Where you go and how you go: Mining to predict destination and future route

Mingqi Lv, Ling Chen, and Gencai Chen
College of Computer Science, Zhejiang University
Hangzhou 310027, P.R. China
lingchen@cs.zju.edu.cn

***Abstract*—The ability to predict the future movement behavior of a person can help to improve many pervasive applications that exploit the locations of their users. In this paper, we propose an approach to predict both the intended destination and the future route of a person. Rather than predicting the destination and future route separately, we have focused on making prediction in an integrated way by exploiting personal movement data collected by GPS. Since personal movement data is more informative than the movement data of vehicles, the proposed approach first detects the significant places of a person using a clustering-based algorithm called FBM, and then extracts movement patterns using an extended CRPM algorithm. Extracted movement patterns are organized as couples of places where a person departs from and goes to. The prediction is made based on a pattern tree built from these movement patterns. Finally, with the real personal movement data of 14 participants, we conducted a number of experiments to evaluate the performance of our system. The results show that our approach can achieve approximately 80% and 60% accuracy in destination prediction and 1-step prediction, respectively.**

Keywords-context-aware computing; location; data mining; route pattern; route prediction

I. INTRODUCTION

People's daily routes show a high degree of temporal and spatial regularity, each individual is characterized by a time-independent travel distance and a significant probability to return to a few highly frequented locations [1], i.e. people always follow more or less the same routes to visit some frequented places. With the popularity of the pervasive mobile devices (e.g. mobile phones, GPS, etc.), it is easy to record a person's movement as a sequence of time-stamped locations, which can be analyzed and used to predict the person's future movement, i.e. destination and future route which are the key context information of a person in the mobile environment. The ability to predict destination and future route can help to improve many pervasive applications. For example, the predicted destination can be used by navigation systems to provide route guidance more intelligently without requiring manual input from the driver, the predicted future route can be used by electronic commerce systems to deliver commercial messages specifically to the persons who will probably pass by the shops, etc.

The potential merits have motivated many research efforts. However, there are two problems with existing movement prediction approaches. First, most existing works use vehicle movement data for analysis [2][3][4][5]. The high degree of regularity of vehicle movement data greatly facilitates the data processing, but it contains less contextual information than personal movement data. For example, the real frequented places of a person can never be found from vehicle movement data (obviously, the places a vehicle frequents are always parking lots or bus stations). Second, most existing works predict the intended destination and future route of a moving object in a separate way which may cause the problem that the predicted future route is irrelevant to the intended destination of a person, and result in predicting a short term future route which may always be not long enough to reach the intended destination [2][6][7].

The problem that we have focused on in this paper is first to analyze the personal movement data collected by GPS to discover the movement patterns which denote the places where the person usually departs from (we call it *origin*) and goes to (we call it *destination*), and the routes which are frequently taken to travel from an origin to a destination. Then we try to predict both the person's intended destination and future route to get there by observing his/her origin and route taken so far based on the movement patterns. We use personal movement data for investigation instead of vehicle movement data. However, personal movement data is more noisy and incomplete than vehicle movement data for various reasons, e.g. a person may take different transportation manners which result in high instability of speed, a person may spend a long time in buildings where GPS loses its signal, a person may not record his/her movement data continuously, etc. Thus, particular data processing approaches are required to counter these problems. On the other hand, the existing movement patterns are mainly repeated routes extracted individually from the movement data. Based on these patterns, we can just predict a short term future route of a moving object. In order to predict destination and future route in an integrated way, the regularities of both the frequented places and the relevant routes should be fused as an integrated pattern representation which is able to capture the implied relationship between them and make the prediction more efficiently.

The remainder of this paper is organized as follows. Section 2 gives a survey of the related work. Section 3 describes the personal movement data preprocessing including filtering the rough GPS data and segmenting the movement data into individual trajectories. Section 4 presents a clustering-based algorithm for extracting candidate origins and destinations. Section 5 details the route pattern mining approach including our space-partitioning technique and the mining algorithm. Section 6 demonstrates

978-1-4244-5328-3/09/$25.00 ©2009 IEEE

how the route patterns are used for predicting destination and future route. The evaluation of the system and the experimental results are reported in Section 7. Finally, we conclude our work and give some future work in Section 8.

II. RELATED WORK

Our work is most related to future movement behavior prediction based on movement patterns. The movement patterns of a person in this paper contain the knowledge about his/her frequented places and repeated routes as well as their implied relationships.

Most previous works applied clustering approaches on a user's location data to discover frequented places. These clustering approaches can be generally divided into three types, i.e. partitioning clustering, density-based clustering and time-based clustering. All partitioning clustering approaches for discovering frequented places are based on the well-known K-Means clustering algorithm [8]. Density-based clustering algorithms which form clusters based on the density of neighborhoods of the points [9] are widely used in places extraction [10][11]. Time-based clustering algorithm works in an incremental manner. It clusters the locations along the time axis, and a new place is found when the new locations which are close to each other but far from the previous place span a significant of time [12]. However, it is impractical to apply existing clustering algorithms directly to personal movement data due to its diverse, noisy and incomplete characteristics.

Previous works on discovering repeated routes mostly applied mining technology to extract patterns from movement data. Mamoulis et al. [13] addressed the problem of discovering periodic patterns in spatio-temporal data. A periodic pattern is a sequence of spatial regions which frequently reappears every regular time periods. Cao et al. [14] adopted lines simplification method to convert the location series into trajectory segments, and used a substring tree to mine for longer patterns. Tao et al. [15] used association rules mining approach to find movement patterns from spatio-temporal data. Giannotti et al. [16] provided a way of building region of interest (ROI) to simplify real trips and designed a mining algorithm to extract trajectory patterns in terms of both space and time. However, these works have not taken origin and destination into consideration, so the extracted patterns can not catch the relationship of each other, and represent only a short term movement regularity of the moving object. Another problem we try to address is the answer loss problem [17] caused by space-partitioning approaches used in many spatio-temporal patterns mining works. Jeung et al. [18] used hidden Markov model (HMM) to explain the relationships between regions and partitioned cells to counter the answer loss problem, but the DBSCAN algorithm used to detect regions is too computationally expensive to run on mobile devices.

