Distrib Parallel Databases (2013) 31:231–258
DOI 10.1007/s10619-012-7115-1

QS-STT: QuadSection clustering and spatial-temporal trajectory model for location prediction

Po-Ruey Lei · Shou-Chung Li · Wen-Chih Peng

Published online: 20 October 2012

Abstract Location prediction is a crucial need for location-aware services and applications. Given an object's recent movement and a future time, the goal of location prediction is to predict the location of the object at the future time specified. Different from traditional location prediction using motion function, some research works have elaborated on mining movement behavior from historical trajectories for location prediction. Without loss of generality, given a set of trajectories of an object, prior works on mining movement behaviors will first extract regions of popularity, in which the object frequently appears, and then discover the sequential relationships among regions. However, the quality of the frequent regions extracted affects the accuracy of the location prediction. Furthermore, trajectory data has both spatial and temporal information. To further enhance the accuracy of location prediction, one could utilize not only spatial information but also temporal information to predict the locations of objects. In this paper, we propose a framework QS-STT (standing for QuadSection clustering and Spatial-Temporal Trajectory model) to capture the movement behaviors of objects for location prediction. Specifically, we have developed QuadSection clustering to extract a reasonable and near-optimal set of frequent regions. Then, based on the set of frequent regions, we propose a spatial-temporal trajectory model to explore the object's movement behavior as a probabilistic suffix tree with both spatial and temporal information of movements. Note that STT is

Communicated by Mohamed Mokbel.

P.-R. Lei
Department of Electrical Engineering, Chinese Naval Academy, Kaohsiung, Taiwan
e-mail: kdboy1225@gmail.com

S.-C. Li · W.-C. Peng (✉)
National Chiao Tung University, Hsinchu, Taiwan
e-mail: wcpeng@cs.nctu.edu.tw

S.-C. Li
e-mail: dreampilot.cs96@g2.nctu.edu.tw

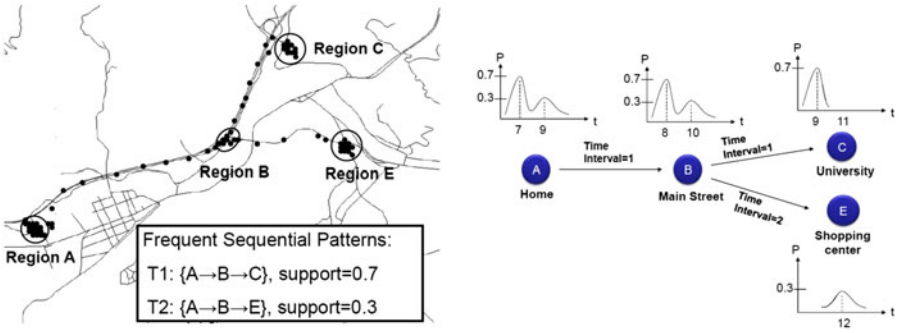
not only able to discover sequential relationships among regions but also derives the corresponding probabilities of time, indicating when the object appears in each region. Based on STT, we further propose an algorithm to traverse STT for location prediction. By enhancing the quality of the frequent region extracted and exploring both the spatial and temporal information of STT, the accuracy of location prediction in QS-STT is improved. QS-STT is designed for individual location prediction. For verifying the effectiveness of QS-STT for location prediction under the different spatial density, we have conducted experiments on four types of real trajectory datasets with different speed. The experimental results show that our proposed QS-STT is able to capture both spatial and temporal patterns of movement behaviors and by exploring QS-STT, our proposed prediction algorithm outperforms existing works.

Keywords Trajectory pattern · Movement behavior mining · Location prediction · Frequent region · Spatial-temporal data

1 Introduction

With the growth of location aware technologies such as mobile devices and GPS applications, it has become possible to track and collect an increasing amount of trajectory data from moving objects. A trajectory is generally considered as the path left behind by a moving object in space and time, i.e, a sequence of location points where each point is a spatial-temporal data corresponding a position in space at a certain timestamp. As a huge amount of trajectories accumulated, there may be some trajectories frequently reappear in the data. Such trajectories are called trajectory patterns. Due to the trajectory data records an object's real movements, the trajectory patterns are supposed to reflect the object's movement behaviors. The trajectory patterns play a fundamental role in helping to analyze the popularity of regions and traveling routes [3, 13, 16, 17, 19], the movement flows of users and location-based social networks [2, 6, 18, 23, 29–31]. In this paper, we target on utilizing trajectory patterns to predict a moving object's location. The problem of location prediction can be generally formulated as: given an object's recent movements and one query time in the future, the location of this object at the future time can be estimated. The location prediction is useful in many applications, such as prefetching for the location-based services, inferring the crowd of a region for tourism recommendations, and estimating the traffic status for transportation management.

The traditional method for location prediction is based on motion function. Given an object's recent movements, one could predict future location by recent movement speed and direction. However, the motion function may provide assistance to predict location in near future time and give an predicted location with large error in far future time. For example, if we know a user was at classroom at 10:00 and he is passing a post office at 10:10 currently, motion function could provide a reasonable predicted location at 10:20. For the location at query time 12:00, the motion function may give an incorrect location. To enhance the prediction accuracy, several works proposed that utilizing trajectory patterns to predict a moving object's future



(a) The original trajectories and trajectory patterns without temporal information

(b) Trajectory patterns with time distribution of frequent regions

Fig. 1 An example of trajectory patterns

location [10, 18, 20, 31]. An object's movements are supposed to follow some trajectory patterns and then those patterns can be utilized as a movement model to predict the future location and improve the prediction accuracy. For example, users usually have their habitual routes from their home to workplace and animals have the annual migration. Without loss of generality, prior works on mining movement behaviors will first extract regions of popularity, in which the object frequently appears, and then discover the movement relationships among regions. It is shown in [10] that by exploring trajectory patterns in form of association rules, the accuracy of location prediction with a far future time is improved compared to the traditional method (i.e., motion function approaches). However, most of the existing works only focus on trajectory pattern discovery in spatial domain. Prediction with spatial based trajectory pattern may result in biased prediction. Consider an example in Fig. 1(a), where a user has two trajectory patterns (i.e., T1 and T2 without considering temporal information). In T1, the user usually has the routine path from his home (i.e., region A) to his study place (i.e., region C) along the main street (i.e., region B) in weekdays. In T2, the user often goes to shopping center (i,e., region E) along the main street from his home in weekends. The sequential patterns compute their support to indicate the number of times they appear in a trajectory dataset. In this example, T1 (i.e., $A \rightarrow B \rightarrow C$) has higher support than T2. Assume that the user's recent movements are regions A and B, and this user's current location is region B around 10:00 am. By these two sequential patterns, the next location of this user is always predicted as region C since T1 has higher support in spatial domain. However, if the trajectory patterns are discovered by considering both spatial and temporal information. As an example shown in Fig. 1(b), one could imply that if a user moves to C, the time will be 11:00 (i.e., the sum of the current time and the time interval between B and C). The appearing probability at region C at 11:00 via the routing path $A \rightarrow B$ is zero. On the other hand, one could infer that this user may appear in region E at 12:00 with the probability larger than zero. we can predict the user has the higher opportunity of moving to region E. Therefore, to further enhance the accuracy of location prediction, we address that one could utilize not only spatial information but also temporal information to estimate the object's future location. Based on above obser-

vations, there are two major issues for our pattern-based location prediction problem as follows:

(1) We suppose that an object's movement behavior follows some patterns and these patterns can be used as predictor to estimate the future location. The representative and effectiveness of the trajectory patterns are critical for prediction accuracy. For accuracy enhancement, the trajectory patterns should be designed and explored by considered in both spatial and temporal domain.
(2) Given the object's recent movements and one query time in the future, how to utilize these discovered spatial-temporal trajectory patterns to predict the user's location at specified future time is a considerable issue.

In this paper, we propose a framework QS-STT (QuadSection clustering and Spatial-Temporal Trajectory model) to explore the movement behavior in form of spatial and temporal model for location prediction. Prior to STT model exploring, the frequent regions where the user often passed by are first discovered by the QuadSection clustering. Then, the algorithm of spatial-temporal trajectory model construction is developed, in which both sequential relationships among regions and the corresponding appearance time information of objects are captured. In light of QS clustering based STT, we further propose an STT prediction algorithm to traverse STT for location prediction.

In frequent region discovery, grid-based clustering is a conventional approach. A cell is identified as a frequent region if the cell contains a sufficient number of trajectories which passed through it. However, determining a proper cell size is the challenging issue of grid-based clustering. The cell size affects the number of patterns and the accuracy of the location prediction [14]. In order to discover the possible number of movement patterns, as many frequent regions as possible are expected to be extracted. Given a minimum number of trajectories, clustering by setting a bigger cell size may easily detect the frequent region but lose the granularity of movement behavior modeling. On the contrary, a smaller cell size may earn granularity but be difficult in frequent region detection and result in fewer regions being discovered. To approach the reasonable and near-optimal set of frequent regions, we propose QuadSection clustering (abbreviated as QS clustering). QS clustering is based on the concept of divisive clustering [25] to detect the maximum number of regions within the optimal cell size in order to improve the completeness of the movement behavior modeling. Based on the frequent regions discovered by QS clustering, The spatial-temporal trajectory model (abbreviated as STT) is designed to discover the sequential relationships among regions in the spatial domain and derives the corresponding time probabilities of the objects which appear in the temporal domain.

