KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 10, NO. 7, Jul. 2016 3150

Vehicle trajectory prediction based on Hidden Markov Model

Ning Ye [1,2], Yingya Zhang [2], Ruchuan Wang [1,2,3] and Reza Malekian[4]

[1] College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China
[2] Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing , 210003, China
[3] Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education,Nanjing University of Post and Telecommunications, Nanjing ,210003,China
[4]Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, 0002, South Africa
[e-mail: reza.malekian@ieee.org]
*Corresponding author: Reza Malekian

Received December 20, 2015; revised March 21, 2016; revised May 1, 2016; accepted May 31, 2016; published July 31, 2016

Abstract

In Intelligent Transportation Systems (ITS), logistics distribution and mobile e-commerce, the real-time, accurate and reliable vehicle trajectory prediction has significant application value. Vehicle trajectory prediction can not only provide accurate location-based services, but also can monitor and predict traffic situation in advance, and then further recommend the optimal route for users. In this paper, firstly, we mine the double layers of hidden states of vehicle historical trajectories, and then determine the parameters of HMM (hidden Markov model) by historical data. Secondly, we adopt Viterbi algorithm to seek the double layers hidden states sequences corresponding to the just driven trajectory. Finally, we propose a new algorithm (DHMTP) for vehicle trajectory prediction based on the hidden Markov model of double layers hidden states, and predict the nearest neighbor unit of location information of the next k stages. The experimental results demonstrate that the prediction accuracy of the proposed algorithm is increased by 18.3% compared with TPMO algorithm and increased by 23.1% compared with Naive algorithm in aspect of predicting the next k phases' trajectories, especially when traffic flow is greater, such as this time from weekday morning to evening. Moreover, the time performance of DHMTP algorithm is also clearly improved compared with TPMO algorithm.

Keywords: Trajectory prediction, hidden Markov model (HMM), double layers hidden states, the nearest neighbor unit

This work is support by National Natural Science Foundation of P. R. China (Grant No.61572260, 61373017 and 61572261), Peak of Six Major Talent in Jiangsu Province(Grant No.2010DZXX026), China Postdoctoral Science Foundation(Grant No.2014M560440), Jiangsu Planned Projects for Postdoctoral Research Funds (Grant No.1302055C), Scientific & Technological Support Project of Jiangsu Province(Grant No. BE2015702), Natural Science Foundation of Jiangsu Province(Grant No. BK20130882).

http://dx.doi.org/10.3837/tiis.2016.07.016 **ISSN : 1976-7277**

1. Introduction

As the rapid development of global location technology and mobile communication technology, mobile handheld devices such as mobile phones, PDA and mobile navigation systems are becoming more and more popular. In many important application fields, such as intelligent transportation system (ITS), intelligent navigation, logistics distribution and mobile e-commerce, users all need to query and analyze the track position information of vehicles. Therefore, Location Based Service (LBS) has been a hot issue of research that has been studied by domestic and foreign scholars. Meanwhile, vehicle trajectory prediction has gradually become one of the hot issues in the research [1]. At present, the digital cities' public transport vehicles, taxis and other vehicles are equipped with GPS and vehicle navigation equipment, which can be used to collect vehicle location information during different time periods, and connect, constitute a complete trajectory in a timed sequence, and then mine its dynamic trajectory. In general, the mobile vehicle periodically sends its position information to the central server [2-5]. However, during the time of two positioning information transmitted, the specific position information and the moving trajectory of the moving vehicle are not known. In addition, the diversification of the moving vehicle environment also makes the problem more complex. How to accurately predict the location information of the driving vehicle is a difficult problem needed to be solved collectively [6]. There are already some research results, such as the clustering of moving objects, the anomaly detection, the location and the prediction of movement trend, where the driving vehicle position prediction technology is continuously improved, but because of the theory and technology of the immature, most models can't be well adapted to the needs of moving vehicle position prediction [7-10].

In Section 2, we review the previous work in vehicle trajectory prediction. Section 3 introduces several main concepts, the framework and working principle of the model are given and then related issues of hidden Markov model are described. We build the HMM associated with vehicle trajectory prediction and present DHMTP algorithm to predicate the nearest neighbor unit of location information of the next k stages in Section 4. Then we illustrate experimental results in Section 5. Finally, section 6 concludes the paper.

