Chen M, Yu X, Liu Y. Mining object similarity for predicting next locations. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 31(4): 649–660 July 2016. DOI 10.1007/s11390-016-1654-2

Mining Object Similarity for Predicting Next Locations

Meng Chen [1], Xiaohui Yu [1,2], *Member, CCF, IEEE*, and Yang Liu [1,*], *Member, CCF, IEEE*

[1] *School of Computer Science and Technology, Shandong University, Jinan 250101, China*
[2] *School of Information Technology, York University, Toronto, M3J 1P3, Canada*

E-mail: chenmeng114@hotmail.com; {xyu, yliu}@sdu.edu.cn

Received February 26, 2016; revised May 27, 2016.

Abstract Next location prediction is of great importance for many location-based applications. With the virtue of solid theoretical foundations, Markov-based approaches have gained success along this direction. In this paper, we seek to enhance the prediction performance by understanding the similarity between objects. In particular, we propose a novel method, called weighted Markov model (weighted-MM), which exploits both the sequence of just-passed locations and the object similarity in mining the mobility patterns. To this end, we first train a Markov model for each object with its own trajectory records, and then quantify the similarities between different objects from two aspects: spatial locality similarity and trajectory similarity. Finally, we incorporate the object similarity into the Markov model by considering the similarity as the weight of the probability of reaching each possible next location, and return the top-rankings as results. We have conducted extensive experiments on a real dataset, and the results demonstrate significant improvements in prediction accuracy over existing solutions.

Keywords weighted Markov model, next location prediction, object similarity

1 Introduction

In recent years, location-based social networks (e.g., Foursquare, Gowalla) have been growing rapidly and users like to post their physical locations in the form of "check-in". A set of check-ins can be regarded as a trajectory because each check-in has a location tag and a time-stamp, corresponding to where and when the check-in is made respectively. Moreover, with the widespread use of positioning technology and the increasing deployment of surveillance infrastructures, it is increasingly possible to track the movement of people and other objects (e.g., vehicles). Both types of trajectory data contain three main attributes: the object, the location and the time-stamp. The availability of such trajectory data in large volume has a strong impact on a wide spectrum of applications such as trajectory search[1], trajectory reduction[2], urban computing[3] and location-based recommendation[4].

One of the fundamental problems in trajectory mining is to predict the next location of a moving object. Next location prediction is of great value to both users and the owners of trajectory datasets. For example, once knowing the next locations that users intend to visit, we may optimize marketing strategies accordingly by pushing promotions to those in the predicted area. In addition, such knowledge may also assist in forecasting traffic conditions and routing the drivers so as to alleviate traffic jams.

As Markov-based methods have performed well in the task of next location prediction[5-7], in this paper, we aim to further enhance the model by considering important factors that have been neglected in previous studies. Xue *et al.*[5] used a Markov model to mine individual mobility patterns for each object based on its own historical trajectories to predict next location.

Regular Paper

Special Section on Data Management and Data Mining 2016

This work was supported in part by the National Basic Research 973 Program of China under Grant No. 2015CB352502, the National Natural Science Foundation of China under Grant Nos. 61272092 and 61572289, the Natural Science Foundation of Shandong Province of China under Grant Nos. ZR2012FZ004 and ZR2015FM002, the Science and Technology Development Program of Shandong Province of China under Grant No. 2014GGE27178, and the NSERC (Natural Sciences and Engineering Research Council of Canada) Discovery Grants.

[*]Corresponding Author

However, the prediction based on individual patterns tends to suffer from data sparsity in many cases, e.g., social check-in and traffic surveillance, where an object only has a small number of past trajectories available, and meaningful moving patterns thus cannot be mined. Chen *et al.*[6-7] proposed variable-order Markov models to consider collective patterns of all available trajectories in making predictions. For example, assume that objects living in the same apartment produce a set of trajectory records as shown in Fig.1, where l_i ($i = 1, \ldots, 9$) indicate different locations, and each line represents a separate trajectory T_j ($j = 1, \ldots, 4$). In this example, we have object 1: $T_1 = (l_1, l_4, l_5, l_8, l_9)$, object 2: $T_2 = (l_1, l_2, l_5, l_6)$, object 3: $T_3 = (l_1, l_4, l_5, l_8)$, object 4: $T_4 = (l_1, l_2, l_5)$. Given a user arriving at l_5, the aforementioned method will take l_8 as the next move, as two out of three trajectories have sequentially passed the two locations in the history. That is, as long as the current location is identical, the method will predict the same next location for all the objects.

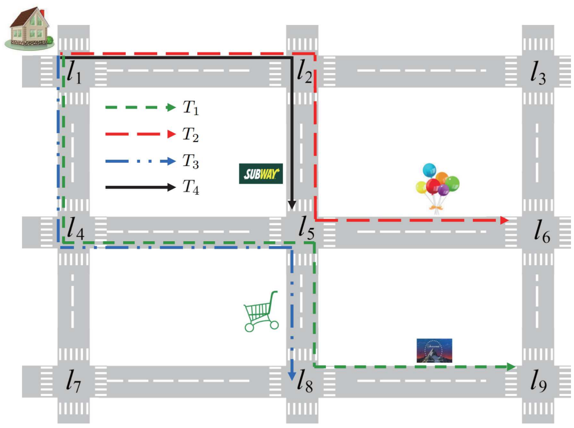


Fig.1. Example of next location prediction.

However, this can be problematic in many cases as it fails to consider the unique characteristics of each object. People's mobility patterns tend to be driven by their intrinsic habits, and different people may have their own travel preferences. As shown in Fig.1, the trajectories of object 2 and object 4 are more similar to each other, as they both share the same sequence (l_1, l_2, l_5). Therefore, to make a prediction for object 4, previous behaviors of object 2 can be more helpful. In consequence, given object 4 at l_5, location l_6 (instead of location l_8) will be selected, as object 2 had a visit at l_6 before. To make a better prediction, it is therefore necessary to give proper consideration to the similarities between objects and quantify them when mining the mobility patterns.

In this paper, we propose a weighted Markov model (weighted-MM) that considers both the sequence of just-passed locations and the object similarity in mining the mobility patterns. Weighted-MM consists of two components: 1) training an individual Markov model for each object with its own trajectories; 2) computing the similarity between any two objects. Further, we propose two methods to measure the object similarity. One is to use the spatial locality similarity based on the assumption that similar objects are more likely to have similar neighborhood, and the other is to calculate an overall trajectory similarity, where a weighted average of all possible similarities between trajectories of the two objects is taken. When making predictions, given the sequence of locations that object o has just passed by, we first find the similar objects of o and quantify the similarities, and then obtain the probability of reaching each possible next location with the individual model of each similar object. Finally, we take the similarity as the weight of the probability of reaching each possible next location, and return the top ranking results as outputs.

