Neurocomputing 173 (2016) 1142–1153

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

The discovery of personally semantic places based on trajectory data mining

Mingqi Lv [a,b], Ling Chen [b,*], Zhenxing Xu [b], Yinglong Li [a], Gencai Chen [b]

[a] College of Computer Science, Zhejiang University of Technology, Hangzhou 310023, PR China
[b] College of Computer Science, Zhejiang University, Hangzhou 310027, PR China

ARTICLE INFO

Article history:
Received 8 October 2014
Received in revised form
26 August 2015
Accepted 26 August 2015
Communicated by Dr. T. Heskes
Available online 3 September 2015

Keywords:
Trajectory data mining
Place extraction
Place recognition
Location-aware computing

时间、空间

ABSTRACT

A personally semantic place is a space that is frequently visited by an individual user and carries important semantic meanings (e.g. home, work, etc.) to the user. Many location-aware applications could be greatly enhanced by the ability of automatic discovery of personally semantic places. The discovery of a user's personally semantic places involves obtaining the physical locations and semantic meanings of these places. In this paper, we propose approaches to address both of the problems. For the physical place extraction problem, a hierarchical clustering algorithm is proposed to firstly extract visit points from the GPS trajectories, and then clusters these visit points to form physical places. For the semantic place recognition problem, the temporal, spatial and sequential features in which the places have been visited are explored to categorize them into pre-defined types. An extensive set of experiments conducted based on a dataset of real-world GPS trajectories has demonstrated the effectiveness of the proposed approaches.

1. Introduction

With the prevalence of mobile devices with positioning ability, there have been an increasing number of location-aware applications. These applications provide users a rich set of location-based services, e.g., directory services (finding the location of the nearest gas station), navigation services (providing the routes to the destination location), geo-coding services (mapping a postal address to a location), social networking services (detecting the locations of your friends), etc. The key element of these applications is "location". However, instead of latitude and longitude coordinates, many emerging location-aware applications require a more semantic notion: "place", which is a human-readable label of space, e.g., home, work, restaurant, etc. Harrison and Dourish [1] have highlighted the critical distinction between place and space, and pointed out that place, rather than space, is a fundamental concept in application design. Using the notion "place" can help to link geographic information to the users' actual needs [2], and enable these location-aware applications to act more intelligently and adaptively. For example, spatial query system can provide contextualized answers according to the places of the users [3], location-based reminder can associate a to-do list with specific places [4], location-based recommender can recommend interesting places [5], route planner can navigate the pedestrian to the destination through a sequence of personalized places [6], location-based messaging system can allow messages to be delivered to specific places [7], location prediction system can predict the user's destination when beginning a trip [8], etc.

According to [9], the sense of place refers to the fact of understanding the properties of places, i.e., their spatial characteristics and social meanings. Thus, a personally semantic place could be defined as a space that is frequently visited by an individual user and carries important semantic meanings to the user, and the discovery of a user's personally semantic places can be divided into two sub-problems, i.e., obtaining the physical locations of the places (i.e., physical place extraction) and assigning semantics to these places (i.e., semantic place recognition). For the former problem, many approaches have been exposed to extract significant locations of users from their historical trajectories [10–14]. However, these significant locations are represented by a geographical point, a point plus radius or a geographical region without semantic meanings. For the latter problem, methods like manual labeling [15,16], reverse geo-coding [17,18] and customized geographic database query [13,19,20] are adopted to give semantic meanings to locations. The shortcomings of these methods are as follows: for manual labeling, it lays extra interaction burden on users, and thus does not scale well when the number of locations becomes large. For reverse geo-coding, the obtained semantics is always represented as postal address (e.g., x Road, y City), which is often as challenging to interpret as raw locations. For customized geographic database query, the personal meaning (e.g., home,

* Corresponding author. Tel.: +86 13606527774.
E-mail addresses: mingqilv@zjut.edu.cn (M. Lv), lingchen@cs.zju.edu.cn (L. Chen), xuzhenxing2010@163.com (Z. Xu), liyinglong@zjut.edu.cn (Y. Li), chengc@zju.edu.cn (G. Chen).

http://dx.doi.org/10.1016/j.neucom.2015.08.071
0925-2312/

workplace) of a place is almost impossible to be drawn. Besides, the same place may carry different personal meanings to different users. Since a place can be roughly classified based on the activity performed there [21], we prefer an approach which can automatically estimate the semantic meanings of personal places by categorizing them into several pre-defined types.

Since GPS is often preferred over its alternatives (e.g., GSM/Wi-Fi based positioning systems) for the above mentioned applications because it is known to be more accurate [22], this paper proposes an approach for automatically discovering a user's personally semantic places (including physical place extraction and semantic place recognition) from the user's GPS trajectory. Our contributions in this paper are summarized as follows:

1. For physical place extraction, a hierarchical clustering algorithm which leverages the advantages of both time-based clustering (which is simple and can work in an incremental way on mobile devices) and distance-based clustering (which can accommodate arbitrary shapes of places) is designed to effectively extract significant locations from GPS trajectory.
2. For semantic place recognition, a comprehensive recognition approach which lies upon the hierarchical clustering algorithm and exploits the temporal, spatial and sequential features of the extracted physical places is proposed to categorize them into pre-defined types.

The remainder of this paper is organized as follows: Section 2 gives a survey of the related work. In Section 3, we present the architecture of our approach. Section 4 proposes the hierarchical clustering framework for physical place extraction. Section 5 details the techniques for semantic place recognition. The evaluation of the approach and the experimental results are reported in Section 6. Finally, we conclude our work and give some future work in Section 7.

2. Related work

2.1. Physical place extraction

Previous works on physical place extraction can be generally divided into two groups, i.e., fingerprint-based approaches and geometry-based approaches. Fingerprint algorithms [23,24] detect stable radio environment that indicates a stay at a place, and the fingerprint of the place is collected during the stay as a vector of visible radio beacons (e.g., cell towers, WiFi APs, etc.). The collected fingerprint is then used to recognize when the user returns to the place. Fingerprint-based algorithms could obtain room-level places because they use data from pervasive radio beacons which have wider coverage in cities than that of GPS signals. However, the major drawback of fingerprint-based approaches is that the physical location cannot be obtained. Although some databases (e.g., PlaceLab [25]) containing the physical location of radio beacons are created based on war driving techniques, the coverage is still limited in practice, especially in rural areas and in the developing countries. Thus, fingerprint-based approaches may not be suitable for many above mentioned applications, where the physical locations of places must be known.

几何方法 Geometry-based approaches represent places by points, circles or polygons based on physical locations (e.g., GPS coordinates). Most existing geometry approaches apply clustering algorithms to find places. These clustering algorithms can be roughly divided into two categories, i.e., point based clustering and trajectory based clustering. For example of point based clustering, Ashbrook and Starner [10] used a variant of K-Means clustering algorithm to cluster the locations (where the GPS signal lost and reappeared after a pre-defined interval) into places. Zhou et al. [12] developed DJ-Cluster, a density-based clustering algorithm, to discover places of arbitrary shape. However, these algorithms do not take the temporal continuity of trajectories into account, and this shortcoming may cause them fail to find some places (e.g., outdoor places where GPS signal is not lost, indoor places with sparse GPS points, etc.). On the other hand, trajectory based clustering algorithms cluster locations by taking advantage of the temporal continuity of trajectories. For example, Kang et al. [11] designed a time-based clustering algorithm to incrementally extract places along the time axis of a trajectory. Palma et al. [13] proposed CB-SMoT, a speed-based clustering algorithm, to detect places which are parts of a trajectory where the speed is lower than in other parts of the same trajectory. However, these algorithms work with single trajectories, and the problem of whether multiple places in different trajectories are the same is not well solved (by simply comparing their distance with a threshold [11] or judging whether they intersect with each other [13]). Besides, the GPS signal loss problem may disturb the continuity of locations in a trajectory, and thus produce false negative results.

2.2. Semantic place recognition

To assign semantic meanings to physical places, location-aware applications such as Reno [15] and Connecto [16] allow users to manually input meaningful labels by interacting with their interfaces, and this information may be further contributed to a global place database for reuse. Apparently, manual labeling requires a great calibration effort, and thus does not scale well when the number of places increases. Some existing works transform physical locations to semantic labels by using reverse geo-coding techniques [17,18]. However, the return of reverse geo-coding services (e.g., Google Map) for a given location is its postal address, which is often difficult to interpret. To acquire colloquial place labels, customized POI (Point of Interest) databases which store the physical locations and semantic meanings of landmarks are used [13,20]. However, these databases only contain the information of public places without personal meaning. For example, a user's home or workplace cannot be identified by querying these databases. Besides, even the same place may have different personal meanings to different users, e.g., a customer has dinner in a restaurant whereas a cook works there.