The spatial-temporal trajectory model is represented as a variant of the probabilistic suffix tree with both spatial and temporal information of movements. Furthermore, the nature of the probabilistic suffix tree can convert a large number of sequential patterns into a compact model and generate a probability of next movement occurring in the recent sequences. Consequently, STT can reflect the moving behavior of an object in both spatial and temporal features, and its structure associated probability can be a predictor for the future location estimation. To evaluate the performance of the proposed QS-STT and location prediction algorithm, we conduct comprehensive

experiments on real data. Specifically, for verifying the effectiveness of QS-STT for location prediction under the different spatial density, we use four types of real trajectory datasets with different speed, including Walk, Run, Bike, and Car. The experimental results show that the proposed QS-STT is able to reflect an object's moving behavior and can predict future movements at a specified time slot efficiently and accurately. The main contributions of this paper are summarized as follows:

1. We propose a location prediction framework *QS-STT*, which extracts the frequent region using *QuadSection clustering* and explores the movement behavior in the form of *Spatial-Temporal Trajectory model*.
2. For frequent region discovery, we propose *QuadSection clustering* to approximate the reasonable and optimal solution for cell size determining in grid-based clustering and to earn the maximum number of discovered frequent regions to improve the movement behavior modeling.
3. We propose *Spatial-Temporal Trajectory model* (STT) to capture movement behavior in both spatial and temporal domains for location prediction.
4. The STT is an efficient compression scheme that converts a large trajectory data into a compact but representative model which reduces the storage size of the trajectory patterns compared to the association rules proposed.
5. We propose an *STT Prediction Algorithm* that provides accurate predictions.
6. The effectiveness and accuracy of our proposed framework QS-STT for location prediction has been demonstrated via extensive experiments on real trajectory datasets.

The rest of the paper is organized as follows. Section 2 discusses related works. An overview of the proposed framework QS-STT is given in Sect. 3. Section 4 describes the STT construction and the algorithm of location prediction using STT model is shown in Sect. 5. Section 6 discusses comprehensive experimental evaluations and Sect. 7 concludes the paper.

2 Related works

Traditional location prediction that uses motion functions to predict next locations of users only has good accuracy if the future time specified in a predictive query is close to the current time [10]. To achieve a better accuracy of location prediction for query time far away from the current time, prior works in [10] have elaborated on mining trajectory pattern for location prediction. The experimental results in [10] show that by exploring trajectory patterns, the accuracy of location prediction for both near time or distant future time is significantly improved. Many researches have put much effort on movement behavior analysis and proposed some algorithms to mine movement patterns. Movement behaviors are represented as different kinds of trajectory patterns in prior works. For example, movement behaviors are defined as sequential patterns [5, 7, 12, 17, 18] and association rules [10, 20]. Without loss of generality, given a set of trajectories, algorithms of mining trajectory pattern will first extract some regions with a certain degree of popularity, which are referred to as frequent regions. Then, original trajectories are transformed into sequences of frequent regions.

With given sequences of frequent regions, movement behaviors are thus defined as trajectory patterns that frequently appear among sequences of frequent regions. Those trajectory patterns can imply that objects usually follow similar movement behaviors. Specifically, the trajectory pattern discovery is a key role of the pattern-based location prediction.

Most research works for trajectory pattern mining use sequential pattern mining techniques [5, 7, 17]. From a set of trajectories, a set of frequent regions are extracted. Then, based on frequent regions, raw trajectories are represented as sequences of frequent regions. Thus, sequential pattern mining techniques are able to discover sequential relationships among frequent regions. Different from most of the existing literature in sequential pattern mining that focuses on the dependent order of frequent regions in the sequence without temporal information, the authors in [4, 5] proposed Temporally-Annotated Sequences (abbreviated as TAS) as trajectory patterns. As such, the results of sequential patterns contain a transition time between consecutive frequent regions along with TAS. Based on the above concept, the authors in [18] utilize TAS for location prediction. The input of location prediction in [18] is the set of recent movements without any query time, which is different from our prediction query. As such, the proposed method in [18] cannot deal with our prediction query.

In [10], the authors proposed a hybrid prediction model. The association rules among frequent regions are explored to represent trajectory patterns. Then, a hybrid prediction model is developed that combines trajectory patterns with motion function to forecast future location of a moving object. If the query time is close to the current time, one could predict the next location by the recent movement speed and direction, i.e, motion function. On the other hand, when the future time is far away from the current time, the authors discovered trajectory patterns in the form of association rules to predict the locations of objects. Hence, mining movement behaviors could provide a powerful way to enhance the accuracy of location predictions. However, the number of ruled patterns are significantly increased with a increasing number of trajectory data.

In the previous works on location prediction, the Markov chain models also have been used to predict the next movement of moving objects [9, 11, 26, 27]. The whole space is divided into cells and the Markov transition probability among cells is derived from a set of trajectories. Given a current location within a certain cell, the next cell is predicted by the transition probability of the Markov chain model that captures movement behaviors. Furthermore, the authors in [11] address the granularity problem of the space partitioning scheme. In particular, the authors utilize density-based clustering techniques to derive frequent regions. Then, the movements could be referred from cells and frequent regions. Thus, a prediction model is based on the hidden Markov process among the frequent regions and cells. However, the above works do not deal with predictive queries in which both the spatial and the temporal predicates are given.

To sum up the related works, there are some deficiencies. First, trajectory pattern is a key role in pattern-based location prediction. Most of the existing works only focus on trajectory pattern discovery in spatial domain. Prediction with spatial based trajectory pattern may result in biased prediction. In order to enhance the prediction

accuracy, both spatial and temporal information should be take into consideration on trajectory pattern discovery. Second, with a huge number of trajectory data, the number of trajectory patterns are significantly increased. To develop a representative compact model is a considerable issue on pattern-based location prediction.

3 The framework of QS-STT for location prediction

In this section, the overview of our proposed framework QS-STT for location-prediction is presented. Given an object's trajectories, our objective is to derive a spatial-temporal trajectory model to capture the movement behaviors of the object. In light of our trajectory model, given recent movements and a future time query, we intend to predict its locations at the specified future time. Thus, our proposed framework is shown in Fig. 2. As can be seen in Fig. 2, the proposed framework consists two modules: STT model construction and STT-location prediction. In the STT model construction module, given a set of raw trajectories, there are three steps as follows:

Step 1. Frequent Region Discovery: In this step, we extract frequent regions from a set of trajectories. Since locations of raw trajectories are GPS data points with uncertain properties in terms of both spatial and time domains, to capture movement behaviors, we need to extract frequent regions. A frequent region contains a sufficient number of trajectories whose data points are within the corresponding region. QuadSection Clustering is proposed to extract good quality frequent regions for movement behavior modeling.

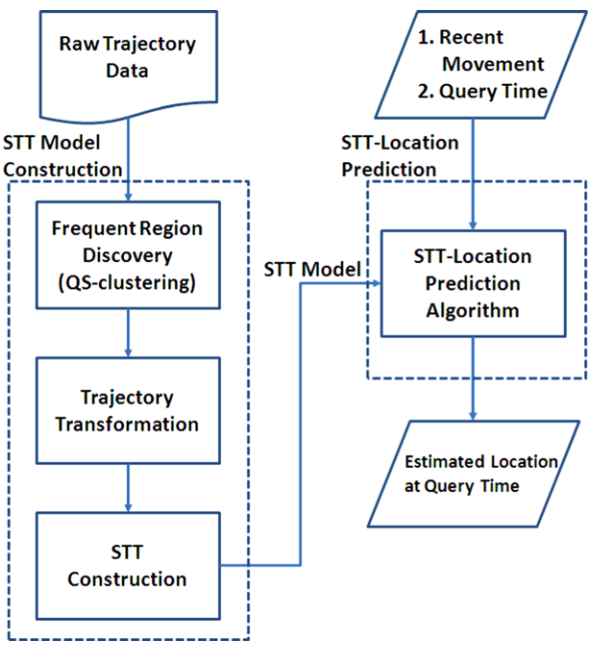


Fig. 2 The framework QS-STT for location prediction

Step 2. Trajectory Transformation: According to the set of frequent regions determined by Step 1, each raw trajectory is transformed into a region-based moving sequence. Location points that are not in frequent regions will be regarded as noise. As such, our STT could capture movement behaviors among frequent regions.

Step 3. STT Model Construction: In this step, we adopt a probabilistic suffix tree with spatial and temporal information to discover movement behaviors. Our proposed STT model contains not only transition probabilities among frequent regions (referred to as the spacial feature of the STT model) but also appearance probabilities within frequent regions (referred to as the temporal feature of the STT model).

In the STT-location prediction module, given the current movements of an object and a future time query, we use the STT model as a location predictor and propose an STT-location prediction algorithm to traverse the model and estimate the future location at the query time.

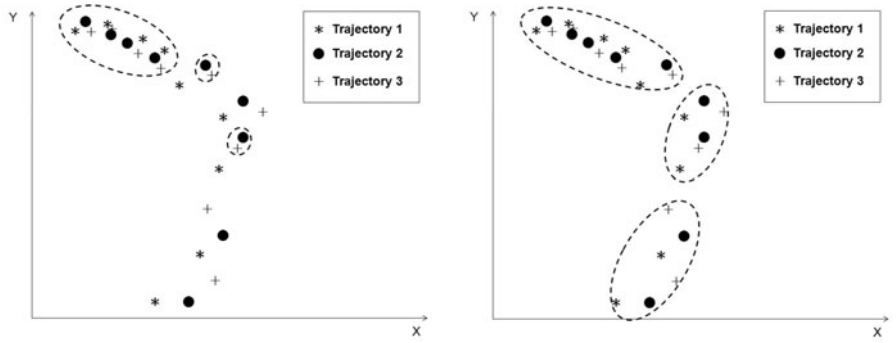
4 Spatial-temporal trajectory model construction

In this section, we detail the procedure for the discovery of frequent regions and the transformation of the raw trajectories to region-based moving sequences. Finally, we describe the process that how to construct an STT model to capture the moving behavior.

