Prediction of Moving Object Location Based on Frequent Trajectories

Mikołaj Morzy

Institute of Computing Science
Poznań University of Technology
Piotrowo 2, 60-965 Poznań, Poland
Mikolaj.Morzy@put.poznan.pl

Abstract. Recent advances in wireless sensors and position technology provide us with unprecedent amount of moving object data. The volume of geospatial data gathered from moving objects defies human ability to analyze the stream of input data. Therefore, new methods for mining and digesting of moving object data are urgently needed. One of the popular services available for moving objects is the prediction of the unknown location of an object. In this paper we present a new method for predicting the location of a moving object. Our method uses the past trajectory of the object and combines it with movement rules discovered in the moving objects database. Our original contribution includes the formulation of the location prediction model, the design of an efficient algorithm for mining movement rules, the proposition of four strategies for movement rule matching with respect to a given object trajectory, and the experimental evaluation of the proposed model.

1 Introduction

Last years have witnessed a tremendous increase in the number of mobile devices available on the market. Advances in position technology and the widespread use of communication standards, such as GPRS, Bluetooth, Wi-Fi, or WiMAX prompt manufacturers to offer mobile devices supplied with high resolution displays and positioning sensors. Global positioning systems (GPS) are becoming affordable and accurate, thus enabling the deployment of position-aware applications. Examples of mobile devices that profit from location-based services and applications include mobile phones, digital cameras, personal digital assistants, vehicles, and many others.

Ongoing adoption of mobile devices results in an increasing demand for location-based services and applications. Most of location-based services require accurate or approximate position of a mobile client to provide functionality. Examples of such services include management of traffic, navigational service, way-finding, location-based advertising, or movement coordination. In a typical scenario, a moving object periodically informs the positioning framework of its current location. Between position disclosures the location of a mobile object remains unknown. Due to the unreliable nature of portable mobile devices and inherent global positioning systems limitations, such as congestions, existence of urban canyons (areas not covered by positioning signals, e.g. subterranean parking garages), or signal losses caused by natural phenomena, the location of a mobile object is often not known for a longer period of time. In such case, an efficient method for predicting possible location of a moving object is required.

Most location-based services demand a fast and reliable location prediction method. The unceasing stream of data generated by positioning sensors, combined with thousands of mobile devices communicating over a wireless channel, make traditional prediction methods obsolete. Sophisticated methods using complex models may yield accurate results, but are computationally unfeasible in mobile environment. For instance, simulation is sometimes used for mimicking the behavior of mobile objects. The quality of simulation depends on numerous parameters governing the movement model. Often, the cost involved in the computation of the model is prohibitively high. In addition, the environment in which mobile objects reside can be dynamic and difficult to capture. For instance, a simple two-dimensional model of a city may not adapt to frequent changes in city topography caused by road construction. Another serious drawback of the currently used location prediction techniques is the fact that most techniques make little or no use of the huge amounts of historical data. Movement data acquired from other moving objects hide valuable knowledge about moving object behavior. In particular, patterns describing popular movement trajectories can be discovered when mining historical data. Alas, data mining techniques are usually (incorrectly) considered too slow and too computationally expensive for real-time location prediction.

Many location-based services may sacrifice prediction accuracy for prediction speed. We follow this paradigm by simplifying our model and resigning from exact modeling of the topography of the movement environment. Instead, we mine historical movement data to discover frequent trajectories traversed by moving objects. These frequent trajectories are further used as an approximate model of the topography. For each object, whose exact location is not known, we perform fast matching of object's trajectory with appropriate frequent trajectories to build a probabilistic model of object location. Our method is fast and efficient, because expensive computations, e.g. mining for frequent trajectories, are performed periodically and offline. Runtime location prediction consists only in trajectory matching, which is a far less arduous task. Another advantage of our approach is the fact that it is independent of a given topography of the movement environment. Our original contribution includes using historical movement data to build an approximate environment model, the design of the *AprioriTraj* algorithm to discover frequent trajectories, the development of four trajectory matching strategies, and the experimental evaluation of the proposal.