2. Related Work

In recent years, the vehicle trajectory prediction has become a hot research topic and these researchers proposed a lot of methods, which are mainly divided into two categories. (1) Mine frequent patterns of trajectories: through mining frequent patterns to identify typical movement patterns [11]. Morzy [12] proposed a new algorithm combining prefix tree (PrefixSpan) and frequent pattern mining (FP-tree) to mine dynamic movement patterns of the moving objects, but the cost of time spending on constructing the prefix tree and FP-tree is very high. Most trajectory prediction methods are based on geographical characteristics of trajectories, while Ying [13] combined with the semantic features and spatial location of trajectories to predict the next location information. But, the shortage of this method is higher cost of calculating Semantic Score for each candidate path. (2) Establish the model of moving vehicles to achieve location prediction. For example, considering the vehicle's movement speed, motion direction and other factors, fitting vehicle's motion function achieves the purpose of prediction.

Qiao [14] proposed HMTP algorithm based on Hidden Markov Model, and extract hidden states and observations states from large amount of trajectory data, according to the different types of trajectories, and then adaptively predict optimal trajectory. Moreover, by analyzing the disadvantages of HMTP, a self-adaptive parameter selection algorithm called HMTP∗ is proposed [15], which captures the parameters necessary for real-world scenarios in terms of objects with dynamically changing speed. Asahara et al. [16] proposed a mixed Markov-chain model (MMM) that has an observable parameter like HMM, but the unobservable parameter of MMM is fixed during the state transition. Experimental results demonstrate that the prediction accuracy of the MMM method is higher than that of the Markov-chain model and HMMs. Gambs et al. [17] have recently extended mobility Markov chainmodel by taking into account the effect of visiting place before n number of states, which is more like a high-order Markov chain. [18] presented a modified Bayesian inference method (MBI) for the limited number of history trajectories and low prediction accuracy. MBI builds Markov Model to quantify the correlation of adjacent location and partition historical trajectories to obtain more accurate Markov Model. However, a disadvantage of this method is that predicting result must be existed trajectory in the database. In our paper, we choose TPMO algorithm and Naive algorithm as compared algorithms, which are also commonly used in the field of predicting vehicular trajectories. TPMO employs hot spot area mining algorithm to partition the trajectory set into different clusters, then predict vehicle trajectory, which proved to be an effective and efficient trajectory prediction method [19]. The Naive algorithm, more simply, computes the transition probability of points between cells and omits the moving direction of trajectories. It functions as follows: It partitions the digital map into cells; then it computes the occurrence probability of the objects; lastly, it employs the transition probability of the cells to obtain the most probable trajectory.

It can be seen that existed methods have some limitations and disadvantage, for these problems, we propose the vehicle trajectory prediction algorithm (DHMTP) based on hidden Markov model of double layers hidden states and define the concept of ε-neighbor. Hidden Markov model and k-step Markov process are constructed by mining double layer hidden states of the historical trajectories. Then the Viterbi algorithm is used to seek the double layers hidden states sequences corresponding to the vehicle's just driven trajectory. Finally, the k-step probability transition matrix is adopted to predict the next k stages' nearest neighbor sets in order to forecast traffic congestion situation in the future, and provide a more optimized scheme for users.

3. Basic Concepts and Model Framework

In order to better describe the algorithm proposed, we first need to describe the trajectory and its related properties. The real-time driving trajectory is acquired by GPS, and GPS can provide vehicles' location information. According to the road network model proposed by [1], these trajectories collected are projected into the road network mode and the distance between two adjacent grids is a unit. Specific work in the literature [1] is illustrated in detail.

Next, we will list the notations used in this paper. As the notations are too many, here we only list them without explanation. But Specific meanings have been given when they emerge in this paper.