4.1 Frequent region discovery and trajectory transformation

An object's trajectory is generally represented as a sequence of spatial-temporal points sampled by a positioning sensor or device. Let a trajectory of a moving object be represented as $T = \{p_1, p_2, \ldots, p_n\}$, where n is the total number of points and a point $p_i = (l_i, t_i)$ is denoted as a location $l_i = (x_i, y_i)$ at a timestamp t_i $(0 \leq i \leq 1)$. We cannot directly use such trajectory data for spatial-temporal trajectory model construction. The reason is that the object will not repeat exactly the same location in every timestamp of each trajectory even if such a set of trajectories has a similar spatial route. Thus, we consider a trajectory to be a sequence of frequent regions. A frequent region is a region which an object often visits. For the discovery of frequent regions from an object's trajectories, density based clustering is adopted by many previous works [5, 10, 11, 17]. In the density based approach, a cluster is viewed as a frequent region if the number of trajectories visited is larger than a predefined threshold. For example, we can apply DBSCAN [8] to discover the clusters, i.e. frequent regions. For each point of a cluster, the neighborhood of a given radius (i.e., *Eps*) has to contain at least a minimum number of points (i.e., *MinPts*), that is, the density in the neighborhood has to exceed a predefined threshold.

However, the clusters that are detected by DBSCAN cannot be applicable to our STT because the clustering parameters are universal. In reality, the density of clusters extracted from trajectories varies widely because the speed of a moving object is not always constant. For example, the location point density of a trajectory for highway driving is lower than that for local city driving. The parameter setting is the constraint of DBSCAN for various density clustering. As an example in Fig. 3(a), if the



(a) Low Eps (b) High Eps

Fig. 3 An example of DBSCAN results with various Eps

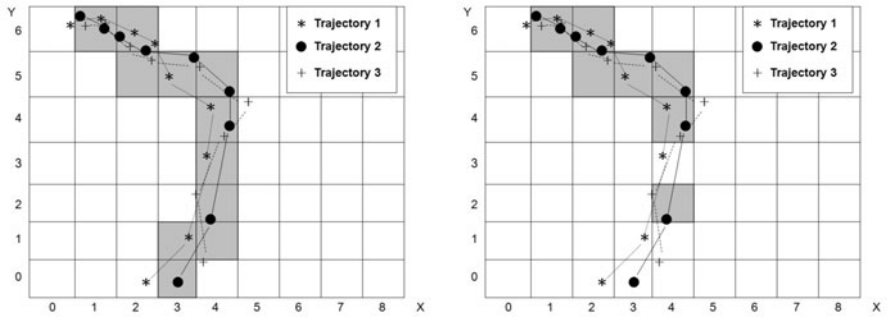
Eps value is set to a smaller value, some points are classified as "noise" and some information would be missed. Using a larger value for Eps causes the granularity problem to represent the trajectory, which is shown in Fig. 3(b).

The key concept of this research is to establish an STT model to capture the moving behavior by describing the relations between frequent regions. The granularity of clusters can affect the explanation of STT considerably. Therefore, we use grid based clustering to discover the frequent regions. The trajectory moving space C is partitioned into k grid cells of the same size. Thus, each region can be represented as a cell c_i with a universal size. The density of a cell is computed by taking the number of trajectory segments which pass through the cell. A trajectory segment $Ts = \{p_i, p_{i+1}\}$, where p_i is the ith location and $1 \leq i < i+1 \leq n$, is a contiguous sub-trajectory of T. We formally define the frequent region as follows:

Definition 1 A *Frequent Region* is a grid cell that contains at least *MinTs* number of trajectory segments passing by the grid cell.

An example of frequent region discovery is provided in Fig. 4(a), where Trajectory1, Trajectory2 and Trajectory3 are trajectories of a moving object passing through a certain space. Given $MinTs = 2$, the frequent regions are regions with gray color. Note that we use the number of segments passing through the cell instead of the number of points enclosed in a cell to decide if the cell is a frequent region. The reason is that the density of points varies with an object's moving speed meaning that it may miss some trajectory information. For example, the frequent regions c_{43}, c_{41}, c_{30} are not detected if the minimum point is set to 2 in a cell as shown in Fig. 4(b).

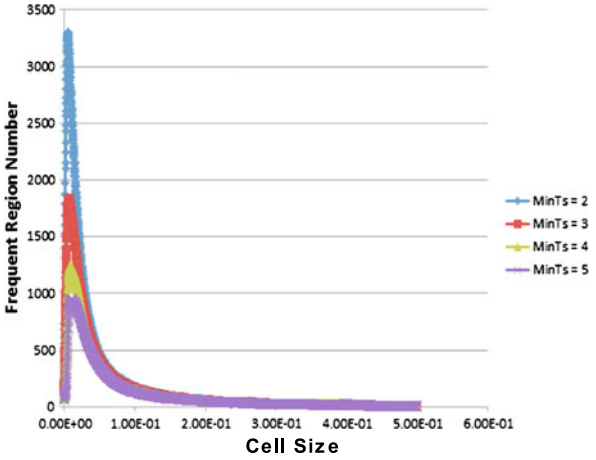
Furthermore, in order to construct a good STT model to capture the movement behavior precisely, as many frequent regions as possible have to be explored. Intuitively, a proper grid cell size can generate many frequent regions and potentially achieve good prediction precision. However, a large number of patterns may improve the prediction precision but increase the storage requirements. Therefore, it is necessary to find a good balance between the accuracy of the prediction and the storage requirements.



(a) Clusters determined by trajectory segments (b) Clusters determined by trajectory points

Fig. 4 An example of frequent region detection by grid-based clustering

Fig. 5 The experimental result of frequent region discovery by greedy testing under various MinTs



4.2 QuadSection clustering

One main challenging issue of grid file management for frequent region discovery is to determine the cell size. The cell size affects the number of patterns and prediction accuracy. Obviously, a large number of explored movement patterns may capture the movement behavior precisely and improve the prediction precision. In order to discover the possible number of movement patterns, as many frequent regions as possible are expected to be extracted. The naive approach is the greedy testing of the grid file with different cell sizes and selecting the one that results in the maximum number of frequent regions. Figure 5 shows the experimental results of greedy testing for frequent region discovery with various cell sizes. Given a *MinTs*, the cell size with the maximum number of discovered frequent regions is selected to be the optimal cell size.

However, greedy testing is an inefficient method to find the optimal cell size. Given a user-defined MinTs, the optimal cell size is defined as the cell size can result in maximal number of frequent regions. The greedy method for frequent region discovery

detects the number of frequent regions under the various cell size by greedy testing and bottom-up clustering. For each cell size, the greedy method will compute all the data point and then extract the number of frequent regions. In this paper, we develop a simplified method, called QuadSection clustering, to approach the optimum solution of determining the cell size and detecting the frequent region under a given *MinTs*. QuadSection clustering (abbreviated as QS clustering) is based on divisive hierarchical clustering [25]. A top-down cluster hierarchy is generated using the divisive clustering method. For frequent region detection, STT with QS clustering has advantages over STT with greedy-based grid clustering, including: the hierarchical approach can reduce and speed up the computation; the top-down clustering is more efficient than bottom-up clustering because we may not have to generate a complete hierarchy all the way down to individual data points.

Given a trajectory dataset, QS Clustering starts at the top rectangular space with all data points as one cluster. The cluster is then split by subdividing it into four cells. A cell is identified as a frequent region and becomes a leaf node if the cell has a maximum capacity, i.e., equal or greater than a user defined MinTs. This procedure is applied recursively to leaf nodes of each level until the number of discovered frequent region reaches the maximum value. The level with the maximum number of nodes is determined as the optimal grid file and the nodes of selected level are retrieved as the frequent regions. The cut-cost is proposed to measure the stopping criterion for divisive clustering if the cut-cost value is equal to or greater than zero. For QuadSection clustering, the cut-cost for level L_i in a divisive hierarchy H_d is defined as follows:

$$\textit{Cut-cost}(L_i) = \log\left(\frac{N_i}{N_{i+1}}\right) \tag{1}$$

In (1), the N_i is the total number of cells that become frequent regions for level L_i. The QuadSection Clustering algorithm is shown in Algorithm 1. Given a set of historical trajectory data D and user defined MinTs, we first bound the data space in rectangular space as the root cluster C_0. C_0 is initialized as level L_0. The algorithm then executes the splitting operation to divide the cluster into four equal rectangular sub-cells. A sub-cell is identified as a frequent region if the number of trajectories contained in the region is equal or greater than MinTs and becomes a leaf node of the next level in the hierarchy H_d. The function cut-cost(L_i) computes the stopping criterion for the splitting operation in each iteration. If the cut-cost(L_i) is equal to or greater than zero, the nodes belonging to L_i are retrieved and outputted as frequent regions.

Figure 6 is an example of QuadSection clustering for $\textit{MinTs} = 2$. The data space is first bounded as a rectangular region as root cluster C_0. The algorithm then splits C_0 into four equal sub-cells. C_2, C_3 and C_4 are identified as frequent regions and become leaf nodes of L_1. Cut-cost(L_0) $= \log(\frac{1}{3}) = -0.477$ and repeats the splitting operation for L_1. L_2 only has two frequent regions C_{42} and C_{13} and meets the stopping criterion due to cut-cost(L_1) $= \log(\frac{3}{2}) = 0.176$. The nodes ($C_2$, C_3 and C_4) belonging to Level 1 are outputted as frequent regions under the optimal cell size. An Effectiveness comparison of QuadSection clustering and the naive approach (grid clustering by greedy approach)is conducted as shown in Table 1. The effectiveness

Algorithm 1: QuadSection Clustering

Input : a set of trajectory data D, MinTs
Output: a set of frequent regions R_f

1 $C_0 \leftarrow \{C | C$ contains points $\in D$ and $count(D) \geq MinTs\}$;
2 Let a H_d has a single root C_0 and $C_0 \in L_0$;
3 **foreach** *node* $n \in L_i$ **do**
4 $S_c \leftarrow Split(n)$;
5 **foreach** *sub cell* s *in* S_c **do**
6 **if** $count(s) \geq MinTs$ **then**
7 Add node s to L_{i+1};
8 **end**
9 **end**
10 **end**
11 **if** $cost\text{-}cut(L_i) < 0$ **then**
12 Repeat line 3;
13 **else**
14 $R_f \leftarrow \{n | n \in L_i\}$;
15 **end**