Most existing works on future movement prediction tried to predict destination and future route of a moving object separately. Ashbrook and Starner [8] presented a system that automatically detected the significant locations from GPS data, and then incorporated these locations into a Markov model to predict the person's next significant location. Krumm and Horvitz [4] proposed a method called Predestination that leveraged both a driver's travel history and an open-world modeling methodology to predict where the driver was going as a trip progressed. However, they only predicted the next significant location (i.e. destination) of a moving object. Froehlich and Krumm [2] calculated the similarity between trips based on a version of the Hausdorff distance algorithm, and merged similar trips to get routes, then made route prediction by looking for the similar route to the current trip. Karimi and Liu [6] used a map-based approach to abstract real trips, and predicted the subsequent movement of a user based on probabilities assigned to the road intersections. Ye et al. [7] predicted a person's future route through the use of a probabilistic tree model built from his/her route patterns which were extracted from historical movement data using a mining algorithm called CRPM. These works were all focus on predicting future route of a user without the ability to predict his/her intended destination as well. The work most similar to ours was presented by Simmons et al. [3] who built a HMM to predict the destination and future route of a driver simultaneously. However, it used vehicle movement data for investigation, so the real origin and destination can not be extracted directly from the movement data, but from the infrastructure of the road network.

To summarize, the major difference between previous works and ours consists in two aspects. First, we predict the intended destination of a person and his/her future route to get there by observing his/her origin and route taken so far. Second, we use a number of mining algorithms designed specifically for processing personal movement data to extract movement patterns which contain the knowledge about both the frequented places and the repeated routes in an integrated way.

III. FROM GPS DATA TO TRAJECTORIES

This section explains the preprocessing of the raw movement data recorded by GPS devices. First, we divide the raw GPS data into discrete trips. Second, we clean the trips by removing the noisy data. The output of this step is a sequence of trajectories.

A. Trip segmentation

The basic objects for our mining algorithms are discrete trajectories with definite origin and destination, but the personal movement data recorded by GPS gives no explicit indication of the beginning and ending of a trip. So the first step of our work is trip segmentation.

The basic criterion for segmenting a trip is based on the interval between two temporally adjacent data points. The segmentation algorithm sorts the raw GPS data incrementally and looks for time gaps between two consecutive recorded data points. If the time gap is greater than the interval threshold, the raw GPS data is split into two trips, the former point becomes the end of the first trip and the latter becomes the beginning of the second trip.

B. Data filtering

Once the GPS data is segmented, the data points within these trips need to be cleaned before further processing due to the uncertainty of GPS which may produce outliers. In order to reserve the diversity of personal movement data, data points scattered in disorder within a small space are not treated as outliers, because these data points may be recorded when the person sticks around a place which may be a candidate origin/destination. We filter out data points which are far away from the reasonable sequence of temporally consecutive data points and result in unreasonable speed by using a clustering-based filter. Fig. 1 shows the data filtering algorithm. For each trip in T (T is the final set of segmented trips), the algorithm puts the consecutive points with a reasonable speed (less than λ_{speed}) together in a cluster (lines 4-5). When an unreasonable speed is detected, a new cluster is created and the points in the previous cluster are removed if the cluster is small (smaller than λ_{size}) in size (lines 6-8). The advantage of the clustering-based filter is that it can detect a group of outliers which are close to each other.

Algorithm 1 Data_Filtering(T, λ_{speed}, λ_{size})

```
1: for each trip t_i in T do
2:     cluster C = Ø
3:     for each point p_j in t_i do
4:         if the speed from last point p_{j-1} to current
   point p_j is less than λ_speed then
5:             Append p_j to C
6:         else
7:             if size(C) < λ_size then
8:                 Remove all the points in C from t_i
9:             C = [p_j]
```

Figure 1. Data filtering algorithm for cleaning the raw trips.

The output of the preprocessing step is a sequence of trajectories with the form $<(x_0, y_0, t_0),\ldots,(x_n, y_n, t_n)>$, where (x_i, y_i) is a longitude-latitude pair and t_i (i=0…n) is a time stamp ($\forall 0 \leqslant k < n, t_k < t_{k+1}$).

IV. CANDIDATE ORIGINS AND DESTINATIONS EXTRACTION

Origins and destinations are significant locations where a person usually departs from and goes to, so the candidate origins and destinations extraction is actually a problem to find a person's significant places that matter to his/her routes. Three kinds of clustering approaches, i.e. partitioning clustering, density-based clustering and time-based clustering could be applied to detect significant places from movement data. However, there are many problems for applying existing clustering algorithms on personal movement data due to its special characteristics.

Partitioning clustering algorithms based on K-Means are not suitable for our work due to several reasons. First, the number of clusters must be specified before the algorithm begins. This is impossible because the system does not have a priori knowledge about the places a person frequents. Second, all points are put into the final clusters. This is unreasonable for personal movement data because it contains on-the-way data points which should not be included in any clusters. Although density-based clustering approach has many advantages, it is also not suitable for processing the personal movement data. This is because personal movement data always contains places with sparse points, e.g. a person stays in a building where GPS loses its signal. Time-based clustering adapts to the temporal characteristic of movement data and is the fundamental approach for our origins and destinations extraction. However, the existing time-based clustering algorithms have a shortcoming that it requires continuous movement data collection [19][20]. As an example showed in Fig. 2(a), assume that there are three potential origins/destinations (i.e. A, B and C), and five trajectories (i.e. trajectory 1 to 5) taken by a person. If the person records all the five trajectories, the time-based clustering algorithm can find all three candidate origins/destinations (as showed in Fig. 2(b)). However, if the person fails to record trajectory 2 and trajectory 4 (as showed in Fig. 2(c)), the whole trajectory is interrupted at the potential origins/destinations and no place will be detected using existing time-based clustering algorithm. We call it a discontinuous trajectories recording problem which is extremely serious when using personal movement data since a person may often fail to record his/her movement data due to many reasons (e.g. forgetting to turn on the GPS receivers, forgetting to recharge the recording devices, etc.).