The paper is organized as follows. Section 2 presents the related work on the subject. In Section 3 we introduce basic definitions used throughout the paper. The *AprioriTraj* algorithm and trajectory matching strategies are presented in Section 4. The results of conducted experiments are reported in Section 5. We conclude in Section 6 with a brief summary and the future work agenda.

2 Related Work

Significant research effort has been undertaken in both mobile computing and spatial data mining domains. Research on tracking of moving objects resulted in several proposals for predicting future object locations. The method presented in [9] uses recent movement history of an object and combines it with recursive motion functions for objects with unknown motion patterns. An interesting proposal of using time-series analysis enriched with travel speed simulation to predict future trajectory of a moving object is formulated in [12]. A complex model that considers location prediction with accuracy guarantees is presented in [11]. A simulation-based approach to future trajectory prediction based on a non-linear movement model can be found in [10].

Since the advent of spatial data mining [6] many methods and algorithms were developed [4]. However, the problem of mining trajectories of moving objects remained relatively unchallenged until recently. Advances in this field include the proposal to cluster similar trajectories [7] and to use periodic patterns appearing in a single trajectory as the basis for location prediction [8]. A very interesting algorithm for mining patterns from imprecise trajectories of moving objects can be found in [13]. All these works extend the basic framework of periodic sequential patterns [5] and frequent sequential patterns [2]. Our work is strongly influenced by the approach presented in [13]. However, [13] deals primarily with uncertainty in moving object trajectories. The authors propose a new match measure for uncertain trajectories and devise the TrajPattern algorithm for mining sequential patterns that is not based on the Apriori algorithm. On the other hand, the method presented in this paper follows the support-based framework for discovering patterns and uses an Apriori-like algorithm to discover simple movement rules.

3 Basic Definitions

Given a database of moving object locations, where the movement of objects is constrained to a specified area. Let $O = \{o_1, \ldots, o_m\}$ denote the set of objects. Let the location of the j-th object during i-th measurement be denoted as $l_j^i = \left(x_j^i, y_j^i\right)$. The domain of location coordinates is continuous, but the smallest unit of location measurement provides a natural discretization of the input data. However, the level of granularity of this natural discretization is too detailed when compared to the number of moving objects or the number of locations registered for a single object. Therefore, any patterns discovered at the raw data level cannot be generalized. To overcome this obstacle we superimpose a grid on the movement area. The grid consists of square cells of the constant size, denoted as *grid_size*. Each edge can be traversed in two directions, vertical edges can be traversed eastwards and westwards, whereas horizontal edges can be traversed northwards and southwards.

An ordered list of consecutive location measurements for a given object constitutes a trajectory of the object, denoted as $t_j = \left\langle l_j^0, l_j^1, \ldots, l_j^n \right\rangle$. Alternatively, the trajectory of an object can be represented as an ordered list of segments, where each segment is defined by two consecutive location measurements of an object, $t_j = \left\langle s_j^0, s_j^1, \dots, s_j^{n-1} \right\rangle$, where $s_j^i = \left(l_j^i, l_j^{i+1} \right)$. Note that each segment can be replaced with an ordered list of edges traversed by that segment. This replacement transforms the original continuous coordinates domain into a discretized domain of edges. Finally, we represent a moving object trajectory as an ordered list of edges traversed by the trajectory. Let e_{pq} denote an edge. Then, the trajectory of the j-th object is $t_j = \left\langle \left(e_{p_0 q_0}, d_0\right)_j, \left(e_{p_1 q_1}, d_1\right)_j, \dots \right\rangle$, where $d_i \in \{ne, sw\}$ denotes the direction in which the edge was traversed (north-east and south-west, respectively).