To estimate personal meanings of places, Liao et al. [26] used relational Markov networks (RMN) to recognize high-level human activities associated with significant locations. The RMN model is extended to incorporate a variety of features including temporal information, spatial information, and global constraints for location-based activity recognition. This approach estimates the semantic meanings of activities performed at each individual significant location. However, the semantic meaning of a place is more sophisticated than that of activity, and it often requires to be visited multiple times before its semantic meaning can be accurately estimated. On the other hand, our approach firstly extracts physical places which may be visited multiple times, and then feeds the mining results (which capture the statistical temporal, spatial and sequential features of these places) into the semantic place recognition model. We have also exploited temporal and spatial information for estimating personal meanings of places in our previous work [31]. Since people usually have certain sequential regularities to visit different types of places, we use sequential patterns as additional information to further improve the estimation accuracy in this paper.

In conclusion, we summarize the difference of our method and the existing work in Table 1.

Table 1
A summary of the difference of our method and the existing work.

Category	Method	Feature
Physical place extraction	Fingerprint detection [23,24]	Detecting relative places (i.e., without physical locations) based on wireless signal fingerprints (e.g., WiFi signal)
	Point based clustering [10,12]	Extracting physical places by directly clustering all the locations in all the trajectories
	Trajectory based clustering [11,13]	Extracting physical places by taking advantage of the temporal continuity of trajectories
	Our method	Extracting physical places based on a two-step clustering by taking into account the signal loss problem of GPS
Semantic place recognition	Manual input [15,16]	Requiring users to manually input semantic place labels
	Reverse geo-coding [17,18]	Transforming physical locations to postal address based on reverse geo-coding technique
	Customized POI database [13,20]	Creating a database that contains the physical locations and semantic labels of places
	Our method	Classifying places into pre-defined categories by exploiting temporal, spatial and sequential features in which they have been visited

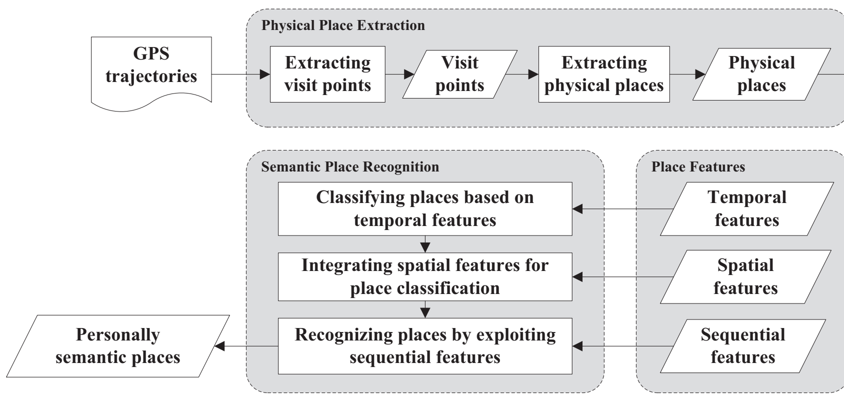


Fig. 1. The system architecture of the personally semantic place discovery approach.

3. System architecture

Fig. 1 gives an overview of the system architecture of our approach, including two major steps, i.e., physical place extraction and semantic place recognition. Given the GPS trajectories of a specific user, our approach firstly uses a hierarchical clustering algorithm to extract the physical places (from GPS trajectories to visit points, and from visit points to physical places), whose particular properties are calculated at the same time. Then, different techniques are employed to explore the place properties for estimating the personal semantics of the extracted physical places, i.e., classification technique for temporal features, geographic database for spatial features, HMM (Hidden Markov Model) for sequential features.

4. Physical place extraction

This paper proposes a hierarchical clustering algorithm to extract physical places where the users have visited based on a three layered model. As shown in Fig. 2, the lowest level is the GPS trajectory which contains all the GPS points sorted by timestamp (see Definition 1), the middle level contains all the visit points (see Definition 2), and the highest level represents the physical places (see Definition 3). The hierarchical clustering algorithm takes the GPS trajectory as input and conducts a time-based clustering to identify visit points, and then extracts physical places from these visit points based on a distance-based clustering.

(Definition 1.) GPS point and GPS trajectory: a GPS point is a pair $p=(lng, lat)$, representing the longitude–latitude location. A GPS trajectory is a sequence of pairs $Traj=<(p_0, t_0),\ldots,(p_n, t_n)>$, in which p_k is a GPS point and t_k $(k=0,\ldots,n)$ is a timestamp $(\forall 0 \le k < n,\ t_k < t_{k+1})$.

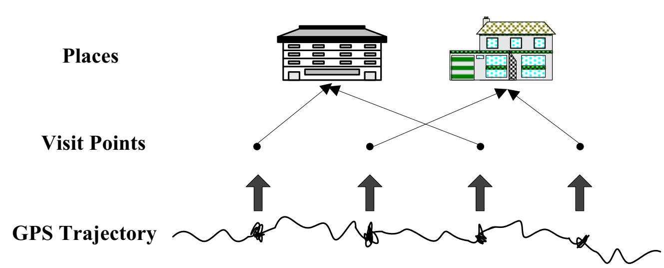


Fig. 2. A three-layered model for physical place extraction.

(Definition 2. Visit point: a visit point is a triple $VP=(p, t_{in}, t_{out})$, where p is a GPS point, t_{in} and t_{out} are timestamps, and the visit point stands for a location p around which the user stays for longer than a time threshold (i.e. $t_{out} - t_{in} > \delta_{time}$).

(Definition 3.) Physical place: a physical place is a collection of visit points $P=\{VP_1,\ldots,VP_n\}$, in which $VP_1.p, \ldots, VP_n.p$ are close to each other.

4.1. Extracting visit points

增量

We use a time-based clustering algorithm to incrementally form location clusters (i.e., a set of GPS points that are spatially close to each other) and identify visit points from these clusters. The criterion of existing time-based clustering algorithm for considering a cluster to be a visit point is that the cluster's time duration should be longer than a threshold δ_{time} [11]. However, this criterion may not work well for GPS trajectories with discontinuous points sampling due to the

GPS signal loss problem. For example, if a user enters a large building from one side A and leaves from another side B (as shown in Fig. 3(a)), the GPS signal will be blocked and the GPS points recorded around A and those recorded around B will form two different clusters (i.e., clusters I and II) because the locations around A are not close enough to those around B. Apparently, neither I nor II can be identified as visit point even if the person stays a long time in the building, and thus produce false negative results. This problem may also exist even when the user enters and leaves the building from the same side, because GPS device often require a period of time to receive signal when the user may leave the building for a relatively long distance (as shown in Fig. 3(b)). We call this problem as the *entrance and exit deviation problem*. 偏差

