

Key techniques for predicting the uncertain trajectories of moving objects with dynamic environment awareness

Shaojie QIAO^{1*}, Xian WANG¹, Lu'an TANG², Liangxu LIU³, Xun GONG¹

1. School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

2. Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana 61801, USA

3. School of Electronics and Information Engineering, Ningbo University of Technology, Ningbo 315016, China

Abstract: Emerging technologies of wireless and mobile communication enable people to accumulate a large volume of time-stamped locations, which appear in the form of a continuous moving object trajectory. How to accurately predict the uncertain mobility of objects becomes an important and challenging problem. Existing algorithms for trajectory prediction in moving objects databases mainly focus on identifying frequent trajectory patterns, and do not take account of the effect of essential dynamic environmental factors. In this study, a general schema for predicting uncertain trajectories of moving objects with dynamic environment awareness is presented, and the key techniques in trajectory prediction are addressed in detail. In order to accurately predict the trajectories, a trajectory prediction algorithm based on continuous time Bayesian networks (CTBNs) is improved and applied, which takes dynamic environmental factors into full consideration. Experiments conducted on synthetic trajectory data verify the effectiveness of the improved algorithm, which also guarantees the time performance as well.

Key words: trajectory prediction; moving objects databases; dynamic environmental factors; continuous time Bayesian networks

© 2011 JMT. All rights reserved.

1. Introduction

The rapid development of wireless communication and location-awareness techniques has gradually improved the capability of tracing the instant spatio-temporal position of continuously moving objects, which promotes the study of intelligent traffic control, intelligent navigation, military command, and fleet management. This emerging environment presents a lot of challenges in moving objects databases. How to accurately predict the uncertain trajectory of moving objects becomes an essential problem. In addition, the mobile environment of objects often changes dynamically. Thus, we cannot only predict the trajectory (or path) of moving objects depending on static environment, i.e., constrained traffic networks. It is important to take into full consideration the following complex dynamic environmental factors: (1) the change of moving speed and direction, for example, running at a high speed or slowing down to turn over; (2) traffic situations, for instance,

traffic lights or traffic jam; and, (3) weather conditions, e.g., fog, snow, or rainstorm.

Dynamic environmental factors include the macro traffic network environment, internal natural environment, and external natural environment that can change dynamically in form and state with time. The main characteristics of dynamic environmental factors are instability and diversification. In this study, dynamic environment awareness is defined to model these dynamically changed environmental factors affecting the location of moving objects, as well as to quantify and formalize these environmental factors that are hard to express. It becomes a research focus to effectively and efficiently predict the uncertain trajectories of moving objects by taking the dynamic environmental factors into full consideration.

It is a difficult and challenging problem to predict the trajectories of moving objects due to the following reasons:

(1) It is not feasible to apply traditional data mining approaches to discovering frequent paths of moving objects, because the trajectory data stream of continuously moving objects obtained by global positioning systems (GPS) is unstable and changes dynamically. We need to develop steady and scalable trajectory mining approaches.

Received Aug. 28, 2011; revision accepted Sep. 19, 2011

*Corresponding author. Tel.: +86-28-86466426

Email: sjqiao@home.swjtu.edu.cn (S.J. QIAO)

© 2011 JMT. All rights reserved

doi: 10.3969/j.issn.2095-087X.2011.03.008

(2) It is necessary to efficiently obtain the positions of moving objects, as the predicting results after a long period of time are useless due to the real-time requirement.

(3) Most existing predicting algorithms depend on several input parameters, which can greatly affect the prediction accuracy. We need to take into consideration important external factors and the knowledge of domain experts.

(4) If the predicting algorithm is not well designed, the computation cost increases in an exponential order as the number of moving objects grows quickly.

The main contribution of this paper includes the following:

(1) We present a generic schema of trajectory prediction with dynamic environment awareness and introduce the key techniques to solve problems in this schema. The proposed schema is appropriate to line as well as region objects besides the point objects. And the proposed index structure can greatly improve the efficiency of predicting.

(2) We identify four important dynamic environmental factors that can exert influence on the motion states of objects.

(3) We apply a trajectory prediction algorithm based on continuous time Bayesian networks (CTBNs), namely PutMode, which takes into full consideration the effect of dynamic environmental factors. And we improve the PutMode algorithm by quantifying the traffic situation and integrating it into the linear function.

(4) Experiments are conducted on synthetic trajectory data generated on real-world maps to demonstrate the effectiveness and efficiency of PutMode by comparing it with the Naive algorithm.