Let E be the set of all edges. The cardinality of the set E is given by

$$|E| = \left\lceil \frac{a}{grid\_size} \right\rceil * \left(\left\lceil \frac{b}{grid\_size} \right\rceil + 1 \right) + \left(\left\lceil \frac{a}{grid\_size} \right\rceil + 1 \right) * \left\lceil \frac{b}{grid\_size} \right\rceil$$

where $a = \max_{ij}\{x_j^i\} - \min_{ij}\{x_j^i\}$, $b = \max_{ij}\{y_j^i\} - \min_{ij}\{y_j^i\}$.

Let D denote the set of all trajectories, $D = \{t_1, t_2, \dots, t_n\}$. The support of an edge e_{pq} is the number of trajectories that traverse the edge in a given direction. Note that each edge has two values of support, northeastward and southwestward. An edge e_{pq} is frequent, if its support exceeds the user-defined threshold of *minsup*. Given a trajectory t_j. The length of the trajectory, denoted as $length\,(t_j)$, is the number of edges constituting the trajectory t_j. Given two trajectories t_i and t_j. The trajectory t_i is a sub-trajectory of t_j (the trajectory t_j contains the trajectory t_i, denoted $t_j \supseteq t_i$) if the list of edges constituting t_i is a continuous sublist of the list of edges in t_j and the directions of traversal of edges in t_i are the same as the directions of traversal of corresponding edges in t_j. Trajectories t_i and t_j are adjacent if there exists a trajectory t_k, such that $length\,(t_k) = 2$ and $\left(e_{p_{max} q_{max}}, d_{max}\right)_i = \left(e_{p_0 q_0}, d_0\right)_k \wedge \left(e_{p_0 q_0}, d_0\right)_j = \left(e_{p_1 q_1}, d_1\right)_k$ (i.e., the last element of the trajectory t_i is the same as the first element of the trajectory t_k and the first element of the trajectory t_j is the same as the second element of the trajectory t_k). The concatenation of trajectories $t_i \| t_j$ is a trajectory $t_l = \left\langle \left(e_{p_0 q_0}, d_0\right)_i, \dots, \left(e_{p_{max} q_{max}}, d_{max}\right)_i, \left(e_{p_0 q_0}, d_0\right)_j, \dots, \left(e_{p_{max} q_{max}}, d_{max}\right)_j \right\rangle$.

The support of the trajectory t_j is the number of trajectories in D that contain t_j. A given trajectory t_j is frequent if the support of t_j exceeds the user-defined threshold of *minsup*. The set of all frequent trajectories is denoted as L. Obviously, a frequent trajectory must consist of frequent edges only, furthermore, each sub-trajectory of a frequent trajectory is also frequent.

A movement rule is an expression of the form $t_i \Rightarrow t_j$ where $t_i, t_j \in L$, t_i and t_j are adjacent trajectories and $t_i \| t_j$ is a frequent trajectory. The trajectory t_i is called the antecedent of the rule, the trajectory t_j is called the consequent of the rule. Contrary to the original formulation of association rule mining we do not require the antecedent and the consequent of a rule to be disjunctive.

The *support* of the movement rule $t_i \Rightarrow t_j$ is defined as the support of $t_i \| t_j$,

$$support\,(t_i \Rightarrow t_j) = \frac{|t_k \in D : t_k \supseteq (t_i \| t_j)|}{|D|}$$

The *confidence* of the movement rule $t_i \Rightarrow t_j$ is the conditional probability of t_j given t_i,

$$\mathit{confidence}\left(t_i \Rightarrow t_j\right) = P\left(t_j | t_i\right) = \frac{\mathit{support}\left(t_i \| t_j\right)}{\mathit{support}\left(t_i\right)}$$

The problem of prediction of location of moving objects based on frequent trajectories can be decomposed into two subproblems:

– generate all movement rules with support and confidence greater than user-defined thresholds of *minsup* and *minconf*, respectively,
– match discovered movement rules with the trajectory of a moving object for which the current location is to be determined.