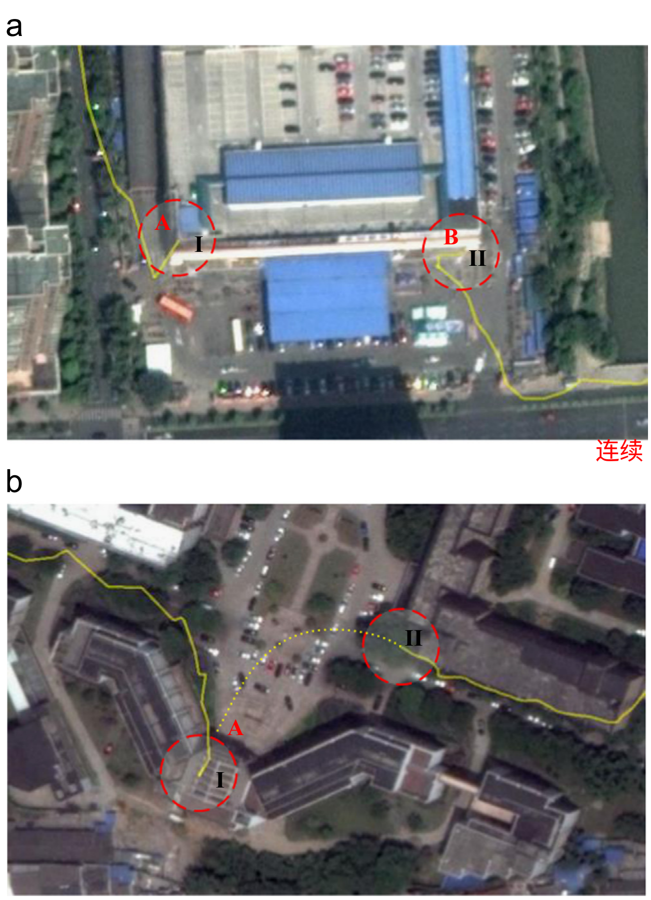
To counter this problem, we refine the existing time-based clustering algorithm to adapt to the discontinuous characteristics of GPS trajectories. The algorithm (as depicted in Algorithm 1) works in an incremental way and processes the GPS points along the time axis. In the algorithm, *CC* and *PC* are the current cluster and the previous cluster, respectively. For each GPS point in *T*, the algorithm compares the distance between it and the centroid of the current cluster with $\delta_{\text{cluster\_distance}}$. If the distance is less than $\delta_{\text{cluster\_distance}}$, this GPS point is added to the current cluster (lines 2–4). Otherwise, the algorithm checks the time duration of the current cluster. If the time duration is longer than δ_{time}, the current cluster is considered as a visit point (lines 6–8). If the time duration is not long enough, the algorithm does not simply ignore it, but checks the time interval (i.e., the interval between the exit time of the previous cluster and the enter time of the current cluster) and the distance (i.e., the distance between the centroids of the previous and the current clusters). If the time interval is longer than δ_{time} and the distance is less than $\delta_{\text{tolerated\_distance}}$, the algorithm combines these two clusters and treats the result as a visit point (lines 10–12).

递增方式

当前集群的质心

Algorithm 1. Visit Point Extraction

INPUT: GPS trajectory *T*, clustering distance threshold $\delta_{\text{cluster\_distance}}$, time threshold δ_{time}, and tolerated distance threshold $\delta_{\text{tolerated\_distance}}$
OUTPUT: A set of visit point *VP*
1: current cluster $CC=\emptyset$, previous cluster $PC=\emptyset$, $VP=\emptyset$
2: **for** each GPS point p_i in *T* **do**
3:   **if** distance(CC, p_i) $< \delta_{\text{cluster\_distance}}$ **then**
4:     Append p_i to *CC*
5:   **else**
6:     **if** duration(CC) $> \delta_{\text{time}}$ **then**
7:       Append *CC* to *VP*
8:       $CC=\emptyset$, $PC=\emptyset$
9:   **else**
10:     **if** interval(CC, PC) $> \delta_{\text{time}}$ **and** distance(CC, PC) $< \delta_{\text{tolerated\_distance}}$ **then**
11:       CC=combine(CC, PC), and Append *CC* to *VP*
12:       $PC=\emptyset$
13:     **else**
14:       $PC=CC$
15:       $CC=\emptyset$
16: **end for**



连续

Fig. 3. Real-world examples of the entrance and exit deviation problems (the solid yellow line represents true GPS trajectories, and the dotted yellow line represents mobility that has not been recorded by GPS). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

When using the algorithm in practice, the parameter $\delta_{\text{tolerated\_distance}}$ should always be set larger than $\delta_{\text{cluster\_distance}}$ in order to tolerate the entrance and exit deviation problem. By using $\delta_{\text{tolerated\_distance}}$, the entrance and exit deviation problem can be greatly alleviated when the GPS sampling is interrupted for a long period of time between two areas which are not far from each other. As the example shown in Fig. 3, cluster I (i.e., GPS points around the entrance) and cluster II (i.e., GPS points around the exit) can be combined as a visit point by using our algorithm as long as they are not too far from each other. Another problem is that it is difficult for the user to specify good absolute values for 绝对值 the parameters without the knowledge of the characteristics of the trajectory. Thus, we use a relative parameter for estimating the absolute values of $\delta_{\text{cluster\_distance}}$ and $\delta_{\text{tolerated\_distance}}$. It regards the distance between consecutive GPS points in a trajectory as a ran- 连续 dom variable *d*, and defines a probability based parameter *area* (with value between 0 and 1). Then, the absolute value of a parameter could be calculated based on the quantile function as $\inf\{d: area \le CDF(d)\}$, where $CDF(d)$ is the cumulative distribution function of the distance variable *d*. In our algorithm, we use two probability based parameters $p_{\text{cluster\_distance}}$ and $p_{\text{tolerated\_distance}}$ to estimate $\delta_{\text{cluster\_distance}}$ and $\delta_{\text{tolerated\_distance}}$ respectively.

4.2. Extracting physical places

The output of the visit point extraction algorithm is a set of clusters, each of which represents a visit point $VP=(p, t_{\text{in}}, t_{\text{out}})$, where *p* is the centroid of the cluster (represented by a GPS point), t_{in} and t_{out} are the timestamps of the first and the last GPS points of the cluster respectively. Because one place may be visited multiple times, we create clusters of visit points to represent physical places as Definition 3.

We use a distance-based clustering algorithm to extract physical places from visit points. The algorithm can form clusters of arbitrary shape like density-based clustering algorithms (e.g., DBSCAN) do. The reason we do not directly apply density-based clustering algorithms is that clusters of extremely low density (i.e., places which have been visited few times) should not be discarded in our condition. Given all GPS points *PS* and the distance parameter *Eps*, the *Neighbor* of a GPS point and the *Joinable* relationship of two neighbors are defined as follows:

(Definition 4.) Neighbour: the neighbour N of a GPS point p, denoted by $N(p)$, is the set of GPS points q in *PS*, such that: the distance of p and q is less than *Eps*.

(Definition 5.) Joinable: we say that $N(p)$ is joinable with $N(q)$, if there exists a GPS point k in *PS*, such that: both $N(p)$ and $N(q)$ contain k.

The algorithm for visit point clustering is depicted in Algorithm 2. In the algorithm, for each point in V, the algorithm calculates its neighbour (lines 2–3). Then it checks all the existing clusters to find those which are joinable with this neighbour, and combines them together (lines 4–7). Finally, the result is appended to the set of clusters. After clustering, the scope of a physical place could be represented as the convex hull of its visit points.

Algorithm 2. Visit Point Clustering.

INPUT: A set of the centroids of visit points V, and distance parameter *Eps*
OUTPUT: A set of visit point clusters *CS*
1: $CS=\emptyset$
2: **for** each point p_i in V **do**
3: N=neighbour(p_i, *Eps*)
4: **for** each cluster C in *CS* **do**
5: **if** is_joinable(N, C) **then**
6: N=combine(N, C)
7: Remove C from *CS*
8: **end for**
9: Append N to *CS*
10: **end for**

5. Semantic place recognition

We estimate the semantic meanings of the extracted physical places by classifying them into several pre-define types. The proposed approach exploits the temporal, spatial and sequential features in which these places have been visited for the classification task.

5.1. Classifying places based on temporal features

Since the addresses returned by reverse geo-coding services carry little personal semantic meanings to individuals, we seek to automatically assign semantic meanings to the extracted physical places by classifying them into several general pre-defined types. Semantic place recognition based on classification has the following advantages: first, places can be categorized in terms of their functions, which are appropriate descriptions for people to interpret the personal meanings of their significant places [27]. Second, place classification can enable many location-aware applications to make adaptive decisions conveniently according to the current place type (e.g., automatically switching the mobile phone to a silent mode when the user enters his/her workplace, reminding the user to buy some items when he/she passes by a supermarket, etc.).

Given that the temporal visiting patterns are profiled according to the place type (e.g., the user tends to stay at home at night for a long period of time), the places can be categorized by looking at the temporal features in which they have been visited. The temporal features of a physical place are extracted as follows: first, we extract visit level features (i.e., the temporal features extracted from each visit point of the physical place). Second, we calculate place level features (i.e., the statistical values of the visit level features from all the visit points belonging to the physical place) to form the feature vector. Finally, we map the feature vectors to predefined place types based on a classifier. The feature extraction process is illustrated in Fig. 4, and we detail the involved features as follows:

Visit level features: these features are extracted from each visit point of a physical place, including *Day of Week*, *Time of Day*, *Duration* and *Response Rate* (as shown in Table 2).

Place level features: these features are calculated for a physical place based on the statistical values of its visit level features. The statistical methods include *Mean*, *Variance*, *Mode*, *Entropy*, *Min*, *Max* and *Frequency*. Mean, Variance, Min and Max are used for visit level features with numeric values (i.e., Duration and Response Rate). Mode and Entropy are used for visit level features with nominal values (i.e., Day of Week and Time of Day). Fig. 4 shows all the combinations of the statistical methods and visit level features. Thus, the final feature vector contains 13 statistical values (e.g.,

Table 2
The details of the visit level features.

