Understanding Mobility Based on GPS Data

Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, Wei-Ying Ma
Microsoft Research Asia
4F, Sigma Building, No.49 Zhichun Road, Haidian District, Beijing 100190, P. R. China
{yuzheng, v-quali, v-yukche, xingx, wyma}@microsoft.com

ABSTRACT
Both recognizing human behavior and understanding a user's mobility from sensor data are critical issues in ubiquitous computing systems. As a kind of user behavior, the transportation modes, such as walking, driving, etc., that a user takes, can enrich the user's mobility with informative knowledge and provide pervasive computing systems with more context information. In this paper, we propose an approach based on supervised learning to infer people's motion modes from their GPS logs. The contribution of this work lies in the following two aspects. On one hand, we identify a set of sophisticated features, which are more robust to traffic condition than those other researchers ever used. On the other hand, we propose a graph-based post-processing algorithm to further improve the inference performance. This algorithm considers both the commonsense constraint of real world and typical user behavior based on location in a probabilistic manner. Using the GPS logs collected by 65 people over a period of 10 months, we evaluated our approach via a set of experiments. As a result, based on the change point-based segmentation method and Decision Tree-based inference model, the new features brought an eight percent improvement in inference accuracy over previous result, and the graph-based post-processing achieve a further four percent enhancement.

Author Keywords
GPS, GeoLife, Machine learning, Recognize human behavior, Infer transportation mode.

ACM Classification Keywords
I.5.2 [Pattern Recognition]: Design Methodology - *Classifier design and evaluation.*

INTRODUCTION

Understanding user mobility from sensor data is a central theme in ubiquitous computing. As an important kind of human behavior, people's transportation mode, such as walking, driving and taking a bus, etc., can endow their mobility with more significance and provide the pervasive computing systems with rich context information.

In the past decades, both coarse and fine-grained sensor data had been used to perform location-driven activity inference. On one hand, a strand of related work [4][9] attempt to recognize individual activity using the data collected by a cluster of wearable sensors. Although the recognition performance is relatively high, the human efforts on carrying many extra sensors are still open challenges. On the other hand, without adding extra overhead on carrying sensors, some researchers [5][11] aim to infer an individual's motion based on coarse-grained sensor data, such as a radio signal of GSM or Wi-Fi. Unfortunately, as the quality of data is not accurate enough, only simple motions like moving and being stationary can be discriminated in these techniques.

In recent years, GPS-phone and GPS-enabled PDA become prevalent in people's daily lives. With such devices people become more capable than ever of tracing their outdoor mobility and using location-based applications. Thus, quite a few projects [6][7][8][10] aiming to understand individual outdoor movements from GPS data have emerged. However, in these projects more attention has been paid on detecting significant locations of a user, predicting the user's movement among these locations and recognizing user-specific activity on each location. But location is only one part of an individual's state. Ideally, we would want to not only predict people's activity at a point location like moving destination but also infer user behavior, such as, the transportation modes, on the way to these point locations. Such high-level behavior would both enable the creation of new computing services that autonomously respond to a person's unspoken needs, and support much more accurate prediction about future behavior.

Another direction of GPS-driven pervasive computing is GPS trajectories sharing applications. In this category of computing services [1], people can record the trajectories of their journey using a GPS-enabled device, and then share life experience with others by exchanging these GPS tracks in an Internet community. At this moment, classifying people's GPS trajectories based on transportation mode is also a critical issue. By understanding the transportation mode of each GPS track, users can absorb rich knowledge from other people's travel experience. They are able to know not only where others have been but also how everyone reaches each location. Moreover, the systems will become more capable of distinguishing transportation mode

UbiComp'08, September 21-24, 2008, Seoul, Korea.

of each GPS trajectory. For instance, system should recommend a bus line rather than a driving way to an individual who is intending to move to somewhere by bus.

Unfortunately, to date, the classification mission of the services mentioned above still relies on users' manually labeling. This is not optimal for the development of these ubiquitous computing systems. Moreover, the identification methods based on simple rules, such as a velocity-based approach, cannot handle this problem with great effect due to the following reasons: 1) people usually change their transportation modes during a trip, i.e., a GPS trajectory may contain more than two kinds of modes. 2) The velocity of different transportation modes is usually vulnerable to traffic conditions and weather. Intuitively, in congestion, the average velocity of driving would be as slow as cycling.

In this paper, we aim to infer transportation modes including driving, walking, taking a bus and riding a bicycle from raw GPS logs based on supervised learning. It is a step toward recognizing human behavior and understanding user mobility for ubiquitous computing. Meanwhile, the work reported here is a step forward of paper [12]. Overall, the contributions of this work lie in:

- We identify a set of valuable features, which are more robust to traffic condition than the features we ever used and help greatly improve the inference accuracy.
- We propose a graph-based post-processing method using probabilistic cues to further improve the inference performance. In this method, we group multiple users' change points into nodes using a density-based clustering algorithm, and build edges between different clusters based on user-generated GPS trajectories. Subsequently, the probability distribution of different transportation modes on each edge, as well as the transition probability between consecutive edges, can be summarized from users' GPS logs. Such knowledge is promising to improve the inference accuracy as it contains both the commonsense constraint of the real world and typical user behavior based on location.

The rest of this paper is organized as follows. First we give a survey on the related work. Second, we introduce how we infer transportation mode from GPS logs. Here, we first present an overview on the framework of our approach, and then describe each step of the approach in details. Third, the experiment design and corresponding results are reported. Finally, we draw a conclusion and offer future work.

RELATED WORKS

***Solutions based on multiple sensors*:** In paper [4] and [9], Parkka et al. aim to recognize human activity, such as walking and running using the data collected by more than 20 kinds of wearable sensors on a person. A user's body condition, such as temperature, heart rate and GPS position, etc, as well as environment situation like environmental humidity and light intensity are employed as input features in a classification model to differentiate the person's everyday activities. However, it is somehow obtrusive and complex for normal users to carry extra sensors in their daily lives. On one hand, the major difference between the technique mentioned above and our work is in that we understand human activities only based on raw GPS data. Thus, it is more promising to be deployed in people's GPS phone without increasing their burden in wearing extra sensors. On the other hand, as GPS data can be a part of sensor ensemble, our approach is also useful to improve the recognition performance of such methods.