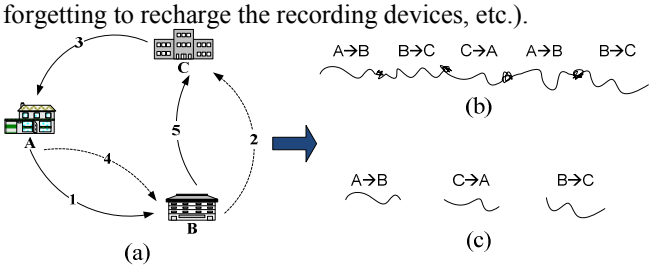

Figure 2. The discontinuous trajectories recording problem: (a) a person's movement among three potential origins/destinations. (b) continuously recorded trajectories. (c) incontinously recorded trajectories.

To counter the discontinuous trajectories recording problem, we propose a variant of time-based clustering algorithm called Forward-Backward Matching (FBM) to find candidate origins/destinations from segmented trajectories. The algorithm is depicted in Fig. 3. For each trajectory in T, the algorithm clusters the locations forward and backward along the time axis to find the first and last clusters (lines 1-5). Then it compares each cluster in forward clusters set FS with each cluster in backward clusters set BS. If the two clusters are close enough to each other (i.e. the distance of their centroids are smaller than $\lambda_{distance}$), they are merged and the result cluster is treated as a candidate origin/destination if its duration is longer than $\lambda_{interval}$ (lines 6-11). When appending a cluster to the final places set, the algorithm checks if the cluster is close enough to one of the existing clusters. If it is true, the two clusters are also merged.

FBM algorithm greatly alleviates the influence of discontinuous trajectories recording problem on candidate origins/destinations extraction. With the previous example showed in Fig. 2(c), by using FBM algorithm, there are three forward clusters (i.e. A, C, B) and three backward clusters

(i.e. B, A, C) are detected and each of them is not treated as a significant place due to their insufficient duration of time, but they are merged as candidate origins/destinations A, B and C in the matching step using FBM algorithm. Therefore, the three potential origins/destinations are still extracted in despite of the discontinuous trajectories recording problem. After discovering all the candidate origins/destinations, we reconstruct the trajectories by labeling the origin-destination annotation on them. The output is a sequence of trajectories with origin-destination annotation.

Algorithm 2 FBM(T, $\lambda_{distance}$, $\lambda_{interval}$)

```
1: for each trajectory t_i in T do
2:     cluster FC = cluster_forward(t_i)
3:     Append FC to forward clusters set FS
4:     cluster BC = cluster_backward(t_i)
5:     Append BC to backward clusters set BS
6: for each C_i in FS do
7:     for each C_j in BS do
8:         if distance(C_i, C_j) < λ_distance then
9:             C = merge(C_i, C_j)
10:            if duration(C) > λ_interval
11:                Append C to final places set PS
```

Figure 3. Forward-backward matching algorithm for discovering candidate origins/destinations.

V. MOVEMENT PATTERNS MINING

In this section, we describe how to extract movement patterns from trajectories with origin-destination annotation. The work can be divided into two steps, i.e. trajectory abstraction based on space partitioning and route patterns mining from abstracted trajectories.

A. Trajectory abstraction

A pattern is composed of frequently repeated elements in sequential data. However, trajectory patterns should not rely strictly on locations because locations do not repeat themselves exactly in every trajectory [13]. It is necessary to abstract the trajectories to adapt to this characteristic of trajectory patterns before they can be mined.

We apply the prevalent space partitioning approach [7][16][18] to abstract the trajectories. Given a person's trajectories, the approach equally divides his/her active area (i.e. the area he/she has visited) to form a grid. Then the person's trajectory can be transformed into a sequence of cells in this grid. This sequence of cells is defined as CTS as follow. The space partitioning approach is illustrated in Fig. 5(a).

Definition 1. (CTS) A Cell-Temporal Sequence (CTS) is a sequence of triples $<(cell_0, enterTime_0, leaveTime_0),\ldots,(cell_n, enterTime_n, leaveTime_n)>$, in which $cell_i$ (i=0…n) is a cell being visited, $enterTime_i$ is the time when the person enters the cell, $leaveTime_i$ is the time when the person leaves the cell, and $enterTime_k \leqslant leaveTime_k < enterTime_{k+1} \leqslant leaveTime_{k+1}$ ($\forall 0 \leqslant k < n$).

The CTSs can tolerate the inherent fuzzy characteristic of location data by regarding slightly different locations as the same, and also simplify the mining process. However, the space partitioning approach brings a problem known as the answer loss problem [17]. As illustrated in Fig. 4(a), suppose the active area of a person is divided into nine cells from 1 to 9, and there are three trajectories (i.e. T_1, T_2 and T_3) move among three origins/destinations (i.e. A, B and C). From the figure, T_1 and T_2 are very similar to each other, but they are expressed by different CTSs (i.e. 4->5->2->6 and 4->5->6) due to the slight deviation from cell 5 of T_1. This problem will seriously disturb the pattern mining procedure, e.g. the pattern 4->5->6 is lost in this example if the mining algorithm requires that a pattern repeats itself at least twice.

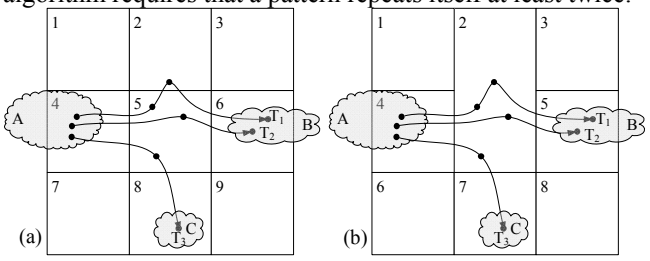

Figure 4. The answer loss problem and our simple solution: (a) the answer loss problem caused by space partitioning. (b) a simple solution based on combining relevant cells.