Feature	Description	Calculation
Day of Week	Whether the visit takes place on weekend of weekday.	1 (for weekday) or 0 (for weekend)
Time of Day	Hour of the day (24 values).	The median of T_{in} and T_{out}
Duration	The time duration of the visit.	T_{out}-T_{in}
Response Rate	The percentage of time that GPS signal is available (used to distinguish between indoor and outdoor places).	$\frac{\text{The duration that GPS signal is available}}{T_{out}-T_{in}}$

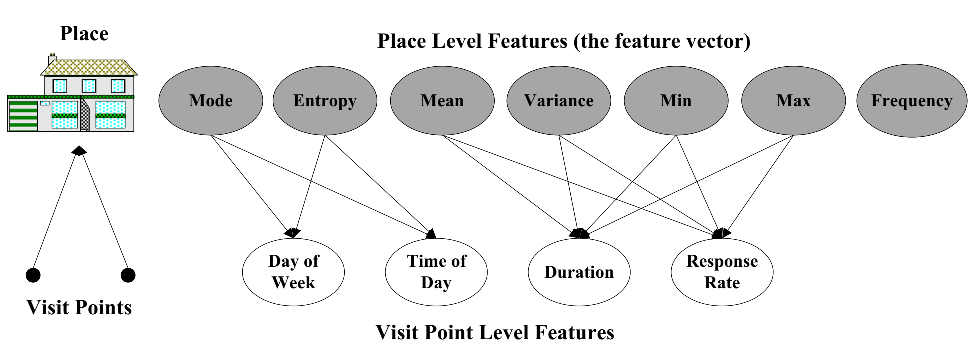


Fig. 4. The feature hierarchy for a place with multiple visit points.

Table 3
The details of the statistical methods of place level features.

Feature	Description	Calculation
Mode	The value that occurs most frequently.	For example, a place has been visited three times, and the "Day of Week" features of these visits are weekday, weekend and weekday respectively. Then, the "Mode" of "Day of Week" is weekday.
Entropy	It reflects the values' degree of uncertainty, where feature S has n values, and p_i is the probability of the cases when the value of S is $value_i$.	$\text{Entropy}(S) = -\sum_{i=1}^{n} P_i \log_2 P_i$
Frequency	It denotes the visit frequency.	$\frac{\text{The number of days when the place is visited}}{\text{The total number of days}}$

"Mode" of "Time of Day", "Mean" of "Duration", etc.). Some of these statistical methods are explained in Table 3 (Mean, Variance, Min, and Max are intuitive, so they are not elaborated).

Some of the place level features should be discretized before feeding into a classifier. For example, Mean, Max and Min of the Duration feature are discretized into long, medium and short. Mean, Max and Min of the Response Rate and the Frequency features are discretized into high, medium and low. After constructing the feature vectors for a number of different types of places which have been correctly labeled, a classifier (e.g., Decision Tree, BN, etc.) can be trained to encode the probability distribution associated to the fact that the user is in a certain type of place for a given temporal feature vector, and the final model can be used to estimate the type of a new place (we called it a *temporal place classifier*). The output of the temporal place classifier for a physical place x is a probability vector $\boldsymbol{PV_x}=\{p_1, p_2,\ldots,p_T\}$, where T is the number of place types for consideration, p_i $(i=1,\ldots,T)$ is the probability that x is classified as type i.

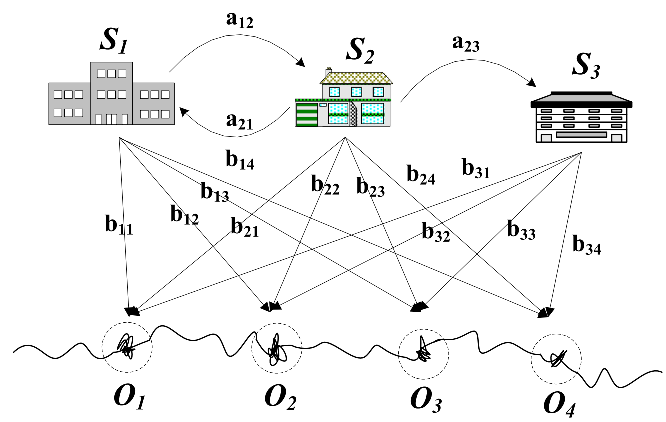


Fig. 5. The Hidden Markov Model for continuous place type prediction.

5.2. Integrating spatial features for place classification

The temporal place classifier categorizes the physical places by exploiting the temporal features in which they have been visited. However, there may be ambiguous place types which cannot be classified with high confidence based only on the temporal features. For example, users may visit a bar and a supermarket with similar temporal features (e.g., long duration in the evening). To alleviate this problem, a customized POI database which contains the spatial features of a corpus of places (including the properties of place location and place type) within the area of interest is used in combination with the temporal classifier to further improve classification accuracy. The customized POI database for the area of interest is constructed by using Google Places API. We create another feature vector for each extracted physical place according to the POIs which intersect with its scope as $\boldsymbol{FV_x}=\{f_1, f_2,\ldots,f_T\}$, where T is the number of place types for consideration, and f_i $(i=1,\ldots,T)$ is the weight of place type i intersects with physical place x, which is defined based on the idea of TF-IDF (i.e. term frequency-inverse document frequency) [28] as follow:

$$f_i = \frac{n_i^x}{n^x} \times \log \frac{N}{N_i} \tag{1}$$

where n_i^x is the number of POIs of place type i intersect with x, n^x is the total number of POIs intersect with x, N_i is the total number of POIs of place types i in the POI database, and N is the total number of POIs in the POI database. Taking both the temporal and spatial features into consideration, we calculate a confidence vector for physical place x $\boldsymbol{CV_x}=\{c_1, c_2,\ldots,c_T\}$, where T is the number of place types for consideration, c_i $(i=1,\ldots,T)$ is the confidence that place x is finally classified as type i.

$$c_i = \alpha \times p_i + (1-\alpha) \times nf_i \tag{2}$$

where p_i is the probability that physical place x is classified as type i based on the temporal classifier, nf_i is the normalized weight of place type i of x (the normalization is to ensure that the weight of every place type for a physical place is summed up to one), and α is the weight to adjust the importance of p_i and nf_i. The physical place is finally classified as the place type with the maximum confidence (we call it an *enhanced place classifier*).

5.3. Recognizing places by exploiting the sequential features

The enhanced place classifier is a static classifier which recognizes each place's type independently. However, people usually have certain sequential regularities to visit different types of places, e.g. people always do not go to another restaurant immediately after already visiting a restaurant. Thus, the types of the previous places can help with the current recognition. To implement this idea, we use the HMM to integrate the sequential dynamics of places been visited with the result of the static classifier. An HMM is a Markov process with a set of hidden states and observations. Each observation has a probability distribution (i.e., *emission probability*) over the possible hidden states, and the HMM also permits transitions among its hidden states which are governed by a set of probabilities (i.e., *transition probability*). As shown in Fig. 5, in our case, the HMM can be formally defined as $M=(S, V, \pi, A, B)$ which is adapted to the place type recognition task.

- S is the set of hidden states, which represent all the concerned place types.
- V is the set of observations. Instead of the raw features extracted from each place, we use the posterior probability vectors of the enhanced place classifier (i.e., the confidence vector mentioned in Section 5.2) as observations. The reason of using the posterior probability vectors is that we can take advantage of the results from the discriminative classifiers (i.e., the enhanced place classifier in this paper) which often outperform generative models (e.g., HMM) in classification task [29].

- π is the initial hidden state distribution. In our model, the initial hidden state distribution is set equal for all the concerned place types.
- $A=\{a_{ij}\}$ is the transition probability distribution, where a_{ij} represents the probability of visiting place type j after visiting place type i in a single step. The transition probability a_{ij} is set based on the ground-truth sequence of place types visited by the user (i.e., the ratio of the number of the cases that place type i transitions to place type j in a single step, to the number of the cases when place type i occurs in the training set).
- $B=\{b_{jk}\}$ is the emission probability distribution, where b_{jk} captures the likelihood of observing the posterior probability vector k produced by the enhanced place classifier conditioned on the current place type j. In our implementation, the emission probability b_{jk} is set as the confidence of type j in the posterior probability vector k. By such setting, we could integrate the result of the enhanced place classifier into the HMM.