Solutions based on coarse-granted sensor data: LOCADIO [5] used a Hidden Markov Model to infer motion of a device using 802.11 radio signals while Timothy et al. [11] attempted to detect the mobility of a user based on GSM signal. Since the observations of radio signals vary in many conditions, such as time, space and number of users in a radio cell, etc., the locations that can be extracted from the signals are quite coarse. Thus, only three simple motions including stationary, walking and driving can be inferred in paper [11], and LOCADIO can only differentiate two fundamental types of activities, Still and Moving. The essential difference between our approach and this strand of work lies on not only the data provided by GPS sensor with higher locality accuracy but also spatial information which is mined from GPS data to improve the recognition performance. Given the fact that a GPS receiver will lose signal indoors, it is promising to enable a more sophisticated approach to recognize user activities by combining the technique mentioned above with ours.

GPS data-driven mining: With the increasing popularity of GPS, GPS data-based activity recognition has received considerable attention during the past years. These works include extracting significant places of an individual [3][7][8], predicting a person's movement [6][10] and modeling a user's transportation routine [8][10]. Patterson *et al.* [10] use GPS tracks to classify a user's mode of transportation as either "bus", "foot", or "car", and to predict his or her most likely route. Similarly, Liao et al. [8] aim to infer an individual's transportation routine given the person's GPS data. Their system first detects a user's set of significant places, and then recognizes the activities like shopping and dining among the significant places. As compared to our approach, the work has the following constraints: 1) It needs the information of road networks, bus stops and parking lots. 2) The model learned from a particular user's historical GPS data is customized for the user. Thus, it is not universal to be deployed in ubiquitous computing systems. However, we mine the knowledge from the raw GPS data collected by multi-users while the knowledge can contribute to both personal use and public use. What's more, we do not need additional information from other sensors or map information like bus stops.

INFERRING TRANSPORTAION MODE

In this section, we first clarify some preliminary knowledge about GPS data, and then give an overview on the framework of our method in which four steps including segmentation, feature extraction, inference and post-processing need to be performed sequentially. Here, we briefly introduce the segmentation and inference method proposed in paper [12] and focus on describing our work on features identification and post-processing.

Preliminary

Before describing the framework of our approach, we have to clarify the following terms: GPS trajectory, segment and change point. Basically, as depicted in the left part of Figure 1, a GPS log is a sequence of GPS points $p_i \in P$, $P=\{p_1, p_2, \ldots, p_n\}$. Each GPS point p_i contains latitude, longitude and timestamp. On a two dimensional plane, we can sequentially connect these GPS points into a trajectory, and divide the trajectory into trips if the time interval of the consecutive points exceeds a certain threshold. A change point stands for a place where people change their transportation modes. For instance, in the right part of Figure 1, a change point was generated when a person transfer transportation mode from driving to walking.

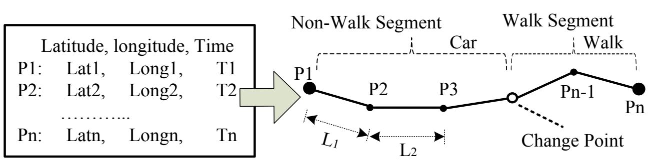

Figure 1. GPS log, segment and change point

Figure 2 depicts how we calculate features from GPS logs. Given two consecutive GPS points, e.g., p_1 and p_2, we can calculate the spatial distance L_1, temporal interval T_1 and heading direction ($p_1.head$) between them. The basis of heading direction is north.

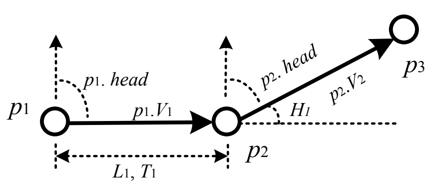

Figure 2. Feature calculation based on GPS logs

Subsequently, the velocity of p_1 can be computed by

$$p_1.V_1 = L_1/T_1; \qquad (1)$$

Then, the heading change, such as H_1, of consecutive three points like p_1, p_2 and p_3 can be calculated as equation 2.

$$H_1 = |p_1.head - p_2.head|; \qquad (2)$$

Further, more features, such as acceleration, expectation of velocity, etc. can be calculated in this way.

Architecture of Our approach

As shown in Figure 3, the architecture of our approach includes two parts, offline learning and online inference. In the offline learning part, on one hand, we first partition GPS trajectories into segments based on change points, and extract features from each segment. Then the features and corresponding ground truth are employed to train a classification model for online inference. On the other hand, using a density-based clustering algorithm, we group the multiple users' change points into clusters, and further build a graph on the clusters using user-generated GPS trajectories. From this graph we can extract some location-sensitive knowledge, such as probability distribution of different transportation modes on different edges, etc. The knowledge can be used as probabilistic cues to improve the inference performance in the post-processing. Further, to enhance the computing efficiency, a spatial index is built on the detected spatial knowledge.

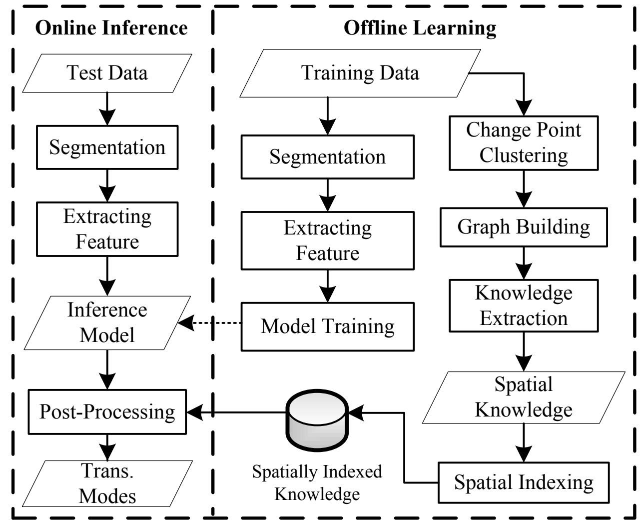

Figure 3 Architecture of our approach

In the online operation, when a GPS trajectory comes, like the offline training process, we first partition it into segments and extract the same features from each segment. Second, given the features, the generative inference model will predict the transportation mode of each segment in a probabilistic manner. Third, given the probability of a segment being different transportation modes, post-processing will improve the inference accuracy by leveraging the spatial knowledge extracted from training data. Finally, the transportation mode with maximum posterior probability will be used as the ultimate result.