In order to alleviate this problem, we propose a simple approach that combines relevant cells to construct regions by taking advantage of the origin-destination knowledge. In our space-partitioning approach, every active cell (i.e. the cell has been visited) maintains a set of couples in the form of (*origin-destination, count*). We call this set an *OD-support set*. An OD-support set of a cell indicates the origin-destination annotations of the trajectories which pass through it and the number of trajectories that support to each origin-destination pair. The OD-support sets are created and updated in the process of transforming trajectories into CTSs. As illustrated in Fig. 4(a), the OD-support set of cell 5 is {(AB: 2), (AC: 1)} (i.e. as for cell 5, there are two trajectories go from A to B through it, and one trajectory goes from A to C through it).

Next, we combine relevant cells to construct regions. To decide whether two cells are relevant, we give the definition of origin-destination containment as follow.

Definition 2. (origin-destination containment) Given a cell C with OD-support set $S = [(OD_0, count_0),\ldots,(OD_n, count_n)]$, and another cell C' with OD-support set $S' = [(OD'_0, count'_0),\ldots,(OD'_m, count'_m)]$, we say that C' is *origin-destination contained* in C if $\forall OD_i$ (i=0…m) in S', $\exists OD_j$ (j=0…n) in S, such that: $OD_i = OD_j$, and $count_i < count_j$.

For each active cell, we test each of its neighbors and check whether it is origin-destination contained in the current cell. If it is true, we combine these two cells and take the new merged cell for further processing. Once there is no neighbor that can be merged into the current cell, it becomes a region. After the regions being constructed, we can convert CTSs into RTSs as the following definition (see Fig. 5(b)).

Definition 3. (RTS) A Region-Temporal Sequence (RTS) is a sequence of triples $<(region_0, enterTime_0, leaveTime_0),\ldots,(region_n, enterTime_n, leaveTime_n)>$, in which $region_i$ (i=0…n) is a region which is a set of relevant cells,

$enterTime_i$ is the time when the person enters the region, $leaveTime_i$ is the time when the person leaves the region, and $enterTime_k \leqslant leaveTime_k < enterTime_{k+1} \leqslant leaveTime_{k+1}$ ($\forall 0 \leqslant k < n$).

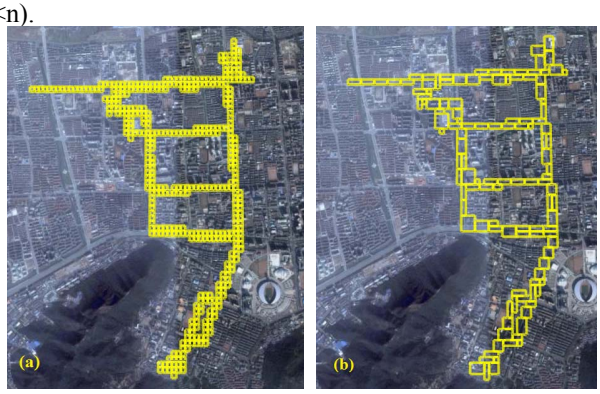

Figure 5. The trajectory abstraction strategy: (a) all the CTSs of a person. (b) all the RTSs of the same person.

By using RTSs to represent the trajectories, we can greatly alleviate the answer loss problem. As showed in Fig. 4, the OD-support set of cell 2, cell 5 and cell 7 are respectively {(*AB: 1*)}, {(*AB: 2*), (*AC: 1*)}, and {(*AC: 1*)}. Because cell 2 is origin-destination contained in cell 5 (with common origin-destination AB and its support count in cell 2 is less than that in cell 5), these two cells are combined into a region (i.e. region 2 showed in Fig. 4(b)). Cell 5 and cell 7 do not satisfy the origin-destination containment condition and will not be combined. As a result, the pattern 4->2->5 is reserved for two times repetitions. Moreover, there is an additional benefit of using RTSs.

B. Route patterns mining

Before describing the route patterns mining procedure, we first give the definition of route pattern used in this paper.

Definition 4. (Route pattern) A Route pattern is a 2-tuple *P = (R, ODS)*, in which R is an ordered sequence of regions, and ODS is an OD-support set with items *(OD, count)*, where *OD* is a origin-destination pair and *count* is the support count of R with origin-destination annotation OD in all RTSs.

We give an example to explain the concept of route pattern used in this paper. For example, a region sequence $<r_1, r_2, r_3>$ repeats four times in RTSs with the origin-destination annotation of AB (i.e. from place A to place B), and repeats six times in RTSs with the annotation of AC, the route pattern can be written as P = ($[r_1r_2r_3]$, {AB: 4, AC: 6}).

Our route patterns mining algorithm takes CRPM [7] as the basis and extends it to capture origin-destination feature of the pattern. The CRPM algorithm is based on PrefixSpan [21] and different from it for introducing the temporal continuity concept. The advantage of the CRPM algorithm is that it can tolerate the diversity in real trips and reserve the continuous properties of trips in patterns, and this characteristic meets the requirement of personal movement data mining. Fig. 6 shows the major part of the extended CRPM algorithm. The main concept used in the algorithm is the *prefix*, which is similar to the one used in PrefixSpan but different in two aspects. First, the adjacent elements in a prefix must satisfy the temporal continuity constraint (line 9). Second, we assign an additional field *OD-support set* to prefix when generating it (line 4), so the pattern contains the knowledge of its origin-destination annotation support condition. The algorithm calls itself recursively to extend the prefix to generate longer patterns like PrefixSpan. When the algorithm calls itself the first time, the initial parameter *projections* is the set of all the RTSs, and the initial prefixes are all the singular frequent regions. In each iteration of recursion, for each projection in the current set of projections, the algorithm searches in the projection for the element corresponding to the last item of the current prefix (line 2), generates new projections based on the element found (line 3), and then generates new sub-prefixes by looking for singular frequent regions in the new projections (line 4). If the sub-prefixes are not empty, the algorithm generates new prefix by concatenating the current prefix and one of the sub-prefixes if the temporal continuity constraint is satisfied (lines 8-10), and then a new route pattern is generated based on the new prefix (lines 11-12).