4 Prediction of Location

To solve the problem of generating movement rules we use a modified version of the well-known *Apriori* algorithm [1]. The outline of our *AprioriTraj* algorithm is depicted in Figure 1. First, we find all frequent trajectories of the length 1 (i.e., all frequent edges). Next, we concatenate adjacent frequent edges to form candidate trajectories of the length 2. We perform a full database scan to determine actual support counts for candidate trajectories and we determine L_2, the set of frequent trajectories of the length 2. Next, we iteratively find sets of frequent trajectories of the length k based on frequent trajectories of the length $(k-1)$ found so far. In each iteration we form a set of k-element candidate trajectories by combining overlapping frequent trajectories. We consider two trajectories t_i and t_j to be overlapping, if the trajectory resulting from removing the first edge from t_i is the same as the trajectory resulting from removing the last edge from t_j (e.g., trajectories $\langle A, B, C, D\rangle$ and $\langle B, C, D, E\rangle$ are overlapping, whereas trajectories $\langle A, C, D, E\rangle$ and $\langle B, C, D, E\rangle$ are not). Two overlapping trajectories t_i and t_j are used to generate a candidate trajectory t_{ij} by concatenating the last edge of t_j to t_i (e.g., the concatenation of $\langle A, B, C, D\rangle$ and $\langle B, C, D, E\rangle$ yields $\langle A, B, C, D, E\rangle$). Contrary to the *Apriori* algorithm, we do not have to verify candidate trajectories for the containment of infrequent sub-trajectories, because the above generation procedure does not produce any superfluous candidate trajectories.

When all frequent trajectories have been found, the task of generating movement rules is straightforward. Each frequent trajectory t_i of the length l can be used to generate movement rules by splitting the trajectory t_i into $(l-1)$ pairs of sub-trajectories (t_i', t_i''), such that for each pair $t_i' \| t_i'' = t_i$ and the splitting point is chosen after i-th $(i = 1, 2, \ldots, l-1)$ element of the trajectory t_i. For each pair $(t_i' \| t_i'')$ we have to verify that $\mathit{confidence}\left(t_i' \Rightarrow t_i''\right) = \mathit{support}\left(t_i\right) / \mathit{support}\left(t_i'\right) \geq \mathit{minconf}$, otherwise we reject the rule $t_i' \Rightarrow t_i''$.

Let us now focus on the problem of matching discovered movement rules with the trajectory of a moving object. Given a moving object q with a trajectory t_q. We are searching for movement rules $t_i \Rightarrow t_j$ such that:

Require: L_1, the set of all frequent trajectories of the length 1

$C_2 = \{t_i \| t_j : t_i \in L_1 \wedge t_j \in L_1 \wedge t_i, t_j \textit{ are adjacent}\}$
for all trajectories $t \in D$ **do**
  for all candidate trajectories $c \in C_2$ **do**
    if $t \supseteq c$ **then**
      $c.count++$;
    end if
  end for
end for
$L_2 = \{t \in C_2 : \textit{support}\,(t) \geq \textit{minsup}\}$
for $k = 3; L_{k-1} \neq \emptyset; k++$ **do**
  for all trajectories $t_i \in L_{k-1}$ **do**
    for all trajectories $t_j \in L_{k-1}$ **do**
      if $\forall n \in \langle 1, k-1 \rangle \left(e_{p_n q_n}, d_n\right)_i = \left(e_{p_{n-1} q_{n-1}}, d_{n-1}\right)_j$ **then**
        $C_k = C_k \cup t_{ij}$, where $t_{ij} = t_i \| \left(e_{p_{max} q_{max}}, d_{max}\right)_j$;
      end if
    end for
  end for
  for all trajectories $t \in D$ **do**
    for all candidate trajectories $c \in C_k$ **do**
      if $t \subseteq c$ **then**
        $c.count++$;
      end if
    end for
  end for
  $L_k = \{t \in C_k : \textit{support}\,(t) \geq \textit{minsup}\}$
end for
$\textit{Answer} = \bigcup_{i=1}^{k} L_i$

Fig. 1. *AprioriTraj* algorithm

- $t_i \supseteq t_q$ and the last edge in both t_i and t_q is the same,
- $t_q \supseteq t_i$ and the last edge in both t_i and t_q is the same,
- t_i and t_q are the same.