Segmentation Method

Using the change point-based segmentation method proposed in paper [12], we partition a given GPS trajectory into several segments. This approach is derived from the following commonsense of the real world, and its effectiveness has been validated by the previous work [12]. 1) Typically, people must stop and then go when changing their transportation modes. 2) Walking should be a transition between different transportation modes. In other words, the start point and end point of a *Walk Segment* could be a change point in very high probability. Therefore, we first retrieve *Walk Segment* and then partition the trajectory into several portions with these *Walk Segments*.

Using Figure 4 as a case, we briefly demonstrate the four steps of the segmentation algorithm. For more information please refer to Section 4 of paper [12].

Step 1: Using a loose upper bound of velocity (V_t) and that of acceleration (a_t) to distinguish all possible *Walk Points* from *non-Walk Points*.
Step 2: If the length of a segment composed by consecutive *Walk Points* or *non-Walk Points* is less than a threshold, merge this one into its backward segment.
***Step 3*:** If the length of a segment exceeds a certain threshold, the segment is regarded as a *Certain Segment*. Otherwise it is deemed as an *Uncertain Segment*. If the number of consecutive *Uncertain Segments* exceeds a certain threshold, these *Uncertain Segments* will be merged into one *non-Walk Segment.*
***Step 4*:** The start point and end point of each *Walk Segment* are potential change points to partition a trip.

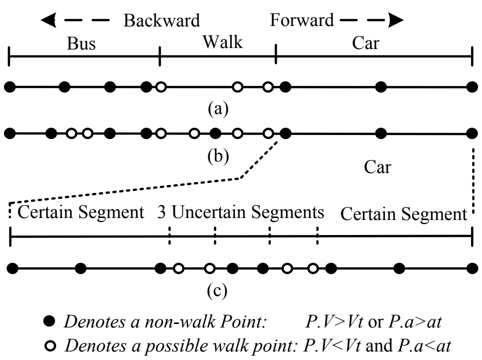

Figure 4. An example of detecting change points

Inference Model

In our previous work [12], four kinds of classification models, including Support Vector Machine (SVM), Decision Tree, Bayesian Net and Conditional Random Field (CRF) were studied. The results show that Decision Tree outperforms others based on the change point-based segmentation. Consequently, we still use this inference model in this paper.

Feature definition

In this subsection, we identify three features, which can be extracted from GPS logs and are more robust to the traffic conditions than those features we ever used.

Heading change rate (HCR): As illustrated in Figure 5, typically, being constrained by a road, people driving a car or taking a bus cannot change its heading direction as flexible as if they are walking or cycling, no matter what the traffic conditions they meet. Moreover, regardless of the traffic condition and weather, people walking or riding a bicycle inevitably and unintentionally wind their ways to a destination although they attempt to create a straight route.

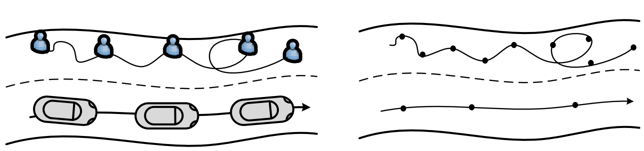

Figure 5 Heading change rate of different modes

In other words, the heading directions of different transportation modes differ greatly in being constrained by the real route while being independent of traffic conditions. Thus, *HCR* modeling this principle is defined as equation 3.

$$HCR = |P_c|/Distance, \qquad (3)$$

where P_c stands for the collection of GPS points at which a user changes his/her heading direction exceeding a certain threshold (*Hc*), and $|P_c|$ represents the number of elements in P_c. After dividing $|P_c|$ by the distance of the segment, *HCR* can be regarded as the frequency that people change their heading direction to some extent within a unit distance.

Stop Rate (SR): Figure 6 presents the typical trend of velocity when people take different transportation modes. We observe that, within a similar distance, people taking a bus are likely to stop more times than driving. Intrinsically, besides waiting at traffic lights at crossroads in a car, a bus would take passengers on or off at bus stops. Meanwhile, people walking on a route would become more likely than other modes to stop somewhere for many reasons, such as talking with passer-by, attracted by surrounding environments, waiting for a bus, etc. These observations motivate us to define two features to differentiate a variety of transportation modes; the one is the stop rate (*SR*) and the other is velocity change rate (*VCR*).

The *SR* stands for the number of GPS points with velocity below a certain threshold within a unit distance. Using a velocity threshold V_s, we can detect a group of GPS points (P_s), whose velocity is smaller than V_s. Then, we can calculate the *SR* as equation 4.

$$SR = |P_s|/Distance \qquad (4)$$

Where $P_s = \{p_i \,|\, p_i \in P, p_i.V < V_s\}$. Obviously, as shown in Figure 6, we can see *SR* (*Walk*) > *SR* (*Bus*) > *SR* (*Driving*).

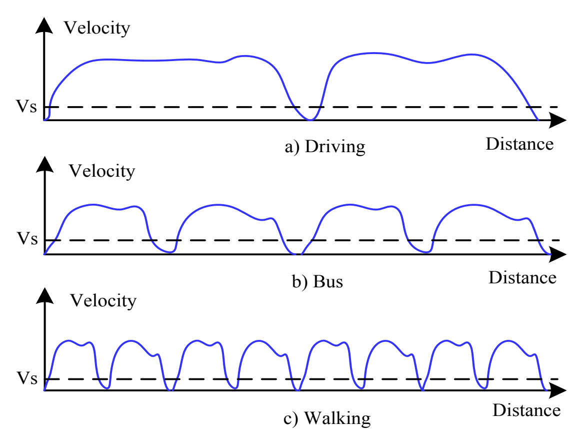

Figure 6 Velocity change rate and stop rate

Velocity Change Rate (VCR): The other hint we get from Figure 6 is to leverage the *VCR* to classify different modes. First, we can calculate the *VRate* of each GPS point as equation (5). Then, we can get the statistics of the number of GPS points whose *VRate* are greater than a certain threshold V_r, and calculate *VCR* according to equation (6).

$$p_1.VRate = |V_2 - V_1|/V_1; \qquad (5)$$

$$VCR = |P_v|/Distance; \qquad (6)$$

Where $P_v = \{p_i \mid p_i \in P, p_i.VRate > V_r\}$. Consequently, we can understand the VCR as the number of GPS points with a velocity change percentage above a certain threshold within a unit distance. The two features clearly capture the difference among variable transportation modes, and get supports from the later experiment results.

