

User identification from mobility traces

Sergio Salomón¹ · Cristina Tîrnăucă² · Rafael Duque²  José Luis Montaña²

Abstract

Geolocation is a powerful source of information through which user patterns can be extracted. User regions-of-interest, along with these patterns, can be used to recognize and imitate user behavior. In this work we develop a methodology for preprocess-ing location data in order to discover the most relevant places the user visits, and we propose a Probabilistic Finite Automaton structure as mobility model. We analyse both location prediction and user identification tasks. Our model is assessed with two evaluation metrics regarding its predictive accuracy and user identification accuracy, and compared against other models.

Keywords Geolocation · User identification · Location Prediction · Probabilistic finite automaton · Learning from observation

1 Introduction

Nowadays geolocation has become a quite common task performed by a variety of electronic devices, such as smartphones, tablet computers, vehicles or wearable devices. Geolocation consists in the estimation of the geographic location of an object (Roxin et al. 2007). To this end, location inference can generate different position data, like geographic coordinates, time stamp, street address, shop names, postal code, country, etc. Later, this information is used to register route tracks, traffic logs, running trails, travel records or geotagged photos. In this domain, the size of these recordings can grow fast as geolocation is made anytime anywhere. As a result, it is a valuable source of trends, statistics, patterns of movement, fitness, traffic and travel behaviors, but its large dimensions must be handled in some way to harness its potential benefits. The field of Trajectory Data Mining (Feng and Zhu 2016) gathers the specific methods to exploit position data. Using those methods, geolocation offers a wide range of interesting applications in ambient

intelligence environments (Cook et al. 2009), like recommender systems, location prediction, activity recognition or individual profiling. Even though these techniques can be applied to other types of individuals (e.g., animals) and objects, in this work we focus on human user geolocation.

Within the analysis of this phenomenon, the extraction of user mobility behavior (known in broader terms as behavior recognition and behavior cloning (Argall et al. 2009)) poses a great challenge. As a user moves through its environment, a complex structure of habits and preferences arises over time. This process is not completely deterministic nor completely random, but involves the appearance of patterns. Therefore, appropriate statistical models can be used to capture the behavior represented in those patterns. After that, the behavior learned is useful not only to predict location but also to produce artificially generated data similar to the user data. We study both tasks in this work, since both could be seen as forms of behavior imitation. The second task allows us to accomplish a more advanced goal: the user identification from geolocation records, since the comparison between generated and real data serves to recognize each user's behavior. In ubiquitous environments, user identification allows technology to adapt to the user automatically, without an explicit interaction, in order to improve the user experience.

In this paper, we propose a methodology to capture individual behavior by preprocessing trajectories and building probabilistic models. Our approach involves detecting visited points, identifying relevant locations and finally constructing a stochastic model that represents the behavior of

✉ Rafael Duque
rafael.duque@unican.es

Sergio Salomón
ssalomong@axpecantabria.com

¹ AXPE Consulting Cantabria, Polígono de la Cros, Camargo, Spain

² Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria, Santander, Spain

the user. The preprocessing steps enable a reduction and discretization of the trajectory data to facilitate the model learning. The stochastic model captures the probability distribution of movement between locations over time (i.e., the model considers the evolution of this behavior), admitting, as well, additional temporal information. Then, we use the trained models to generate new trajectories, and next we perform user identification by comparing them with real traces. This way, by comparing the imitated traces to the original trajectories, we determine if the model trained is good enough to describe the user’s daily routines. For this comparison, we propose an evaluation metric more appropriate to behavior recognition and user identification goals. In the experimentation phase, we employ an open repository of geolocated routes published by Microsoft Research Asia (Zheng et al. 2009).

The paper is structured as follows. Section 2 introduces some definitions we use throughout this document. Section 3 highlights the contributions in Trajectory Data Mining and Behavior Inference that are relevant to our development. Section 4 describes our methodology for the formulated problem. In Sect. 5 we report on the results obtained by conducting experiments on a real dataset. Finally, in Sect. 6 we present some concluding remarks and further work ideas.

2 Theoretical background

In this article, we use several concepts from geolocation and mobility. For the reader’s convenience, we include here those definitions that are essential for a better comprehension of our work.

Definition 1 Location (also geographical point) is a tuple of coordinates $p = (\textit{latitude}, \textit{longitude})$ that identifies a point on the Earth’s surface unambiguously according to the Global Positioning System (GPS).

Definition 2 Trajectory is a sequence of tuples $\pi = [(p_1, ts_1), \dots, (p_n, ts_n)]$ ordered chronologically by the time stamp ts_i , where each p_i is a location. For simplicity, we will refer to each component of a trajectory point as $\pi_i.\textit{lat}$, $\pi_i.\textit{lon}$ and $\pi_i.\textit{ts}$.

Definition 3 Semantic trajectory is an enhanced trajectory π such that each component has assigned a discrete label or class that we denominate $\pi_i.C$. The labels could be street addresses, road names, Points-of-Interest or POI (useful locations for one or more users), Regions-of-Interest or ROI (regions that may contain multiple POI) or other marks with semantic relevance (for example, user-specific spaces like “home” and “workplace”).

Definition 4 Stay point (also “stationary point”, “stay” or “stop”) is a 4-tuple $s = (\textit{lat}, \textit{lon}, ts_{\textit{arv}}, ts_{\textit{lev}})$ that represents a location $(\textit{lat}, \textit{lon})$ where the user stayed within a distance radius $d < \textit{distLim}$, and for a time lapse $t > \textit{timeLim}$ between the arrival and leaving time stamps, $ts_{\textit{arv}}$ and $ts_{\textit{lev}}$, respectively. The thresholds $\textit{distLim}$ and $\textit{timeLim}$ depend on the particular application and data.

Definition 5 Location routine is a sequence of locations (p_1, \dots, p_k) of size k that appears with relevant frequency among the set of user’s trajectories and that is repeated on a daily basis (contrary to a weekly pattern, monthly pattern, etc.).