We present experimental results on a real dataset consisting of the vehicle passage records over a period of 31 days in a metropolitan city. The experimental results confirm the superiority of the proposed methods over existing methods.

The contributions of this paper can be summarized as follows.

- We present weighted-MM, a weighted Markov model that considers both the sequence of just-passed locations and the object similarity in predicting next locations. Specifically, we measure the similarity between any two objects and obtain the probability of reaching each possible next location with the individual Markov models for each object; then we regard the similarity as the weight of the probability and return the top-ranked ones as answers in the task of next location prediction.
- We propose two methods to measure the similarity between two objects from the perspective of spatial locality similarity and trajectory similarity respectively. One is to represent the spatial locality of an object with the frequency distribution over global locations, and measure the object similarity using the Kullback-Leibler divergence; the other is to estimate the similarity between two objects in terms of a weighted average of all possible similarities in trajectories that belong to them.

• We conduct extensive experiments with real-world traffic data to investigate the effectiveness of the proposed models, showing remarkable improvement as compared with baselines in predicting next locations.

The rest of the paper is organized as follows. We review the related work in Section 2, and give the preliminaries of our work in Section 3. In Section 4, we introduce the Markov model for next location prediction. In Section 5, we incorporate the object similarity into the Markov model to generate a weighted Markov model. We present the experimental results and the performance analysis in Section 6, and conclude this paper in Section 7.

2 Related Work

2.1 Next Location Prediction

Prediction with Individual Patterns. There exist an array of studies that use individual histories to predict the next locations, and we focus on methods with Markov-based solutions[5,8-9]. Xue *et al.*[5] used taxi traces to construct a Probabilistic Suffix Tree and predicted short-term routes with Variable-Order Markov models. Simmons *et al.*[8] built a hidden Markov model (HMM) for every driver, which predicts the future destination and route of each target. Liao *et al.*[9] introduced a hierarchical Markov model that infers a user's daily movements through an urban community. All these studies focus on predicting the destinations of specific individuals based on their own habits and historical trajectories. However, such methods will not work properly while a new user arrives or his/her trajectory history cannot be located.

Prediction with Collective Patterns. Mining the mobility patterns with historical movements of all the moving objects has been widely investigated[10-12]. Monreale *et al.*[10] built a T-pattern tree with all the trajectories to make future location predictions. Morzy[11] used all the moving objects' locations to discover frequent trajectories and movement rules with the PrefixSpan algorithm. Xue *et al.*[12] decomposed historical trajectories into sub-trajectories, connected the sub-trajectories into "synthesised" trajectories, and then predicted the destination with a Markov model. Besides, Chen *et al.*[6] trained an integrated Markov model with different trajectory sets to mine both individual and collective movement patterns to predict next locations. However, the above methods based on collective patterns make predictions at too coarse a granularity.

Prediction with External Information. Pushing further from the historical trajectories, there are studies[13-16] that focus on dynamic environments and improve prediction performance with external information (e.g., semantic features, driving speed and direction, expert knowledge). Zhou *et al.*[3] extracted a small set of reference trajectories for each target trajectory and trained a local model for prediction. Ye *et al.*[14] first discriminated the category of user activities with a hidden Markov model and then calculated the most likely locations given the estimated category distribution. Pan *et al.*[15] incorporated the historical traffic data with the real-time event, and proposed H-ARIMA+ to predict the traffic in the presence of incidents. Zhang *et al.*[16] extracted the underlying correlation between human mobility patterns and cellular call patterns and used it for the location prediction from temporal and spatial perspectives. However, as such external knowledge is not always available, the above methods can only be applied to some specific applications. Without loss of generality, in this paper, we aim to solve the next location prediction problem with trajectory data containing three main attributes, namely, object, location, and time-stamp.

In summary, none of the above studies investigate object similarity in mining mobility patterns. To the best of our knowledge, it is the first time to exploit both the sequence of just-passed locations and the object similarity in predicting next locations.

2.2 POI Recommendation

Some recent studies on POI recommendation in location-based social networks are also related to our work[17-18], in which any unvisited POIs can be recommended to users. Ye *et al.*[17] proposed an unified POI recommendation framework based on user-based CF to incorporate user preference, social influence, and geographical influence to compute the recommendation score of a candidate POI. Yuan *et al.*[18] attempted to enhance the user-based CF by considering the temporal information, and developed a collaborative time-aware recommendation model to recommend locations where users have not visited before.

The above methods are based on the user-based collaborative filtering (CF) method, which is widely adopted for recommender systems. Given a user, user-based CF first measures the similarities between the user and others, and then computes the recommendation score of a POI by taking a weighted combination of the other users' check-in records on the POI. We implemented the user-based CF method in [18] and applied it for next location prediction, but it performed much worse than Markov-based methods. One potential reason is that user-based CF method makes predictions without considering the just-passed locations. That is, the method will always predict the identical next location on the condition that the object is the same.

2.3 Trajectory Similarity

A trajectory consists of a time-ordered sequence of locations, and is a kind of time series data. Researchers have proposed a large volume of similarity measures to cope with different lengths of trajectories. For example, Vlachos *et al.*[19] first formalized non-metric similarity functions based on the longest common subsequence and then provided an intuitive notion of similarity between trajectories by giving more weight to the similar portions of the sequences. Dynamic time warping (DTW) is a much more robust distance measure for time series data. Keogh[20] proposed a novel technique for the exact indexing of DTW. Chen *et al.*[21] introduced a novel distance function, Edit Distance on Real sequence (EDR), which was robust against noise, shifts and scaling of trajectory data.

Furthermore, there are studies that try to obtain semantic information from the trajectories to help measure trajectory similarities. For example, Ying *et al.*[22] thought that trajectories which were geographically close might not be similar and thus introduced a novel trajectory similarity measurement named Maximal Semantic Trajectory Pattern Similarity (MSTP-Similarity), which measures the semantic similarity between trajectories. Velpula and Prasad[23] proposed a multi-viewpoint similarity measure which considers both the movement and the speed of the objects, along with the semantic features for clustering trajectories. However, the above methods are not suitable to compute trajectory similarity in this paper, because the semantic information is missing in our trajectory data.

3 Preliminaries

In this section, we first introduce some concepts which are required for the subsequent discussion, and then give an overview of the problem addressed in this paper.