Spatial Knowledge Extraction

Figure 7 illustrates the four steps toward extracting spatial knowledge from users' GPS logs. The knowledge includes a change point-based graph and the probability distribution on each edge of the graph.

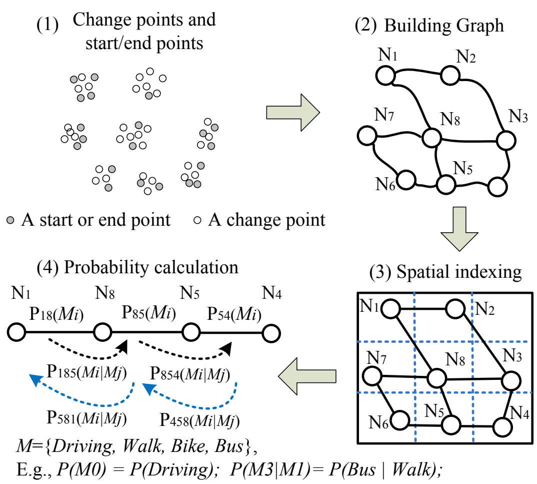

Figure 7. Extract spatial knowledge from GPS logs.

First, given GPS logs with labeled ground truth, we can get the special points including change points and the start/end points of each GPS trajectory. These special points were subsequently grouped into several nodes (clusters) using a density-based clustering algorithm. The reasons we prefer to use density-based clustering instead of agglomerative methods, such as K-Means, lie in two aspects. One is a density-based approach is capable of detecting clusters with irregular structures, which may stand for bus stops or parking places. The other is that it can help us discover popular places where most people change their transportation modes, while removing sparse change points representing places with low access frequency.

Second, with the GPS trajectories from multiple users' GPS logs, we can construct an undirected graph with nodes being clusters of the special points mentioned above and edges being the transportation between nodes. Here we do not differentiate different trajectories with similar start/end points for simplicity, i.e., all the trajectories passing two graph nodes are regarded as similar.

Third, we build a spatial index over the graph to improve the efficiency of accessing the information of each node and edge. Fourth, we can calculate the probability distribution of different transportation modes on each edge. For instance, as depicted in the fourth step of Figure 7, $P_{18}(Bus)$ stands for the likelihood of the event that people take bus on the edge between node 1 and node 8. Further, the conditional probability between different transportation modes can also be computed based on the graph. E.g., $P_{185}(Bus|Walk)$ represents the transition probability from Walk to Bus between edge 18 and edge 85. In other words, it denotes the likelihood of the event that a user walks from node 5 to node 8 based on the observed occurrence that the user takes bus from node 1 to node 8.

Such knowledge mentioned above is promising in improving the inference accuracy due to the following reasons. 1) It implies people's typical motion among different places. The clusters of change points represent the places many people change their transportation modes. Usually, these places could be bus stops, parking lots and railway stations, etc. We can take into account user behavior among these nodes as probabilistic cues when we infer other trajectories passing these two nodes. Naturally, for example, if most of users take a bus between two nodes, we can get suggestions that the two nodes could be bus stops, and the edge between them could have a very high probability of being a bus line. 2) The probability on each edge implies constraint of the real world. For instance, buses only take passengers on at bus stops, cars are left in parking lots, and cars and buses only travel on streets, *etc.*

Graph-based Post-Processing

The graph-based post-processing algorithm takes the preliminary inference result and the spatial knowledge mentioned above as input, and aims to generate an improved prediction result. To distinguish this algorithm from the post-processing method in paper [12], we call the previous one normal post-processing. The graph-based post-processing includes three components: normal post-processing, prior probability-based enhancement and transition probability-based enhancement.

Framework of post-processing

Figure 8 shows the flowchart of the graph-based post-processing we designed. When an inferred segment of GPS trajectory appears, we first search the spatial index to quickly match its vertexes against graph nodes. If we cannot find graph nodes close to the vertex, the normal post-processing algorithm is employed to enhance the inference performance. Otherwise, the prior probability-based enhancement or the transition probability-based method would be used. If the maximum posterior probability of the segment being a kind of transportation given feature X exceeds a certain threshold T_1, we believe the transportation mode corresponding to the maximum probability would be a correct inference. Subsequently, we can leverage this segment to revise its adjacent segments using the transition probability-based enhance method. If the situation is opposed to the above assumption, we will further check whether its maximum probability is smaller than another threshold $T2$. If the condition holds, the prior probability-based enhancement method is performed. Otherwise, the normal post-processing method is used. When a segment holds the conditions being processed by both the prior probability-based and the transition probability-based enhancement approaches, the latter is preferable to be used. Finally, we select the transportation mode with maximum probability as the prediction result of each segment.

The idea of post-processing is motivated by the observation that when a segment's maximum $P(m_i|X)$ exceeds a threshold ($T1$ in Figure 8), it is very likely to be a correct prediction. Hence, it can be used to fix the potentially false inference adjacent to it in a probabilistic manner. Instead, if the maximum posterior probability of a segment is less than a certain threshold T_2, it would be a false inference in a very high probability. Consequently, it deserves our revision. With the threshold T_1 and T_2, we are more likely to correct the false prediction while maintaining the accurate inference results. (See Figure 17 for evidences.)

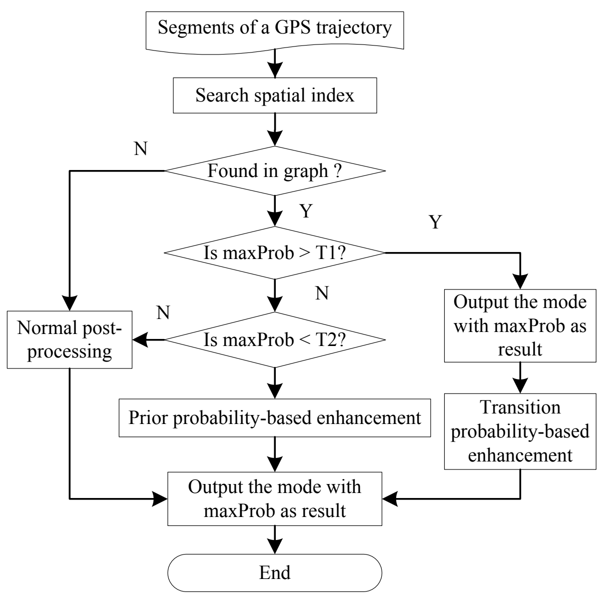

Figure 8 Flowchart of the post-processing

Normal post-processing

Using the trajectory depicted in Figure 8 as an example, we briefly introduce the normal post-processing proposed in paper [12]. It attempts to improve the prediction accuracy by considering the conditional probability between different transportation modes. After implementing the preliminary inference, we can get the predicted posterior probability of each segment being different transportation modes given feature X. If we directly select the transportation mode with maximum posterior probability as the final results, the prediction would be Driving→Bike→Bike while the ground truth is Driving→ Walk→ Bike, i.e., a prediction error occurred. On this occasion, if a segment, e.g., segment i-1, whose posterior probability being a kind of transportation mode exceeds threshold T_1 (0.6 used in our experiment), we select the transportation mode as the final prediction, and use it to revise the inference results of its adjacent segments. Then, according to equation (6) and (7), we can respectively re-calculate the posterior probability of segment i being different transportation modes conditioned by the transportation mode of segment i-1.