Definition 6 We use **mobility model** to refer to a probabilistic model that estimates the probability distribution of the user’s locations and it is capable of imitating the user’s behavior (i.e., the model is generative (Bishop and Lasserre 2007), so it can reproduce the user’s daily movements).

3 Related work

Behavior inference is a problem with multiple applications from different fields, like robotics (Billard et al. 2008), autonomous driving (Pomerleau 1989) and computer games (Ontañón et al. 2008). For this reason, it has been studied within several domains, with different terminology, as stated in (Argall et al. 2009) and (Ontañón et al. 2014). Learning from Observation (LfO), Learning from Demonstration, Programming by Demonstration, Imitation learning, Behavioral Cloning (BC) or Behavioral Recognition (BR) are just a few terms that have been used to refer to the problem of behavior inference: extraction and imitation of an individual behavior from observational traces. This approach differs from the Reinforcement Learning strategy, where the behavior is learned through reward optimization instead of directly cloning the patterns in user data. Within the LfO paradigm, the behavior has been modeled through Hidden Markov Models in a real-time strategy video game (Dereszynski et al. 2011), Dynamic Bayesian Networks and Probabilistic Finite Automata to replicate vacuum cleaner robot strategies (Ontañón et al. 2014; Tîrnăucă et al. 2016) or through Neural Networks to imitate a set of tasks (Duan et al. 2017).

In the geolocation context, there are many studies that aim at understanding mobility patterns and mobility behavior. The authors of (González et al. 2008) show that human mobility follows reproducible patterns with regularity; their study is based on the analysis of passive data from Call Detail Records (CDR). Also, (Lu et al. 2013) find that it is possible to obtain very high, but limited, accuracy in predicting location with CDR data, and they accomplish this using Markov models.

In the field of Communication Systems, research about mobility models (see (Munjal et al. 2012) for a survey) attempts to capture human movement in order to build adaptive mobile networks (e.g., Opportunistic Networks (Batabyal and Bhaumik 2015)). These are classified mainly as synthetic (mathematical models that do not extract data patterns) or trace-based (models constructed from real movement traces). All of these models are designed to generate artificial traces, trying to simulate human-like behavior of users as a community. However, the problem we address in this paper requires capturing the particularities of the users behavior in order to be able to later identify each of them from their individual traces.

In pattern recognition from spatio-temporal traces (with GPS and WiFi sources), there are several different approaches for data preprocessing as well as user modeling. One approach uses association rule learning methods to extract patterns and routines (Giannotti et al. 2007; Amirat et al. 2018), representing the user as a set of rules instead of a probabilistic model. (Gambs et al. 2012) use adapted Markov Chains to predict the next location. On the same problem, (Do et al. 2015) apply a probabilistic kernel method to estimate the probability distribution of future location depending on temporal and spatial information. (Liu et al. 2016) propose a Recurrent Neural Network designed specifically for spatio-temporal data and location prediction. (Lv et al. 2017) build two Hidden Markov Models to predict current location at a specific time, and the next destination for a specific location, respectively.

To the best of our knowledge, there are no current studies of mobility from an LfO perspective where the user model aims to replicate the learned behavior from the observations. In this research, we work with continuous spatio-temporal data and, after a preprocessing phase, we train for each user a stochastic model (a Probabilistic Finite Automaton) to replicate the user’s mobility behavior. We test our mobility model on location prediction, as well as on user identification through the comparison of replicated and real traces.

4 Methodology

Our objective is to model user mobility behavior from user location history. To achieve this goal, we first transform the sequence of trajectories into semantic trajectories, out of

$$r * 2 * \arcsin \left(\sqrt{\sin^2 \frac{lat_2 - lat_1}{2} + \cos(lat_1) * \cos(lat_2) * \sin^2 \frac{lon_2 - lon_1}{2}} \right)$$

which our mobility model extracts location routines. The semantic trajectories will allow us to process the behavior with discretized locations in a meaningful way. This transformation includes the steps of stay point detection,

regions-of-interest discovery and time segmentation. After that, we build our mobility model and assess it with two different evaluation metrics, one of them being the usual for location prediction, and the other one being more appropriate for the goal of user identification.

4.1 Detection of stay points

The first step in our methodology is designed to remove, for each trajectory π , transit points (those places covered by the user while moving between locations) and points corrupted with noise (such as GPS errors), therefore keeping only visited places for the remaining of the preprocessing phase. This gives us sequences of points when the user is at home, shopping at the mall, in the workplace, etc. Thus, we reduce data size significantly without losing relevant information of user routines. We address the task of stay points detection with an algorithm initially introduced in (Li et al. 2008) that has a certain relevance in Trajectory Data Mining literature (Lin and Hsu 2014; Zheng 2015; Jiang et al. 2017). The input of the algorithm consists in a trajectory and two parameters: a distance threshold $distLim$ and a time elapsed threshold $timeLim$. This procedure groups every sequence of one or more points within $distLim$ distance with respect to the first of them and as long as the difference between the first point following this sequence and the initial one is greater than $timeLim$, and generates a stay point as the centroid of the group. The two thresholds will be set depending on the application. So, whenever the user moves around the same area enough time, this will be marked as stay point. Our adaptation of this method is presented in Fig. 1. We have decided to keep the number of points in each group as a measure of density for the stay point. Another difference with respect to the original algorithm is that we do not include in the *stay group* the point that exceeds the thresholds in order to avoid possible outliers introducing high noise to the position of the centroid. Also, we allow single points in data to be considered as a potential stay point. This algorithm has a worst-case time complexity of $\mathcal{O}(m^2)$ and a space complexity of $\mathcal{O}(m)$, where m is the number of points in the input trajectory.

The *distance* function in the Stay Points Detection Algorithm is calculated using the haversine formula:

where (lat_1, lon_1) and (lat_2, lon_2) are the coordinates in radians of two geographical points (latitudes range from $-\pi/2$ to $\pi/2$ and longitudes range from $-\pi$ to π) and r is the Earth’s radius (6371 for kilometers). The haversine formula was introduced in (Rios et al. 1797).