Definition 1 (Location). *An object o passes through a set of locations, where each location l is defined as a point or a region where the position of o is recorded.*

Definition 2 (Trajectory). *The* trajectory T *is defined as a time-ordered sequence of locations:* $(l_1, l_2, \dots, l_n)$.

Definition 3 (Prefix Sequence). *For a location l_i and a given trajectory $T = (l_1, \dots, l_n)$, its prefix sequence L_i^j refers to a length-j subsequence of T ending with l_i.*

Definition 4 (Candidate Next Locations). *For location l_i, we define location l_j as a candidate next location of l_i if an object can arrive at l_j from l_i without going through another location first.*

The set of candidate next locations can be obtained either by prior knowledge (e.g., locations of the surveillance cameras combined with the road network graph), or by the induction from historical trajectories of moving objects.

Given a trajectory sequence $T = (l_1, l_2, \dots, l_n)$, the next location prediction problem is to predict the location that the moving object will arrive at next. That is, given T, to predict the next location l_{n+1}.

4 Markov Modeling

A Markov model is a stochastic model used to model randomly changing systems, and it performs well for prediction[6,24], thereby we also choose it to predict next locations. A naive approach is to train a separate Markov model for each object using its past trajectories. The Markov model regards the mobility patterns of an individual as a discrete stochastic process. Specifically, a state in the Markov model corresponds to a location, and a state transition corresponds to moving from one location to another.

Let T be an object's trajectory of length n (i.e., it contains n locations), and let $p\left(l_{n+1}|T\right)$ be the probability that the object will arrive at location l_{n+1} next. The location l_{n+1} is given by

$$\begin{aligned} l_{n+1} &= \underset{l \in \mathcal{L}}{\arg\max} \left\{p\left(l_{n+1} = l|T\right)\right\} \\ &= \underset{l \in \mathcal{L}}{\arg\max} \left\{p\left(l_{n+1} = l|l_1, l_2, \dots, l_n\right)\right\}, \end{aligned}$$

where $\mathcal{L}$ is the set of all the locations. Essentially, this approach for each location l computes its probability of next visit, and selects the one that has the highest possibility.

However, the probability of arriving at l actually only depends on a small set of m preceding locations, instead of all of them in the trajectory. It is clear that an object's current location will shed more light on its next move, and the most recent preceding steps will have more impacts on the object's future decisions than those that have passed much earlier. Therefore, the location l_{n+1} that the object will arrive at next can be given by

$$l_{n+1} = \arg\max_{l \in \mathcal{L}} \left\{ p\left(l_{n+1} = l | l_{n-(m-1)}, \ldots, l_n\right) \right\},$$

where m is the order of the Markov model.

In order to use the m-th order Markov model, we learn l_{n+1} for each prefix sequence $L_n^m = \left(l_{n-(m-1)}, \ldots, l_{n-1}, l_n\right)$ containing m locations, by estimating the conditional probability $p\left(l_{n+1} = l | L_n^m\right)$. The most commonly used method for estimating this value is to use the maximum likelihood principle, and the conditional probability $p\left(l_i | L_n^m\right)$ therefore can be computed by

$$p\left(l_i | L_n^m\right) = \frac{\sharp(L_n^m, l_i)}{\sharp(L_n^m)},$$

where $\sharp(L_n^m)$ is the number of times that prefix sequence L_n^m occurs in the training set, and $\sharp(L_n^m, l_i)$ is the number of times that location l_i occurs immediately after L_n^m.

However, the m-th order Markov model may suffer from a few problems. First, fixing m for all cases makes the model too rigid to account for the variability of different locations. Second, for a particular location l_i, trajectories containing l_i with a length less than m cannot be applied. To deal with problems, we therefore build a variable-order Markov model. Starting from the first, we sequentially increase the order of Markov models, until the m-th order model has been developed. In the end, we have m Markov models for each object in total. To make a prediction, for a given sequence of locations that object o has just passed by, we adopt the principle of the longest match. In particular, we go through the previously constructed m models in decreasing order. If the current trajectory does not contain the corresponding states, we opt to continue with another model with a smaller order.

5 Considering the Object Similarity

Trajectory data tend to be sparse, as most objects are likely to visit only a small amount of locations. As a result, when an object goes to some place that has never been visited before, its own Markov models will not work. To solve this problem, we consider locating objects that share similar patterns to the target, and use their mobility patterns to assist the prediction, as there is a reason to believe that objects demonstrating similar behaviors in the past are likely to take the same route in the future. In addition, we propose a weighted Markov model (weighted-MM) which quantifies the subtle difference in similarities and integrates them into the Markov framework.

Training weighted-MM consists of two parts: 1) constructing a variable-order Markov model for each object with its own trajectories; 2) evaluating the similarity between any two objects. The first part can be achieved by adopting the method introduced in Section 4. We focus on measuring the similarity between two objects in this section.

5.1 Spatial Locality Similarity

Intuitively, objects localized to the same area are likely to have a similar spatial visiting pattern. For example, people living in the same apartment tend to visit the same gas station or grocery store nearby. The spatial locality of an object thus can be expressed as the frequency distribution over different visited locations, and formalized with global location frequency.

Definition 5 (Global Location Frequency). *For an object o, the global location frequency of location $l_i \in \mathcal{L}$ is defined as*

$$f\left(o, l_i\right) = \frac{\sharp\left(l_i\right) + \alpha}{\sum_{l \in \mathcal{L}}(\sharp\left(l\right) + \alpha)}.$$

In the above definition, $\mathcal{L}$ is the set of sampling locations and $\sharp\left(\cdot\right)$ is the visited number of the corresponding locations. If object o has not visited location l_i, we set $\sharp\left(l_i\right)$ to 0. To avoid zero values, we choose additive smoothing[25] to smooth global location frequency, where α is the smoothing parameter and we set α to 1 here.

We then build an object spatial locality matrix $\boldsymbol{F}$ with the above definition. Each row of $\boldsymbol{F}$ represents an object; each column stands for a location; each item in $\boldsymbol{F}$ corresponds to a visit frequency of the object. A row vector reflects the frequency that one object arrives at every location. Formally, we have

$$\boldsymbol{F} = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1l^*} \\ f_{21} & f_{22} & \cdots & f_{2l^*} \\ \vdots & \vdots & \ldots & \vdots \\ f_{o^*1} & f_{o^*2} & \cdots & f_{o^*l^*} \end{pmatrix},$$

$$f_{ij} = f(i,j),$$
$$\sum_{j=1}^{l^*} f_{ij} = 1, (i = 1, 2, ..., o^*),$$

where o^* is the number of objects and l^* is the number of sampling locations. f_{ij} stands for the global location frequency of object i arriving at location j.