Table 1. The notations used in this paper

$TD=\{Traj_1,Traj_2,...,Traj_r\}$	$Traj=\{P_1,P_2,...,P_d\}$	$P_i=\{(x_i,y_i)\}$
$N_\varepsilon(S_i)=\{U \mid dist(P_i,P)<\varepsilon\}$	$S=\{S_1,S_2,\dots,S_N\}$	$O=\{O_1,O_2,\dots,O_M\}$
$P(S_i)$	$a_{ij}=P(S_j \mid S_i),1\le i,j\le N$	S_j
$b_{ij}=P(O_i \mid S_j)$	$\lambda=(S,O,A,B,\pi)$	$\delta_t(j)$
$\varphi_t(j)$	q_t^*	$\delta_1(i)=\pi_i b_i(O_{k1})$
$b_i(O_{k_1})$	U_k	S_{ij}^2
$P_i=\frac{Num(S_i)}{\sum_{j=1}^{N}Num(S_j)}$	$P(S_i \mid S_{i-1})=\frac{Num(S_{i-1},S_i)}{Num(S_{i-1})}$	$P(O_j \mid S_i)=\frac{Num(S_i,O_j)}{Num(S_i)}$
$t_{n+1},t_{n+2},...,t_{n+k}$	$\varepsilon-NS$	$RT(PT_2,PT_2,...,PT_n)$
$N_\varepsilon(N_\varepsilon(P_1),N_\varepsilon(P_2),...,N_\varepsilon(P_n))$	$N_\varepsilon(P_n)$	P_{ij}^m

3.1 Related Definitions

Trajectory database TD stores a large number of vehicles' location information of different timestamps, and the ordered set of vehicles' location information of timestamp is called trajectory. $TD=\{Traj_1,Traj_2,...,Traj_r\}$ represents trajectory set and |TD|=r is the number of trajectories.

Definition 1. Trace sequence: $Traj=\{P_1,P_2,...,P_d\}$ is composed of some multi-dimensional discrete trajectory points according to the time sequence, where discrete trajectory point $P_i=\{(x_i,y_i)\}$, $i\in[1,d]$, (x_i,y_i) represents the coordinate point of the vehicle trajectory point P_i when occuring at timestamp t_i in the two-dimensional road network model.

Definition 2. Transition point: a transition point refers to the intersection's coordinate of the roads where the vehicle is driving. Features of the transition point: two adjacent transition points are on the same line; the transition point must be the intersection of roads.

According to the above road network model, we projected **Fig. 1** of the trajectory from Sanpailou campus of NJUPT to Gongningyuan Quater into model. To simplify the expression, we only express the starting point, the ending point and the transition point of the trajectory. The trajectory is represented in the two dimensional road network model as $traj=\{(2,1),(2,4),(1,4),(1,3),(0,3)\}$, as shown in **Fig. 2**. This trajectory is through Nanrui Road, Heilongjiang Road, Zhongfu Road, and Fujian Road, the starting point is (2,1), and the ending point is (0,3). (2,4) is the intersection (transition point) between Nanrui Road and Heilongjiang Road and (2,1) and (2,4) are in a straight line; (1,4) is the intersection (transition point) between Heilongjiang Road and Zhongfu Road and (2,4) and (1,4) are in a straight line; (1,3) is the intersection (transition point) between Zhongfu Road and Fujian Road and (1,4) and (1,3) are in a straight line, meanwhile (1,3) and (0,3) are also in a straight line.

Fig. 1. A driving trajectory from the East Gate of NJUPT to Gongningyuan Quarter

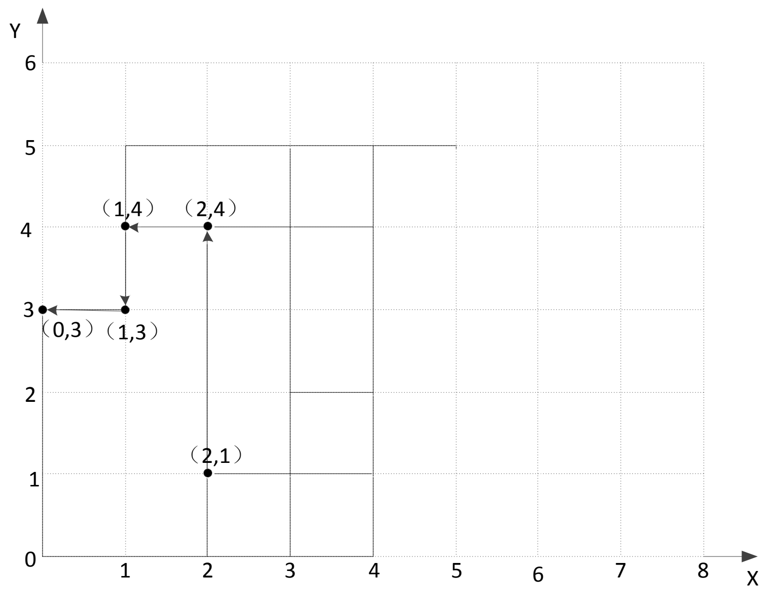

Fig. 2. A trajectory projected in road network model.