Fig. 1 Stay points detection algorithm

```

1: procedure STAYPOINTSDETECTION( $\pi = [(p_1, ts_1), \dots, (p_m, ts_m)]$ ,  $distLim$ ,  $timeLim$ )
2:    $staypoints := \emptyset$ 
3:    $i := 1$ 
4:   if  $m = 1$  then
5:     Add  $(\pi_1.lat, \pi_1.lon, \pi_1.ts, \pi_1.ts, 1)$  to  $staypoints$ 
6:   end if
7:   while  $i < m$  do
8:      $j := i + 1$ 
9:     while  $j \leq m$  do
10:       $dist := distance((\pi_i.lat, \pi_i.lon), (\pi_j.lat, \pi_j.lon))$ 
11:      if  $dist > distLim$  then
12:         $tspan := \pi_j.ts - \pi_i.ts$ 
13:        if  $tspan > timeLim$  then
14:           $lat := \frac{\pi_i.lat + \dots + \pi_{j-1}.lat}{j-i}$ 
15:           $lon := \frac{\pi_i.lon + \dots + \pi_{j-1}.lon}{j-i}$ 
16:          Add  $(lat, lon, \pi_i.ts, \pi_{j-1}.ts, j - i)$  to  $staypoints$ 
17:        end if
18:       $i := j$ 
19:    break
20:  end if
21:   $j := j + 1$ 
22: end while
23:  $i := i + 1$ 
24: end while
25: return  $staypoints$ 
26: end procedure

```

Instead of applying this method to the whole trajectory data from a user directly, or separately to the trajectory of each calendar day in data, we use subtrajectories without too much uncertainty for input, i.e., sequences with no lapses bigger than *maxElapse* without information. This way we avoid making assumptions over long periods of inactivity. Again, the *maxElapse* parameter has to be set such as to serve the needs of a specific application.

Thus, as the final step of the stay point detection phase, we replace every subtrajectory $\pi = [(p_1, ts_1), \dots, (p_m, ts_m)]$ by a sequence $sp = [s_1, \dots, s_n]$ with each stay point as $s_i = (lat, lon, ts_{arv}, ts_{lev}, density)$.

4.2 Discovery of regions of interest

The stay points extracted could be stops at the user's home, workplace, etc. However, that high-level information is not directly observable from the dataset, as our observations only include the latitude, longitude and time stamps of the stay points. We wish to categorize the continuous spatial features into classes (or, equivalently, assign them labels) in order to make the study of the user's location patterns feasible. We can assume that multiple entries of stay points will occur at the user's relevant places. Thus, we generate the classes in terms of data distribution and density. Therefore, we apply the density-based clustering algorithm DBSCAN (*Density Based Spatial Clustering of Applications with Noise*), introduced by (Ester et al. 1996), to the spatial features (latitude and longitude) to detect dense groups of stay points as Regions-of-Interest (ROI)

for the user. This algorithm assigns stay points very close in space to the same cluster. To run the DBSCAN algorithm, it is required to set at least one parameter: the maximum distance radius between points of the same cluster. Optionally, the minimum number of points of a cluster can be specified. The values of these parameters depend on the data and the specific application. The stay points that cannot be grouped in a cluster are considered outliers (we understand that these points are just occasional stops). As a result, with this method, we group visits to the same place at different times with the same label of ROI, and ignore isolated stay points.

The next step is intended to filter irrelevant locations, given that the DBSCAN algorithm retrieves even places visited just a few times (for example, locations from a vacation trip), which cannot possibly lead to valuable patterns in the user's daily routine. For this, we compute the number of different days and the total amount of points for each cluster (we obtain this number by adding together the densities of the stay points in the cluster), and we only keep those clusters that contain enough days and points within.

As a result of this step, the set $SP = \bigcup_j sp^j$ of all stay points of a given user will be partitioned into k clusters C_1, \dots, C_k . Moreover, we denote by c_i the centroid of cluster C_i and by r_i its radius, with i in $\{1, \dots, k\}$.

$$c_i = \left(\frac{\sum_{s \in C_i} s.lat}{|C_i|}, \frac{\sum_{s \in C_i} s.lon}{|C_i|} \right)$$

$$r_i = \max_{s \in C_i} distance(c_i, (s.lat, s.lon))$$

Lastly, to be able to compare different users' behaviors, it is advisable to unify the regions of interest of all users in a single set without duplicated regions. This way, a region label C_i will be the same region of interest for every user. For that purpose, the centroids of all detected regions of interest are grouped together with DBSCAN clustering to establish the full label set $C = \{C_1, \dots, C_K\}$.

4.3 Segmentation by time intervals

Once again, we want to discretize continuous values (in this case, time values) by assigning them different labels. Thus, we group initial time stamps with small differences and facilitate the generation of a mobility model according to daily routines. We introduced in (Salomón et al. 2017) three different strategies to address segmentation based on the time interval of the day: by fixed size values (e.g., each interval of one-hour size), by means of data distribution (e.g., computing percentiles to obtain intervals with an equally distributed number of records) or by means of data density (e.g., using a clustering algorithm). In Example 1 we present the three strategies applied to the same time stamps set.

Example 1 Let us assume that we want to have three time intervals and that the time stamps recorded for one user in one particular day are 10:00, 11:00, 15:00, 21:00, 22:00 and 23:00. Then, the three different approaches mentioned above will result in three distinct intervals (see Fig. 2).

In this work we employ a fixed equally sized segmentation, as we wish to finally accomplish user identification through the mobility models and this approach will give us the same intervals for all the users in the dataset. Thus, we set $\mathcal{I} = [I_1, \dots, I_H]$ to be the partition of the $[0 : 00 : 00, 23 : 59 : 59]$ range into H intervals of equal length. We assign the initial and final time stamps of each stay point to the corresponding intervals that we refer to as I_{arv} and I_{lev} ($s_i \cdot ts_{arv} \in I_{arv}$ and $s_i \cdot ts_{lev} \in I_{lev}$). Therefore, we can work on the same trajectory data with variable granularity, depending on the chosen size of the intervals.