Once a hidden state has been assigned to each observation in the training set, Baum–Welch algorithm [30] can be used to obtain an estimation of the parameters of the model. Once the HMM is trained, given a input sequence of observations $X=\{x_1, x_2,\ldots,x_T\}$ (x_t is the posterior probability vector at time instant t), it can recognize the hidden state at time instant T (i.e., the place type for the posterior probability vector at time T) to maximize the function $P(q_T=q_i \mid X, M)$. Note that each element x_t in X is corresponding to a visit point, and X is corresponding to a sequence of visits to different places during a specific time interval (e.g., a day), so the HMM estimates place types on visit level. On the other hand, the static classification is conducted on place level, so the visit points to the same place have identical posterior probability vector, but may be recognized as different place types. We call it a *continuous place recognizer*. By incorporating the results of the enhanced place classifier with the HMM model, the continuous place recognizer can smooth the place recognition process, and improve the place recognition accuracy.

6. Experiment

In order to evaluate the performance of our approach, we conducted a number of experiments based on real GPS trajectories collected from 10 participants for nearly four months. All participants involved in the experiment are students and faculties of our lab, and their family members. They live in different areas of Hangzhou city, China. During the data collection phase, each participant carried a GPS enabled mobile phone. A logging program running on the mobile phone recorded the GPS readings at 1 Hz.

All participants were instructed to carry out the experiment in an open-ended way to make the recorded GPS trajectories reflect their daily lives as truly as possible, i.e., we did not set up any constraints on the data recording environment, and the participants could take the recording devices as they went about their normal lives (e.g., going to work, going for shopping, going for dinner, etc.). The participants were asked to log the place type by choosing from a pre-defined list in the logging program and write a textual description about the current place whenever they entered it (note that all the logged places are building-level places). They were also asked to provide the exact location of every logged place by clicking on a digital map. The place logs provided by the participants can be used as the ground-truth information about the places they have visited.

Before applying the physical place extraction algorithm, the GPS trajectories need to be cleaned due to the uncertainty of GPS devices which may produce outliers. The GPS points which are far away from the reasonable sequence of temporally consecutive GPS points and result in unreasonable speed are removed [8]. Before presenting the details, we give an overview about all the experiments in Table 4.

6.1. Physical place extraction experiment

In this section, we tried to evaluate the performance of the hierarchical clustering algorithm for physical place extraction. The algorithm has totally found 3473 visit points and 489 places from the GPS trajectories of the 10 participants. In the first experiment, we tried to evaluate the effectiveness of our time-based clustering algorithm for extracting visit points from GPS trajectories. A visit point *VP* is correctly extracted if there is a logging timestamp t_{log} provided by the participant, satisfying $VP.t_{in} \le t_{log} \le VP.t_{out}$. For the incorrect results made by the algorithm, we distinguish them as either *false negative* (i.e., the algorithm reports the user is moving while he/she is actually visiting a place) or *false positive* (i.e., the algorithm reports the user is visiting a place while he/she is actually moving). We compared our work with the existing time-based clustering algorithm [11] (we refer it as *ETC*) and speed-based clustering algorithm [13] (we refer it as *CB-SMOT*), and tuned the parameters in order to reduce both false 调整参数
negative and false positive. The ETC algorithm does not have the parameter $\delta_{tolerated\_distance}$, and simply ignores the cluster whose time duration is not long enough. The CB-SMOT algorithm extends the DBSCAN algorithm to adapt to trajectory data. It expands a core point (i.e., the trajectory has a low speed within the neighborhood of it) to form visit point. Both ETC and CB-SMOT have a clustering distance parameter (i.e., $\delta_{cluster\_distance}$ for ETC and *Eps* for CB-SMOT) and a time duration parameter δ_{time}. The parameter $\delta_{cluster\_distance}$ and *Eps* are estimated by using a probability based parameter $p_{cluster\_distance}$ by using the approach mentioned in Section 4.1. We fixed $\delta_{time}=300$ s based on the experiences gained in [11], and monitored the performance of the two algorithms by adjusting $p_{cluster\_distance}$.

Fig. 6(a) shows the influence of $p_{cluster\_distance}$ on the *false positive ratio* (FPR, i.e., the ratio of the number of false positive to the number of extracted visit points) and the *false negative ratio* (FNR, i.e., the ratio of the number of false negative to the number of logged visit points) of the two algorithms. Note that we use extremely high values for $p_{cluster\_distance}$. This is because that the GPS points were sampled at a high rate in the experiment, and thus the distance between most consecutive GPS points is too short to be used for clustering. It can be found from the figure that setting $p_{cluster\_distance}$ too low will result in high FNR, and setting $p_{cluster\_distance}$ too high will result in high FPR for the two algorithms. The reason is that setting $p_{cluster\_distance}$ too low will cause

Table 4
An overview of all the experiments.

Evaluated method	Experiment purpose
Physical place extraction	Evaluating the time-based clustering algorithm for extracting visit points from GPS trajectories (the first experiment in Section 6.1)
	Evaluating the distance-based clustering algorithm for extracting physical places from visit points (the second experiment in Section 6.1)
Semantic place recognition	Evaluating the place classification method based only on temporal features (the first experiment in Section 6.2)
	Evaluating the place classification method based on temporal and spatial features (the second experiment in Section 6.2)
	Evaluating the place classification method based on temporal, spatial and sequential features (the third experiment in Section 6.2)

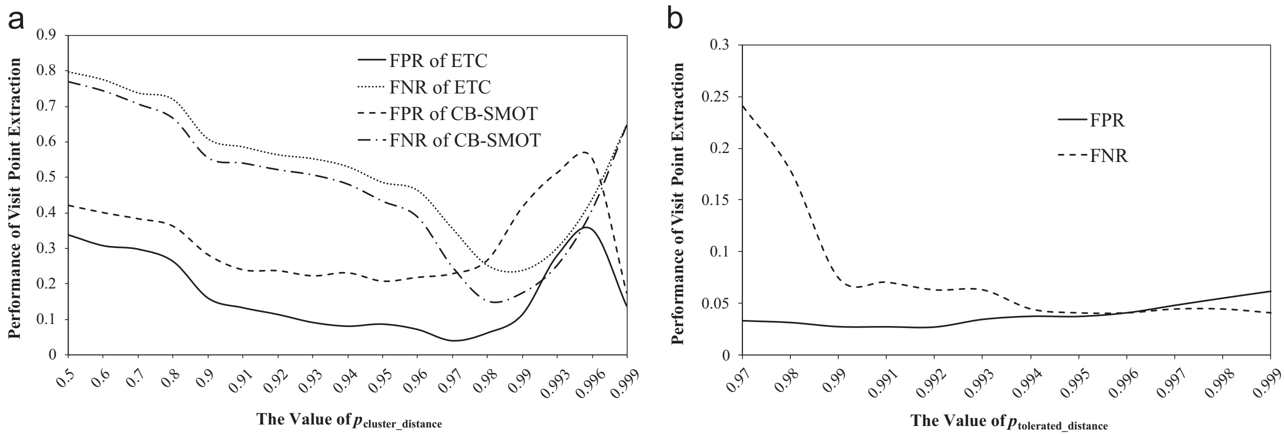


Fig. 6. The performance of extracting visit points from GPS trajectories: (a) the performance of the existing clustering algorithms by adjusting $p_{\text{cluster\_distance}}$; (b) the performance of our time-based clustering algorithm by adjusting $p_{\text{tolerated\_distance}}$.