Definition3. (ε-neighbor): For a given neighbor threshold ε and each coordinate point of the trajectory $P_i = (x_i, y_i)$, if there is a neighbor unit U near it whose coordinate points is $P = (x, y)$, the ε-neighbor of P_i is denoted as $N_\varepsilon(S_i) = \{U \mid dist(P_i, P) < \varepsilon\}$,otherwise, the ε-neighbor set of P_i is denoted as corresponding road brand. Note that ε-neighbor set of trajectory coordinate points is given in definition3 that reflects the transition point marker information.

3.2 Trajectory prediction method and route recommendation system framework based on HMM

Trajectory prediction has a broad application prospect in various aspects, such as intelligent transportation system (ITS), the recommendation of tourism, advertising push, automatic setting of vehicle navigation and intelligent traffic management, where users need to query and analyze the vehicle trajectory location information [20-21]. The purposes of mining the information on the trajectory of moving vehicles are: in a short time, it can remind the driver of the security at the intersection under traffic congestion situation; in a long time, it can forecast the area that may occur traffic congestion, and so as to timely make a manner to guide the traffic and remind drivers to make a timely route adjustment. Therefore, we propose vehicle trajectory prediction method and the framework of route recommendation system based on HMM.

The system framework proposed in this paper is shown in **Fig. 3**, and its working principle is divided into four steps:

(1) Use the ETL technology to transform historical trajectories into vectors stored in the database;

(2) Excavate trajectories' double layers hidden states, and train HMM to determine the relative parameters of it. Input just-driven trajectories, and then use Viterbi algorithm to obtain the most likely double layers hidden states sequences corresponding to the known observation sequences. The initial probability matrix and the state transition matrix of HMM are used as the parameters of the k-step Markov chain to predict the location information of k stages in the future.

(3) Estimate congestion condition of each road and further recommend optimal routes to users, according to the future vehicle trajectory predicted.

(4) Update the database. Next with the increase of vehicles driving trajectories of the system, the number of data set for training HMM is also increasing, so the accuracy of the HMM model for the prediction of the vehicle future trajectories will be increased. The main function of this step is to update HMM according to the continuing inputted trajectories, which makes the accuracy rate of prediction increase.

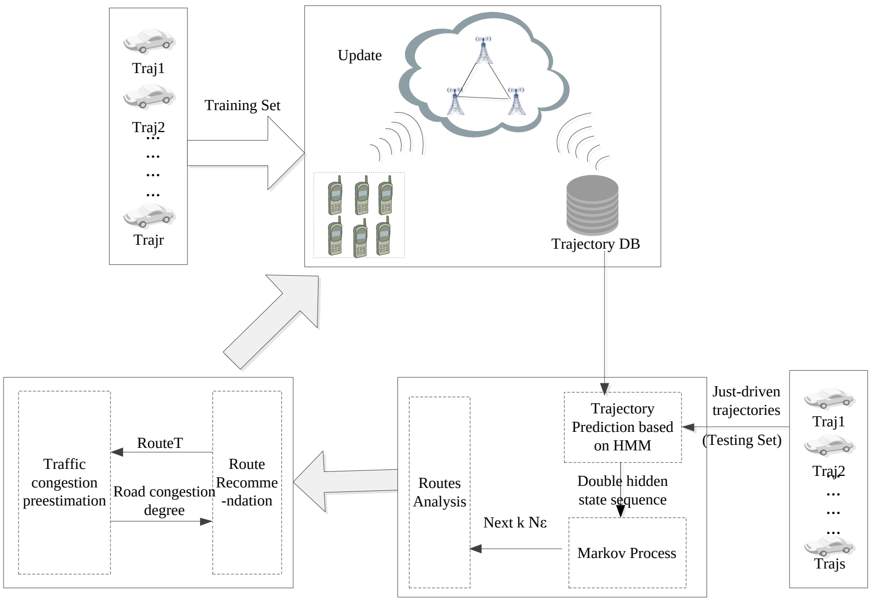

Fig. 3. Trajectory prediction and route recommendation system framework based on HMM

3.3 Related issues of hidden Markov model

In this section we briefly introduce the basic concept of hidden Markov model, the process of Viterbi algorithm to solve the hidden states sequence, k-step Markov process.