Algorithm 3 Pattern_Generation(*prefix*, *projections*, λ_{time})

```
1: for each projection_i in projections do
2:    lastRTelem = search_lastRTelem(prefix, projection_i)
3:    new_projs = generate_proj(projection_i, lastRTelem)
4:    sub_prefixes = generate_1_size_freq_item(new_projs)
5:    if size(sub_prefixes) == 0 then
6:       return
7:    else
8:       for each RTelem_j in sub_prefixes do
9:          if RTelem_j.enterTime – lastRTelem.leaveTime <
              λ_time then
10:            new_prefix = concatenate(prefix, RTelem_j)
11:            pattern = generate_pattern(new_prefix)
12:            Append pattern to final patterns set Patterns
13:            Pattern_Generation(new_prefix, new_projs, λ_time)
```

Figure 6. The major part of the extended CRPM algorithm for generating route patterns.

The route patterns mining algorithm based on PrefixSpan is too computational expensive to run directly on mobile phones. To counter this problem, we run the algorithm on a server, and a user can send his/her RTSs to server for route patterns mining to take advantage of the C/S architecture. This solution may cause little privacy problem because the RTSs with only region identifiers contain no real geographical information of the user.

VI. DESTINATION AND ROUTE PREDICTION

Our destination and route prediction approach based on the route patterns extracted in the previous step can be divided into three steps. First, we use a tree structure to represent the final route patterns. Second, we match the online movement information (i.e. the origin and the recent routes implied in the GPS traces recorded so far) of a person to the route patterns by stepping down the tree. Third, we predict the person's intended destination and continuous future routes according to the matching result.

A. Pattern tree building

The pattern tree organizes all the patterns by their prefixes. To build the tree, we scan all the route patterns. In seeing a new pattern, we search the tree to find that if there is any branch that corresponds to the longest prefix of the pattern. If it is found, we concatenate the rest of the elements of the pattern to this branch and update the confidence of each origin-destination annotation being supported. Otherwise, the pattern is inserted to the tree as a new branch.

For example, assume that there are fifteen trajectories, three candidate origins/destinations (i.e. A, B, C) and five active regions (i.e. r_1, r_2, r_3, r_4, r_5). Three distinct route patterns has been extracted, i.e. $P_1 = ([r_1r_2r_3], \{AB: 4\})$, $P_2 = ([r_1r_2r_4], \{BC: 6\})$, $P_3 = ([r_1r_5], \{AC: 5\})$. The pattern tree for the three route patterns is partially portrayed in Fig. 7. Every circular node (except the root) of the tree represents an active region (the number outside the parentheses is the region's identifier, and the number inside the parentheses is the support count of the pattern), every rectangular node of the tree represents the confidence of each origin-destination annotation supported by the pattern, and every branch of the tree represents a route pattern. For example, the leftmost branch represents the route pattern P1 supporting to origin-destination annotation AB with 100% confidence. Note that the route pattern's origin-destination support condition is represented by confidence instead of support count after the construction of pattern tree, e.g. P_1 is now written as $P_1 = ([r_1r_2r_3], \{AB: 100\%\})$. Obviously, there are more than three route patterns being exhibited in the tree, because all the sub-patterns which are derived from the distinct route patterns are also inserted into the tree (btw, the sub-patterns are also generated by the route patterns mining algorithm). Though the tree expands with redundant data, it facilitates the route pattern matching which will be discussed in the next section.

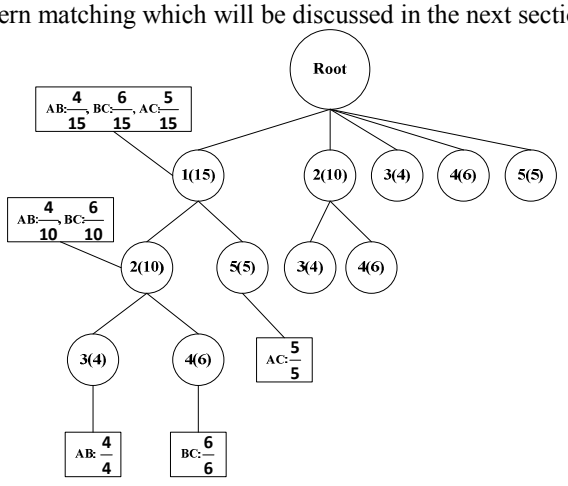

Figure 7. Example of the pattern tree.

B. Route pattern matching

Given the online movement information of a person, the route pattern matching procedure is to find the candidate route patterns whose prefix matches the person's recent region sequence. The advantage of the pattern tree structure proposed in the previous section is that route pattern matching can be conducted on the tree without searching for an entry. The direct children of the root node include all the active regions in the final route patterns because they are inserted into the pattern tree as singular route patterns, and all the patterns that take them as prefix can be found by directly stepping down the node. For example, given the input region sequence $<r_1, r_2>$, we look for the matching pattern by directly stepping down the first child (i.e. region r_1) of the root node, and two candidate matching route patterns $P_1 = ([r_1r_2r_3], \{AB: 100\%\})$ and $P_2 = ([r_1r_2r_4], \{BC: 100\%\})$ can be found.