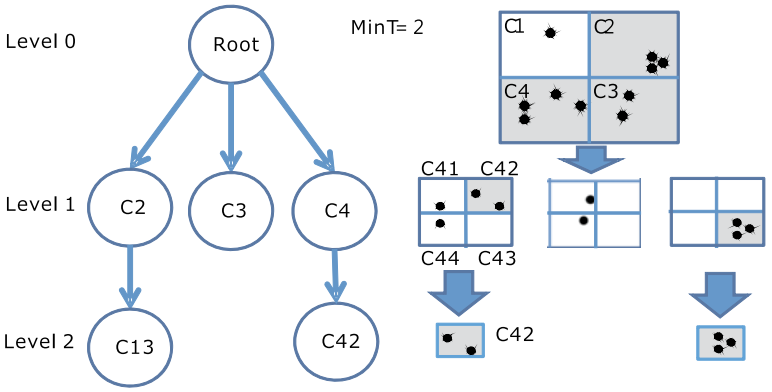
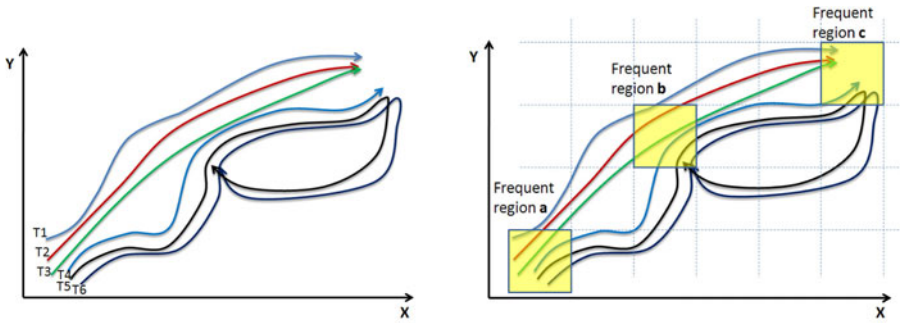


Fig. 6 QuadSection clustering example for $MinTs = 2$

Table 1 Comparison of QuadSection clustering and grid clustering by greedy approach

	QuadSection	Greedy approach
$MinTs = 2$	3240	3299
$MinTs = 3$	1707	1833
$MinTs = 4$	1215	1240
$MinTs = 5$	934	950



(a) Raw trajectories (b) Frequent region detection

T1: (a, 08:00) —30 mins→ (b, 08:30) —60 mins→ (c, 09:30)

T2: (a, 08:00) —30 mins→ (b, 08:30) —60 mins→ (c, 09:30)

T3: (a, 08:00) —20 mins→ (b, 08:20) —70 mins→ (c, 09:30)

T4: (a, 08:00) —40 mins→ (b, 08:40) —50 mins→ (c, 09:30)

T5: (a, 17:00) —30 mins→ (b, 17:30) —60 mins→ (c, 18:30) —60 mins→ (b, 19:30)

T6: (a, 17:00) —30 mins→ (b, 17:30) —60 mins→ (c, 18:30) —60 mins→ (b, 19:30)

(c) Trajectory transformation with time information

Fig. 7 An example of frequent region detection and trajectory transformation

is measured by the average approaching rate. The average approaching rate is defined as the average ratio between the number of frequent regions discovered by QS clustering and naive approach. The average approaching rate of Table 1 is 96.8 %. The experimental result shows that QS clustering is able to effectively discover the frequent regions with the approaching optimal cell size.

After extracting the frequent regions, each raw trajectory is transformed into a sequence of frequent regions. Such a sequence is called a region-based moving sequence and is defined as follows:

Definition 2 A *Region-based Moving Sequence* (moving sequence) is a sequence of frequent regions $S_r = \{(r_1, t_1), (r_2, t_2), \ldots, (r_n, t_m)\}$ with time constraint $t_1 < t_2 < \cdots < t_m$, where (r_i, t_j) indicates that object visits the frequent region r_i at timestamp t_j.

As can be seen in Fig. 4(a), three trajectories are transformed into region-based moving sequences $\{(c_{16}, t_1), (c_{26}, t_2), (c_{25}, t_3), (c_{35}, t_4), (c_{45}, t_5), (c_{44}, t_6), (c_{43}, t_7), (c_{42}, t_8), (c_{41}, t_9), (c_{31}, t_{10}), (c_{30}, t_{11})\}$.

Figure 7 shows a running example of frequent region discovery and trajectory transformation. There are six trajectories of a data set. The whole space is partitioned into cells. Given a *MinTs*, a frequent region is extracted if a cell is passed by *MinTs*

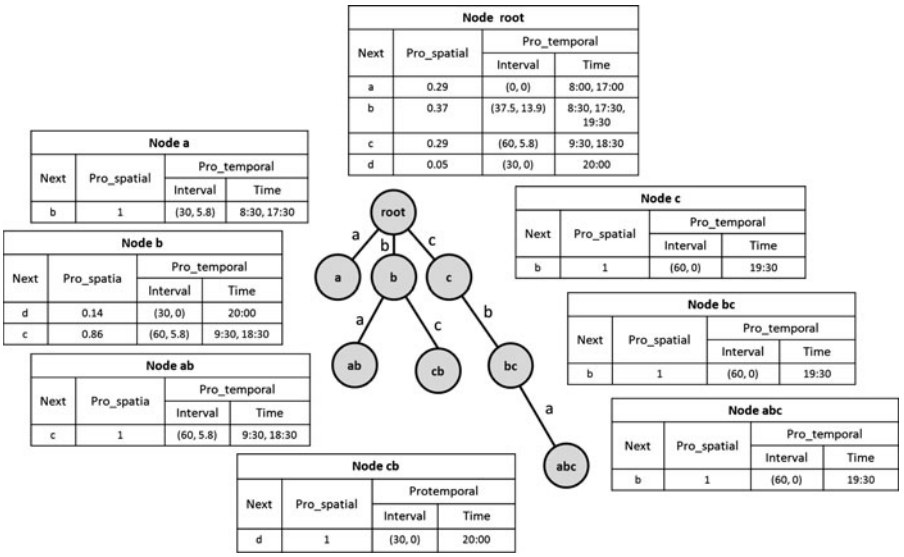


Fig. 8 An example of the STT model

trajectories. Finally, the raw trajectories are transformed into region-based moving sequences.

4.3 Spatial-temporal trajectory model construction

After trajectories are transformed into region-based moving sequences, we construct a spatial-temporal trajectory model to capture the moving behavior of an object. STT model is a variant of the probabilistic suffix tree. PST is a compact representation of a variable-order MarKov chain which draws the probability distribution for discrete events occurring in sequences. PST can be used to predict the next event by given the preceding sub-sequence [1, 22, 24, 28]. The original PST only considers the order of the sequence in the spatial domain but disregards the temporal information. For dealing with our location prediction at a specific future time, we propose the spatial-temporal trajectory model, which is an extension of PST.

Figure 8 shows an example of the STT model, which builds an STT over a symbol set $\Sigma = \{r_1, r_2, \ldots, r_n\}$, where symbol r_i is represented as a frequent region and n is the total number of frequent regions. Each edge in the tree is labeled by a frequent region that indicates one movement from one frequent region to the other. Each tree node is labeled by a sequence which represents a path from the node to the root. For example, a tree node which is labeled as $r_k \cdots r_2 r_1$ can be located by traversing from the root along the path $root \rightarrow r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_k$. A predictive table is associated with each node to maintain both spatial and temporal correlation between the region and the next movement: the spatial probability $Pro_{spatial}$ and the temporal probability $Pro_{temporal}$. $Pro_{spatial}$ is denoted as the conditional probability $P(r_{k+1} \mid r_1 r_2 \cdots r_k)$ of the next frequent region r_{k+1} that follows the label of the tree node $r_1 r_2 \cdots r_k$. $Pro_{temporal}$ is represented as transition interval i_{k+1} and representative timestamps

t_{k+1}. The transition interval is defined as vector $i_{k+1} = (mean, sd)$, where mean is the average transition interval from r_k to r_{k+1} and sd is the standard deviation of the transition interval. Representative timestamp t_{k+1} is donated as the representative time when the moving object often passes by r_{k+1}. For example, the predictive table of node ab in Fig. 8 shows that the next movement region is region c, $P(c \mid ab) = 1$, $i_c = (60, 5.8)$ and $t_c = 9{:}30, 18{:}00$, i.e., the conditional probability in the spatial domain that from ab moving to c is 0.6, the object often spends 60 minutes from ab moving to c and the standard deviation is 5.8 minutes, and the time to pass by c is usually 9:30 and 18:00.

Now, we will present how to construct the STT model. STT construction includes two steps: (1) for spatial domain, construct a PST to discover the frequent sequential patterns over frequent regions. (2) for the temporal domain, extract the transition interval vector and representative time of the next movement region for each tree node. Before the construction of STT, *minimal support*, denoted as *MinSup*, is specified to decide whether a frequent region r_k should generate a tree node as a child of a parent node or not. At the beginning of the first step, we hold an STT consisting of a single root node with the counts of each frequent region appearing in the trajectories. If the count of frequent region r_k is larger than the predefined *MinSup*, one tree node labeled as r_k will be created as a child node of the root. Then, tree node r_k will maintain the conditional probability of the frequent region r_{k+1} with the prefix segment of node r_k in the predictive table. For each sequence of frequent regions $r_1 r_2 \cdots r_k$, if a frequent region r_{k+1} appears behind it, the statistical information of all nodes labeled with the suffix of $r_1 r_2 \cdots r_k$ should be updated accordingly. After constructing the relationship between the frequent regions in the spatial feature from the historical trajectory dataset, the STT construction enters the second step to retrieve the temporal information of moving behavior. Each predictive table of tree nodes of STT is extended by aggregating the relevant temporal information. In this paper, the time distribution of a region for a moving object frequently visited is approximately either a single Gaussian distribution or mixture Gaussian distribution. Therefore, we assume that the time distribution of a frequent region can be approximated as a Gaussian mixture model (GMM), where each component of the GMM represents a temporal feature of the frequent cluster. The determination of the transition interval and representative timestamp of each frequent region can be approached by the GMM parameter estimation. In general, the parameters of the GMM model can be extracted by an Expectation-Maximization (EM) algorithm. However, using the EM algorithm for analysis of the GMM model may result in some drawbacks: it does not guarantee completeness due to unfortunate initialization; some smaller dense regions may be absorbed by larger and denser regions if they are too close to them. In order to extract the temporal feature, we build a histogram of the frequency distribution to approach the time distribution of each frequent region. Finally, we estimate the parameters of each component as a transition interval vector and representative timestamp. The whole process of STT construction is described in Algorithm 2. The algorithm of STT construction extracts significant sequential patterns and prunes infrequent patterns during tree construction, and then generates a STT. The input of the algorithm includes the STT parameter *MinSup* and a set of region-based sequence t_p. *MinSup* is the minimal occurrence of a pattern, i.e, the criteria for creating a child