Given an object o_u, we use an l^*-dimensional vector $\boldsymbol{F}_u = (f_{u1}, f_{u2}, ..., f_{ul^*})$ to represent its spatial locality, where l^* is the total number of sampling locations. We measure the spatial locality similarity of two objects o_u and o_v using the Kullback-Leibler divergence[26]:

$$Osim_{uv} = 1 - \frac{1}{2}\sum_{w=1}^{l^*}\left(f_{uw}\log\frac{f_{uw}}{f_{vw}} + f_{vw}\log\frac{f_{vw}}{f_{uw}}\right).$$

Here, note that other distance metrics such as the cosine similarity can also be used to measure the similarity between objects.

5.2 Trajectory Similarity

To further quantify the subtle difference between objects demonstrating similar patterns, we evaluate the object similarity with their corresponding trajectories. To be more specific, we estimate the difference between objects o_u and o_v in terms of a weighted average of all possible similarities in trajectories that belong to them respectively. That is,

$$Osim_{uv} = \frac{1}{|\mathcal{T}_{o_u}| \times |\mathcal{T}_{o_v}|}\sum_{T_p \in \mathcal{T}_{o_u}}\sum_{T_q \in \mathcal{T}_{o_v}} Tsim(T_p, T_q),$$

where $\mathcal{T}_{o_u}$ is the trajectory set of object o_u, and $Tsim(T_p, T_q)$ is the similarity between two trajectories T_p and T_q.

An object's visit at each sampling location has a unique time-stamp; therefore the corresponding trajectories naturally formulate a collection of time series. Their distance can be calculated with a large volume of existing measures, e.g., Jaccard similarity (JS), longest common subsequence (LCS) and dynamic time warping (DTW)[21]. As each distance measure has its own superiorities, we explore the use of the three metrics, and compare their performance in our empirical study.

5.2.1 *Jaccard Similarity*

The Jaccard similarity is a common index for binary variables. It is defined as the quotient between the intersection and the union of the pairwise compared variables among two objects. In our study, each trajectory T can be represented with a set of locations, and we have:

$$Tsim(T_p, T_q) = \frac{|set(T_p) \cap set(T_q)|}{|set(T_p) \cup set(T_q)|},$$

where $set(T_p)$ is the location set of trajectory T_p.

5.2.2 *Longest Common Subsequence*

Trajectories can be considered as similar if they have common parts, and we thus adopt the longest common subsequence (LCS) to indicate their resemblance, and the similarity of two trajectories T_p and T_q can be defined as

$$Tsim(T_p, T_q) = \frac{LCS(T_p, T_q)}{\max(|T_p|, |T_q|)},$$

where $|T|$ is the length of trajectory T. For instance, given a trajectory $T_p = (l_1, l_2, l_3, l_5)$ and a trajectory $T_q = (l_1, l_3, l_5)$, their longest common part is $LCS(T_p, T_q) = (l_1, l_3, l_5)$.

5.2.3 *Dynamic Time Warping*

In time series analysis, dynamic time warping (DTW) measures an optimal match similarity between two temporal sequences which may vary in time or speed. For example, similarities in walking patterns could be detected using DTW, even if one person is walking faster than the other, or if there are accelerations and decelerations during the course of an observation. In our study, for trajectories T_p and T_q:

$$DTW(T_p, T_q) = \begin{cases} 0, & \text{if } |T_p| = |T_q| = 0, \\ \infty, & \text{if } |T_p| = 0 \text{ or } |T_q| = 0, \\ dist(l_1^p, l_1^q) + \min\{ & \\ DTW(T_p, Rest(T_q)), & \\ DTW(Rest(T_p), T_q), & \\ DTW(Rest(T_p), Rest(T_q))\}, & \text{otherwise}, \end{cases}$$

where $|T|$ is the length of trajectory T and $Rest(T)$ is the subsequence of T without the first location. $dist(l_1^p, l_1^q)$ is the distance between a pair of locations l_1^p and l_1^q, in which l_1^p and l_1^q represent the first location in the trajectories T_p and T_q respectively. In our experiments, we set $dist(l_1^p, l_1^q)$ to 1 if l_1^p is not the same as l_1^q and 0 otherwise. The similarity of T_p and T_q is defined as

$$Tsim(T_p, T_q) = 1 - \frac{DTW(T_p, T_q)}{\max(|T_p|, |T_q|)}.$$

5.3 Next Location Prediction

In order to consider both the sequence of just-passed locations and the object similarity in mining the mobility patterns, we propose to incorporate the similarity into the Markov model. Specifically, we quantify the similarity between any two objects and obtain the probability of reaching each possible next location with the individual Markov models for each object; then we regard the similarity as the weight of the probability and return the top-ranked ones as answers.

Given a trajectory sequence of object o_k, the location l_{n+1} that it will arrive at next is given by

$$\begin{aligned} &l_{n+1} \\ &= \arg\max_{l_i \in \mathcal{L}} \left\{ \sum_{o_j \in \mathcal{O}} Osim_{kj} \times p\left(l_{n+1} = l_i | L_n^m, o_j\right) \right\}, \\ &p\left(l_{n+1} = l_i | L_n^m, o_j\right) \\ &= \arg\max_{l_i \in \mathcal{L}} \left\{ \frac{\sharp(L_n^m, l_i, \mathcal{T}_{o_j})}{\sharp(L_n^m, \mathcal{T}_{o_j})} \right\}, \end{aligned}$$

where $\mathcal{T}_{o_j}$ is the trajectory set of object o_j and $Osim_{kj}$ represents the similarity between objects o_k and o_j. Here we directly use the object similarity $Osim_{kj}$ as the weight of the Markov model. We also can process the similarity with some functions (e.g., exponential function), and use $exp^{Osim_{kj}}$ as the weight according to the features of data.

Note that the similarity between any two objects and the Markov model of any object can be computed offline in advance. When predicting next locations, we do not take all of the objects into consideration, but only consider the similar objects. The reasons are mainly two-fold: 1) including more objects would bring higher time and space cost in training the weighted Markov model; 2) introducing irrelevant objects may deteriorate the overall prediction accuracy. We will evaluate the effect of the number of similar objects in the experiments.

6 Performance Evaluation

We present experiments using a real vehicle passage dataset to evaluate the proposed models. In this section, we first introduce the data and settings in our empirical study, and then present the results of our proposal. In addition, we compare our models with other state-of-the-art strategies.