C. Destination and continuous route prediction

After finding the candidate matching route patterns to the online region sequence, we predict the person's destination and future route by taking the probability of both the next region and origin-destination annotation into consideration. Our algorithm for prediction is depicted in Fig. 8. Given all route patterns *PS*, the algorithm first constructs pattern tree based on route patterns and gets input information from online movement data *online_data* (lines 1-2), then it conducts the route pattern matching procedure (lines 3-8). Since it may probably fail to get a matched pattern in the tree, in this case, we shorten the input region sequence with the earliest region (lines 9-10) to conduct the matching procedure again until finding a match (there is definitely a match when the input region sequence is shortened to just one region). When a matched pattern is found in the tree, the algorithm will step down the tree to predict the destination and future route by taking the probability of both the origin-destination annotation and next region into consideration (lines 12-14). One thing that is worth mentioning is that sometimes the online movement data may not contain origin information, our algorithm can also adapt to this situation by treating the output *true_origin* as arbitrary origin/destination in the prediction procedure, i.e. the condition "*origin == true_origin*" in line 12 always returns true.

Algorithm 4 Predict(*PS*, *online_data*)

```
1:  tree = build_tree(PS)
2:  true_origin, recent_regs = get_online_info(online_data)
3:  current_node = tree.root
4:  for each reg in recent_regs do
5:      if current_node.has_child(reg) return true then
6:          current_node = current_node.get_child(reg)
7:      else
8:          break
9:  if reg is not the last element in recent_regs then
10:     shorthen recent_regs with the earliest region and go
back to line 3
11: else
12:     get item (origin-destination: confidence) satisfying
            origin == true_origin with highest confidence in the
            rectangular node of current_node
13:     pred_destination = get_destination(item)
14:     iteratively step down tree until reaching a leaf node to
find next_region with the highest support to (true_origin,
pred_destination) annotation
```

Figure 8. The destination and route prediction algorithm.

Because the algorithm predicts the future route based on the matched route patterns, the length of prediction (i.e. how many future regions can be predicted) may not be long enough to reach the destination. In order to get longer route prediction, when one round of prediction is completed, we append the predicted route to the online movement data and run the prediction algorithm again until the latest predicted route is close enough to the destination, there is no more future region can be predicted, or a pre-defined iteration count is reached. We give an example to illustrate how the prediction algorithm works based on the pattern tree showed in Fig. 7. Suppose a person starts a new trip from origin A, and the recent region sequence is $<r_5, r_1, r_2>$, obviously no match can't be found in the pattern tree, so we shorten the recent region sequence to $<r_1, r_2>$. Now two matched patterns can be found, they are respectively P_1 = ($[r_1r_2r_3]$, $\{AB: 4\}$) and P_2 = ($[r_1r_2r_4]$, $\{BC: 6\}$). Because the rectangular node of tree node r_2 in the left branch is {AB: 4/10, BC: 6/10} and the true origin of the person is A, the prediction algorithm predicts the destination as B. Then with the origin-destination annotation AB, the prediction of next future region is r_3 since the pattern P_1 support to the origin-destination annotation AB with 100% confidence.

VII. PERFORMANCE EVALUATION

In order to evaluate the performance of the system, we conduct a number of tests by using real personal movement data collected from 14 participants. All participants involved in the experiment are students or faculties of Zhejiang University or their family members. To collect real personal movement data, we choose the widely used mobile phones as the recording devices. A program running on the mobile phone can connect to an external GPS receiver through Bluetooth and record GPS positions at an adaptive mode, i.e. the program dynamically adjusts the sampling frequency based on current movement speed (the sampling frequency is increased when the speed is getting higher).

All participants were instructed to carry out the experiment in an open-ended way to make the recorded movement data reflect their daily lives as truly as possible, i.e. they could take the recording devices during their daily life for arbitrary trips, e.g. going to work, going for shopping, going for a drive, etc. The final dataset contains over 1.5 million positions and nearly 1000 trajectories collected from the 14 participants for more than one month.

A. Movement data mining results

As mentioned previously, based on the origin-destination knowledge, the answer loss problem can be greatly alleviated by combining relevant cells together into regions. This will result in discovering longer route patterns from the same movement data. To demonstrate it, we run our mining algorithm on both CTSs and RTSs of the same movement data and compare the outputs. Before giving the results of the experiment, we first define "Maximum Pattern Length Proportion" as follow.

Definition 5. (MPLP) Given a set of route patterns extracted from a set of CTSs/RTSs, the Maximum Pattern Length Proportion is the ratio of the length of the longest route pattern to the average length of CTSs/RTSs the origin-destination annotation of which can be found in the OD-support set field of the longest route pattern.

MPLP reflects the relative length of the route patterns with respect to the real trajectories, and we take it as the evaluation metric. Fig. 9 compares the MPLP of route patterns extracted respectively from CTSs and RTSs of each participant's trajectories. From the figure, it can be found that the route patterns extracted from RTSs have much higher MPLP than that extracted from CTSs. This result means that by combining relevant cells to construct regions, our route patterns mining algorithm can derive much longer patterns with respect to the real trajectories from the same movement data.

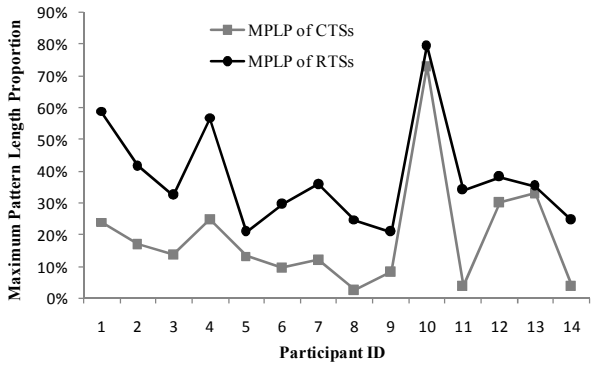

Figure 9. Comparison of the MPLP of CTSs and RTSs.

B. Destination and route prediction results

For the destination and route prediction evaluation, we use 10-fold cross validation where the route patterns are extracted from 90% of the movement data and the other 10% is used for test. This is repeated 10 times and the average performance is reported. We evaluate the performance of the prediction system from two aspects, which include the accuracy of destination and 1-step prediction, and the capability of continuous prediction.