2. Related work

Existing works relevant to trajectory prediction mainly focus on discovering frequent trajectory patterns, indexing and querying continuously moving objects, uncertainty management of moving objects, and trajectory prediction. The typical works are presented as follows.

- *Discovering frequent trajectory patterns.* Tao et al. [1] proposed a schema to monitor and index moving objects and developed a frequent trajectory patterns mining algorithm called STP-tree, which can query objects in an effective manner. In order to discover frequent trajectory patterns, Morzy [2] proposed an efficient frequent trajectory mining algorithm, namely Traj-PrefixSpan, which uses a probabilistic model to compute the most possible location of objects based on moving rules. However, the typical works introduced in [3-5] need to scan trajectory databases for several times, which is time-consuming.

- *Indexing and querying moving objects.* It is an essential phase to index moving objects in trajectory prediction. The indexing techniques can be classified into two categories: indexing historical trajectories, and indexing current and future trajectories [6]. Ding et al. [7] proposed the UTR-tree to index uncertain trajectories of moving objects in constrained road networks. TPR-tree [8] is a widely used indexing method, which models the motion state of moving objects in linear functions and uses the linear interpolation method to depict the complex moving behavior. Another work is by Pfoser and Jensen [9]. They proposed to transform 2-dimensional moving curves into 1-dimensional set of segments, and then map these segments into the corresponding trajectory. Li et al. [10] proposed an improved UTR-tree, which indexes static constrained road networks by R*-tree and employs R*-tree and Hash array to manage dynamically changed location information. As for indexing the current and future location of moving objects, there are several works based on R-tree. R^R-tree [11] is a typical structure, which contains two operations, including the update operation and the grouping operation. Tao et al. [12] proposed the TPR*-tree index structure that takes into full consideration the unique features of dynamic objects by a set of improved construction algorithms. ANR-tree [13] is a recently proposed index updating approach, which uses adaptive unit to group neighboring objects with similar movement patterns. The main problem in indexing moving objects is that the update cost of index is high. R^R-tree and TPR*-tree are alternative solutions for efficient indexing, but these approaches are constrained by the fixed networks. We have to improve these algorithms to adapt to unfixed networks or propose generic indexing structures. The newly proposed querying processing models for moving objects are detailedly discussed in [14-16].

- *Uncertainty management of moving objects.* There are several uncertainty factors that can affect the position of moving objects, such as, uncertain location, appearing or disappearing, and uncertain moving behaviors. Currently, researchers mainly concentrated on managing the uncertain location of moving objects. Trajcevski et al. [17] proposed the probability density functions to describe the uncertainty associated with the location, and identified syntactic variants for systematic incorporation of uncertainty in the continuous nearest-neighbor queries. Pfoser et al. [18] used the error ellipse to represent the positions of moving-point objects and quantified the imprecisions in the proposed representation. Ding et al. [19] employed the distance threshold that affects the update of locations, the speed threshold, and the speed in some time point to compute the most possible positions in a $POS \times T$ plane, in which

POS represents the position dimension and T is the time dimension.

● *Trajectory prediction of moving objects.* Recently, this research topic has attracted a number of researchers. Jeung et al. [20] proposed a hybrid trajectory prediction model called TPT-tree based on trajectory patterns and moving functions, which employs the node update strategy of Signature tree, and uses the premise similarity measures, forward query processing, and backward query processing to predict the frequent trajectory patterns in an accurate fashion. Parker et al. [21] proposed an efficient and effective mobility destination prediction approach based on distinct axioms that can be employed to measure the difference between the predicted itineraries and the real ones. Guo et al. [22] proposed novel uncertain trajectory measures, and used them to discover the uncertain trajectory patterns. Song et al. [23] introduced the limits of predictability in human mobility in *Science*. They found a 93% predictability in user mobility across the whole user base by calculating the entropy of each individual's trajectory. In addition, they proved that users' regular mobility path is independent of the distance users cover. Jeung et al. [24] proposed a network mobility model to capture the turning patterns at junctions and the travel speeds on road segments of objects. However, the proposed method is only applicable to constraint road networks. Chen et al. [25] presented a personal

route prediction system using a probabilistic model based on historical trajectory data. This system employed Markov model to obtain the continuous route patterns using frequent trajectory pattern mining method with low time cost. These existing trajectory prediction algorithms often overlook the effect of some important and complex dynamic environmental factors, such as speed, moving direction, and users' travel experiences. The existing methods lack generality as moving objects may follow the dynamic motion patterns in real life. In [26], Qiao et al. applied CTBNs to predict the uncertain trajectory of moving objects. However, they did not index the objects before predicting.