In all above cases the consequent t_j is used as a prediction of the location of the moving object q in subsequent moments of time. Unfortunately, for a given trajectory t_q too many movement rules can be matched for making an informed decision about possible location of q. A naive approach of ranking matched movement rules based solely on confidence fails, because it does not consider lengths of the rules, thus prefers very short rules that are often useless for predicting the location of a moving object. Furthermore, this approach does not consider the coverage of matched rules with the trajectory t_q. This motivates us to devise new strategies of matched movement rule selection, presented briefly below.

4.1 Simple strategy

$$\arg\max_{t_i \Rightarrow t_j} \frac{|t_i|}{|t_q|} * \mathit{confidence}\,(t_i \Rightarrow t_j)$$

The drawback of the simple strategy is the fact that it does not consider the length of the consequent of the movement rule. Also, the simple strategy treats the length of the consequent linearly, which may lead to the situation where a long covering movement rule with low confidence is preferred to a shorter but more credible movement rule. Nevertheless, when the trajectory t_q is very short (the history of movement of the object q is almost unknown), then the simple strategy is appropriate.

4.2 Polynomial strategy

$$\arg\max_{t_i \Rightarrow t_j} \frac{1}{2}\left(\sqrt{\frac{|t_i|}{|c_1|}} + \sqrt{\frac{|t_j|}{|c_2|}}\right) * \mathit{confidence}\,(t_i \Rightarrow t_j)$$

where c_1 and c_2 are the lengths of the longest antecedent and consequent in the rule set, respectively. This strategy is fair and balanced, it represents a reasonable compromise between the simplicity of the simple strategy and the complexity of the logarithmic strategy presented next.

4.3 Logarithmic strategy

$$\arg\max_{t_i \Rightarrow t_j} \left(w_1 + w_2 * \log_{|c_1|}|t_i| + w_3 * \log_{|c_2|}|t_j|\right) * \mathit{confidence}\,(t_i \Rightarrow t_j)$$

where $w_1 + w_2 + w_3 = 1$. Weights w_1, w_2, w_3 are used to shift emphasis on the confidence factor, the relative length of the antecedent, or the relative length of the consequent of the movement rule. The use of the logarithm smooths the differences between long movement rules and, at the same time, accentuates the differences between short movement rules.

4.4 Aggregate strategy

Often, discovered movement rules can be grouped into sets of similar rules. In particular, rules sharing the same antecedent can be regarded as a family of rules that predict the movement in a given direction with the confidence of prediction diminishing with the length of the consequent. Let us assume that all movement rules have been grouped according to their antecedent. The aggregate strategy selects the following movement rule

$$\arg\max_{t_i \Rightarrow t_j} \frac{|t_i|}{|t_q|} * \frac{|t_j|}{|c_2|} * \frac{\sum_{t_x \Rightarrow t_y \in G} |t_y| * \mathit{confidence}\,(t_x \Rightarrow t_y)}{\sum_{t_x \Rightarrow t_y \in G} |t_y|}$$

where G denotes the group to which belongs the rule $t_i \Rightarrow t_j$ and c_2 denotes the length of the longest consequent in the rule set. The aggregate strategy considers the coverage factor, the relative length of the antecedent, and the predictive power of the group of related movement rules. The main drawback of this strategy is the computational cost involved in grouping of the movement rules. It is worth emphasizing that the four strategies of movement rule selection presented above are complementary and can be used simultaneously depending on the nature of the trajectory t_q.