$$Segment[i].P(Bike) = Segment[i].P(Bike) \times P(Bike|Driving), \quad (6)$$

$$Segment[i].P(Walk) = Segment[i].P(Walk) \times P(Walk|Driving) \quad (7)$$

where $P(Bike|Driving)$ and $P(Walk|Driving)$ stands for the transition probability from driving to riding a bike and that from driving to walking. $Segment[i].P(Bike)$ represents the probability of riding bike on the segment i. After the calculation, we use the transportation mode with maximum probability as the final results. In the case depicted in Figure 8, since the transition probability from driving to riding bike is very small, the probability of Bike will drop behind Walk after the multiplication as equation (6) and (7).

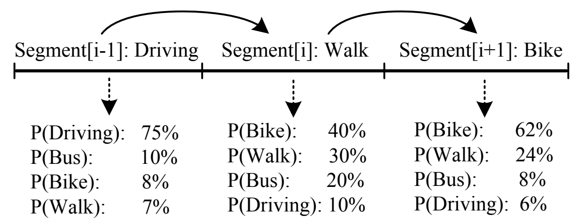

Figure 9. Perform normal post-processing on a GPS trajectory

Prior probability-based enhancement on graph

When a segment is detected to pass two graph nodes (i, j), we can use the prior probability distribution on the corresponding edge (E_{ij}) to reduce the incorrect prediction. Actually, what we want to predict is the posterior probability of transportation mode (m_i) on E_{ij} based on observed feature X, i.e., $P(m_i|X, E_{ij})$. Since E_{ij} is involved in the condition part, the probability is more discriminative and informative than $P(m_i|X)$ used in the preliminary inference model. As shown in equation (8), by applying Bayesian rule, we decompose $P(m_i|X, E_{ij})$ into prior probability and the likelihood of seeing (X, E_{ij}) given transportation mode m_i. Then, using the assumption X is independent of E_{ij}, we can further decompose $P(X, E_{ij} \mid m_i)$ into $P(X \mid m_i)$ and $P(E_{ij} \mid m_i)$.

$$\begin{aligned}
P(m_i|X, E_{ij}) &= \frac{P(X, E_{ij} \mid m_i)P(m_i)}{P(X, E_{ij})} \\
&= \frac{P(X \mid m_i)P(E_{ij} \mid m_i)P(m_i)}{P(X)P(E_{ij})} \\
&= \frac{P(m_i|X)P(X)}{P(m_i)} \cdot \frac{P(m_i|E_{ij})P(E_{ij})}{P(m_i)} \cdot \frac{P(m_i)}{P(X)P(E_{ij})} \\
&= \frac{P(m_i|X) \cdot P(m_i|E_{ij})}{P(m_i)} \qquad (8)
\end{aligned}$$

Figure 10 presents how to calculate each element of equation (8) after the transformation. As we can see, $P(m_i|E_{ij})$ and $P(m_i)$ can be summarized from multiple users' dataset while $P(m_i|X)$ is difficult to be directly calculated since the elements of X may not be independent of each other. So, in this work, we use the posterior probability generated by the preliminary inference model as an approximate substitute. From the theoretic perspective, we would face some risks in computing $P(m_i|X, E_{ij})$ since the assumption of independence between X and E_{ij} is subject to challenge, and $P(m_i | X)$ is substituted by an approximate value. Thus, we need to use it carefully to ensure the effectiveness of this approximate calculation. That is another reason why we need threshold T_1 and T_2.

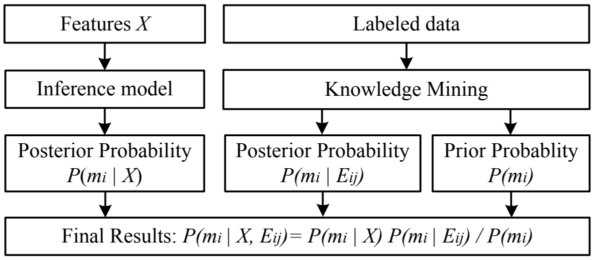

Figure 10. Prior probability-based enhancement method

Transition probability-based enhancement on the graph

This process is performed only when we find consecutive segments on the graph and one segment's $P(m_i | X)$ exceeds the threshold T_1. The process is similar to the normal post-processing algorithm while the difference is in that the transition probability between different transportation modes is based on the graph. Since the probability is location-constraint and contains more commonsense information of the real world, it is more useful and informative than the normal transition probability summarized from all users' ground truth regardless of location.

EXPERIMENTS

In this section, we first describe the framework of the experiments we performed. Second, the experiment setup including GPS devices, GPS data and toolkits are presented. Third, we respectively describe how we select the parameters for each algorithm. Finally, we report detailed experimental results with corresponding discussions.

Framework of Experiments

Figure 11 shows the framework of the experiments, in which we focus on evaluating the effectiveness of the new features including *HCR*, *SR* and *VCR*, and testing the effect of the graph-based post-processing. For simplicity's sake, we call the features used in [12] *Basic Features*.

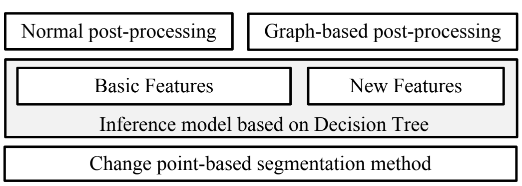

Figure 11. Framework of the experiment we performed

Segmentation and inference model: In our previous work [12], we conclude that the change point-based method outperforms other competitors, and the Decision Tree shows its advantages beyond others in inferring transportation mode over the segmentation method. Thus, we also employ the change point-based method to partition GPS trajectories into segments, and use the Decision Tree-based model to perform predictions. The parameters of the segmentation method take the same values as the work [12].