3. Schema of trajectory prediction with dynamic environment awareness

In this study, we aim to solve the following important problems of uncertain trajectory prediction of objects with dynamic environment awareness:

- (1) designing a generic schema of uncertain management, which is applicable to various kinds of moving objects and can automatically determine different management strategies for the corresponding objects;
- (2) developing new spatio-temporal index structures, which are efficiently updated and are not constrained by the fixed networks as well; and,

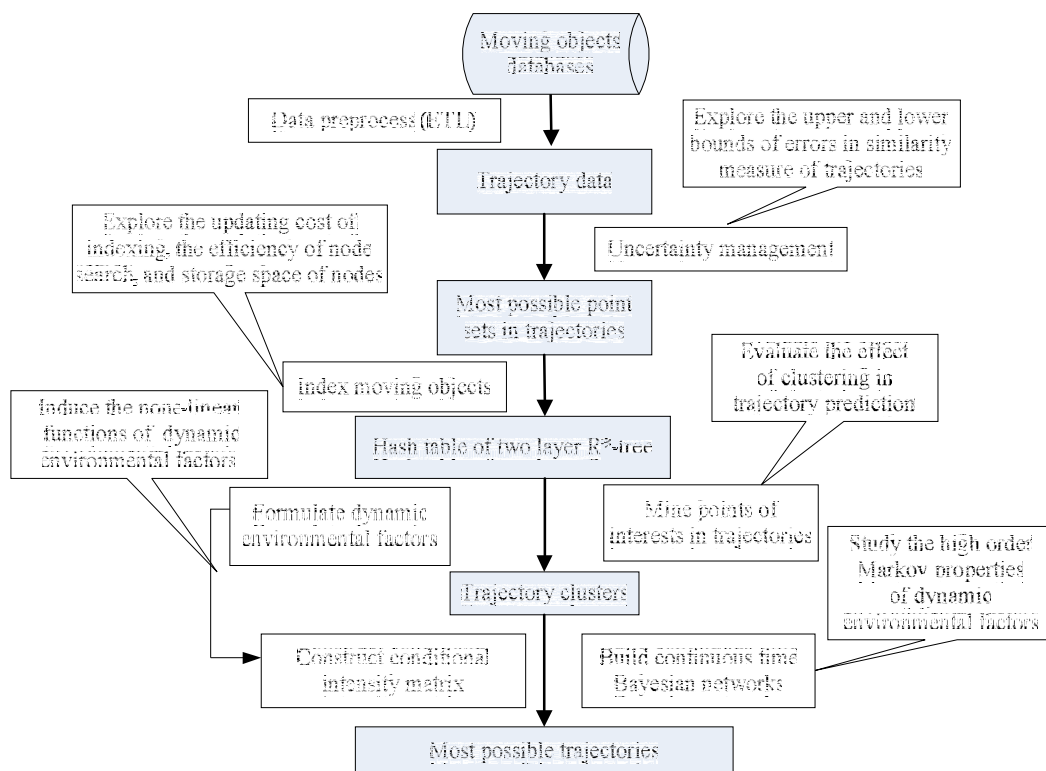


Fig. 1 Schema of trajectory prediction under dynamic environment

(3) predicting uncertain trajectories of moving objects by including a full consideration of the dynamic environmental factors in an effective and efficient manner.

We propose the schema as shown in Fig. 1 to handle these three problems, where the annotations in the rectangles represent the strategies to solve the corresponding problems.

4. Key techniques for predicting uncertain trajectories

For solving the key problems addressed in Section 3, we use the following strategies in this study.

4.1. Managing the uncertain motion of objects

We use the following models corresponding to different kinds of moving objects to depict the possible motion curve of objects. In general, the objects are classified into three categories: point, line, and region objects. We design different trajectory volumes as shown in Fig. 2 to approximate the possible motion curve of different kinds of objects.

Here we take point objects as an example to introduce the theory of managing uncertain motion [26].