4.4 Mobility model

Once the preprocessing steps are carried out, we can extract user behavior from the obtained semantic trajectories. For this task, we train a Probabilistic Finite Automaton (PFA), which is a 5-tuple $\mathcal{M} = (\Sigma, Q, \phi, \beta, \gamma)$ where Σ is a finite set of symbols, Q is the set of states, $\phi : Q \times \Sigma \times Q \rightarrow [0, 1]$ is the transition probability function (i.e., $\phi(q, s, q')$ will give the emission probability of the symbol s in the transition between the states q, q'), $\beta : Q \rightarrow [0, 1]$ is the initial state probability function and $\gamma : Q \rightarrow [0, 1]$ is the final state probability function.

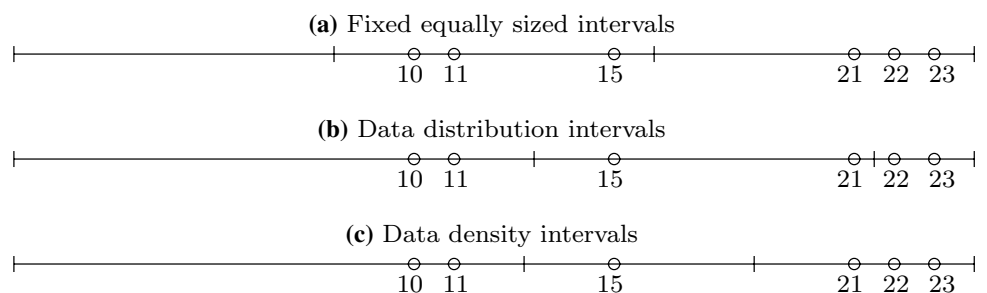
To build a PFA that simulates a user's mobility behavior, we define Q to be the set C of detected regions-of-interest, Σ to be the time intervals \mathcal{I} derived from time segmentation, and ϕ, β, γ to be different probability distributions over the regions: the transition distribution between each pair of regions, the distribution of initial regions on each daily trajectory, and the distribution of final regions on each daily trajectory, respectively. In PFA training we will have to determine those distributions from the sequences of stay points recorded each day by the respective user. More precisely, let us assume that after the preprocessing phase, a trajectory of a given user on day j is mapped into the following sequence of stay points: $\{s_1^j, s_2^j, \dots, s_{k_j}^j\}$. We build the trace T^j of this user on day j by keeping only partial information: $[(I_1^j, C_1^j), \dots, (I_{k_j}^j, C_{k_j}^j)]$, where I_i^j is the time interval to which $s_i^j \cdot ts_{lev}$ belongs, and C_i^j is the cluster to which s_i^j belongs. We thus create sequences of observations T^1, \dots, T^L as learning traces for the L different days in data.

Next, we compute $\#(C, I, C')$, the number of occurrences of transitions between regions C and C' at time interval I , as follows.

$$\begin{aligned} \#(C, I, C') &= |\{(j, i) / I_i^j \\ &= I, C_i^j = C, C_{i+1}^j \\ &= C', 1 \leq j \leq L, 1 \leq i \leq k_j - 1\}| \end{aligned}$$

With the previous definition, we compute transition distribution $\phi(C, I, C')$ by means of the Maximum Likelihood

Fig. 2 Three strategies for time segmentation



Estimation (MLE). We use Laplace smoothing in this distribution to avoid zero probability values.

$$\begin{aligned}\phi(C, I, C') &= \frac{\#(C, I, C') + \alpha}{\sum_{I' \in \mathcal{I}} \sum_{C'' \in \mathcal{C}} (\#(C, I', C'') + \alpha)} \\ &= \frac{\#(C, I, C') + \alpha}{\alpha KH + \sum_{I' \in \mathcal{I}} \sum_{C'' \in \mathcal{C}} \#(C, I', C'')}\end{aligned}$$

Thus, $\phi(C, I, C')$ represents the probability of moving to the next location given the current location and time interval. Also, $\sum_{C' \in \mathcal{C}} \sum_{I \in \mathcal{I}} \phi(C, I, C')$ equals one for any fixed C (i.e., the probability of going anywhere anytime adds up to one for any given origin).

The initial and final state distribution are computed, as well, based on the MLE principle.

$$\begin{aligned}\beta(C) &= \frac{|\{j \in \{1, \dots, L\} / C_i^j = C\}|}{L} \\ \gamma(C) &= \frac{|\{j \in \{1, \dots, L\} / C_{k_j}^j = C\}|}{L}\end{aligned}$$

Note that for $H = 1$ (only one time interval), the PFA is equivalent to a first-order Markov chain.

In Example 2 we show how a PFA describing a user's mobility model is built from semantic trajectories.

Example 2 Assume that we have the following data from three consecutive days of a certain user.

- Day 1: $[(I_1, C_1), (I_1, C_2), (I_2, C_3), (I_3, C_1)]$
- Day 2: $[(I_1, C_1), (I_2, C_3), (I_3, C_1), (I_3, C_1)]$
- Day 3: $[(I_1, C_1), (I_2, C_2), (I_3, C_3)]$

We can observe three different regions of interest (C_1, C_2 and C_3) and three time intervals (I_1, I_2 and I_3), with probability values given by ϕ , in this case calculated without smoothing to make the example easier to follow (see Table 1 and Fig. 3).

For instance, the transition $C_2 \rightarrow C_3$ appears within the intervals I_1 and I_2 with probability $1/2$ and it never appears in the I_3 interval. Although initial states are not indicated, the regions where the user begins the trajectories would be given by the probability distribution $\beta(C_i)$. For the final states, $\gamma(C_i)$ distribution would be used.