6.1 Datasets and Settings

In our study, we exploit a real vehicle passage dataset which is collected over the traffic surveillance system in a major metropolitan city. We accumulate 10 344 058 records from the data center during a period of 31 days. Each record contains a vehicle ID, a location of the surveillance camera, and a time-stamp of passage. We pre-process them to formulate trajectories, and only keep those that contain at least three locations to make the model more robust. The cumulative distribution functions (cdf) of the number of trajectories and the number of candidate next locations are shown in Fig.2. It is clear that about 80% of objects have less than 35 trajectories, which explicitly demonstrates the data sparsity issue mentioned above. Moreover, about 86.3% of the locations have more than 10 candidate next locations, and the average number of candidate next locations is about 43.

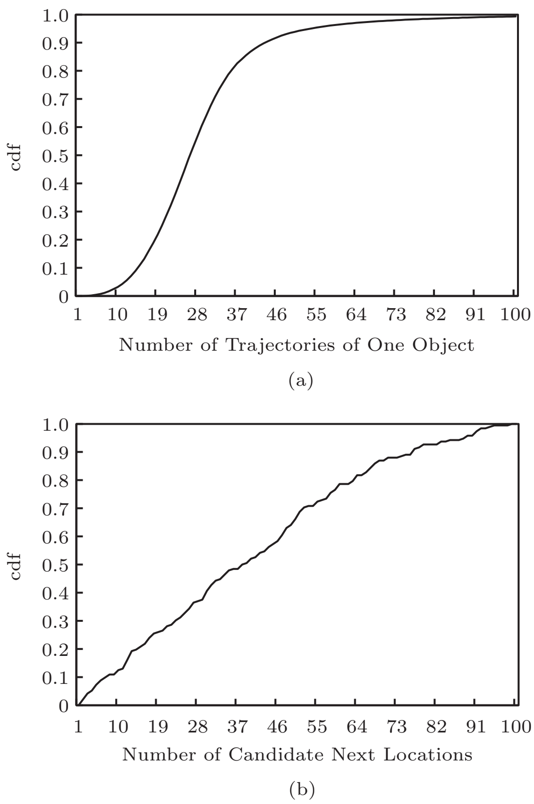


Fig.2. Characteristics of the trajectory data.

In the task of predicting next locations, we first select trajectories in the first 20 days to construct the models, and adopt the data in the next seven days to fine tune the parameters. Then 746 790 trajectories of

30 134 vehicles in the first 27 days are used to train the final models with the tuned parameters. Finally, we use the remaining 4-day 104 129 trajectories to formulate the test dataset.

To compare different Markov-based methods, we use two evaluation metrics, namely, accuracy and average precision. Accuracy is defined as the ratio of the number of trajectories for which the model is able to correctly predict to the total number of trajectories in the test set. That is,

$$accuracy = \frac{1}{|\mathcal{T}'|} \sum p\,(l)\,,$$

where $|\mathcal{T}'|$ is the number of trajectories in the testing set, and $p\,(l)$ is 1 if l is the true next location and 0 otherwise.

Average precision is defined as

$$ap = \frac{1}{|\mathcal{T}'|} \sum \frac{p\,(l_i)}{i},$$

where i denotes the position in the predicted list, and $p\,(l_i)$ takes the value of 1 if the predicted location at the i-th position in the list is the actual next location and 0 otherwise.

6.2 Parameter Settings and Tuning

In this subsection, we evaluate the performance of our proposed models, namely, weighted-MM(SL) (which stands for the weighted Markov model with spatial locality), weighted-MM(JS) (which stands for the weighted Markov model with Jaccard similarity), weighted-MM(LCS) (which stands for the weighted Markov model with the longest common subsequence) and weighted-MM(DTW) (which stands for the weighted Markov model with dynamic time warping). In the proposed models, there exist three parameters: the order of Markov model, the threshold of similarity (weighted-MM(SL)) and the number of similar objects (weighted-MM(JS), weighted-MM(LCS) and weighted-MM(DTW)). We evaluate the effect of the parameters in the following experiments, and report the average results of 50 runs for each experiment.

We study the effect of the order of Markov model by varying it from 1 to 5, and apply the order to weighted-MM(SL) (the threshold of similarity is 0.8), weighted-MM(JS), weighted-MM(LCS), and weighted-MM(DTW) (the number of similar objects is 300). Here we summarize the proposed models. We first propose weighted-MM(SL), which uses the frequency distribution over global locations to express the spatial locality of one object and measures the object similarity with them. Further, we propose three methods (weighted-MM(LCS), weighted-MM(DTW) and weighted-MM(JS)) which measure the object similarity in terms of a weighted average of all possible similarities in trajectories that belong to them separately. In weighted-MM(LCS), we adopt the longest common subsequence (LCS) to indicate the resemblance of trajectories. The completely same sequences of two trajectories are used to measure the similarity, and the constraint is pretty rigorous. Weighted-MM(DTW) has loose constraint, and the partly same sequences can be used to compute the trajectory similarity. In weighted-MM(JS), we represent a trajectory with a set of locations, and compute the similarity with Jaccard distance.

We demonstrate the performances using top-1 accuracy, top-5 accuracy and top-5 average precision respectively in Fig.3. On one hand, clearly, the accuracy and the average precision of all the models have a significant improvement when the Markov order increases from 1 to 2, but start to decrease slightly and keep stable as we further increase the order. On the other hand, weighted-MM(LCS) performs the worst, as it measures the similarity with the completely same sequences of two trajectories, and the constraint is too rigorous to find the real similar ones for an object; weighted-MM(DTW) performs better than weighted-MM(LCS), as it relaxes the constraint and computes the similarity with the partly same sequences; weighted-MM(JS) represents a trajectory with a set of locations, and computes the similarity with Jaccard distance, and it performs better than weighted-MM(DTW); weighted-MM(SL) performs the best, as it measures the object similarity with the spatial locality, which could reflect objects' travel preference. In addition, the size of the model grows with respect to the Markov order, and a higher-order model may suffer from a high training cost in terms of time and space. Thus we set the order of Markov model to 2 in the following experiments.

For weighted-MM(SL), we take the weight of similarity between two objects into consideration. Here we manually set various similarity thresholds to filter similar objects in developing the weighted-MM(SL), and then examine the result in Fig.4. Initially an increase in the threshold is accompanied by an increase in the accuracy and average precision, and they reach the peak values when the threshold approaches 0.8. However, as we further increase the value, both accuracy and average precision decrease dramatically. This is due to the fact that most objects will be removed from the candidate list given a high threshold.