3.3.1 The description of HMM

Hidden Markov model is a statistical model used to describe the Markov process with unknown parameters. It is often used to look for some changing patterns in a period of time and analysis a system. The state which we hope to predict is hidden in the appearance, and is not what we observed, for example, by observing the appearance of algae to predict the change of weather. Here, there are two kinds of state, observed state (the state of algae), hidden state (the state of weather). The difficulty is to determine the implicit parameters of the process from the observable parameters, and then use these parameters to do further analysis.

(1) The hidden states $S=\{S_1,S_2,\ldots,S_N\}$, which meet the Markov property, where N indicates the number of hidden states.

(2) The observed states $O=\{O_1,O_2,\ldots,O_M\}$, associated with hidden states in the model, which can be obtained by direct observation (the number of observed states is not necessarily equal to the number of hidden states), where M is denoted as the number of observable states.

(3) The initial state probability matrix π represents hidden state probability matrix when in the initial timestamp t = 1. For example, when t = 1, $P(S_1)=\pi_1$, $P(S_2)=\pi_2$ *and* $P(S_3)=\pi_3$, initial state probability matrix π=[π_1, π_2, π_3].

(4) The transition probability matrix A of hidden states, describes the transition probability between hidden states in HMM, where $a_{ij}=P(S_j \mid S_i), 1\le i,j\le N$, indicates that in timestamp t+1, the probability of state S_j is a_{ij}, in the condition of state S_i in timestamp t.

(5) The Confusion Matrix B of observed states, describes the transition probability between the hidden states and observed states in HMM, where $b_{ij}=P(O_i \mid S_j)$ $(1\le i\le M, 1\le j\le N)$ represents what the probability of observed state O_i is in the condition of hidden state S_j in timestamp t.

Fig. 4 is a state transition diagram of HMM, where $S=\{S_1,S_2,S_3\}$ are hidden states, $O=\{O_1,O_2\}$ are observed states, where a represents the state transition probability of hidden states, and b represents the transition probability between the hidden states and the observed states.

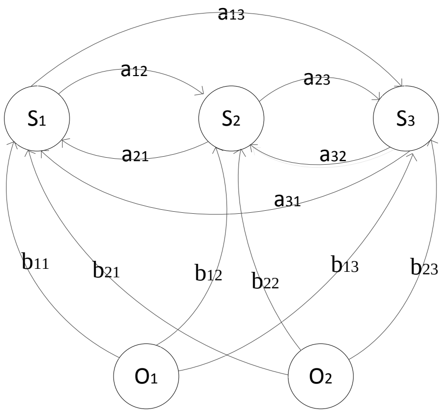

Fig. 4. State transition diagram

3.3.2 Viterbi Algorithm

Viterbi algorithm is a method used to solve HMM decoding, that is, given the observation sequence $O = O_{k_1}, O_{k_2}, \ldots, O_{k_t}$ and model parameter $\lambda = (S, O, A, B, \pi)$, how to find the optimal hidden state sequence RT, which meets the observation sequence, and the most commonly used algorithm is the Viterbi algorithm in this step. To solve this problem, we first introduce three symbols:

$\delta_t(j)$ represents the maximum probability that produces $O = O_{k_1}, O_{k_2}, \ldots, O_{k_t}$ and along the path $q_1, q_2, ..q_t$, when observed state is O_j at timestamp t.

$\varphi_t(j)$ represents a status value, which saves the last optimal state leading to the current state.

q_t^* saves the final choice of optimal hidden state at timestamp t.

The steps of Viterbi algorithm to solve the optimal hidden state sequence:

(1) Initialize the probability of each state producing O_{k_1} . t = 1, $\delta_1(i) = \pi_i b_i(O_{k_1}), 1 \le i \le N$, and $\varphi_1(i) = 0, 1 \le i \le N$ where $b_i(O_{k_1})$ represents the probability of producing the observed state O_{k_1} on the premise that hidden state is S_i .Therefore, the probability is denoted as $b_i(O_{k_1}) = b_{k_1 i}$.