Features: Table 1 presents the features we explored in this experiment. Besides the top features proposed in paper [12], we are more curious about the bottom three features.

Features	Significance
Dist	Distance of a segment
MaxVi	The ith maximum velocity of a segment
MaxAi	The ith maximum acceleration of a segment
AV	Average velocity of a segment
EV	Expectation of velocity of GPS points in a segment
DV	Variance of velocity of GPS points in a segment
HCR	Heading Change Rate
SR	Stop Rate
VCR	Velocity Change Rate

Table 1. The features we explored in the experiment

Post-processing: In this step, we aim to evaluate the effectiveness of graph-based post-processing, and compare it with the normal post-processing algorithm. At first, we group multiple users' change points using OPTICS [2], a density-based clustering algorithm, which can help us detect data clusters with irregular structures. Further, a graph is built based on these clusters and GPS trajectories.

Evaluation: To validate the effectiveness of each approach, both the inference accuracy of transportation mode and that of change points are investigated. With regard to the prediction accuracy of transportation mode, we focus on the *Accuracy by Segment* (A_S) and *Accuracy by Distance* (A_D).

$$A_S = m/N \; ; \quad (9)$$

$$A_D = \frac{\sum_{j=0}^{m} CorrectSegment\ [j].Distance}{\sum_{i=0}^{N} Segment\ [i].Distance} \; ; \quad (10)$$

where N stands for the total number of the segments, while m denotes the number of segments being correctly predicted. Intuitively, it is more important to correctly predict a segment with long distance than that with short distance. So, A_D is more objective to measure the inference accuracy. Regarding the inference accuracy of change points, we explore their recall and precision while their recall has higher priority over its precision. If the distance between an inferred change point and its ground truth is within 150 meters, we regard the change point as a correct inference.

Settings

GPS devices: Figure 12 shows the GPS devices we chose to collect data in a frequency of one record every two seconds. They are comprised of stand-alone GPS receivers and GPS phones. Subjects carry such devices in their daily lives as long as they are outdoors, and travel wherever they like.

GPS Data: Figure 13 shows the distribution of the GPS data we used in the experiments. It is collected by 65 users over a period of 10 months. In each day of the data collection, to help data-creators manually label their GPS trajectories, we respectively visualize each person's traces on a map and show the timestamp of each GPS point. Given the short time span between data created and data labeled, users are enabled to reflect on when and where they change transportation modes based on their reliable memory. Therefore, users can label their data in the following way, 2:35:02– 2:55:24 pm, bike; 2:55:26-3:02:04pm, walk.

Figure 12 GPS devices used in the experiments

The data covers 18 cities, and its total length has exceeded 30,000 kilometer. For each user, about 70 percent of the data are used as a training set while the rest is used as test data. Each trajectory is segmented into trips if the interval between consecutive GPS points exceeds 20 minutes.

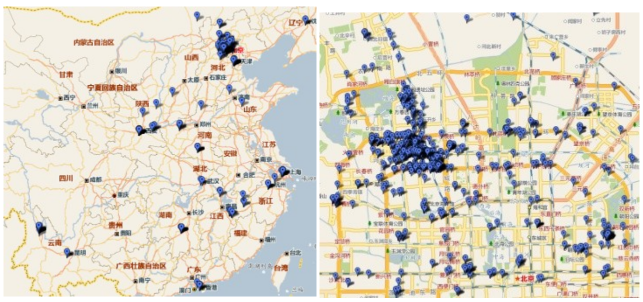

Figure 13 Distribution of the data used in the experiments

Parameter selection

Selecting parameter for SR, VCR and HCR

HCR: Figure 14 shows the inference accuracy changing over the threshold (*Hc*) when *HCR* is used alone to differentiate different modes. The curve painted in Figure 14 presents us with the prior knowledge that *HCR* becomes the most discriminative when *Hc* is set to 19 degree. In other words, when a user changes his/her heading direction by a angle greater than 19 degrees, the corresponding GPS point will be sampled into collection *Pc*, and further calculate *HCR* according to equation 3.

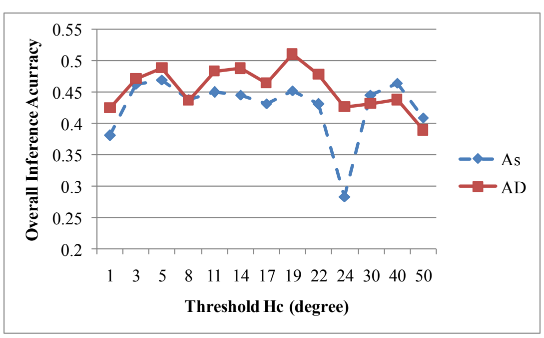

Figure 14. Selecting threshold *Hc* for *HCR*. The inference is performed only based on *HCR* without post-processing

SR: In Figure 15, we study the effect of *SR* in stand-alone predicting transportation mode. Obviously, we can get the suggestion that, as compared to other candidates, when *Vs* equals to 3.4, *SR* shows its greatest advantages in classifying different transportation modes.

VCR: Figure 16 depicts the inference accuracy changing over the threshold V_r when we employ *VCR* alone. We found evidence that when V_r is set to 0.26, i.e., a GPS point would be sampled into collection *Pv* if the velocity of this GPS point changes beyond 26 percent over its predecessor, *VCR* become the most powerful beyond other candidates.

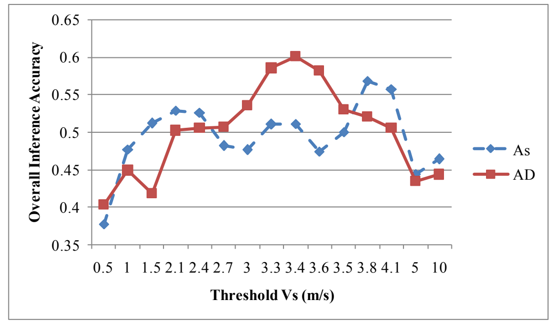

Figure 15. Selecting threshold (*Vs*) for *SR*. *SR* is the only feature used in the inference model without post-processing

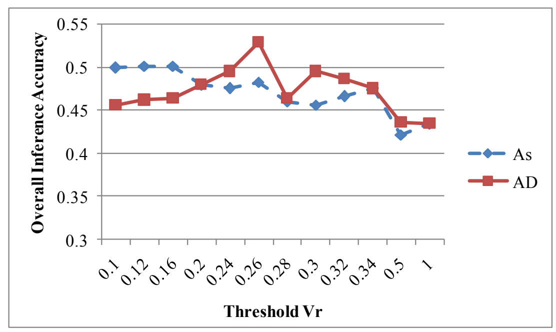

Figure 16. Selecting threshold (*Vr*) for *VCR*. *VCR* is the only feature used in the inference model without post-processing

Consequently, based on the experiment results we can select *Hc*=19 for *HCR*, *Vs*=3.4 for *SR,* and *Vr*=0.26 for *VCR*.