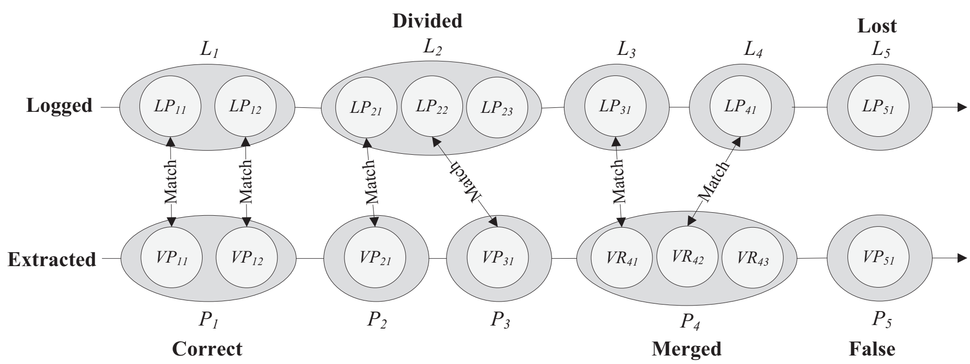


Fig. 7. An example of evaluation metrics for physical place extraction.

the extracted clusters to be too small, and thus a visit point may be ignored due to the short time duration of the small cluster. Setting $p_{\text{cluster\_distance}}$ too high will make the extracted clusters grow too large, and thus the time duration of a cluster extracted when the user is moving may even be long enough to be considered as a visit point. Besides, ETC has better performance on FPR, and CB-SMOT has better performance on FNR. This might be because that CB-SMOT tends to expand the clusters to a great extent. Therefore, it has less chance to miss a visit points, and may be also more likely to find incorrect ones. Apparently, we cannot simultaneously minimize FPR and FNR for the two algorithms, so we configure $p_{\text{cluster\_distance}}$ to minimize Eq. (3), which considers both FPR and FNR using F-measure, i.e., $p_{\text{cluster\_distance}}=0.97$ (corresponding to 45 m on average), $F=0.097$ for ETC, and $p_{\text{cluster\_distance}}=0.98$ (corresponding to 70 m on average), $F=0.138$ for CB-SMOT.

$$F = \frac{2 \times \text{FPR} \times \text{FNR}}{\text{FPR} + \text{FNR}} \tag{3}$$

On the other hand, our time-based clustering algorithm adopts a tolerated distance parameter to alleviate the entrance and exit deviation problems, and this means that our algorithm tends not to miss visit points even when $\delta_{\text{cluster\_distance}}$ is small. We use two probability parameters $p_{\text{cluster\_distance}}$ and $p_{\text{tolerated\_distance}}$ to estimate $\delta_{\text{cluster\_distance}}$ and $\delta_{\text{tolerated\_distance}}$. We fix $p_{\text{cluster\_distance}}$ to minimize FPR and tune $p_{\text{tolerated\_distance}}$ to reduce FNR. As shown in Fig. 6(b), we fix $p_{\text{cluster\_distance}}=0.97$ to minimize FPR (FPR=6.05%), and FNR is significantly decreased as increasing $p_{\text{tolerated\_distance}}$. However, setting $\delta_{\text{tolerated\_distance}}$ too high will make the algorithm suffer from noises. For example, the algorithm may mistakenly find a visit point when the user turns on the mobile phone far away from the location where he/she turns it off after a long period of time. Thus, we set a reasonable value for $p_{\text{tolerated\_distance}}$ to minimize Eq. (3) ($p_{\text{tolerated\_distance}}=0.994$ (corresponding to 350 m on average), $F=0.041$). In conclusion, the best visit point extraction performances of our time-based clustering algorithm, ETC, and CB-SMOT are $F=0.041$, $F=0.097$, and $F=0.138$, respectively.

In the second experiment, we evaluated the effectiveness of the physical place extraction algorithm for extracting physical places from visit points. Places extracted by the algorithm are called *Extracted*, and extracted place x is represented by a collection of visit points $P_x=\{VP_{x1},\ldots,VP_{xn}\}$ (see Definition 3). Places recorded in the place logs are called *Logged*. Since a place y may be recorded several times due to multiple visits, it can be represented by a collection of log points $L_y=\{LP_{y1},\ldots,LP_{ym}\}$. LP is a couple $LP=(l, t)$, where l is the location of the logged place, and t is the timestamp of the log. Before giving the evaluation metrics, we first define the "*Match*" relationship between visit points and log points.

(Definition 6.) Match: a visit point VP matches a log point LP if the distance between VP and the LP is smaller than a threshold (i.e., $\textbf{distance}(VP.p, LP.l) < \delta_{\text{match\_distance}}$) and LP is recorded during VP (i.e., $VP.t_{\text{in}} \le LP.t \le VP.t_{\text{out}}$).

As illustrated in Fig. 7, an *Extracted* place x is *Correct* if each visit point of x matches one of the log points of *Logged* place y, and vise versa (i.e., each log point of y matches one of the visit points of x). A *Logged* place y is *Divided* if the log points of y match the visit points of different *Extracted* places (or there remain y's log points which cannot find matched visit points). An *Extracted* place x is *Merged* if the visit points of x match the log points of different *Logged* places (or there remain x's visit points which cannot find matched log points). An *Extracted* place x is *False* if none of the visit points of x can find a matched log point among all the *Logged* places. A *Logged* place y is *Lost* if none of the log points of y can find a matched visit point among all the *Extracted* places. To evaluate the physical place extraction performance, we define the following metrics (# stands for "number of"), where *FR*, *LR*, *MR*, *DR*, *P*, *R* and *F* denote False Rate, Lost Rate, Merged Rate, Divided Rate, Precision, Recall and F-measure, respectively.

$$FR = \frac{\#\,False}{\#\,Extracted},\ LR = \frac{\#\,Lost}{\#\,Logged} \tag{4}$$

$$MR = \frac{\#\,Merged}{\#\,Extracted},\ DR = \frac{\#\,Divided}{\#\,Logged} \tag{5}$$

$$P = \frac{\#\,Correct}{\#\,Extracted},\ R = \frac{\#\,Correct}{\#\,Logged},\ F = \frac{2 \times P \times R}{P + R} \tag{6}$$

Since several kinds of clustering algorithms can be employed, we compared the performance of three algorithms for extracting physical places from visit points, i.e., *Merging* [11] (it checks the distance between the newly discovered visit point and each of the existing visit point sets, merges the visit point to one set if the distance between them is less than a threshold $\delta_{\text{merge\_distance}}$, and the final visit point sets are judged as physical places), *Partitioning* [10] (a variant of K-Means clustering algorithm to group visit points into physical places) and *Distance-based* (algorithm used in this paper). The parameters of the three algorithms were tuned to output the best performance ($\delta_{\text{merge\_distance}}$ of Merging is set to 100 m, *radius* of K-Means is set to 100 m, *Eps* of our algorithm is set to 50 m, $\delta_{\text{match\_distance}} = 50$ m). Table 5 shows the results obtained from the three algorithms. From the table, it can be found that Merging algorithm has the lowest performance because it simply merges the newly discovered visit point to an existing visit point set to create physical place without a clustering procedure. As compared to Partitioning, Distance-based algorithm has a higher MR value because it merges the overlapping visit points together to a great extent. However, Distance-based algorithm has the lowest DR value because it can detect physical place with arbitrary shape which can accommodate all its visit points, and thus achieves the best overall performance.

6.2. Semantic place recognition experiment

This experiment tried to verify the results of the place recognition algorithm. In our experiment, we have asked the participants to log among eight types of places, i.e., home, work, restaurant (to indicate any kind of dining place), supermarket, shop (to indicate any kind of small-size store, e.g., grocery, clothes store, etc.), recreation (to indicate any kind of place for recreation, e.g., cinema, club, etc.), business (to indicate any kind of place for public service, e.g., administration, hospital, etc.) and tour (to indicate any kind of outdoor touring place, e.g., beach, park, etc.). This classification is rather arbitrary, and applications of different domain may have different criteria of place classification.

In the first experiment, we tried to evaluate the performance of place classification based only on temporal features. We used three state-of-the-art classifiers, i.e., BN (Bayesian Network), LR (Logistic Regression) and RF (Random Forest), as our temporal place classifier. For evaluation, we adopted both user-specific validation (i.e., classifier was trained based on each participant's physical places and tested on the same participant based on 5-fold cross validation) and leave-user-out validation (i.e., classifier was trained based on the physical places of eight of the participants and tested on the other two participants left out). We also rearranged the test set to ensure that both the training and testing set contain all the place types. All the validation processes were repeated for the 10 participants, and the summary results are presented in Fig. 8.

Table 5
Physical place extraction performance of different clustering algorithms.

	Merging	Partitioning	Distance-based
FR	0.105	0.094	**0.085**
LR	0.054	0.054	0.054
MR	0.118	**0.092**	0.165
DR	0.214	0.181	**0.053**
P	0.513	**0.747**	0.742
R	0.656	0.737	**0.783**
F	0.576	0.742	**0.762**

The RF classifier achieves the best overall performance for both user-specific validation (94.81%) and leave-user-out validation (76.62%). The high accuracy of user-specific validation demonstrates that it is feasible to classify a user's places based on the temporal features. However, the relatively low accuracy of leave-user-out validation indicates that different users may have fairly different temporal features even they visit the same type of places. Fig. 8 also shows the accuracy of classifying different types of places. It can be found from the results that the temporal place classifier achieves considerably high performance for places that have strong personal meaning (e.g., home, work, etc.), and has unsatisfactory performance for public places (e.g., business, supermarket, etc.) based on leave-user-out validation. This means that these participants share common temporal features to visit personal places, and have flexible temporal features to visit public places.