For each point (x, y, t) along a trajectory, the uncertain area is a disk with radius r . A possible motion curve P_{tr} of trajectory tr can be expressed by a function $f_{P_{tr}}: t \rightarrow r^2$ for any time $t \in [t_1, t_n]$. Assuming that an object moves at a constant speed v , the 3D point $(f_{P_{tr}}, t)$ inside an uncertain area of the expected location at a given time instant t satisfies

$$(x - (x_i + v_i^x t))^2 + (y - (y_i + v_i^y t))^2 \leq r^2. \quad (1)$$

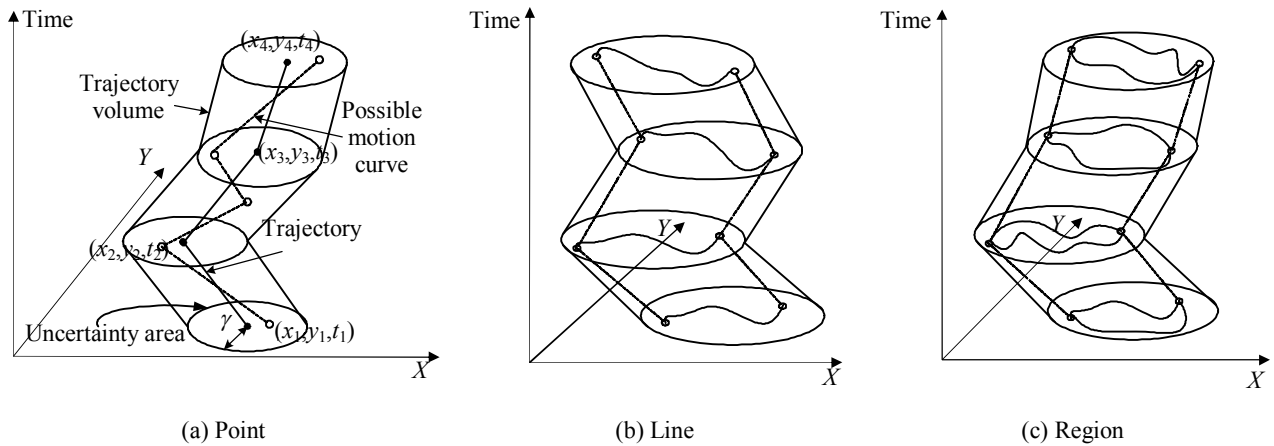


Fig. 2 Trajectory volumes of different kinds of moving objects

4.2. Indexing the current and future positions of moving objects

We design a new index structure called UMO-tree with low updating cost as shown in Fig. 3. UMO-tree is partitioned into two layers, where the upper layer R_{road} is a two dimensional R*-tree that is used to index the streets in static road networks, and the lower layer R*-tree that indexes the moving objects denoted by R_{object} in the leaf node of R_{road} . In contrast to traditional R*-tree indexing structures, the leaf nodes in R_{road} contain the road information based on their spatial positions, which can help reduce the number of nodes, the dead space for storing nodes, and the overlapped area. H_{data} is a Hash table that is used to store the instant information of objects, e.g., moving speed and location.

The advantage of the proposed index structure is that it uses R*-tree to index the static road information with less change. In addition, UMO-tree uses R*-tree and Hash table to manage the dynamic moving information in order to efficiently index the current and future positions of objects.

4.3. Predicting uncertain trajectories of moving objects with dynamic environment awareness

We have to quantify the dynamic environmental factors including the location, speed, direction, and traffic situation. These dynamic environmental factors are formalized by the following methods:

- Location: it is quantified by the identification number of each street.
- Speed: it is divided into several levels according to the speed criterion. For example, it is represented by $\{s_1, s_2, \dots, s_5\}$ for cars.