Given the region where a person is currently located, destination prediction is to predict the candidate destination and 1-step prediction is to predict the very next region of the current region. Fig. 10 shows the correct rate of destination and 1-step prediction given the origin and the recent visited regions of a person. The overall accuracy of 10-fold cross validation is 79.6% for destination prediction and 58.4% for 1-step prediction.

This accuracy is not fairly high as compared with existing works which use vehicle movement data for analysis [3]. The main reason for this consists in the inherent characteristic of personal movement data which often contains irregular trajectories that have rarely been taken by the person. However, a vehicle such as bus always navigates along highly fixed paths. This fact can be demonstrated by Fig. 11 which shows the accuracy of destination and 1-step prediction for each round of the 10-fold cross validation. We first sort the rounds of 10-fold cross validation by the degree of their testing trajectories' irregularity, i.e. the round with smaller index is conducted on testing trajectories with higher

regularity. Then we track each round of the 10-fold cross validation and the accuracies are averaged over the experimental results of all the 14 participants. Fig. 11 clearly demonstrates that the system has fairly high performance on several rounds with smaller indices (the accuracy is roughly over 90% for destination prediction and over 80% for 1-step prediction), but the performance drops with the round's index. This means that in spite of the moderate performance on all movement data, the system works fairly well on regular trajectories, and has relatively low performance on irregular trajectories.

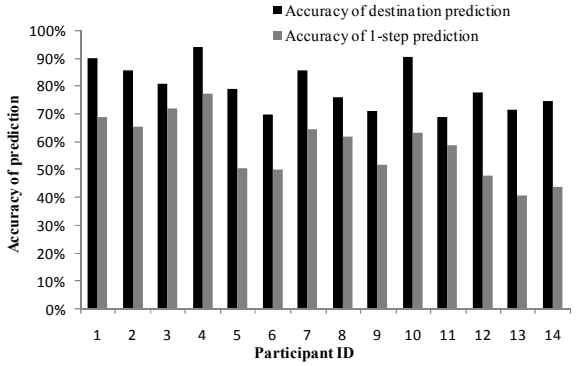

Figure 10. The overall accuracy of destination and 1-step prediction for each participant.

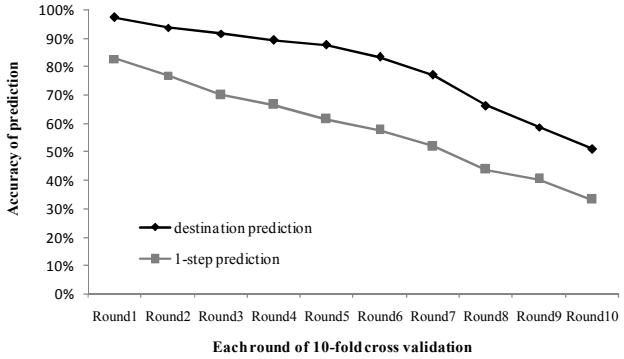

Figure 11. Average destination and 1-step prediction accuracy of each round of the 10-fold cross validation.

Another important performance metric of the prediction system is the capability of continuous route prediction. In order to evaluate it, we measure the similarity of the predictive future routes and the real future routes. The Hausdorff distance [22] is used as the similarity measure in the experiment. As illustrated in Fig. 12, first, given the predictive and real future routes represented by sequences of regions (see Fig. 12(a)), we find the centroids of all the predictive regions and real regions, and then connect them with lines respectively. The outputs are two polylines representing the predictive and real future routes (see Fig. 12(b)). Second, we adopt the Hausdorff distance algorithm to calculate the distance between these two polylines (see Fig. 12(c)). The value of the result distance reflects the average deviation of predictive route from real route. A small distance value indicates a high degree of similarity between predictive and real future routes, and means the prediction is more accurate.

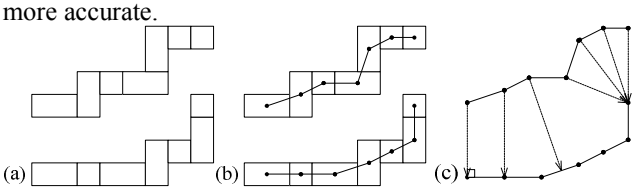

Figure 12. The Hausdorff distance algorithm for continuous route prediction evaluation: (a) two routes represented by sequence of regions. (b) connecting the centroids of the regions with lines. (c) using Hausdorff distance algorithm to calculate the distance between the two routes.

Fig. 13 shows the 10-fold cross validation results of the Hausdorff distances of the continuous route prediction for each participant. The average Hausdorff distance of all participants is 59.6 meters. This means the average deviation of continuous route prediction from real trajectories is approximately 60 meters. Similar to the destination and 1-step prediction, the continuous route prediction algorithm also has much better performance on regular trajectories than irregular trajectories. As portrayed in Fig. 14, with a similar experimental setting as that of the destination and 1-step prediction, the average Hausdorff distances over the experimental results of the 14 participants increase significantly with the round's index, and the distance value varies from a dozen meters to a hundred meters or more. The average Hausdorff distance of continuous route prediction on regular trajectories is about 20 meters.

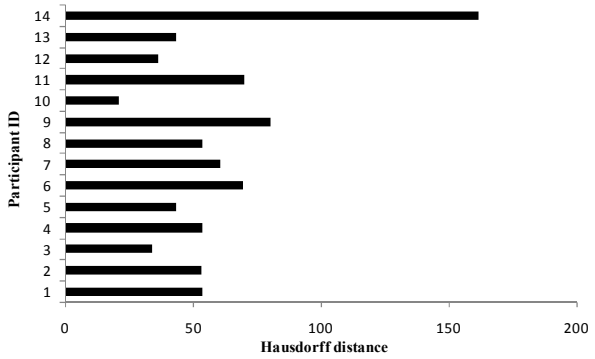

Figure 13. The overall Hausdorff distance for each participant.