Algorithm 2: STT Construction

```
   Input  : a set of region-based sequence tp, MinSup
   Output: a STT model T̄
1  Initialization: Let a T̄ has a single root and k = 1;
2  S_k ← {σ | σ ∈ t_p and count(σ) ≥ MinSup};
3  while S_k is not empty do
4      foreach element s in S_k do
5          Add node s to T̄;
           // Build the predictive table of node s
6          SpatialPrability (σ|s);
7          TransitionInterval(σ, s);
8          RepretativeTimes(σ, s);
9          if there exists a σ' ∈ t_p such that count(σ's) ≥ MinSup then
10             Add σ's to S_{k+1};
11     end
12 end
```

node of a subsequence s. The algorithm starts by initializing $\bar{T}$ and then extracts the set S_k of candidate patterns with length 1 (Lines 1–2). As shown in Lines 3–12, the algorithm checks whether each candidate is qualified to be a node in the tree. If a candidate s is qualified, the functions SpatialPrability, TransitionInterval, and RepretativeTimes (Lines 6–8) return the spatial-temporal information and update the corresponding data in predictive table. Next, the algorithm extends the candidates, as shown in Lines 9–10, and iterates the procedure until the candidate set is empty.

Although the execution efficiency for QS-STT is dependent on the size of trajectory data, QS-STT has some advantages to improve the execution efficiency. In QS clustering, the top-down hierarchical clustering approach can reduce and speed up the computation. Furthermore, STT model is a variant of PST, which is a successful and efficient model to capture the significant sequential patterns and organize those patterns into a tree structure.

5 Location prediction using STT model

In this section, we present how to predict future location using the trajectory moving profile for a given moving object's recent movements and query time. The STT represents a keyword dictionary of frequent trajectory patterns which are associating with conditional probability entries for each possible next movement. The concept of our prediction is to find the best next movement literally. We first encode the recent movements into a query sequence. Specifically, the query sequence s_q is a sequence of frequent regions which the object has visited. The tree node of STT will be located by the best *movement similarity* of its labeled pattern and s_q. The predictive table associated with the node will decide *moving potential* of each candidate for next movement in the table. The candidate with the highest moving potential will be

the next movement region and added into s_q. Such prediction procedure will be repeated literally until the query time is reached. The last movement is selected as the possible location at query time.

5.1 Movement similarity

The movement similarity is used to measure the similarity of a labeled sequence of a tree node n_k of STT and the moving sequence s_q. The objective is to search a node of STT model whose labeled sequence is the most similar to the moving sequence. Intuitively, the more recent movements have greater effect on future movements. Thus, we assign more weight to the frequent region where it is closer to the query time. The movement similarity of a tree node labeled n_k and the query sequence s_q is defined follows:

$$MS(n_k, s_q) = \sum_{i=1}^{Size(n_k \& s_q)} \frac{i^2}{\sum_{j=1}^{Size(s_q)} j^2} \tag{2}$$

where $0 \leq MS \leq 1$ and $(n_k \& s_q)$ is the longest common suffix of n_k and s_q.

For example, assume $s_q = abc$ and there are the patterns a, b, c, bc and ab in the tree. Considering the similarity between abc and those nodes, the similarity values are 0.07, 0.28, 0.64, 0.93, and 0.36 respectively. The node labeled bc has the best movement similarity with the query sequence abc.

5.2 Moving potential

After the best similar node is located, the moving potential of each next movement candidate is calculated to decide the next movement. The region with larger spatial probability in the predictive table has a greater chance of an object moving to it. However, such prediction would not be an effective prediction for a moving object because it only considers the sequential relationship in the spatial domain and ignores the effect of the temporal domain. For example, the probability that a user starting from home to school around 7:00 am is 0.7 and the probability of going from home to the park around 4:00 pm is 0.3. If the user's recent movement is home at 3:50 pm, the answer of location prediction is always school when we consider the spatial domain only. However, the correct location prediction should be the park. Thus, by considering both spatial and temporal information, the accuracy of location prediction could be enhanced. To reflect this idea, we propose a moving potential that takes both spatial and temporal information into consideration for location prediction.

For a next movement r_{k+1} of node n_k, the moving potential Pro_{ST}, is measured as follows:

$$Pro_{ST} = Pro_{spatial} \times Pro_{temporal} \tag{3}$$

$Pro_{spatial}$ is computed according to the conditional probabilities of seeing the symbol r_{k+1} right after the node n_k labeled string $r_1 r_2 \cdots r_k$ in the dataset.

To take the time feature into account for location prediction, we have to measure $Pro_{temporal}$. We use Chebyshev's inequality [21] to compute the upper bound of probability of time error as a temporal similarity measurement. For a candidate of next movement r_{k+1} of node n_k, we estimate the arrival time of next movement

t_e which is the sum of the current time t_c and the average transition interval *mean*. The minimum difference of t_e and the representative time t_{k+1} of the next movement candidates is defined as *temporal error*.

$$Pro_{temporal} = \frac{sd^2}{|\min\{t_e - t_{k+1}\}|^2} \tag{4}$$

For example, assume that temporal features of the next movement with node n_k are $i_{k+1} = (5, 2)$ and $t_{k+1} = \{12{:}00, 15{:}00, 17{:}00\}$. If the current time is 11:52, we evaluate that arrival time of region r_{k+1} of the moving object is 11:57 because the mean of the transition interval is 5. Minimum error among t_{k+1} is $|11{:}57 - 12{:}00| = 3$ minutes because 12:00 is the closest time to the arrival time. Therefore, $Pro_{temporal}$ is bound by 0.44. To ensure the sum of probability is 1, we normalize the estimation value to reassign the adjusted value as its temporal probability after estimating the upper bound of probability of time.

5.3 Location prediction

Given a recent movement and query time, the algorithm initializes the query sequence by encoding the recent movements into a sequence of frequent regions. The tree node of STT with the best movement similarity of its labeled pattern and s_q will be located. The predictive table associated with the node will decide the moving potential of each candidate for next movement in the table. The candidate with the highest moving potential will be the next movement region and added into s_q. Such prediction procedure will be repeated literally until the query time is reached. Therefore, we design a prediction algorithm which is a dynamic programming with time complexity of $O(nd)$ where n is the number of tree nodes and d is the number of time intervals. Let $F(c, t)$ be the highest probability to reach the next region among the candidates of next node. The recursive solution and initial condition are defined as follows:

$$F(c, t) = \begin{cases} 1, & \text{for } t \leq 0 \quad (5) \\ \max_{c'}(F(cc', t - m_{c'}) \times Pro_{c'}), & \text{for } t > 0, \quad (6) \end{cases}$$

where c is denoted as a query sequence, t is traveling time, c' is one candidate of the next nodes of c, and $m_{c'}$ is the mean of the transition time between them.

The sub-problem, $F(cc', t - m_{c'})$, is calculated recursively. The initial condition is set $F(c, t)$ to 1 for t is too short to go continuously. An example is shown in Fig. 9. If we want to find the answer of $F(c_1, 10)$, we have to calculate $F(c_2, 5)$ and $F(c_3, 5)$ first. The value of $F(c_1, 10)$ will be set to the maximum probability between $F(c_2, 5)$ and $F(c_3, 5)$, the value of $F(c_3, 5)$ is decided by $F(c_7, 0)$, and so on. As a result, $F(c_1, 10)$ is 0.4.

The STT prediction processing algorithm follows a general backward recursion procedure of a tabular method and is shown in Algorithm 3. In the beginning, we initialize the query sequence with recent movements and locate the node c with best movement similarity with s_q. Then, we run the procedure STT-prediction(c, t_q) to locate the predicted node c'. For each problem STT-prediction$(c, time)$ with *time* larger than 0, we first check whether it has computed or not. If it has not been computed yet, we scan all of the candidate next node c' of c to find the answer to the problem and return the computed result.

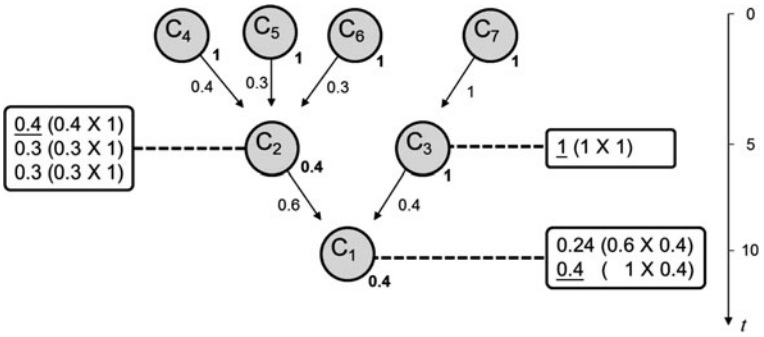


Fig. 9 An example of recursive solution for STT prediction

Algorithm 3: STT Location Prediction

Input : query pattern s_q, query time t_q
Output: predicted location

Set the best match node c by searching STT corresponding to s_q;
Get predicted node c' by calculating **STT-Prediction**(c, t_q);
Return **Location**(c');
`// A backward recursion function`
function STT-Prediction(c, time)
if *time* ≤ 0 **then**
    Return pattern c;
else
    if ***STT-Prediction****(c, time) hasn't compute* **then**
        Return **Max**(**STT-Prediction**(cc', *time-mean*$_{c'}$), **Proc**'), where $c' \in next_c$;
    else
        Return the computed result;
    end
end
end function

6 Experiments

In this section, extensive experiments are performed to evaluate the effectiveness and efficiency of our proposed location prediction method.