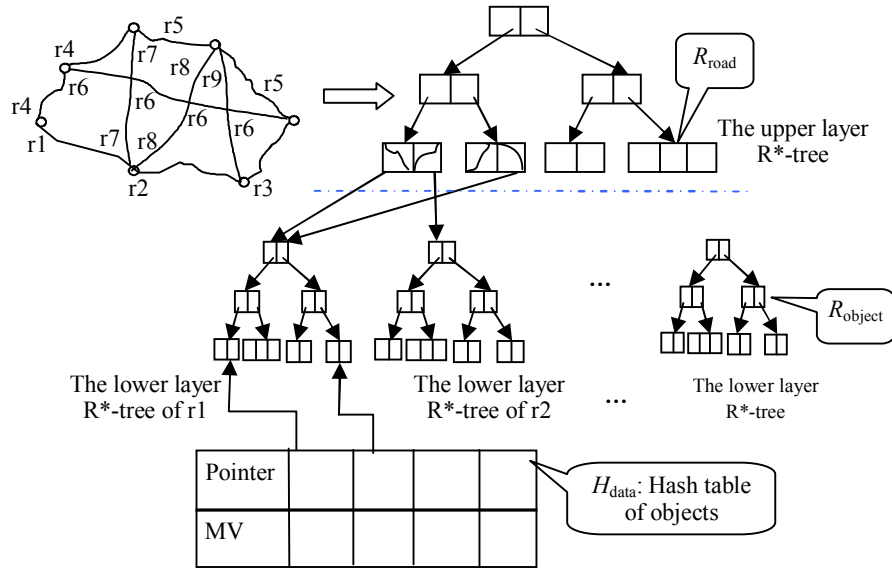


Fig. 3 Two layer indexing structure based on R*-tree

c) Direction: objects often move forward, turn left or right, and move back, which can greatly affect the motion of objects. In addition, the direction contains some second-order information which takes into consideration specific situations. For instance, if we know the previous street, we could roughly estimate the direction at the current street. The detail is presented in [26].

d) Traffic situation: it contains the cases of traffic lights and traffic jam. As for traffic lights, we divide them into four cases, i.e., green in South and North, red in East and West; yellow in South and North, red in East and West; green in East and West, red in South and North; yellow in East and West, red in South and North. As for traffic jam, we classify the traffic situation into three levels: no jam, jam, and serious jam.

Then, we use the numerical approach proposed in [26] to quantify the above dynamic environmental factors, and design a linear function to model these factors. In [26], the authors have verified that the linear function can be employed to map the motion state of objects to one unique value that can be used to construct the conditional intensity matrix (CIM) of trajectories.

In this study, we use continuous time Bayesian networks (CTBNs) to calculate the state transition probability changing continuously over time. It is reasonable to depict a trajectory as a CTBN, because the dynamic environmental factors in the same state have the property of continuous time. Due to space limitation, please refer to [27] for the detail of CTBNs.

The trajectory prediction algorithm is borrowed from [26] and contains the following phases:

(1) Use the linear function $F(X)$ to compute the state values in each trajectory by integrating these four dynamic environmental factors, which are denoted by a

four-tuple vector $X = \{(x_1, x_2, x_3, x_4) \mid x_i \in [1, 4]\}$ represents location, speed, direction, and traffic situation, where the finite domain of each variable is $\text{dom}(x_i)$.

(2) Create the conditional intensity matrix based on the given state values. Let L be the combined state of four variables with regard to each dynamic environmental factor. The domain of L is $f(L) = \{l_1, l_2, \dots, l_n\}$, and L is determined by its previous state $L'(t)$. A CIM of a trajectory is defined as the following matrix:

$$\mathbf{M}_{L/L'} = \begin{bmatrix} -p_{l_1}^l(L') & p_{l_{12}}^l(L') & \dots & p_{l_{1n}}^l(L') \\ p_{l_{21}}^l(L') & -p_{l_2}^l(L') & \dots & p_{l_{2n}}^l(L') \\ \vdots & \vdots & \ddots & \vdots \\ p_{l_{n1}}^l(L') & p_{l_{n2}}^l(L') & \dots & -p_{l_n}^l(L') \end{bmatrix}, \quad (2)$$

where the element $p_{ij}^l(L')$ in a CIM represents the instantaneous probability of state transition from l_i to l_j . The transition probability can be calculated by

$$p_{ij}^l(L') = f_{ij}^l(L') / \sum_{i \neq j} f_{ij}^l(L'), \quad (3)$$

in which $f_{ij}^l(L')$ is the occurrence frequency of visiting a street in a trajectory where the state of an object changes from l_i to l_j , and the frequency is calculated based on the historical data.

(3) Finally, compute the visiting probability of each trajectory by multiplying the state values in a trajectory, and find the possible trajectories. The algorithm is given below [26]. The trajectory prediction algorithm contains three essential phases [26]:

i) Initialize a null state chain C_0 and add s_0 to it (lines 1–2);

- ii) Find the state chain whose product of the previous transition probabilities is greater than ε (lines 3–9);
- iii) Compute the xy -coordinate of each state and the time interval between two states in each trajectory and finally output the predicted results (lines 10–15).

Algorithm 1 Trajectory prediction based on CTBNs

Input: A CIM M , an initial state s_0 of a object, a probability threshold ε , a radius r of a trajectory volume.

Output: a set of possible trajectories $T = \{T_1, T_2, \dots, T_n\}$.