(2) At the next each timestamp:

$$\delta_t(j) = \max_{1 \le i \le n}[\delta_{t-1}(i) a_{ij}] b_j(O_t) \tag{1}$$

$$\varphi_t(j) = \arg\max_{1 \le i \le n}[\delta_{t-1}(i) a_{ij}] \tag{2}$$

$$q_t^* = \varphi_{t+1}(q_{t+1}^*) \tag{3}$$

By the formula (1-3), we can solve the hidden state sequence corresponding to the observed state sequence, and the results are preserved in q_t^*.

3.3.3 Markov Process

If the vehicle is running on the road, only when it comes across the intersection of roads, its direction may change, and the change of direction is based on the direction of the crossed roads, which means when the vehicle traveling on a road, its next direction has nothing to do with just driven roads. Therefore, we can regard the vehicle trajectory as a Markov process. It is assumed that the former states are $X_0, X_1, ..., X_n$, and then the probability of x state that the trajectory's state is at timestamp n+1 can be obtained by the formula (4).

$$P(X_{n+1} = x_j \mid X_0 = x_0, X_1 = x_1, ..., X_n = x_i) = P(X_{n+1} = x \mid X_n = x_i) = P_{ij} \quad (4)$$

The k-step transition probability of Markov chain:

$$P_{ij}^{(k)} = P(X_{m+k} = x_j \mid X_m = x_i), m \geq 0, k \geq 1 \quad (5)$$

where $P_{ij}^{(k)}$ is k-step transition probability of Markov chain, and $P^{(k)} = P_{ij}^{(k)}$ is called k-step transition matrix of Markov chain, and $P_{ij}^{(k)} \geq 0, \sum_{\forall j} P_{ij}^{(k)} = 1, P(k) = P^k$.

That which road may be chosen in the transition point is based on just-driven trajectory points and the statistical data of historical trajectories. Obviously, the closer to the present timestamp the time of historical state is, the greater impact it may has on the next transition point, and early historical state can be neglected. So we can based on the experience, retain the closest to now k historical trajectory points, the rest points are neglected. And the probability of each nearest neighbor unit mark at next moment can be computed based on the Markov chain through weighted by (6).

$$P(X_n) = a_1 X_{n-1} P + a_2 X_{n-2} P^2 + ... + a_k X_{n-k} P^k \quad (6)$$

where the X_n is a one-dimensional state matrix, $P(X_n)$ is the probability of each nearest neighbor unit mark at next timestamp. $X_j, n-k \leq j \leq n-1$ represents the state at timestamp j, which is also a matrix of $1 \times n$, where the value of first row and j-th column is 1, others are 0. $a_1, a_2, ..., a_k$ are weights, which respectively represents the impact of the former the timestamp of 1,2,...k on what decision that the next timestamp's state may make, which are also empirical values. We assume $a_1 \geq a_2 \geq ... \geq a_k$, and note that the weights are relative values, do not have to comply with the condition that they sum is 1.

4. Build the HMM associated with vehicle trajectory prediction

On the basis of analyzing vehicle routing problem, we model for this problem, and abstract HMM, that is to build a HMM associated with vehicle trajectory prediction. The process of using hidden Markov model to predict the vehicle trajectory consists of two phases of training and prediction. In the training phase, we mainly study and mine historical trajectory information, and then construct the HMM; in the predicting phase, based on the training model, input queried vehicle's just-driven trajectory, adopt Viterbi algorithm to determine the most likely double layers hidden sequences corresponding the known observation sequence. Finally, the k-step probability transition matrix can be calculated according to the parameters of HMM model, and next analyze and predict the moving trajectory. The process is shown in **Fig. 5**.

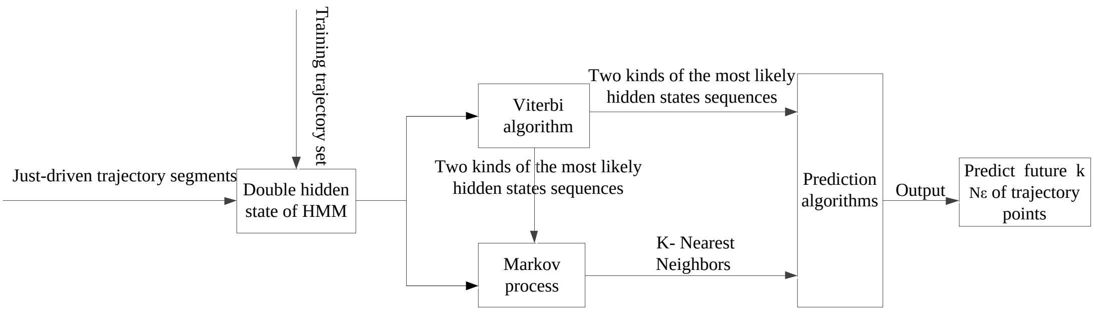

Fig. 5. The process of vehicle trajectory prediction based on HMM.

