications Review*, 7(4):64–65, Oct. 2003.
 25. I. Witten, E. Frank, and M. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition, 2011.
 26. Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining Individual Life Pattern Based on Location History. In *10th Intl. Conf. on Mobile Data Management: Systems, Services and Middleware (MDM'09)*. IEEE, May 2009.