6.1 Experimental setting

In our experiments, we use real trajectory datasets from CarWeb [15], which is a traffic data collection platform, in which users record their own location every five seconds and upload their trajectories to the CarWeb server. We extract the data of

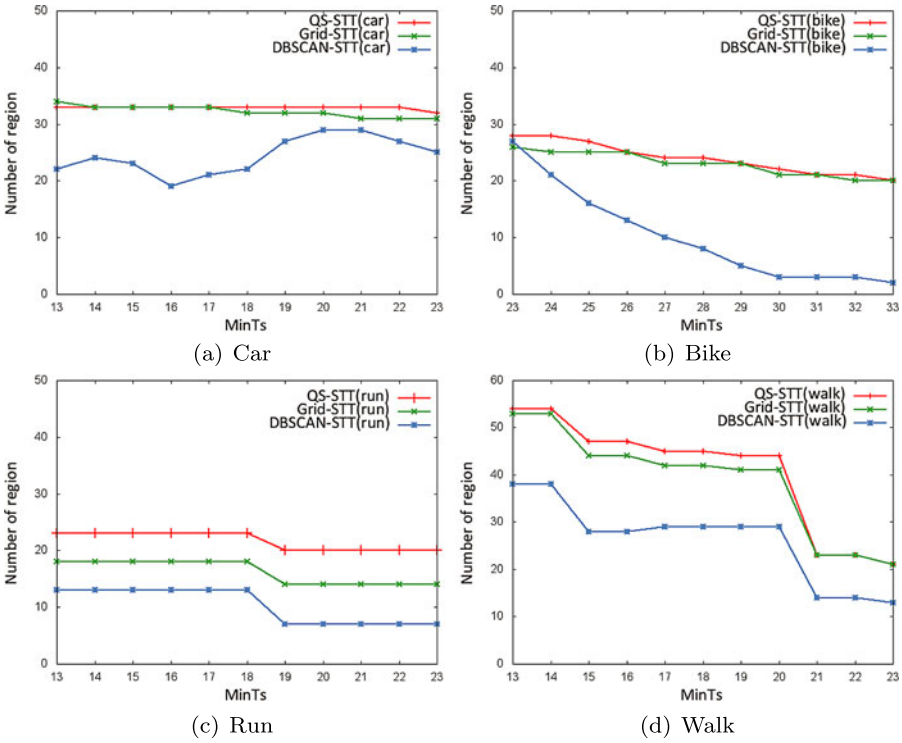


Fig. 10 Number of discovered patterns comparison by different clustering methods

one car of which the moving behavior has more than one similar path data as our major dataset for the experiments. The movements of a vehicle were obtained by a GPS equipped car while it followed the traffic network in Hsinchu city, Taiwan, over a period of two months. Since CarWeb is our own platform, the ground truth is easy to verify. Moreover, in order to evaluate the effectiveness and efficiency of our proposed method under various conditions, we use three trajectory data sets in different kinds of moving behaviors from RunSaturday http://www.runsaturday.com, which is a website server to collect training paths of sports hobbyists and the uploading rate is around tens of seconds to several minutes. The three data sets are summarized as follows:

– Walk: A walker's movements were collected over a period of seven months and 24 trajectories were extracted. The man has two types of walking behavior: walking in the street near his home and going hiking in Pohjois-Savo, Ita-Suomi, Finland.
– Run: A runner's movements were collected over a period of two months and 35 trajectories were extracted. The man has three types of running behavior: running in the playground near his home, running along an urban outer road, and running along the coastline.
– Bike: A bike's movements were collected over a period of two months and 16 trajectories were extracted. The data was recorded while the bike went

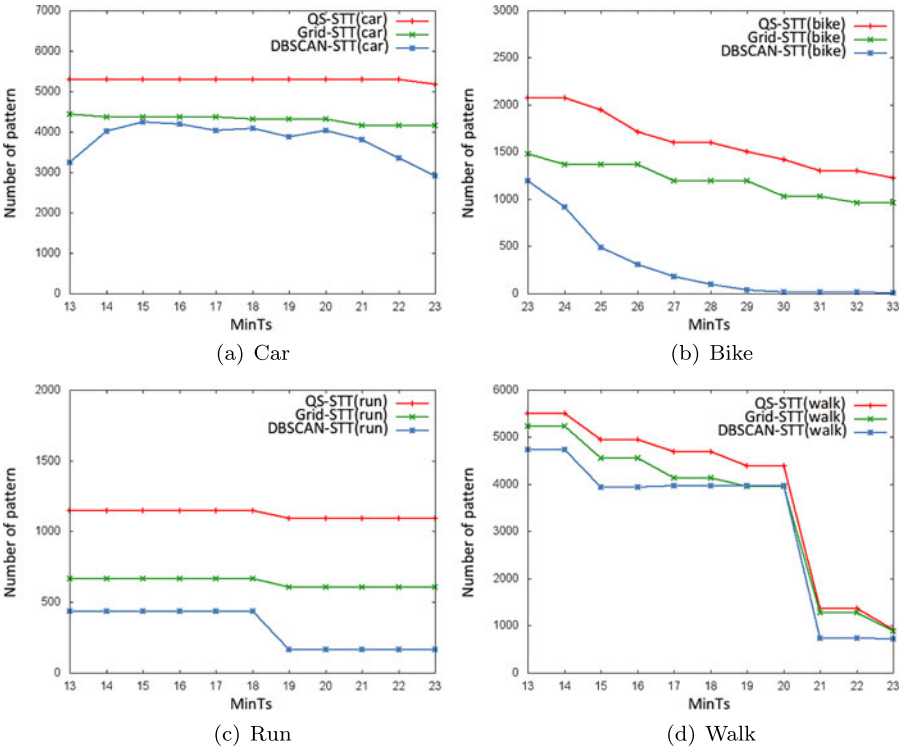


Fig. 11 Number of discovered regions comparison by different clustering methods

from a start position, cruising, and back to the start position. The moving behaviors on weekdays are different from on weekends such that the time periods on weekends were longer than on weekdays, so the trajectories around a larger circle. Those movements were recorded over a period of almost two hour.

The larger datasets are generated by perturbing the original datasets: every trajectory is duplicated with a small ($\leq \pm 5$ %) random value added to the original value. The number of trajectories (points) is 100 (20000). The test dataset was randomly selected from 20 % of the original trajectories.

6.2 Performance comparison of clustering algorithm

In this section, in order to verify that the prediction accuracy of STT can be improved by using QuadSection clustering (QS-STT), the performance of the proposed QS-STT, STT prediction model with density-based clustering (DBSCAN-STT), and grid-based clustering (Grid-STT) [14] is compared. Under various *MinpTs*, the best experimental *Eps* is set for DBSCAN-STT and the greedy testing is conducted to find the optimal cell size with maximum discovered regions for Grid-STT while the near-optimal cell size of QS-STT is able to be auto-decided. Given $MinSup = 2$ for

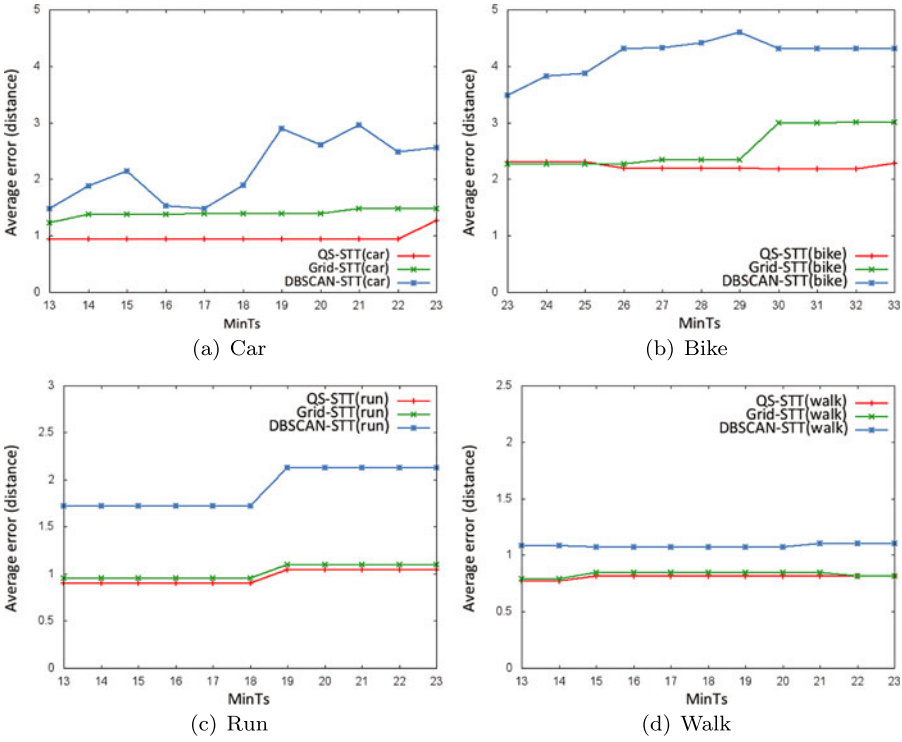
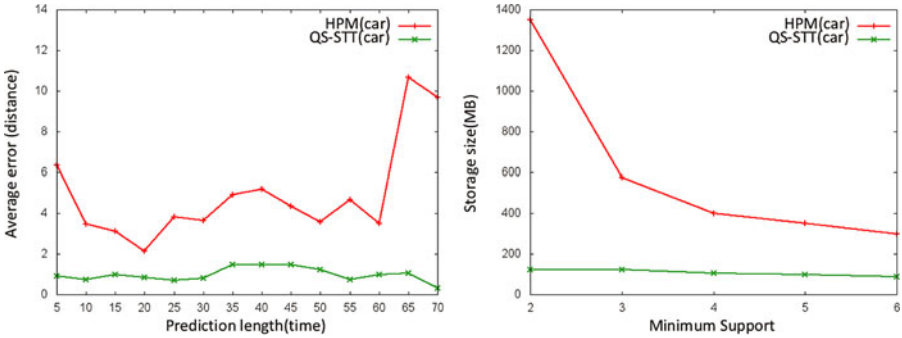