In the second experiment, we verified the enhanced place classifier by taking advantage of both the temporal place classifier and a customized POI database. We used Google Places API to collect public places for the area where the experiment was carried out and stored the results (including each place's name, location, type and address) to the customized POI database. We adopted the leave-user-out validation to evaluate the enhanced place classifier, and monitored the performance by adjusting the value of α. The enhanced place classifier gained the best average performance (83.12%) when setting $\alpha = 0.8$ (i.e., 80% influence from the temporal place classifier and 20% influence from the POI database). However, the variation rules of classification performance of different place types with respect to α are greatly different from each other. As shown in Fig. 9, the classification performance of places "home", "work" and "restaurant" is monotone increasing with respect to α, while the classification performance of places "business" is monotone decreasing with respect to α. This means that the participants tended to have strong temporal regularity to visit places "home", "work" and "restaurant", and have very weak temporal regularity to visit places "business". Besides, places "recreation", "supermarket", "shop" and "tour" have sophisticated variation rules of classification performance with respect to α. This is because that people usually have much weaker regularity of visiting public places (e.g., shop, supermarket, etc.) than personal places (i.e., home and work). On the other hand, some public place types in this experiment contain multiple fine-grained place types (e.g., "recreation" contains "cinema", "club", etc.), and this fact further complicates the recognition of these places. These experimental results can give heuristics to help location-aware applications that consider different types of places to decide how to set α. For example, the application can set $\alpha = 1.0$ to find places "home" and "work" based only on the temporal classifier, and set $\alpha = 0.0$ to find places "business" based only on the POI database.

In the third experiment, we compared the performance of three place recognition approaches, i.e., the enhanced place classifier (abbreviated as EPC), the continuous place recognizer (abbreviated as CPR), and the existing technique proposed in [26] (abbreviated as LAR). The parameter α is set to 0.8 for EPC and CPR. LAR also exploits the temporal, spatial and sequential features to recognize activity performed at certain locations. However, LAR trains the recognizer based on features from each individual visit point without place extraction procedure, while CPR computes statistical features for each

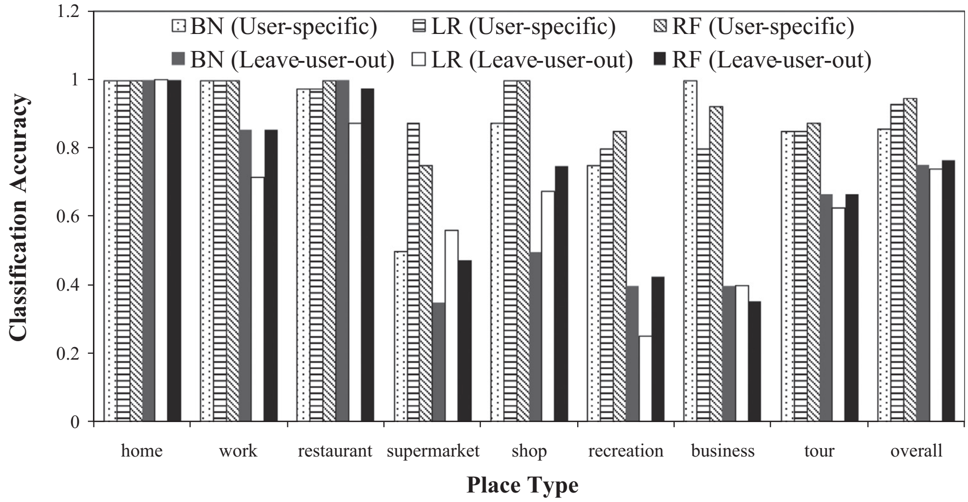


Fig. 8. The classification accuracy of different place types and different classifiers.

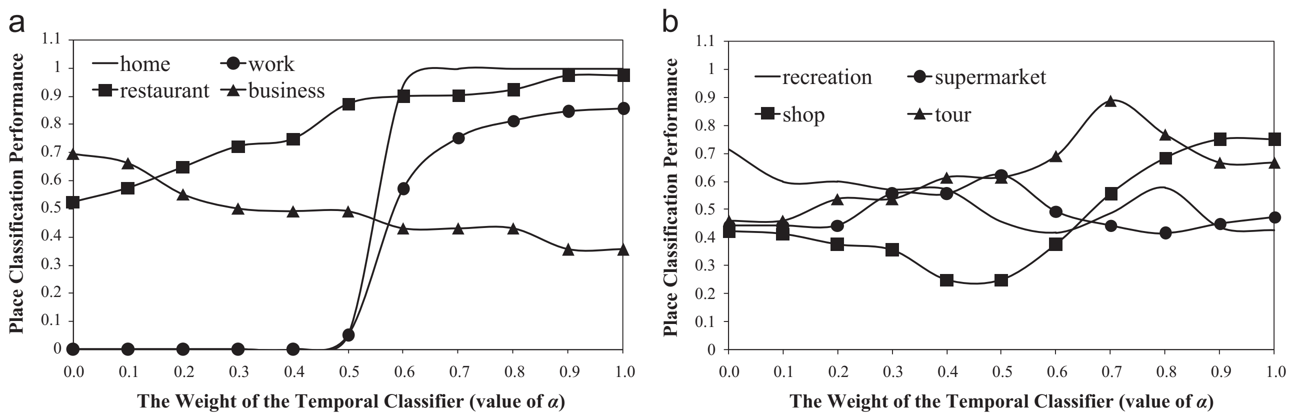


Fig. 9. The performance of the enhanced place classifier: (a) effect of parameter α on the place classification performance (with monotone variation rules); (b) the effect of parameter α on the place classification performance (with sophisticated variation rules).

place which is comprised of multiple visit points and takes advantage of the posterior probabilities produced by the discriminative model. We use temporal features (i.e., Day of Week, Time of Day and Duration), geographic features (i.e., whether a type of POI is within a certain range from the visit point), and sequential features extracted from each visit point to implement LAR.

Note that both CPR and LAR estimate the place types on visit level. The visit points which belong to the same place have identical posterior probability vector for CPR, but have different feature vectors for LAR. Both CPR and LAR may recognize visit points belonging to the same place as different place types. On the contrary, EPC is performed on place level, so all the visit points belonging to the same place are recognized as the same place type. We used leave-user-out validation to evaluate these approaches. The test cases for CPR and LAR are sequences of visit points for specific days in the testing set. Fig. 10 compares the accuracy of place recognition by using the three approaches. It can be found from the figure that CPR achieves the best overall performance. CPR outperforms EPC means that the participants do have certain sequential regularities to visit these types of places. LAR performs comparably to CPR for place types with consistent visiting patterns (e.g., work, restaurant), and CPR has better performance for place types with inconsistent visiting patterns, which are difficult to be captured from a single visit (e.g., supermarket, business). Therefore, the advantage of CPR trained based on statistical features from multiple visit points is that it could better recognize place types with inconsistent visiting patterns. This fact can also be demonstrated by the phenomenon that EPC achieves better classification performance for visit points than that for places (this indicates that places which have more visit points are always better classified than those have less visit points).

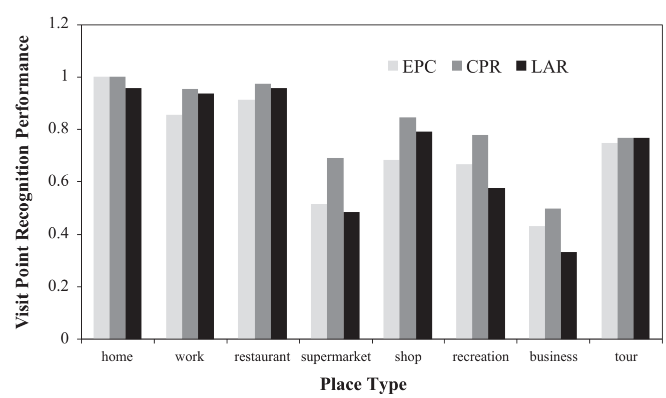


Fig. 10. The accuracy of visit point level place classification based on different approaches.

7. Conclusions and future work

In this paper, we propose a framework to discover users' personally semantic places from their GPS trajectories. The framework includes obtaining the physical locations of the places he/she has visited and estimating the pre-defined types of these places. Extensive experiments have been conducted to investigate various aspects of the system performance and the characteristics of people visiting their personal places in real-world environment. The experiment results have demonstrated that our approach outperforms existing approaches.