5 Experimental Results

The data used in our experiments have been generated by the Network-based Generator of Moving Objects [3]. Maximum velocity has been set to 150 and the number of time units has been set to 100. For each distinct number of moving objects 30 different instances of the database have been generated. All experiments were conducted on a Pentium Centrino 1.8 GHz computer with 1GB RAM running Windows XP.

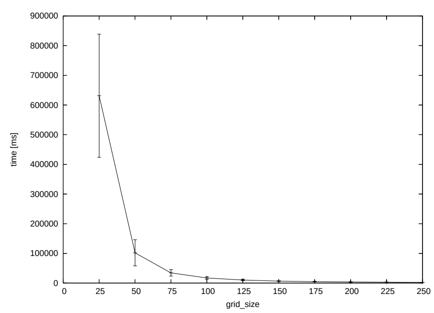

Fig. 2. time vs grid_size

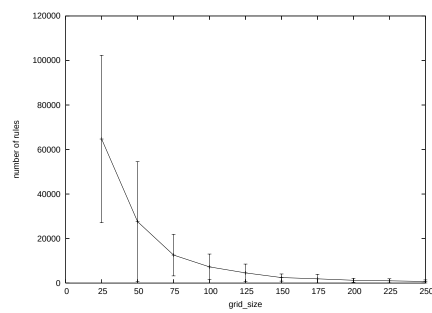

Fig. 3. rules vs. grid_size

Figures 2 and 3 present the execution time and the number of discovered rules when varying the *grid_size* parameter in the range $\langle 25 \div 250 \rangle$ for a fixed number of 300 moving objects and *minsup* $= 0.03$. As can be easily seen, low values of the *grid_size* parameter produce large numbers of movement rules. We also observe a significant variation in the averaged results over 30 database instances. For small values of *grid_size* discovered rules are too detailed and do not generalize well for prediction of movement of other objects. Interestingly, the execution time and the number of discovered movement rules drop significantly for larger values of *grid_size*.

The impact of the number of moving objects on the execution time and the number of discovered movement rules is depicted in Figures 4 and 5 (*grid_size* $=$ 250, *minsup* $= 0.01$). A linear dependence of the execution time on the number of moving objects guarantees scalability of the proposed solution. The decrease of

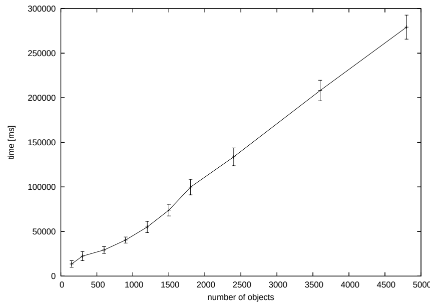

Fig. 4. time vs. number of objects

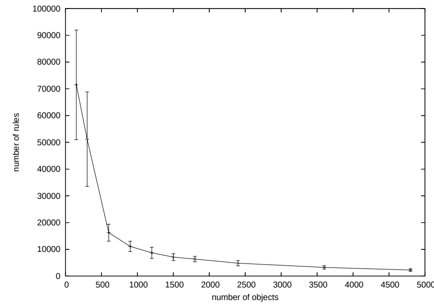

Fig. 5. rules vs. number of objects

the number of discovered movement rules is caused by the increase in the absolute value of *minsup*, because adding new moving objects increases the minimum number of objects that have to traverse an edge to make it frequent. New objects are spread over the movement area uniformly, effectively decreasing support counts of most edges.