Fig. 12 Prediction error with different clustering methods

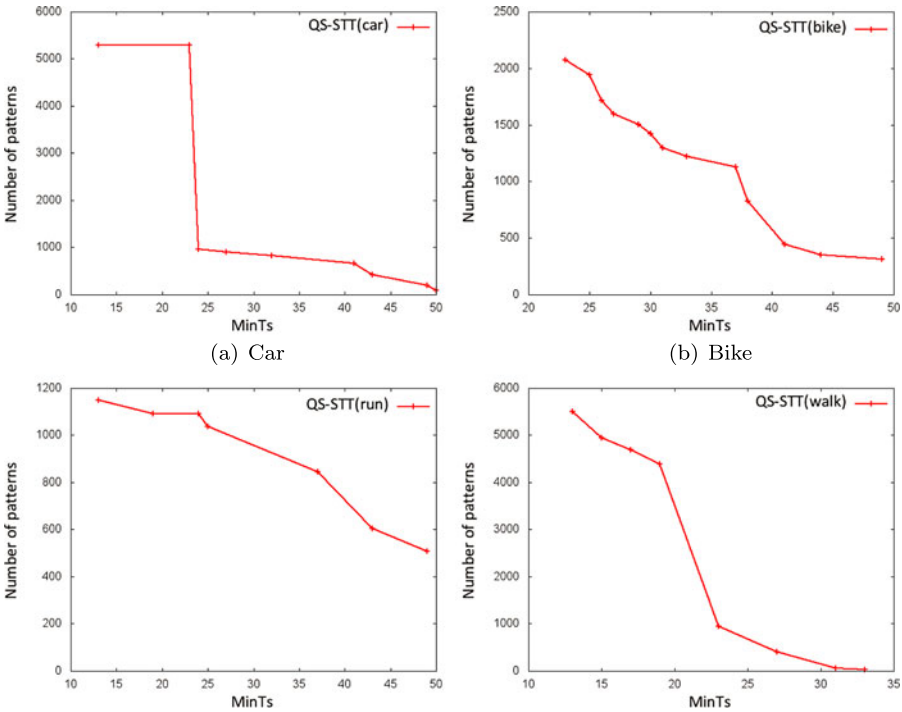
STT construction, we compare the number of regions and patterns discovered and the prediction error of different clustering methods. The number of regions discovered by the different clustering methods is compared in Fig. 10. The number of discovered regions decreases as the value of *MinpTs* grows. The more frequent regions that are extracted, the more trajectory patterns may be generated. As shown in Fig. 11, the experimental result indicates that the QuadSection clustering is able to detect the more patterns than STT-DBSCAN and Grid-STT.

Prediction accuracy is measured by the prediction errors, which is defined as the distance between a prediction location and its true location at a given query time. We tested 150 queries by giving different query times and averaging their errors. The prediction accuracy comparison with varied MinTs is shown in Fig. 12. For each MinTs, the prediction errors under different query time are averaged. The experimental result shows that QS-STT is more precise than other methods. QS-STT has very low errors regardless of prediction length while the errors of DBSCAN-STT rise significantly as the prediction length changes and this is because that region (cluster) discovered by DBSCAN-STT results in various sizes, which may cause granularity problems and lose accuracy when the predictive location falls in a big region. QS-STT is able to detect the frequent region with reasonable and near-optimal cell size compared with Grid-STT. This implies that QS-STT has more precise and complete information than DBSCAN-STT does in capturing moving behavior. Thus, the QuadSection clustering



(a) Prediction performance comparison (b) Storage requirements comparison

Fig. 13 Prediction model comparison

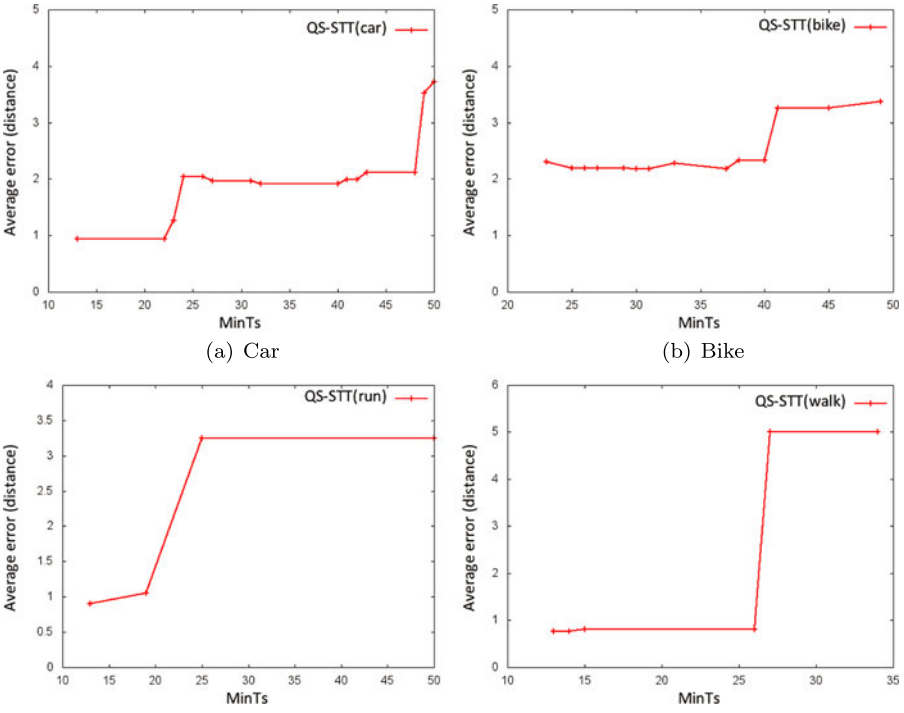


(a) Car (b) Bike

(c) Run (d) Walk

Fig. 14 Effect of MinTs on number of patterns

method preserves the moving behaviors well for the STT location prediction model. Note that the average errors of bike is higher than those of car is because the bike has smaller number of patterns. The smaller number of patterns will reduce the prediction accuracy.



(a) Car (b) Bike

(c) Run (d) Walk

Fig. 15 Effect of MinTs on prediction error

6.3 Performance comparison of location prediction

We compare the location prediction accuracy and storage requirements of our method (QS-STT) with that of the hybrid prediction model (HPM) [10], which uses the association rule-based pattern prediction approach. The parameters are set for best performance in terms of accuracy based on the experimental results. The QS-STT model is constructed by the near-optimal cell size under $MinTs = 13$. The HPM for our dataset: the frequent regions are decided by DBSCAN $Eps = 0.0055$, $MinPts = 4$, and *minimum confidence* $= 0.3$ for association rule discovery. The prediction comparison is under the various prediction temporal lengths. As expected, Fig. 13(a) indicates that QS-STT has lower errors than HPM in location prediction. HPM has higher errors caused by the fact that the frequent regions discovered by DBSCAN clustering may result in clusters of arbitrary shapes and sizes while the error of QS-STT is restricted by the fixed cell size.

We next study the storage requirements comparison under various *MinSup*. As expected, Fig. 13(b) demonstrates that our method has smaller storage size than HPM. While the storage size of HPM dramatically grows as the number of frequent regions increases, our method QS-STT still maintains a small storage size with tiny changes. The reason is because HPM using association rule based patterns generates an exponential number of rules as the number of frequent regions increases. On the