In the future, this work could be extended from several aspects. First, the proposed approach is designed based on GPS trajectories. We could extend it to incorporate trajectories obtained based on other kinds of positioning infrastructures (e.g., cellular network, wireless network, etc.). Second, the proposed approach can extract building-level places. We could extend it to extract room-level places by leveraging various kinds of mobile phone sensing data (e.g., WiFi signal, Bluetooth signal, accelerations, etc.). Third, we could incorporate various kinds of contextual information (e.g., the user's occupation, gender, etc.) to further improve the semantic place recognition accuracy.

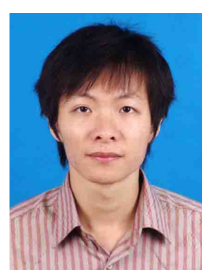
Besides, it would be an interesting problem to compare the semantic place recognition performance between our trajectory mining based method and the traditional computer vision based methods. In our study, we do not have data to conduct the comparison, because it is impractical to ask the participants to frequently turn on the camera to capture images. However, in other settings (e.g., using vehicle mounted devices), this is a promising research problem.

Acknowledgment

This work was supported by National Natural Science Foundation of China (Nos. 61202282, 61332017, 61502421), the Zhejiang Provincial Natural Science Foundation of China (Nos. LY15F020025, LY15F020026), China Knowledge Centre for Engineering Sciences and Technology (No. CKCEST-2014-1-5), and the Science and Technology Department of Zhejiang Province (Nos. 2011C13042 and 2015C33002).

References

[1] S. Harrison, P. Dourish. Re-place-ing space: the role of place and space in collaborative systems, in: Proceedings of CSCW, 1996, pp. 67–76.
[2] I. Arminen, Social functions of location in mobile telephony, Pers. Ubiquitous Comput. 10 (5) (2006) 319–323.
[3] J. Zhang, M. Zhu, D. Papadias, Y. Tao, D.L. Lee. Location-based spatial queries, in: Proceedings of SIGMOD, 2003, pp. 443–454.
[4] Y. Tu, L. Chen, M. Lv, Y. Ye, W. Huang, G. Chen, iReminder: an intuitive location-based reminder that knows where you are going, Int. J. Human-Comput. Interact. 29 (12) (2013) 838–850.
[5] K. Jiang, H. Yin, P. Wang, N. Yu, Learning from contextual information of geo-tagged web photos to rank personalized tourism attractions, Neurocomputing 119 (7) (2013) 17–25.
[6] Z. Fang, Q. Li, X. Zhang, S. Shaw, A GIS data model for landmark-based pedestrian navigation, Int. J. Geogr. Inf. Sci. 26 (5) (2012) 817–838.
[7] Y. Jung, P. Persson, J. Blom. Dede: design and evaluation of a context-enhanced mobile messaging system, in: Proceedings of CHI, 2005, pp. 351–360.
[8] L. Chen, M. Lv, G. Chen., A system for destination and future route prediction based on trajectory mining, Pervasive Mobile Comput. 6 (6) (2010) 657–676.
[9] L. Lentini, F. Decortis, Space and places: where interacting with and in physical space becomes a meaningful experience, Pers. Ubiquitous Comput. 14 (5) (2010) 407–415.
[10] D. Ashbrook, T. Starner, Using GPS to learn significant locations and predict movement across multiple users, Pers. Ubiquitous Comput. 7 (5) (2003) 275–286.
[11] J.H. Kang, W. Welbourne, B. Stewart, G. Borriello. Extracting places from traces of locations, in: Proceedings of WMASH, 2004, pp. 110–118.
[12] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, L. Terveen, Discovering personally meaningful places: an interactive clustering approach, ACM Trans. Inf. Syst. 25 (3) (2007).
[13] A.T. Palma, V. Bogorny, B. Kuijpers, L.O. Alvares. A clustering-based approach for discovering interesting places in trajectories, in: Proceedings of SAC, 2008, pp. 863–868.
[14] Y. Ye, Y. Zheng, Y. Chen, J. Feng, X. Xie. Mining individual life pattern based on location history, in: Proceedings of MDM, 2009, pp. 1–10.
[15] G. Iachello, I. Smith, S. Consolvo, M. Chen, G.D. Abowd. Developing privacy guidelines for social location disclosure applications and services, in: Proceedings of SOUP, 2005, pp. 65–76.
[16] L. Barkhuus, B. Brown, M. Bell, S. Sherwood, M. Hall, M. Chalmers. From awareness to repartee: sharing location within social groups, in: Proceedings of CHI, 2008, pp. 497–506.
[17] J. Liu, Q. Wolfson, H. Yin. Extracting semantic location from outdoor positioning systems, in: Proceedings of MDM, 2006.
[18] X. Xiao, Q. Luo, X. Xie. Finding similar users using category-based location history, in: Proceedings of GIS, 2010, pp. 442–445.
[19] V. Bogorny, B. Kuijpers, L.O. Alvares, ST-DMQL: a semantic trajectory data mining query language, Int. J. Geogr. Inf. Sci. 23 (10) (2009) 1245–1276.
[20] J. Ying, E. Lu, W.C. Lee, V.S. Tseng. Mining user similarity from semantic trajectories, in: Proceedings of LBSN, 2010, pp. 19–26.
[21] P.J. Ludford, R. Priedhorsky, K. Reily, L. Terveen. Capturing, sharing, and using local place information, in: Proceedings of CHI, 2007, pp. 1235–1244.
[22] J. Paek, J. Kim, R. Govindan. Energy-efficient rate-adaptive GPS-based positioning for smartphones, in: Proceedings of MOBISYS, 2010, pp. 299–314.
[23] J. Hightower, S. Consolvo, A. Lamarca, I. Smith, J. Hughes. Learning and recognizing the places we go, in: Proceedings of UBICOMP, 2005, pp. 159–176.
[24] D.H. Kim, J. Hightower, R. Govindan, D. Estrin. Discovering semantically meaningful places from pervasive RF-beacons, in: Proceedings of UBICOMP, 2009, pp. 21–30.
[25] A. LaMarca, Y. Chawathe, S. Consolvo et al. Place lab: device positioning using radio beacons in the wild, in: Proceedings of PERVASIVE, 2005, pp. 116–133.
[26] L. Liao, D. Fox, H. Kautz, Extracting places and activities from GPS traces using hierarchical conditional random fields, Int. J. Robot. Res. 26 (1) (2007) 119–134.
[27] J. Lin, G. Xiang, J.I. Hong, N. Sadeh. Modeling people's place naming preferences in location sharing, in: Proceedings of UBICOMP, 2010, pp. 75–84.
[28] G. Salton, C. Buckley, Term-weighting approaches sin automatic text retrieval, Inf. Process. Manag. 24 (5) (1988) 513–523.
[29] A. Ng, M. Jordan., On discriminative vs. generative classifier: a comparison of logistic regression and naïve bayes, Adv. Neural Inf. Proc. Syst. 16 (2002) 841–848.
[30] E. Trentin, M. Gori., A survey of hybrid ANN/HMM models for automatic speech recognition, Neurocomputing 37 (1-4) (2001) 91–126.
[31] M. Lv, L. Chen, G. Chen. Discovering personally semantic places from GPS trajectories, in: Proceedings of CIKM, 2012, pp. 1152–1156.



Mingqi Lv received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2012. He is currently a lecturer with the College of Computer Science and Technology, Zhejiang University of Technology. His research interests include ubiquitous computing, data mining and human computer interaction.

Ling Chen received the B.S. and Ph.D. degrees in computer science from Zhejiang University in 1999 and 2004, respectively. He is currently an associate professor with the College of Computer Science and Technology, Zhejiang University. His research interests include ubiquitous computing, human computer interaction and pattern recognition.

Zhenxing Xu is currently a Ph.D. candidate in the college of computer science and technology at Zhejiang University, Hangzhou, China. His research interests include location-based services and social media data mining.

Yinglong Li received the Ph. D. degree in computer science from Renmin University of China, Beijing, China in 2014. He is currently a lecturer in Zhejiang University of Technology, Hangzhou. His research interests include routing and data processing in wireless sensor networks.

Gencai Chen graduated from Physics Department, Hangzhou University, Hangzhou, China, in 1973. He is currently a professor with the College of Computer Science and Technology, Zhejiang University. His research interests include database, data mining and computer supported cooperative work.