1. $N \leftarrow \emptyset$; $C_0 \leftarrow \emptyset$;
 2. $C_p.put(s_0)$; $C_0.prob = 1$; $N.put(C_0)$;
 3. **for** each $C_p \in N$ **do**
 4. $\{i \leftarrow C_p.lastIndex();\}$
 5. **for** (each state j in the i th row of M & $M(i,j) \neq 0$)
 6. **if** ($C_p.prob * M(i,j) \geq \varepsilon$) **then**
 7. $\{C_p = C_p; C_p.put(j);\}$
 8. $C_p.prob \leftarrow C_p.prob * M(i,j)$;
 9. $N.put(C_p)$;
 10. **for** each $C_p \in N$ **do**
 11. $\{T_p = CalTraj(C_p);\}$
 12. **for** each state κ in C_p **do**
 13. $\{\tau \leftarrow CalTime(tr);\}$
 14. $T_p.put(t_s, t_e, \tau);\}$
 15. **Output**(T_p);
-

5. Experiments

In order to evaluate the effectiveness and efficiency of the trajectory prediction algorithm based on CTBNs that takes the dynamic environmental factors into full consideration, we conducted experiments to compare the proposed PutMode [26] with the Naive prediction algorithm (Naive for short) that does not take into account the effect of the presented four dynamic environmental factors. We have improved PutMode algorithm by quantifying the traffic situation and integrating it into the linear function. In addition, by extensive experiments, we find that the improved PutMode algorithm is better than the old-fashioned one.

Both algorithms were implemented in Java and the experiments were performed on an AMD Athlon 5000+, 3.0 GHz CPU with 2.0 GHz of main memory, running on Ubuntu Linux 8.04 operating system.

All experiments were run on the data set that is generated by Brinkhoff's network-based generator [28]. The data were generated with real-world maps by the network-based spatio-temporal data generating approach [29]. The data set was obtained from the New York state map consisting of 3 287 nodes and 3 997 edges. The probability threshold ε and the radius r are important input parameters in Algorithm 1, we have to tune these parameters before conducting experiments. The detail of the parameter tuning process is available in [26].

5.1. Effectiveness comparison of trajectory prediction

The prediction accuracy is defined by [26]:

$$Accuracy = \frac{n}{N}, \quad (4)$$

where n is the number of the *hits* set of trajectories and N is the number of the real-world trajectories during a given time interval. The implication of *hits* is defined as follows [26].

For $t = (s_1, \tau_1) \in Tr$ and $t^* = (s_2, \tau_2) \in Tr^*$, where $\{s_1, s_2\} \in S$, $\{\tau_1, \tau_2\} \in \tau$, s is the set of streets and τ is the set of timestamps, t should satisfy the τ -containment [30] which means t is contained in t^* satisfying

- 1) $\forall k \in [0, n], s_{1,k} \subseteq s_{2,k}$,
- 2) $\forall k \subseteq [1, n], |\tau_{1,k} - \tau_{2,k}| \leq \tau$,

where $\tau_{*,k} = \sum_{j=i_{k-1}}^{i_k} \tau_{2,j}$, $\forall k \in [1, n]$.

In this set of experiments, we use the accuracy measure to compare the effectiveness of trajectory prediction between PutMode and Naive algorithms in the New York data set. The results are given in Fig. 4.

One can see from Fig. 4 that PutMode outperforms Naive in any case with an average gap of 10.3% as the number of moving objects grows. This is because PutMode takes account of the dynamic environmental factors including the current location, direction, speed, and traffic situation.

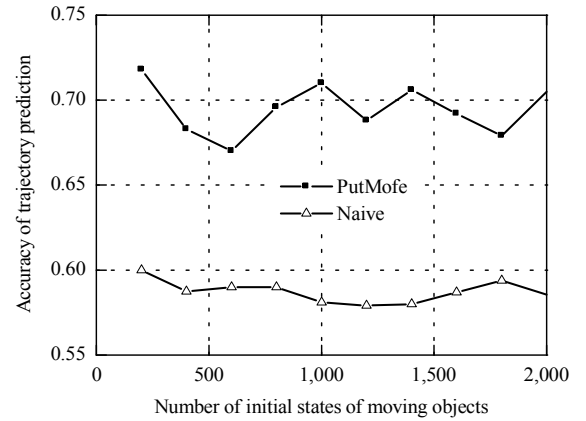


Fig. 4 Prediction accuracy comparison between PutMode and Naive algorithms

5.2. Efficiency comparison of trajectory prediction

In order to further verify the time performance of PutMode, we compare the runtime of trajectory prediction with the number of initial states of objects on the New York trajectory data set. The results are shown in Fig. 5.