Table 1 The values of the probability distribution function ϕ

| | I_1 | C_1 | C_2 | C_3 | I_2 | C_1 | C_2 | C_3 | I_3 | C_1 | C_2 | C_3 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| C_1 | | 0 | 2 | 1 | C_1 | 0 | 0 | 0 | C_1 | 1 | 0 | 0 |
| C_2 | | 0 | 0 | 1 | C_2 | 0 | 0 | 1 | C_2 | 0 | 0 | 0 |
| C_3 | | 0 | 0 | 0 | C_3 | 2 | 0 | 0 | C_3 | 0 | 0 | 0 |

$$\beta = \left[\frac{3}{3}, 0, 0 \right], \gamma = \left[\frac{2}{3}, 0, \frac{1}{3} \right]$$

4.5 Model evaluation

We evaluated the trained mobility model with respect to two different tasks: prediction of the next user location and imitation of user's behavior. The first task is a standard evaluation method for statistical models, where the performance is measured through the accuracy of the model and the error made. The second one is more usual within the LfO field, the goal being to measure the similarity between artificially generated data and real user traces.

For predictive accuracy, the model should guess the next location from the available set of regions of interest. The input features for prediction are represented by the current time interval and current region. The prediction is made for each real stay point in the testing set. Finally, accuracy is computed for each trajectory as $acc = \frac{\# \text{ hits}}{\# \text{ points}}$, and then a user average accuracy (evaluated considering the total number of available trajectories for that user) is used to determine the average accuracy for the model.

While predictive accuracy is a rather usual metric for classification problems, we observe that it is not the ideal measure to assess how well the model can reproduce the user mobility behavior. We can assume user location to be a stochastic phenomenon, as user movements are not deterministic or too complex to be seen as deterministic. For example, if we

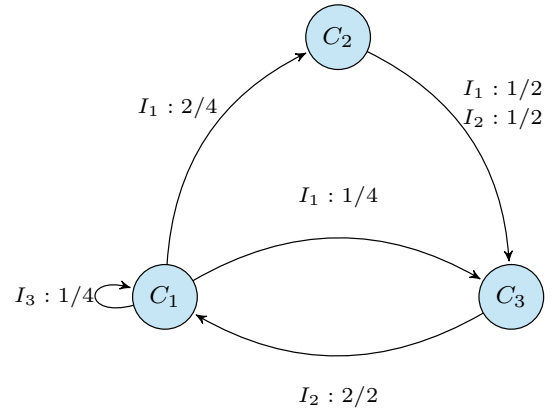


Fig. 3 Example of a PFA mobility model with time intervals $\{I_1, I_2, I_3\}$ and regions $\{C_1, C_2, C_3\}$

consider the stay points after leaving a very frequent region like the user’s home, the next location may be different even in the same time interval. But in this scenario, the location with highest probability will always be the same, and by using predictive accuracy as evaluation metric we implicitly evaluate a stochastic process from a deterministic perspective.

Therefore, we believe that comparing real data with artificially generated data represents a more appropriate test to evaluate a mobility model. Thus, for this second task, the model will generate a new trace for each original daily trajectory through the sampling of its estimated probability distribution. Just like in the earlier case, predicting the next location will be done knowing the current time interval and region of interest. The distance used to measure how much the generated traces depart from the real data is a Monte Carlo approximation of the cross entropy between probability distributions, also known as Vapnik’s risk (Tirnáucă et al. 2016). This metric estimates empirically the divergence between the original data distribution and the one learned by the model in the absence of full knowledge about the exact probability distribution behind the user’s behavior. When the data generated by our model is more similar to the real one, the computed Monte Carlo distance will be smaller. Ideally, the model learned from a given user will generate data at minimal distance to the user’s training data when compared with distances with respect to other user’s data. This way, we can identify, for each trained model, the most similar user, and measure the success at recognizing the correct one. The Monte Carlo distance is calculated with the following formula:

$$\text{dist}(T, T') = -\frac{1}{n} \sum_{i=1}^n \log \left[\frac{1}{m} \sum_{j=1}^m \mathbb{1}_{\{o'_j\}}(o_i) \right]$$

Here, $T = [o_1, \dots, o_n]$ and $T' = [o'_1, \dots, o'_m]$ are two traces to be compared, with observations for each trace $o_i = (I_i, C_i)$ and $o'_j = (I'_j, C'_j)$, $I_i, I'_j \in \mathcal{I}$, $C_i, C'_j \in \mathcal{C}$. $\mathbb{1}_{\{o'_j\}}$ is the indicator function of the tuple o'_j . If we apply Laplace smoothing, the Monte Carlo distance becomes:

$$\text{dist}(T, T') = -\frac{1}{n} \sum_{i=1}^n \log \left[\frac{1}{m} \sum_{j=1}^m \frac{\mathbb{1}_{\{o'_j\}}(o_i) + 1}{m + |\mathcal{C}| \times |\mathcal{I}|} \right]$$

5 Results

We apply the proposed methodology to the *GeoLife GPS trajectory dataset* (Zheng et al. 2009). “*GeoLife*” is an open data repository from a location-based social network, collected by the Urban computing group of Microsoft Research Asia. This dataset provides 18670 trajectories from 182 different users over five years, with a total of over 24 million points. Data is distributed mainly around China and most of the records

are located in Beijing, but there are also trajectories located in Europe and USA. It contains multiple kinds of activities like life routines, shopping, traveling and sports. The dataset includes the features of latitude and longitude (in decimal degrees), altitude (in feet) and date-time values (UTC).

In the next subsections we describe the parameter selection for our methodology, some remarks on the preprocessing results, and the evaluation of the mobility model with both predictive accuracy and Monte Carlo distance. The PFA mobility model is assessed against three other benchmark models for comparison purposes: a random choice of the user’s regions (referred to as RND), the most frequent region choice (referred to as MODE) and a first-order Markov chain model (referred to as MC) as one of the most used state-of-the-art models (Lu et al. 2013; Jahromi et al. 2016).