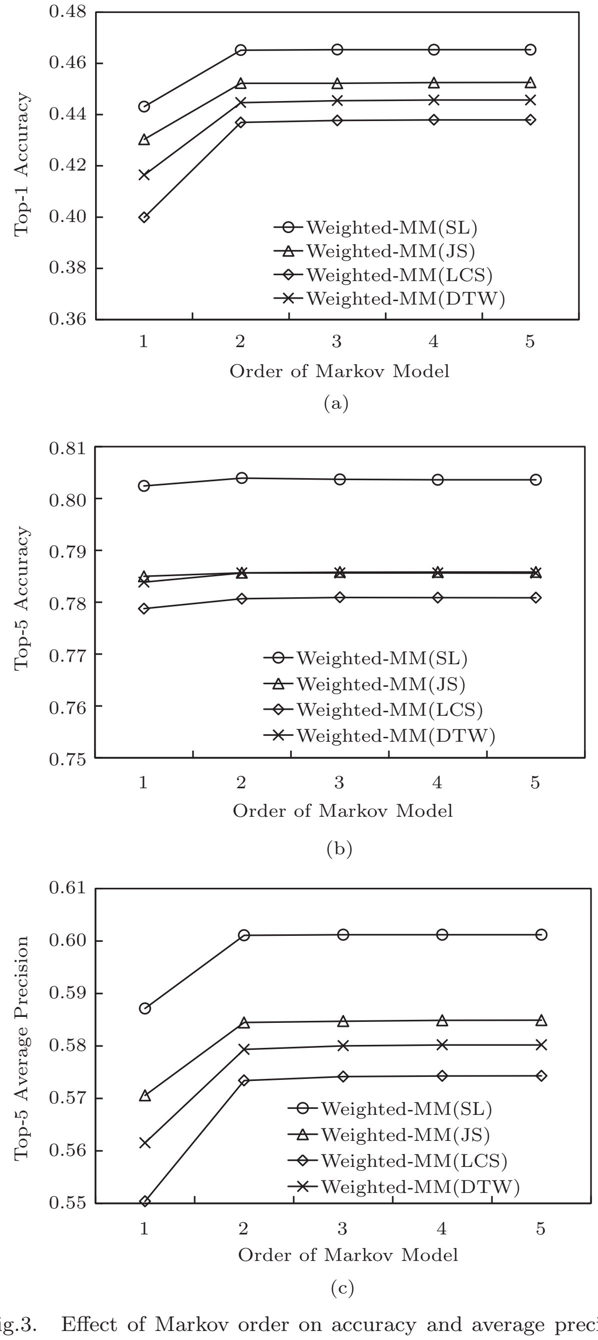


Fig.3. Effect of Markov order on accuracy and average precision. (a) Top-1 accuracy. (b) Top-5 accuracy. (c) Top-5 average precision.

For weighted-MM(JS), weighted-MM(LCS) and weighted-MM(DTW), we first rank the similarity values, and then pick up similar objects with a decreasing order. We vary the number of similar objects from 100 to 5 000 to evaluate its effect, and report the results in Fig.5. Note that the performance improves for all three models as the number of similar objects increases. Nonetheless, both average precision and accuracy gradually decrease after a certain value, which might be caused by introducing more noise with a larger amount of similar objects. Besides, we notice that weighted-MM(JS) achieves the best performance among three methods. Therefore, we adopt weighted-MM(JS) for the comparison with other alternative solutions.

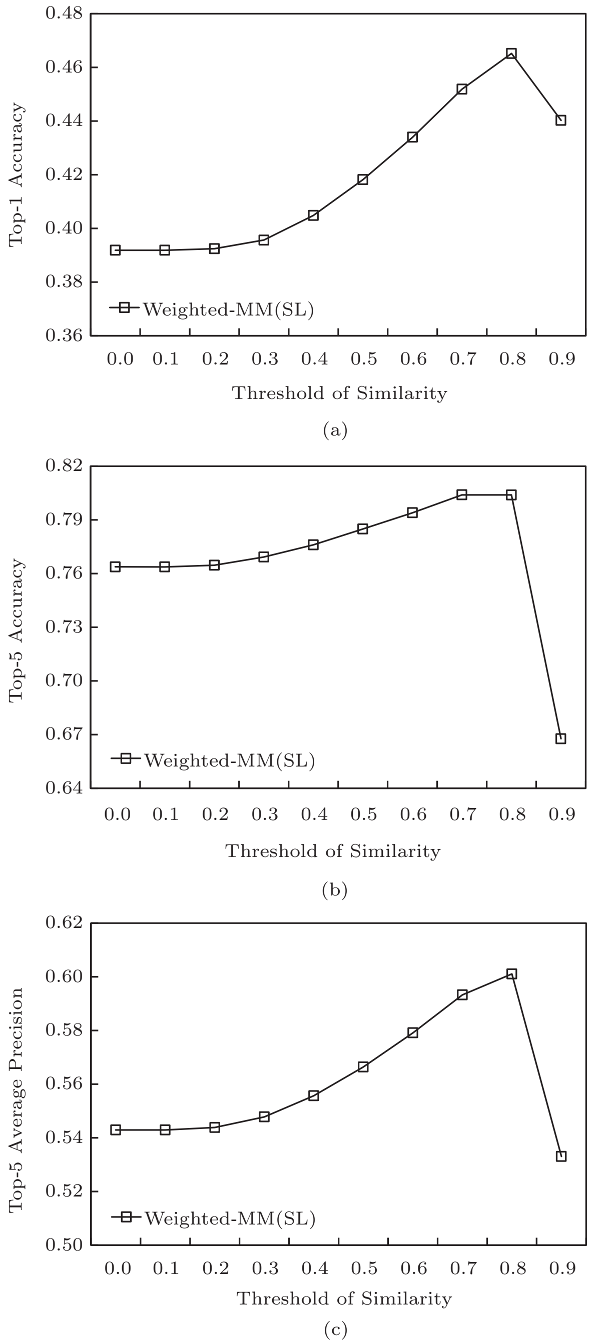


Fig.4. Effect of the threshold of similarity. (a) Top-1 accuracy. (b) Top-5 accuracy. (c) Top-5 average precision.