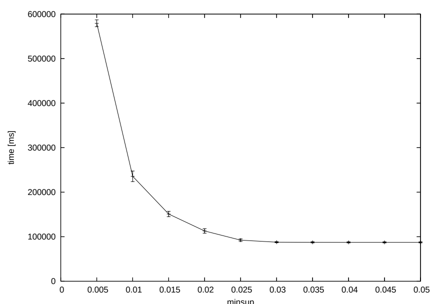

Fig. 6. time vs. *minsup*

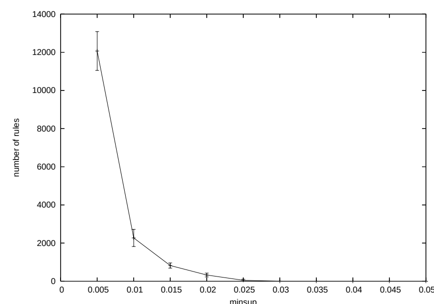

Fig. 7. rules vs. *minsup*

The influence of the varying *minsup* parameter on the execution time and the number of discovered movement rules is presented in Figures 6 and 7 ($grid\_size$ = 250, 4800 moving objects). Starting from $minsup = 0.025$ the execution time remains almost constant, because the number of discovered movement rules falls below 50 for larger values of *minsup*.

We have also conducted experiments on predicting the location of an object for different movement rule matching strategies. We refrain from presenting the results of these experiments due to their unreliable nature. The results were skewed by the properties of the synthetic data. The uniform distribution of moving objects across the examined area caused no evident differences between employed matching strategies.

6 Conclusions

In this paper we have presented a new model of movement rules discovered from moving object data. Movement rules provide a simplification and generalization of a large set of moving objects and allow for predicting the location of a moving object. This paper intends to open the research in the field of mining movement rules. Our future work agenda includes experimental verification and comparison of the proposed movement rule matching strategies using a real world dataset, extending movement rule framework to handle temporal aspects, and combining movement rules with spatial data (location of gas stations, shops, advertisement billboards) to make informed decisions based on the intensity of traffic.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB'94, September 12-15, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.
2. R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE'95, Taipei, Taiwan, March 6-10*, pages 3–14. IEEE Computer Society, 1995.
3. T. Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6(2):153–180, 2002.
4. M. Ester, A. Frommelt, H.-P. Kriegel, and J. Sander. Spatial data mining: Database primitives, algorithms and efficient dbms support. *Data Mininig and Knowledge Discovery*, 4(2/3):193–216, 2000.
5. J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. In *ICDE'99, Sydney, Austrialia, 23-26 March*, pages 106–115. IEEE Computer Society, 1999.
6. K. Koperski and J. Han. Discovery of spatial association rules in geographic databases. In *SSD'95, Portland, Maine, August 6-9*, pages 47–66. Springer, 1995.
7. Y. Li, J. Han, and J. Yang. Clustering moving objects. In *ACM SIGKDD'04, Seattle, Washington, USA, August 22-25*, pages 617–622. ACM, 2004.
8. N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. Cheung. Mining, indexing, and querying historical spatiotemporal data. In *ACM SIGKDD'04, Seattle, Washington, USA, August 22-25*, pages 236–245. ACM, 2004.
9. Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and indexing of moving objects with unknown motion patterns. In *ACM SIGMOD'04, Paris, France, June 13-18*, pages 611–622. ACM, 2004.
10. G. Trajcevski, O. Wolfson, B. Xu, and P. Nelson. Real-time traffic updates in moving objects databases. In *DEXA'02, Aix-en-Provence, France, 2-6 September*, pages 698–704. IEEE Computer Society, 2002.
11. O. Wolfson and H. Yin. Accuracy and resource concumption in tracking and location prediction. In *SSTD'03, Santorini, Greece*, pages 325–343. Springer, 2003.
12. B. Xu and O. Wolfson. Time-series prediction with applications to traffic and moving objects databases. In *MobiDE 2003, September 19, 2003, San Diego, California, USA*, pages 56–60. ACM, 2003.
13. J. Yang and M. Hu. Trajpattern: Mining sequential patterns from imprecise trajectories of mobile objects. In *EDBT'06, Munich, Germany, March 26-31*, pages 664–681. Springer, 2006.