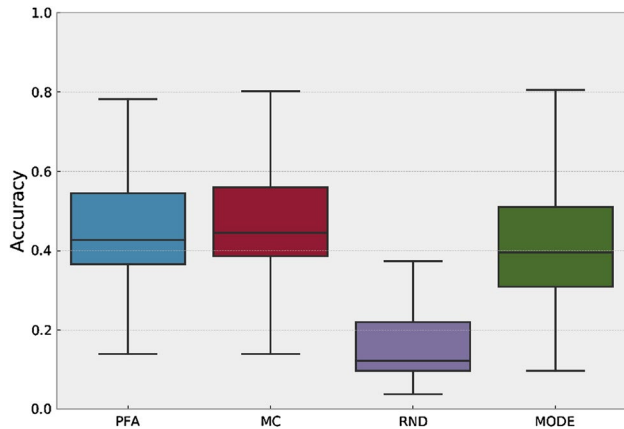
5.1 Preprocessing and parameter selection

Since the altitude sometimes presents high levels of noise, we discard it and keep the latitude, longitude and time stamp variables. We also remove instances with missing data (if any) and points where one of the coordinates has an invalid value ($-180 \leq \text{longitude} \leq 180$, $-90 \leq \text{latitude} \leq 90$). Finally, we convert the time value in the date field to use the time zone from Beijing (UTC+8), since this is the place that most of the data comes from.

Choosing the right values for all the parameters involved is by far not a trivial task. For the Stay Points Detection algorithm, we have experimented with different thresholds (some used in the literature, others chosen by us), and finally fixed *distLim* to 200 m and *timeLim* to 10 min (these values were suggested in (Jiang et al. 2017)), which avoid capturing false stay points (e.g., when waiting at traffic lights), while keeping a reasonable amount of interesting regions. The data is split by a *maxElapse* value of 6 h (which means that if we do not have data for more than 6 h, the algorithm continues with the next subsequence). In identification and unification of regions-of-interest, we choose experimentally a maximum radius of 50 m for DBSCAN, since larger values tend to generate clusters that are too wide. Given that we want to ignore irrelevant regions, we also set five (because most countries have a five day workweek) as the minimum number of points to form a cluster, as afterwards we use a minimum number of five days to consider relevant regions (and a cluster with less than five points cannot possibly have enough different days). For this dataset, we do not use a threshold for the minimal amount of points in each cluster because some users have very few data. In time segmentation, we choose to have eight intervals ($H = 8$), as we consider this number of intervals to provide enough detail capturing user routines. Higher number of intervals would require much more data in order to avoid having mostly “close-to-zero” probability distribution values, and a smaller number may oversight the

Table 2 Summary of the preprocessing results

| | Min | Max | Mean | Median |
|----------------------------------|-------|---------|----------|---------|
| Number of available days (L) | 1 | 1251 | 61.275 | 21 |
| Number of initial points (M) | 17 | 2156994 | 136686.7 | 35182 |
| Mean sampling period (secs) | 1.128 | 547.348 | 65.239 | 14.623 |
| Number of stay points (N) | 0 | 2494 | 186.3 | 49 |
| Number of clusters (k) | 0 | 30 | 2.49 | 0 |
| Compression ratio N/M | 0.0 | 0.0678 | 0.00692 | 0.00163 |
| Compression ratio k/N | 0.0 | 0.0625 | 0.00795 | 0.0 |

**Fig. 4** Accuracy of each model (from left to right: PFA, MC, RND and MODE) on the prediction of the next location

user’s movement habits. Finally, we keep those users with enough data (i.e., at least three different regions-of-interest and ten different days).

As one can see from Table 2, there is a high heterogeneity in the data. Initially, there are users with very few days (the minimum being one) or with more than three years of data (maximum is 1251 days). Also, the number of initial points varies between 17 and more than two millions. After the preprocessing phase, the resulting dataset is considerably reduced, the maximum number of stay points for a user amounting to almost 2500 and the number of classes not exceeding 30 per user.

Table 3 Example users with significant accuracy differences between MC and PFA and best accuracy scores

| User | L | M | N | k | MODE | MC | PFA |
|----------|-----|---------|------|-----|-------|--------------|--------------|
| u_3 | 237 | 485226 | 1348 | 21 | 0.565 | 0.570 | 0.586 |
| u_5 | 62 | 109046 | 247 | 4 | 0.805 | 0.802 | 0.673 |
| u_{35} | 69 | 312042 | 412 | 9 | 0.295 | 0.580 | 0.715 |
| u_{41} | 138 | 1057043 | 910 | 12 | 0.745 | 0.746 | 0.709 |
| u_{44} | 52 | 76846 | 187 | 3 | 0.709 | 0.713 | 0.782 |
| u_{71} | 61 | 123850 | 142 | 3 | 0.778 | 0.796 | 0.668 |

Example users with significant accuracy differences between MC and PFA (maximum score between the two models in bold)

5.2 Model analysis

The evaluation of predictive accuracy of the next location for all the compared models give us the results shown in Fig. 4.

Here, we present the mean accuracy for all users of each model. We can see that, while MODE already significantly improves RND accuracy, the best results are obtained from the Markov model and the PFA model. The relatively high accuracy obtained using MODE is due to the fact that the most common region (as could be the user’s home or workplace) appears very frequently in most of the users of the dataset. Between MC and PFA models we see that the Markov model reaches a higher mean accuracy, but both get very similar results with a significant variance in the accuracy. We find by inspecting the data that the reason for this is user and data heterogeneity. While the MC and the MODE models work better in the case of more regular users, the PFA seems to have higher predictive ability for more complex users. This is shown in Table 3, where we present six users with relevant accuracy differences between MC and PFA, and with the best three accuracy results for each model.

For the user identification evaluation, we first compute the Monte Carlo distances between the user’s real traces and the data generated by each model. In Fig. 5 we observe the mean distances distributions. In this case, PFA and MC give better and almost identical distance results.

In Fig. 6 we can see the results obtained when performing user identification.

The diagonal of the distance matrix between the user (on the abscissa axis) and the trained model (on the ordinate axis) represents correct user identification. The identification outcomes of the two models are very similar. Note that the distance matrix is nearly-symmetric, which means that if the artificially generated trace from the model trained using data from user k is similar to the trace of user h , then the converse also holds (i.e., the artificially generated trace from the model trained using data from user h is similar to the trace of user k).