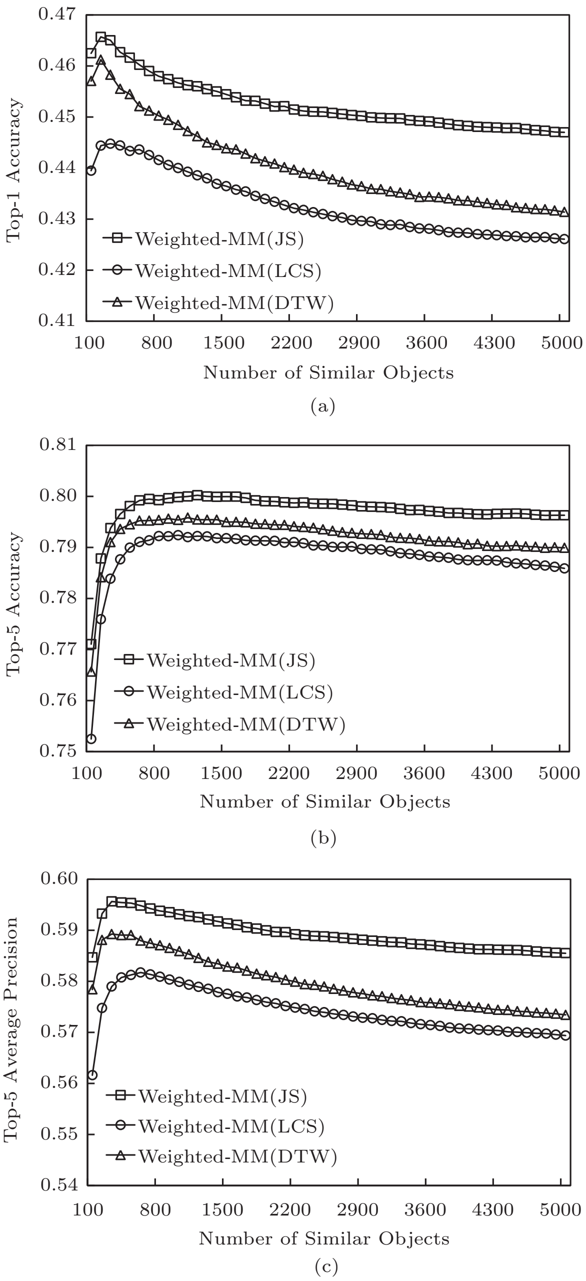


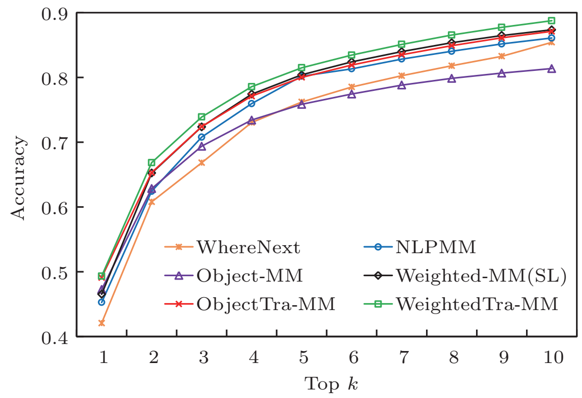
Fig.5. Effect of the number of similar objects. (a) Top-1 accuracy. (b) Top-5 accuracy. (c) Top-5 average precision.

6.3 Comparisons with Baselines

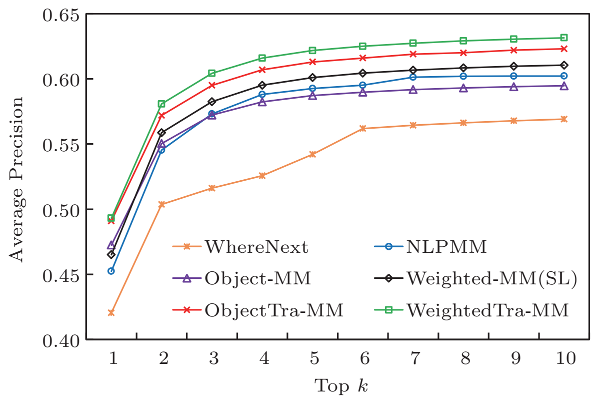
To evaluate the effectiveness of the proposed models, we compare against some start-of-the-art approaches including WhereNext[10], NLPMM[7] and objectTra-MM[27]. WhereNext uses the previous movements of all moving objects to make future location predictions. NLPMM uses logistic regression to combine the individual and the collective movement patterns in making predictions. ObjectTra-MM consists of two models: object-MM and tra-MM. Object-MM first clusters similar objects based on their spatial localities, and then builds variable-order Markov models with the trajectories of objects in the same cluster; tra-MM clusters trajectories using a given similarity metric, and trains a series of Markov models with trajectories in each cluster. As they concentrate on different aspects of the movement patterns (object-MM considers the object similarity and tra-MM considers the trajectory similarity), they are integrated through logistic regression to obtain objectTra-MM. Similarly, we also integrate weighted-MM(SL) and tra-MM to generate the final predictor weightedTra-MM to obtain a better prediction accuracy.

We first empirically obtain the optimal parameters for these approaches in our dataset, and then apply them in the comparison. For WhereNext, the support for constructing T-pattern tree is 20. For NLPMM, the Markov order is 2. For objectTra-MM, the cluster number in object-MM is 80, and the distance threshold in tra-MM is 8. The threshold of similarity in weighted-MM(SL) is 0.8.

We predict the top k next locations with all the models, and demonstrate the performances in Fig.6. We can see that: 1) Both accuracies and average precisions improve as k increases for all the models. 2) WhereNext performs the worst, because it uses the trajectories of all the objects to mine the collective patterns and is coarse grained. NLPMM considers both the individual patterns and the collective patterns in making predictions, and it performs better than WhereNext. 3) When predicting the top-1 next location, object-MM performs better than WhereNext and NLPMM, as it clusters similar objects and learns fine grained patterns. The weakness of object-MM is that it cannot discover collective patterns and performs dissatisfactorily in predicting top-10 next locations, and this might be due to the fact that each cluster only has a part of objects and all the objects in the same cluster are treated equally. 4) Though weighted-MM(SL) and object-MM organize the same information, weighted-MM(SL) performs better than object-MM, since it quantifies the spatial locality similarity between any two objects and sets larger weights to the more similar objects in mining the mobility patterns. 5) ObjectTra-MM considers both the object similarity and the trajectory similarity in predicting next locations, and it performs better than weighted-MM(SL). Further, as weighted-MM(SL) performs better than object-MM, the integrated predictor weightedTra-MM also outperforms objectTra-MM. It is worth mentioning that the top-10 accuracy of 0.89 is a significant improvement as the average number of candidate next locations is about 43 (meaning there are 43 possibilities).



(a)



(b)

Fig.6. Performance comparison with baselines. (a) Accuracy. (b) Average precision.