Determining T_1 and T_2 for post-processing

Figure 17 shows the statistical results we performed based on the preliminary inference results.

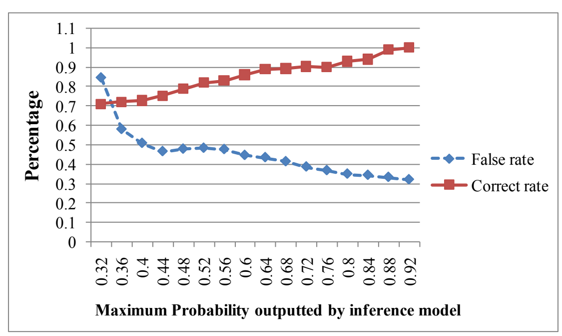

Figure 17. The statistic results of relationship between maximum $P(m_i|\, X)$ and inference accuracy

Using the relationship between the value of maximum posterior probability $P(m_i|\, X)$ of a segment being a kind of transportation mode given feature X and its inference result, we found the following evidences. When the value of maximum $P(m_i | X)$ is smaller than 0.36, the rate of false inference exceeds 60 percent. Instead, when the value is greater than 0.6, the rate of correct inference outscores 90 percent. Thus, we set T_1 to 0.6 and T_2 to 0.36 to reduce the risk of modifying a correct prediction.

Selecting parameter for OPTICS Clustering

Using the GPS data within a given geographic region, Figure 18 presents the clustering results of OPTICS changing over the number of GPS trajectories. As a result, the number of clusters within the region does not keep on increasing with the incrementally added GPS trajectories. It proves that the number of places where most people change their transportation in a given region is limited, and is constrained by the real world. It also provides positive support on the feasibility of our graph-based post-processing. With regard to the OPTICS algorithm, its result depends on two parameters, core-distance (*CorDist*) and minimal points (*minPts*) within the core-distance. According to the commonsense knowledge of real world and experimental evaluation, we found that when *CorDist=25* and *minPts=5,* the distribution of clusters make more sense than that based on other parameters.

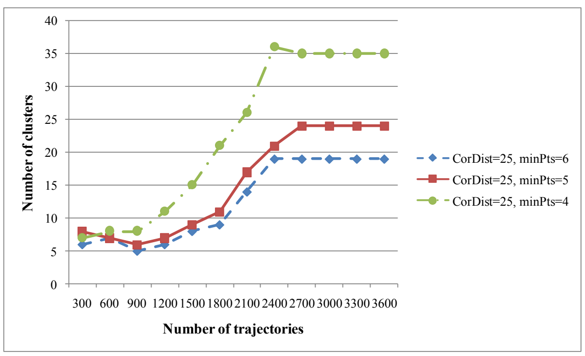

Figure 18. Change point-base clusters changing over the number of GPS trajectories.

Figure 19 paints the change point-based graph on a map, and displays the most popular transportation mode on each graph edge. Each purple circle stands for a cluster and the line between two circles represents a graph edge.

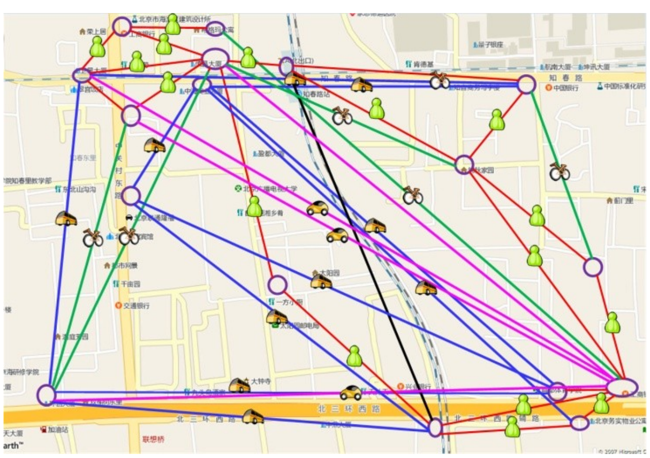

Figure 19 A change point-based graph within a region

Results

Single feature exploration: Using the inference accuracy without post-processing, Table 2 shows the capability of each feature in stand-alone discriminating different transportation modes. As we can observe, *SR* and *HCR* outperform other features, and *VCR* also show its advantages over others competitors except *AV*.

.	Feature	A_S	A_D		Feature	A_S	A_D
1	***SR***	0.35	**0.561**	8	*DV*	0.27	0.357
2	***HCR***	0.34	**0.561**	9	*MaxV2*	0.32	0.344
3	*AV*	0.39	0.547	10	*MaxV1*	0.29	0.257
4	***VCR***	0.34	**0.526**	11	*MaxA2*	0.24	0.217
5	*EV*	0.40	0.523	12	*MaxA1*	0.26	0.208
6	*Dist*	0.30	0.499	13	*MaxA3*	0.27	0.197
7	*MaxV3*	0.33	0.365				

Table 2 Inference accuracy using each feature alone

Effectiveness of feature combination: Using a subset feature selection method, we evaluate the performance of different feature combination. Table 3 presents some results. Both the accuracy of predicted transportation mode as well as precision (CP/P) and recall (CP/R) of change point are explored. The *Basic Feature* includes *MaxV1, MaxV2, MaxV3, MaxA1, MaxA2, MaxA3, AV, EV, DV* and *Dist,* while the *Enhanced Feature,* with which we got the best performance in this experiment, contains *HCR, VCR, SR, MaxV1, MaxV2, MaxA1, MaxA2, AV, EV* and *DV*. The inference result of *Basic features* is slightly less than that reported in paper [12] due to the increased test data.