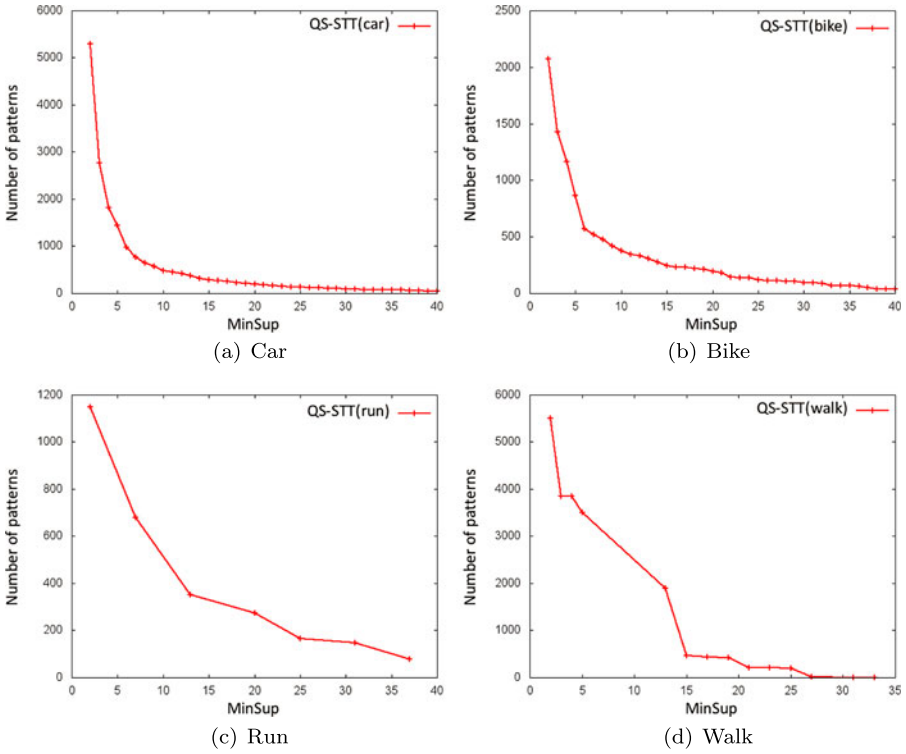
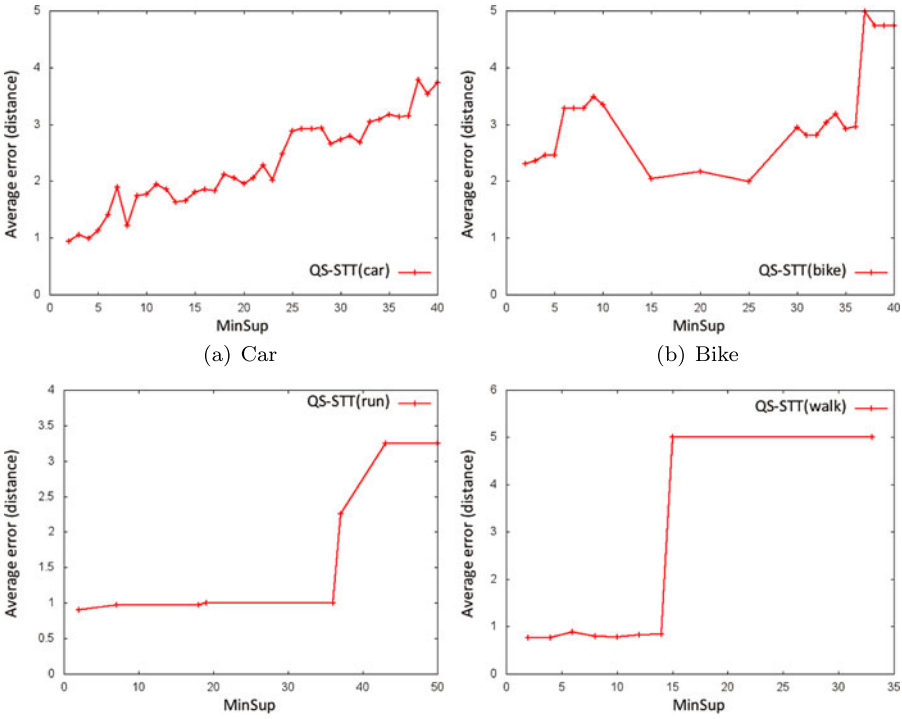


Fig. 16 Effect of MinSup on number of patterns

other hand, QS-STT using uses the suffix tree data structure which can compress the number of sequential patterns into a compact model.

6.4 Sensitivity analysis of parameters

In this section, we examine the effect of *MinTs* and *MinSup* to our model and prediction. *MinTs* is the parameter of QuadSection clustering and *MinSup* is the parameter of STT construction. We first study the effect of *MinTs* on frequent region discovery and the accuracy of the location prediction. In our definition, a frequent region is decided by *MinTs* number of trajectories which pass the region in a cell size. Therefore, a high value of *MinTs* may cause a small number of frequent regions and trajectory patterns. Prediction based on trajectory patterns could be affected by *MinTs*. According to the experimental results shown in Fig. 14, the number of trajectory patterns is reduced as the number of *MinTs* increases. The prediction error increases significantly due to the small number of trajectory patterns as shown in Fig. 15. We also investigate the effect of *MinSup*. For STT construction, a movement sequence is defined as a trajectory pattern if the sequence has *MinSup* number of same movement sequences appearing in a trajectory data set. Figure 16 presents the experimental results with *MinSup* varied. The number of trajectory patterns decreases dramatically as the value of *MinSup* grows. Furthermore, the prediction error affected by *MinSup*



(a) Car (b) Bike

(c) Run (d) Walk

Fig. 17 Effect of MinSup on prediction error

is provided in Fig. 17. The prediction error will potentially rise as the value of *MinTs* grows.

In general, *MinTs* and *MinSup* are two parameters which mainly determine the number of frequent regions and trajectory patterns which can be discovered. The more frequent regions that are extracted, the more trajectory patterns may be generated. From the experimental results above we can observe that the model can potentially achieve good prediction precision as many frequent regions and trajectory patterns which are discovered.

7 Conclusion

In this paper, we presented a pattern-based approach to predicting an object's future locations. We not only focus on how to discover frequent movement patterns and manage these patterns to answer predictive queries but also aim to propose a model that can reduce the pattern storage size. To achieve this goal, we proposed Quad-Section clustering to extract the maximum number of frequent regions with the reasonable and near-optimal cell size, and develop a spatial-temporal trajectory model to capture the movement behaviors of objects. QS-STT model could be a predictor for location prediction and enhance the prediction accuracy. The experimental results show that the QS-STT model is able to reflect an object's moving behavior with a

smaller storage size compared to existing patten-based approaches, while still guaranteeing the accuracy of location prediction.

Acknowledgements Wen-Chih Peng was supported in part by the National Science Council, Project No. 100-2218-E-009-016-MY3 and 100-2218-E-009-013-MY3, by Taiwan MoE ATU Program, by ITRI JRC, Project No. B301EA3300, by D-Link and by Microsoft.

References

1. Bejerano, G., Yona, G.: Modeling protein families using probabilistic suffix trees. In: Proc. of RECOMB, pp. 15–24 (1999)
2. Hung, C.-W.C.C.-C., Peng, W.-C.: Mining trajectory profiles for discovering user communities. In: Proc. of GIS-LBSN (2009)
3. Cao, X., Cong, G., Jensen, C.S.: Mining significant semantic locations from gps data. PVLDB **3**(1), 1009–1020 (2010)
4. Giannotti, F., Nanni, M., Pedreschi, D.: Efficient mining of temporally annotated sequences. In: Proc. of SDM (2006)
5. Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D.: Trajectory pattern mining. In: Proc. of KDD (2007)
6. Gonzalez, M., Hidalgo, C., Barabási, A.: Understanding individual human mobility patterns. Nature **453**(7196), 779–782 (2008)
7. Guyet, T., Quiniou, R.: Mining temporal patterns with quantitative intervals. In: Proc. of ICDM Workshops, pp. 218–227 (2008)
8. Hinneburg, A., Keim, D.A.: An efficient approach to clustering in large multimedia databases with noise. In: Proc. of KDD, pp. 58–65 (1998)
9. Ishikawa, Y., Tsukamoto, Y., Kitagawa, H.: Extracting mobility statistics from indexed spatio-temporal datasets. In: Proc. of STDBM, pp. 9–16 (2004)
10. Jeung, H., Liu, Q., Shen, H.T., Zhou, X.: A hybrid prediction model for moving objects. In: Proc. of ICDE (2008)
11. Jeung, H., Shen, H.T., Zhou, X.: Mining trajectory patterns using hidden Markov models. In: Proc. of DaWaK (2007)
12. Krumm, J., Horvitz, E.: Predestination: inferring destinations from partial trajectories. In: Proc. of UbiComp (2006)
13. Lee, J.-G., Han, J., Li, X., Gonzalez, H.: TraClass: trajectory classification using hierarchical region-based and trajectory-based clustering. PVLDB **1**(1), 1081–1094 (2008)
14. Lei, P.-R., Shen, T.-J., Peng, W.-C., Su, I.-J.: Exploring spatial-temporal trajectory model for location prediction. In: Proc. of MDM, pp. 58–67 (2011)
15. Lo, C.-H., Peng, W.-C., Chen, C.-W., Lin, T.-Y., Lin, C.-S.: CarWeb: A traffic data collection platform. In: Proc. of MDM (2008)
16. Lu, C.-T., Lei, P.-R., Peng, W.-C., Su, I.-J.: A framework of mining semantic regions from trajectories. In: Proc. of DASFAA, pp. 193–207 (2011)
17. Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., Cheung, D.W.: Mining, indexing, and querying historical spatiotemporal data. In: Proc. of KDD (2004)
18. Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F.: Wherenext: a location predictor on trajectory pattern mining. In: Proc. of KDD, pp. 637–646 (2009)
19. Montoliu, R., Gatica-Perez, D.: Discovering human places of interest from multimodal mobile phone data. In: Proc. of MUM, pp. 12–21 (2010)
20. Morzy, M.: Mining frequent trajectories of moving objects for location prediction. In: Proc. of MLDM, pp. 667–680 (2007)
21. Ostle, B., Malone, L.: Statistics in Research: Basic Concepts and Techniques for Research Workers. Iowa State University Press, Ames (1988)
22. Sun, S.C.P., Arunasalam, B.: Mining for outliers in sequential databases. In: Proc. of SDM (2006)
23. Peng, W.-C., Ko, Y.-Z., Lee, W.-C.: On mining moving patterns for object tracking sensor networks. In: Proc. of MDM (2006)
24. Ron, D., Singer, Y., Tishby, N.: The power of amnesia: learning probabilistic automata with variable memory length. Mach. Learn. **25**(2–3), 117–149 (1996)

25. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining, 1st edn. Addison-Wesley, Boston (2005)
26. Tsai, H.-P., Yang, D.-N., Peng, W.-C., Chen, M.-S.: Exploring group moving pattern for an energy-constrained object tracking sensor network. In: Proc. of PAKDD (2007)
27. Ishikawa, Y.T.Y., Kitagawa, H.: Extracting mobility statistics from indexed spatio-temporal datasets. In: Proc. of STDBM, pp. 9–16 (2004)
28. Yang, J., Wang, W.: Agile: A general approach to detect transitions in evolving data streams. In: Proc. of ICDM (2004)
29. Ying, J.J.-C., Lu, E.H.-C., Lee, W.-C., Weng, T.-C., Tseng, V.S.: Mining user similarity from semantic trajectories. In: Proc. of GIS-LBSN, pp. 19–26 (2010)
30. Yu, X., Pan, A., Tang, L.A., Li, Z., Han, J.: Geo-friends recommendation in gps-based cyber-physical social network. In: Proc. of ASONAM, pp. 361–368 (2011)
31. Zheng, K., Trajcevski, G., Zhou, X., Scheuermann, P.: Probabilistic range queries for uncertain trajectories on road networks. In: Proc. of EDBT, pp. 283–294 (2011)