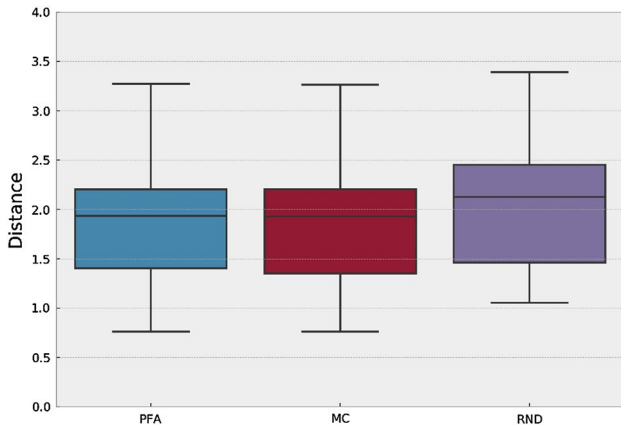


Fig. 5 Cross entropy between each model (from left to right: PFA, MC and RND) and real data distribution

In Table 4 we present the identification results for each model according to the identified and misidentified users.

6 Conclusions and future work

Geolocation technology in our daily lives is a valuable source of information on our routines and life style preferences. Nevertheless, these records are quite sensitive to noise and tend to reach huge dimensions. In this work, we present a methodology to preprocess this type of data in order to infer hidden and useful information such as relevant regions-of-interest. From our automatic data transformation,

we build a mobility model able to predict the user's next location and to generate new location traces by means of the estimated probability distribution given by a Probabilistic Finite Automaton. Besides standard predictive accuracy, we propose another evaluation metric using an approximation of the cross entropy between user data distribution and the artificially generated data from the mobility model. This alternative metric is more useful for performing user identification based on the similarity between users and models. We compared our model with a first-order Markov model, observing many similarities in user identification accuracy (recall that the Markov model is equivalent to a PFA trained with a unique time interval), while the PFA turned out to be more precise whenever the user's routines were dependent on the specific time interval. For ubiquitous environments, our methodology allows technology to recognize the user while unobtrusively tracking his moves.

As future work, we plan to include other latent variables in our methodology, such as the user activity or intentionality. Additionally, we would like to perform a more in-depth analysis about the user heterogeneity and regularity.

Table 4 User identification by each model

| | MC | PFA |
|------------------------|------|------|
| Correct identification | 39 | 41 |
| Wrong identification | 5 | 3 |
| Accuracy | 0.89 | 0.93 |

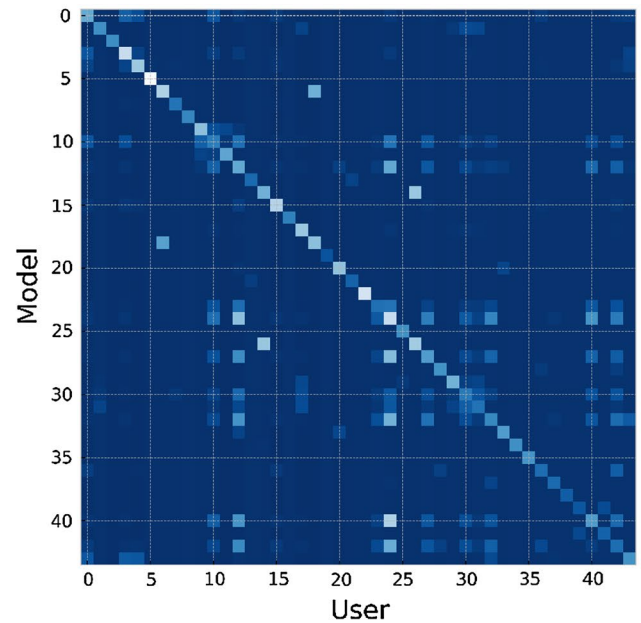
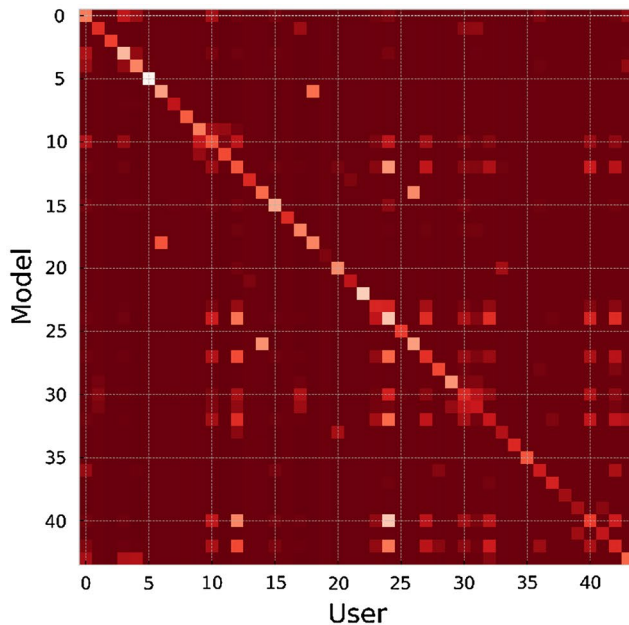


Fig. 6 Distance matrix for user identification with the MC model (left) and the PFA model (right). The color indicates the distance value (brighter means lower)

Acknowledgements The authors gratefully acknowledge the financial support from FEDER (Fondo Europeo de Desarrollo Regional) and SODERCAN (Sociedad para el Desarrollo Regional de Cantabria) for the project TI16-IN-007 within the program “I+C=+C 2016—PROYECTOS DE I+D EN EL ÁMBITO DE LAS TIC, LÍNEA SMART”, and from Ministerio de Ciencia e Innovación (MICINN), Spain for the project PAC::LFO (MTM2014-55262-P).