	A_S	A_D	CP/P	CP/R
MaxA1+MaxA2+MaXA3	0.460	0.461	0.279	0.718
MaxV3+MaxA3+AV	0.508	0.542	0.297	0.728
MaxV1+MaxV2+MaxV3	0.521	0.585	0.211	0.726
SR+HCR+VCR	0.532	0.667	0.274	0.738
Basic Features	0.562	0.648	0.244	0.781
Enhanced Features	**0.583**	**0.728**	**0.278**	**0.789**

Table 3 Inference performance of some feature combination without implementing post-processing

From the results shown in Table 3, on one hand, we can get the suggestion that the combination of *SR+HCR+VCR* is more effective than that of velocity and acceleration in predicting users' motion modes. It justified our statement that these three features are more robust to traffic condition than *Basic Features*. On the other hand, by combining *SR+HCR+VCR* with some of the *Basic Features*, we score the highest accuracy. This evidence further proves that our feature is discriminative and has little correlation between existing features. However, adding more features like *MaxV3* to *Enhanced Features*, we cannot improve the inference performance as we expect. Instead, the accuracy drops behind 0.7. This phenomenon may be caused by the correlation between original features, such as MaxV3 and MaxV1, etc. Meanwhile, the data we used may not be sufficient enough to allow the inference model to learn a more robust model, although the scale of data we collected is much larger than related research projects.

Effect of post-processing: Table 4 shows the comparison of inference performance with and without post-processing. We observe that the normal post-processing has achieved almost 2 percent improvement in accuracy (A_D) based on the inference model using *Enhanced Features,* while the graph-based post-processing outperforms the normal method by bringing a 4 percent promotion over the preliminary inference result. The detailed inference results, including precision and recall of each kind of transportation mode by distance, are presented in Table 5.

	A_D	CP/P	CP/R
Enhanced Features (*EF*)	0.728	0.278	0.789
EF + normal post-processing	0.741	0.314	0.778
EF + graph-based post-processing	**0.762**	**0.342**	**0.771**

Table 4 Comparison between normal post-processing and graph-based post-processing

Ground truth	Predicted Results (KM)					
	Walk	*Driving*	*Bus*	*Bike*		
Walk	1026.4	122.1	386.5	357.3	0.543	**Recall**
Driving	42.6	2477.3	458.5	235.1	0.771	
Bus	34.8	164.7	1752.4	46.2	0.877	
Bike	49.3	113.5	31.9	1234.3	0.864	
	0.891	0.861	0.666	0.659	**0.762**	
	Precision					

Table 5. Fusion matrix of final inference results with graph-based post-processing

With a large-scale dataset, we believe that the graph-based post-processing would bring a greater improvement to current experimental results. On one hand, with more GPS data, the change point-based graph could cover more places where people log their trajectory with GPS data. Thus, more inferred GPS trajectories can be matched on the graph, and further be processed by the graph-based method. On the other hand, the probability distribution on the graph would become more capable of representing typical user behavior on locations.

CONCLUSION

In this paper, we report the work aiming to infer transportation modes from GPS logs based on supervised learning. In the approach described here, we identify a set of sophisticated features including heading change rate, velocity change rate and stop rate. Beyond simple velocity and acceleration, they are more robust to traffic condition and contain more information of users' motion. Subsequently, we propose a graph-based post-processing algorithm to further improve the inference performance. This algorithm considers the probabilistic cues including the commonsense constraint of real world and typical user behavior based on location. Using the GPS logs collected by 65 people over a period of 10 months, we evaluated our approach via a set of experiments. As a result, *SR* outperformed other features ever used in previous work, and the combination of *SR+HCR+VCR* also show its advantage beyond others. Based on the change point-based segmentation method and Decision Tree-based inference model, we got 72.8 percent inference accuracy by employing *SR, HCR* and *VCR*. Further, the graph-based post-processing promoted the inference performance (by distance) to 76.2 percent with 4 points improvements. With a larger scale of dataset, we believe the graph-based method would bring more improvement to inference performance.

Future work in this area includes exploring more knowledge from the change point-based graph and incorporation of additional prediction features. As example of the former, temporal dimension could be considered in the graph, while for the later road network and point of interest could be useful.

REFERENCES

1. GPS Track log route exchange forum: http://www.gpsxchange.com
2. Ankerst, M. Breunig, M.M., Kriegel, H., Sander, J., OPTICS: Ordering points to identify the clustering structure. In *Proc.* of *SIGMOD 99*, ACM Press (1999): 49-60.
3. Ashbrook, D., Starner, T., Using GPS to learn significant locations and predict movement across multiple users. Personal and Ubiquitous Computing 7, 5(2003), 275-286.
4. Ermes, M., Parkka, J., Mantyjarvi, J., Korhonen I., Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions, IEEE Transactions on Information Technology in Biomedicine 12, 1(2006), 20-26.
5. Krumm, J., Horvitz, E., LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths. In *Proc.* of *Mobiquitous 2004*, IEEE Press (2004), 4-13.
6. Krumm, J., Horvitz, E., Predestination: Inferring Destinations from Partial Trajectories. In *Proc.* of *UBICOMP'06*, Springer-Verlag Press(2003), 243-260
7. Liao L., Patterson, D.J., Fox, D., Kautz, H., Building Personal Maps from GPS Data. *IJCAI MOO05*, Springer Press(2005), 249-265
8. Liao L., Fox, D., Kautz, H., Learning and Inferring Transportation Routines. In *Proc*. of *AI 2004*. AAAI Press (2004), 348-353.
9. Parkka, J., Ermes, M., Korpipaa P., Mantyjarvi J., Peltola, J., Activity classification using realistic data from wearable sensors, IEEE Transactions on Information Technology in Biomedicine 10, 1 (2006), 119-128.
10. Patterson, D.J., Liao, L., Fox, D., Kautz, H., Inferring High-Level Behavior from Low-Level Sensors. In *Proc. of UBICOMP '03*, Springer Press (2003), 73-89
11. Timothy, S., Varshavsky, A., LaMarca A., Chen M. Y., Choudhury T., Mobility detection using everyday GSM traces. In *Proc. Ubicomp 2006*, Springer Press (2006).
12. Zheng, Y., Liu, L., Wang, L., Xie, X, Learning transportation mode from raw GPS data for geographic applications on the Web. In Proc. *WWW 2008*, ACM Press (2008), 247-256.