From Fig. 5, we observe that Naive wins in all cases, because it only uses the transfer from one street to another to represent the state transition [26]. However, PutMode uses the linear function integrating four dynamic environmental factors to determine one state transformation that is time intensive. In addition, it is interesting to see that the prediction time of PutMode grows linearly with the number of initial states of objects. The reason is that the prediction time is determined by the number of state chains (see Algorithm 1 for detail), which in turn depends on the number of initial states of moving objects [26].

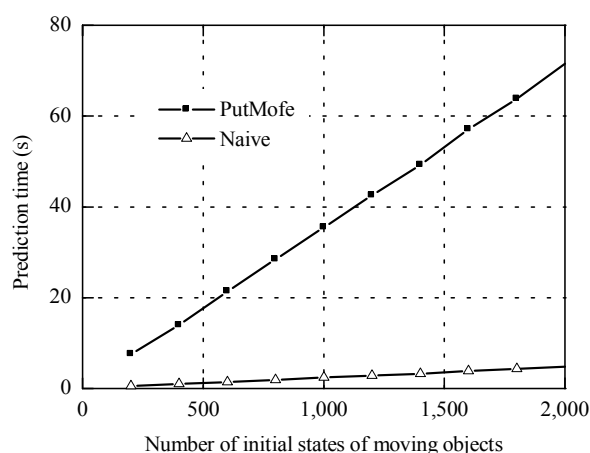


Fig. 5 Prediction time comparison between PutMode and Naive algorithms

6. Conclusion

In order to cope with the problem of effectively predicting the uncertain trajectories of moving objects, we have proposed a schema of trajectory prediction under a dynamic environment. We presented the key techniques for managing the uncertain motion of objects, indexing the current and future positions of moving objects, and predicting the moving trajectories. Then, we improved a trajectory prediction algorithm based on CTBNs by quantifying the traffic situation and integrating it into the linear function, and employed it to predict the uncertain trajectories of moving objects under a dynamic environment. Finally, we carried out experiments to estimate the accuracy and time performance of the introduced algorithm.

There are several challenges that we plan to study in the future. One is to improve the efficiency of the PutMode algorithm by reducing the useless states of moving objects. Another is to propose another accurate model to represent these dynamic environmental factors. Finally, we plan to apply the PutMode algorithm and the proposed schema to tracing and predicting human mobility or travel itineraries.

Acknowledgements

The research is supported by the National Natural Science Foundation of China (Nos.61100045, 61165013, 61003142, 60902023, and 61171096), the China Postdoctoral Science Foundation (Nos.20090461346, 201104697), the Youth Foundation for Humanities and Social Sciences of Ministry of Education of China (No.10YJCZH117), the Fundamental Research Funds for the Central Universities (Nos.SWJTU09CX035, SWJTU11ZT08), Zhejiang Provincial Natural Science Foundation of China (Nos.Y1100589, Y1080123), and the Natural Science Foundation of Ningbo, China (No.2011A610175).

References

- [1] Y. Tao, C. Faloutsos, D. Papadias, et al., Prediction and indexing of moving objects with unknown motion patterns, In: *Proceedings of SIGMOD 2004*, Paris, France, ACM, 2004: 611-622.
- [2] M. Morzy, Mining frequent trajectories of moving objects for location prediction, In: *Proceedings of MLDM 2007*, Leipzig, Germany, LNCS 4571, Springer-Verlag, 2007: 667-680.
- [3] F. Giannotti, M. Nanni, F. Pinelli, et al., Trajectory pattern mining, In: *Proceedings of KDD 2007*, San Jose, CA, ACM, 2007: 330-339.
- [4] G. Trajcevski, O. Wolfson, K. Hinrichs, et al., Managing uncertainty in moving objects databases, *ACM Transactions on Database Systems*, 2004, **29**(3): 463-507.
- [5] H.M.O. Mokhtar, J. Su, Universal trajectory queries for moving object databases, In: *Proceedings of MDM 2004*, Los Alamitos, USA, IEEE Computer Society, 2004: 133-144.
- [6] L. Liu, *Research on key technologies of spatial-temporal data Management in moving object database* [Dissertation], Shanghai: Dong Hua University, 2008.
- [7] Z. Ding, B. Yu, M. Li, et al., Indexing the Uncertain Trajectories of Network-constrained Moving Objects, *Computer Science*, 2008, **35**(3): 79-83.
- [8] S. Saltenis, C. Jensen, S. Leutenegger, M. Lopez, Indexing the positions of continuously moving objects, In: *Proceedings of SIGMOD 2000*, Dallas, Texas, USA, ACM, 2000: 331-342.
- [9] D. Pfoser, C.S. Jensen, Indexing of network constrained moving objects, In: *Proceedings of GIS 2003*, New Orleans, Louisiana, USA, ACM, 2003: 25-32.
- [10] H. Li, C. Tang, S. Qiao, et al., UTR*-tree: an uncertain trajectories model for indexing moving objects in constrained networks, *Journal of Sichuan University (Engineering Science Edition)*, 2010, **42**(2): 118-125.
- [11] L. Biveinis, S. Saltenis, C.S. Jensen, Main-memory operation buffering for efficient R-tree update, In: *Proceedings of VLDB 2007*, Vienna, Austria, VLDB Endowment, 2007: 591-602.
- [12] Y. Tao, D. Papadias, J. Sun, The TPR*-tree: an optimized spatio-temporal access method for predictive queries, In:

- Proceedings of VLDB 2003*, Berlin, Germany, VLDB Endowment, 2003: 790-801.
- [13] J. Chen, X. Meng, Update-efficient indexing of moving objects in road networks, *Geoinformatica*, 2009, **13**(4): 397-424.
 - [14] R.H. Güting, T. Behr, J. Xu, Efficient k-nearest neighbor search on moving object trajectories, *The VLDB Journal*, 2010, **19**(5): 687-714.
 - [15] Y. Gao, B. Zheng, G. Chen, et al., Efficient mutual nearest neighbor query processing for moving object trajectories, *Information Sciences*, 2010, **180**(11): 2176-2195.
 - [16] Z. Chen, H.T. Shen, X. Zhou, et al., Searching trajectories by locations: An efficiency study, In: *Proceedings of SIGMOD 2010*, Indianapolis, Indiana, USA, 2010: 255-266.
 - [17] G. Trajcevski, R. Tamassia, H. Ding, et al., Continuous probabilistic nearest-neighbor queries for uncertain trajectories, In: *Proceedings of EDBT 2009*, Saint Petersburg, Russia, ACM, 2009: 874-885.
 - [18] D. Pfoser, C.S. Jensen, Capturing the uncertainty of moving-objects representations, In: *Proceedings of SSD 1999*, Hong Kong, LNCS 1651, Springer-Verlag, 1999: 111-131.
 - [19] Z. Ding, X. Li, B. Yu, Indexing the historical, current, and future locations of network-constrained moving objects, *Journal of Software*, 2009, **20**(12): 3193-3204.
 - [20] H. Jeung, Q. Liu, H. Shen, et al., A hybrid prediction model for moving objects, In: *Proceedings of ICDE 2008*, Piscataway, USA, IEEE Computer Society, 2008: 70-79.
 - [21] V.S. Parker, J.G. Subrahmanian, Fast and accurate prediction of the destination of moving objects, In: *Proceedings of SUM 2009*, Washington DC, LNAI 5785, Springer-Verlag, 2009: 180-192.
 - [22] L. Guo, Z. Ding, Z. Hu, et al., Uncertain path prediction of moving objects on road networks, *Journal of Computer Research and Development*, 2010, **47**(1): 104-112.
 - [23] C. Song, Z. Qu, N. Blumm, et al., Limits of predictability in human mobility, *Science*, 2010, **327**(5968): 1018-1021.
 - [24] H. Jeung, M.L. Yiu, X. Zhou, et al., Path prediction and predictive range querying in road network databases, *The VLDB Journal*, 2010, **19**(4): 585-602.
 - [25] L. Chen, M. Lv, Q. Ye, et al., A personal route prediction system based on trajectory data mining, *Information Sciences*, 2011, **181**(7): 1264-1284.
 - [26] S. Qiao, C. Tang, H. Jin, et al., PutMode: prediction of uncertain trajectories in moving objects databases, *Applied Intelligence*, 2010, **33**(3): 370-386.
 - [27] U. Nodelman, C.R. Shelton, D. Koller, Learning continuous time Bayesian networks, In: *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, San Francisco: Morgan Kaufmann, 2003: 451-458.
 - [28] T. Brinkhoff, Network-based generator of moving objects, <http://iapg.jade-hs.de/personen/brinkhoff/generator/>, April 19, 2005.
 - [29] T. Brinkhoff, A framework for generating network-based moving objects, *Geoinformatica*, 2002, **6**(2): 153-180.
 - [30] F. Giannotti, M. Nanni, D. Pedreschi, Efficient mining of temporally annotated sequences, In: *Proceedings of the 6th SIAM International Conference on Data Mining*, SIAM, Bethesda, 2006: 346-357.

(Editor: Dongju CHEN)