References

- Amirat H, Benslimane A, Fournier-Viger P, Lagraa N (2018) Locrec: Rule-based successive location recommendation in lbsn. In: 2018 IEEE International Conference on Communications (ICC), pp 1–6
- Argall BD, Chernova S, Veloso M, Browning B (2009) A survey of robot learning from demonstration. *Robot Auton Syst* 57(5):469–483
- Batabyal S, Bhaumik P (2015) Mobility models, traces and impact of mobility on opportunistic routing algorithms: a survey. *IEEE Commun Surv Tutor* 17(3):1679–1707
- Billard A, Calinon S, Dillmann R, Schaal S (2008) Robot programming by demonstration. Springer, Berlin, Heidelberg, pp 1371–1394
- Bishop CM, Lasserre J (2007) Generative or discriminative? Getting the best of both worlds. Oxford University Press, Oxford, pp 3–24
- Cook DJ, Augusto JC, Jakkula VR (2009) Ambient intelligence: technologies, applications, and opportunities. *Pervasive Mob Comput* 5(4):277–298
- Derezynski E, Hostetler J, Fern A, Dietterich T, Hoang TT, Udarbe M (2011) Learning probabilistic behavior models in real-time strategy games. In: Proceedings of the seventh AAAI conference on artificial intelligence and interactive digital entertainment, AAAI Press, AIIDE’11, pp 20–25
- Do TMT, Dousse O, Miettinen M, Gatica-Perez D (2015) A probabilistic kernel method for human mobility prediction with smartphones. *Pervasive Mob Comput* 20(C):13–28
- Duan Y, Andrychowicz M, Stadie B, Jonathan Ho O, Schneider J, Sutskever I, Abbeel P, Zaremba W (2017) One-shot imitation learning. In: Advances in neural information processing systems 30, Curran Associates, Inc., pp 1087–1098
- Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the second international conference on knowledge discovery and data mining, AAAI Press, KDD’96, pp 226–231
- Feng Z, Zhu Y (2016) A survey on trajectory data mining: techniques and applications. *IEEE Access* 4:2056–2067
- Gambis S, Killijian MO, del Prado Cortez MNN (2012) Next place prediction using mobility markov chains. In: Proceedings of the first workshop on measurement, privacy, and mobility, ACM, New York, NY, USA, MPM ’12, pp 3:1–3:6
- Giannotti F, Nanni M, Pinelli F, Pedreschi D (2007) Trajectory pattern mining. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, New York, NY, USA, KDD ’07, pp 330–339
- González M, Hidalgo C, Barabási A L (2008) Understanding individual human mobility patterns. *Nature* 453(June):779–782
- Jahromi KK, Zignani M, Gaito S, Rossi GP (2016) Simulating human mobility patterns in urban areas. *Simul Model Pract Theory* 62:137–156
- Jiang S, Ferreira J, Gonzalez MC (2017) Activity-based human mobility patterns inferred from mobile phone data: a case study of singapore. *IEEE Trans Big Data* 3(2):208–219
- Li Q, Zheng Y, Xie X, Chen Y, Liu W, Ma WY (2008) Mining user similarity based on location history. In: Proceedings of the 16th ACM SIGSPATIAL international conference on advances in geographic information systems, ACM, New York, NY, USA, GIS ’08, pp 34:1–34:10
- Lin M, Hsu WJ (2014) Mining gps data for mobility patterns: a survey. *Pervasive Mob Comput* 12:1–16
- Liu Q, Wu S, Wang L, Tan T (2016) Predicting the next location: a recurrent model with spatial and temporal contexts. In: Proceedings of the Thirtieth AAAI conference on artificial intelligence, AAAI Press, AAAI’16, pp 194–200
- Lu X, Wetter E, Bharti N, Tatem AJ, Bengtsson L (2013) Approaching the limit of predictability in human mobility. *Sci Rep* 3:2923
- Lv Q, Qiao Y, Ansari N, Liu J, Yang J (2017) Big data driven hidden markov model based individual mobility prediction at points of interest. *IEEE Trans Veh Technol* 66(6):5204–5216
- Munjal A, Camp T, Aschenbruck N (2012) Changing trends in modeling mobility. *J Electric Comput Eng* 2012:372572:1–372572:16
- Ontañón S, Mishra K, Sugandh N, Ram A (2008) Learning from demonstration and case-based planning for real-time strategy games. In: Prasad B (eds) Soft computing applications in industry. Studies in fuzziness and soft computing, vol 226. Springer, Berlin, Heidelberg
- Ontañón S, Montaña JL, Gonzalez AJ (2014) A dynamic-bayesian network framework for modeling and evaluating learning from observation. *Expert Syst Appl* 41(11):5212–5226
- Pomerleau DA (1989) Alvin: an autonomous land vehicle in a neural network. *Adv Neural Inf Process Syst* 1:305–313
- Rios JDMY, Banks J, Cavendish H (1797) Recherches sur les principaux problemes de l’astronomie nautique. Par Don Josef de Mendoza y Rios, F. R. S. Communicated by Sir Joseph Banks, Bart. K. B. P. R. S. *Philos Trans Royal Soc London Series I* 87:43–122
- Roxin A, Gaber J, Wack M, Nait-Sidi-Moh A (2007) Survey of wireless geolocation techniques. In: 2007 IEEE Globecom Workshops, pp 1–9
- Salomón S, Tîrnăucă C, Duque R, Montaña JL (2017) Daily routines inference based on location history. In: Ubiquitous computing and ambient intelligence. Springer International Publishing, Cham, pp 828–839
- Tîrnăucă C, Montaña JL, Ontañón S, González AJ, Pardo LM (2016) Behavioral modeling based on probabilistic finite automata: an empirical study. *Sensors* 16(7):958
- Zheng Y (2015) Trajectory data mining: an overview. *ACM Trans Intell Syst Technol* 6(3):29:1–29:41
- Zheng Y, Chen Y, Xie X, Ma WY (2009) Geolife2.0: A location-based social networking service. In: 2009 tenth international conference on mobile data management: systems, services and middleware, pp 357–358