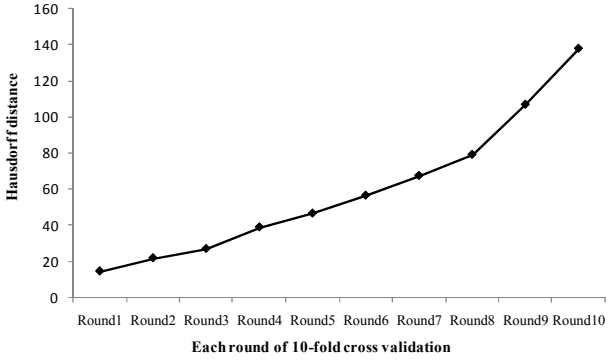

Figure 14. Average Hausdorff distance of each round of the 10-fold cross validation.

To summarize, our destination and route prediction system has satisfactory performance on all movement data (i.e. the overall accuracy is about 80% for destination prediction and 60% for 1-step prediction, and the average Hausdorff distance is about 60 meters), and has fairly high performance on regular trajectories (i.e. the accuracy is over 90% for destination prediction and over 80% for 1-step prediction, and the Hausdorff distance is about 20 meters).

VIII. CONCLUSION AND FUTURE WORK

This paper presents an approach for predicting both the intended destination and the continuous future route to reach the destination based on route patterns extracted from the rough personal movement data recorded by GPS without any a priori knowledge about the environment (e.g. road network). The experimental results drawn from a dataset of real personal movement data show that our system has satisfactory performance on the full set of movement data which contains irregular trajectories, and has fairly high performance on the regular trajectories. Furthermore, we have implemented a prototype for the Symbian S60 platform based on Python by using the algorithms proposed in this paper. The prototype extracts patterns from user's historical movement data and predicts his/her destination and future route on the fly.

There is still much future work left to be done. First, our prediction system is only based on the consideration of movement data, i.e. the context information of geographical locations. In the future, we can incorporate more context information such as time of day, day of week, traffic condition etc. into the system to provide more sophisticated prediction that can better adapt to the actual environment. Second, our system has relatively low performance on irregular trajectories. In the future, we can explore some techniques to bypass or alleviate the influence of the uncertainty of the system on unknown routes. Third, the intended destination and future route prediction is one kind of prediction of human behavior. By analyzing personal movement data, we can find much information that reflects a person's daily living habit, e.g. where he/she often has dines, what kind of transportation he/she often chooses, etc. In the future, we can utilize more learning techniques to find more knowledge than just the destination and routes.

REFERENCES

[1] M. C. González, C. A. Hidalgo, and A. L. Barabási, "Understanding individual human mobility patterns," Nature, vol. 453, no. 7196, 2008, pp. 779–782.

[2] J. Froehlich and J. Krumm, "Route prediction from trip observations," Proc. Society of Automotive Engineers World Congress, 2008.

[3] R. Simmons, B. Browning, Y. Zhang, and V. Sadekar, "Learning to predict driver route and destination intent," Proc. Intelligent Transportation Systems Conference, 2006, pp. 127-132.

[4] J. Krumm and E. Horvitz, "Predestination: inferring destinations from partial trajectories," Proc. Eighth International Conference on Ubiquitous Computing, 2006.

[5] T. Nakata and J. Takeuchi, "Mining traffic data from probe-car system for travel time prediction," Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 817-822.

[6] H. A. Karimi and X. Liu, "A predictive location model for location-based services," Proc. ACM Internation Symposium on Advances in Geographic Information Systems, 2003, pp. 126-133.

[7] Q. Ye, L. Chen, and G. C. Chen, "Predict personal continuous route," Journal of Zhejiang University, vol. 10, 2009, pp. 221-231.

[8] D. Ashbrook and T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," Personal and Ubiquitous Computing, vol. 7, no. 5, 2003, pp. 275-286.

[9] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Proc. International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226-231.

[10] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personally meaningful places: a interactive clustering approach," ACM Transaction on Information Systems, vol. 25, no. 3, 2007.

[11] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, "A clustering-based approach for discovering interesting places in trajectories," Proc. ACM Symposium on Applied Computing, 2008, pp. 863-868.

[12] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," Proc. ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, 2004, pp. 110-118.

[13] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. Cheung, "Mining, indexing, and querying historical spatiotemporal data," Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 236-245.

[14] H. Cao, N. Mamoulis, and D. W. Cheung, "Mining frequent spatio-temporal sequential patterns," Proc. IEEE International Conference on Data Mining, 2005, pp. 82-89.

[15] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias, "Spatio-temporal aggregation using sketches" Proc. International Conference on Data Engineering, 2004, pp. 214-225.

[16] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 330-339.

[17] C. S. Jensen, D. Lin, B. C. Ooi, and R. Zhang, "Effective density queries on continuously moving objects," Proc. International Conference on Data Engineering, 2006, pp. 71.

[18] H. Jeung, H. T. Shen, and X. Zhou, "Mining trajectory patterns using hidden Markov models," Proc. International Conference on Warehousing and Knowledge Discovery, 2007, pp. 470-480.

[19] N. Bicocchi, G. Castelli, M. Mamei, A. Rosi, and F. Zambonelli, "Supporting location-aware services for mobile users with the whereabouts diary," Proc. International Conference on Mobile Wireless Middleware, Operating Systems, and Applications, 2008.

[20] L. Liao, D. Fox, and H. Kautz, "Extracting places and activities from GPS traces using hierarchical conditional random fields," International Journal of Robotics Research, vol. 26, no. 1, 2007, pp. 119-134.

[21] J. Pei, J. Han, B. Mortazavi-Asl, and H. Pinto, "PrefixSpan: mining sequential patterns efficiently by prefix-projected growth," Proc. International Conference on Data Engineering, 2001, pp. 215-224.

[22] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge, "Comparing images using the Hausdorff distance," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 9, 1993, pp. 850-863.