7 Conclusions

In this paper, we proposed a weighted Markov model (weighted-MM) which enhances the Markov model for next location prediction by quantifying the similarities between objects. Weighted-MM consists of two parts: 1) training an individual Markov model for each object with its own trajectories; 2) quantifying the similarity between two objects. We proposed two methods to measure the object similarity from the perspective of spatial locality similarity and trajectory similarity respectively. We evaluated the proposed models using a real vehicle passage record dataset, and the experiments showed that the proposed models significantly outperform the state-of-the-art methods.

References

[1] Chen W, Zhao L, Xu J J, Liu G F, Zheng K, Zhou X. Trip oriented search on activity trajectory. *Journal of Computer Science and Technology*, 2015, 30(4): 745-761.

[2] Park H, Lee Y J, Chae J, Choi W. Online approach for spatio-temporal trajectory data reduction for portable devices. *Journal of Computer Science and Technology*, 2013, 28(4): 597-604.

[3] Hsieh H P, Lin S D, Zheng Y. Inferring air quality for station location recommendation based on urban big data. In *Proc. the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2015, pp.437-446.

[4] Gao H J, Tang J L, Hu X, Liu H. Content-aware point of interest recommendation on location-based social networks. In *Proc. the 29th AAAI Conference on Artificial Intelligence*, January 2015, pp.1721-1727.

[5] Xue G, Li Z, Zhu H, Liu Y. Traffic-known urban vehicular route prediction based on partial mobility patterns. In *Proc. International Conference on Parallel and Distributed Systems*, December 2009, pp.369-375.

[6] Chen M, Liu Y, Yu X. NLPMM: A next location predictor with Markov modeling. In *Proc. the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, May 2014, pp.186-197.

[7] Chen M, Yu X, Liu Y. Mining moving patterns for predicting next location. *Information Systems*, 2015, 54: 156-168.

[8] Simmons R, Browning B, Zhang Y, Sadekar V. Learning to predict driver route and destination intent. In *Proc. International Conference on Intelligent Transportation Systems*, September 2006, pp.127-132.

[9] Liao L, Patterson D J, Fox D, Kautz H. Learning and inferring transportation routines. *Artificial Intelligence*, 2007, 171(5/6): 311-331.

[10] Monreale A, Pinelli F, Trasarti R, Giannotti F. WhereNext: A location predictor on trajectory pattern mining. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, June 28-July 1, 2009, pp.637-646.

[11] Morzy M. Mining frequent trajectories of moving objects for location prediction. In *Proc. International Conference on Machine Learning and Data Mining*, July 2007, pp.667-680.

[12] Xue A Y, Zhang R, Zheng Y, Xie X, Huang J, Xu Z. Destination prediction by subtrajectory synthesis and privacy protection against such prediction. In *Proc. IEEE International Conference on Data Engineering*, April 2013, pp.254-265.

[13] Zhou J, Tung A K, Wu W, Ng W S. A “semi-lazy” approach to probabilistic path prediction in dynamic environments. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2013, pp.748-756.

[14] Ye J, Zhu Z, Cheng H. What’s your next move: User activity prediction in location-based social networks. In *Proc. SIAM International Conference on Data Mining*, May 2013, pp.171-179.

[15] Pan B, Demiryurek U, Shahabi C. Utilizing real-world transportation data for accurate traffic prediction. In *Proc. the 12th IEEE International Conference on Data Mining*, December 2012, pp.595-604.

[16] Zhang D, Zhao S, Yang L T, Chen M, Wang Y, Liu H. NextMe: Localization using cellular traces in Internet of things. *IEEE Transactions on Industrial Informatics*, 2015, 11(2): 302-312.

[17] Ye M, Yin P, Lee W C, Lee D L. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proc. International Conference on Research on Development in Information Retrieval*, July 2011, pp.325-334.

[18] Yuan Q, Cong G, Ma Z, Sun A, Magnenat-Thalmann N. Time-aware point-of-interest recommendation. In *Proc. the 36th International Conference on Research on Development in Information Retrieval*, July 2013, pp.363-372.

[19] Vlachos M, Kollios G, Gunopulos D. Discovering similar multidimensional trajectories. In *Proc. the 18th IEEE International Conference on Data Engineering*, February 26-March 1, 2002, pp.673-684.

[20] Keogh E. Exact indexing of dynamic time warping. In *Proc. the 28th International Conference on Very Large Data Bases*, August 2002, pp.406-417.

[21] Chen L, Özsu M T, Oria V. Robust and fast similarity search for moving object trajectories. In *Proc. International Conference on Management of Data*, June 2005, pp.491-502.

[22] Ying J C, Lu H C, Lee W C, Weng T C, Tseng V S. Mining user similarity from semantic trajectories. In *Proc. ACM SIGSPATIAL International Workshop on Location Based Social Networks*, November 2010, pp.19-26.

[23] Velpula V B, Prasad M K. TCMVS: A novel trajectory clustering technique based on multi-view similarity. *Cybernetics Information Technologies*, 2015, 15(2): 53-62.

[24] Wu Q, Ng M K, Ye Y. Cotransfer learning using coupled Markov chains with restart. *IEEE Intelligent Systems*, 2013, 29(4): 26-33.

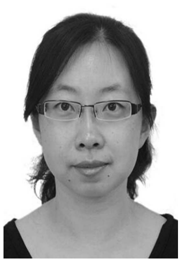
[25] Miller F P, Vandome A F, McBrewster J (eds.). Additive Smoothing. Alphascript Publishing, 2010.

[26] Ertoz L, Steinbach M, Kumar V. A new shared nearest neighbor clustering algorithm and its applications. In *Proc. the Workshop on Clustering High Dimensional Data and Its Applications at the 2nd SIAM International Conference on Data Mining*, April 2002, pp.105-115.

[27] Chen M, Liu Y, Yu X. Predicting next locations with object clustering and trajectory clustering. In *Proc. the 19th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, May 2015, pp.344-356.

Meng Chen received his B.S. degree in software engineering in 2011 from Shandong University, Jinan. He is currently a Ph.D. candidate in the School of Computer Science and Technology, Shandong University, Jinan. His research interest is in the area of data mining.

Xiaohui Yu received his Ph.D. degree in computer science in 2006 from the University of Toronto, Canada. His research interests are in the areas of database systems and data mining. He is a professor in the School of Computer Science and Technology, Shandong University, Jinan, and an associate professor in the School of Information Technology, York University, Canada (on leave).



Yang Liu received her Ph.D. degree in computer science and engineering in 2008 from York University, Canada. She is currently an associate professor in the School of Computer Science and Technology, Shandong University, Jinan. Her main areas of research are data mining and information retrieval.