4.1 Mining the second layer hidden states of HMM

According to the description of Hidden Markov Model, we need to determine the parameters of HMM. First, we define observed states set O and hidden states set S. Here, we dig out double layers hidden states set S_1 and S_2, and then respectively determine the parameters of HMM under two kinds of hidden states sets. We regard trajectories' coordinate points as observed states, and the most likely trajectory sequence corresponding to each coordinate point as the first layer of hidden state, and the appropriate nearest neighbor set of each coordinate point as the second layer of hidden state. Therefore, the input is vehicle traveling trajectory composed of some coordinate points, the output are the most likely trajectory sequence and the appropriate nearest neighbor set corresponding to each coordinate point. The definition of the hidden state is relatively complex in this model. However, after analyzing each trajectory of the training set, we use the computational vector of [1] to determine the first layer of hidden state corresponding to each coordinate point, which is not elaborated in this paper and the detail process seen in [1]. In this paper, we focus on how to explore the second layer hidden state of the trajectory. It is known that observed state sequence and the hidden state sequence are related with probability. So we can model the process as a Hidden Markov process. Supposed that three vehicle trajectories $traj_1 = \{(2,1)\ (2,4)\ (1,4)\ (1,3)\ (0,3)\}$ $traj_2 = \{(5,5)\ (4,5)\ (1,5)\ (1,4)\ (1,3)\ (0,3)\ (0,2)\}$, $traj_3 = \{(0,3)\ (1,3)\ (1,5)\ (4,5)\}$ .The $traj_1$ is from the East Gate of NJUPT to Gongningyuan village; $traj_2$ is from NanJing Railway

Station to Hongqiao Center; $traj_3$ is from the Hospital of Nanjing University of Finances and Economics to Jinqiao Market. We can define all the coordinate point set P in road network model as the observed state set O. So the observed state are coordinate points (2,1), (2,4), (1,4), (1,3), (0,3), (5,5), (4,5)..... Next, we should mine the second layers hidden states corresponding to trajectory points, which is the appropriate nearest neighbor set.

The algorithm of mining the second layer hidden states (seeking the right nearest neighbor set) is as follows:

Input: Training database TD, for example, where the three trajectories are contained: traj1, traj2, traj3.

Output: The second layer hidden states of each coordinate point of |TD| traveling trajectories can be denoted as $S_{ij}{}^2$, where i represents the serial number of the trajectory in training set, and j represents the number of coordinate point element in the $traj_i$. When ε=100m, the hidden states corresponds to driving $traj_1$ are $S_{11}{}^2$ →{the East Gate of NJUPT, NanRui Group Corporation}, $S_{12}{}^2$ →{Jinling Mansion, YuQin Hotel}, $S_{13}{}^2$ →{WuYue International Apartment, LonMenJu , Jinling Mansion}, $S_{14}{}^2$ →{No.1 of Zhongfu Road, Nanjing Tumor Hospital}, $S_{15}{}^2$ →{Gongningyuan village, Nanjing University of Finances and Economics}.The hidden states of $traj_2$ are respectively: $S_{21}{}^2$ →{NanJing Railway Station, Shuguang International Hotel, Line 1 of Metro}, $S_{22}{}^2$ →{ Line 9 of Metro, Agricultural Bank of China }, $S_{23}{}^2$ →{ The City River village, Yihe International Hotel}, $S_{24}{}^2$ →{Jinling Mansion, LongMenJu, WuYue International Apartment}, $S_{25}{}^2$ →{No.1 of Zhongfu Road, Nanjing Tumor Hospital}, $S_{26}{}^2$ →{Songdeli District, Tianhe Garden}, $S_{27}{}^2$ →{HongQiao Center}. The hidden states of $traj_3$ are respectively: $S_{31}{}^2$ →{Songdeli District, Jiangsu Materials Building, LuXunYuan village}, $S_{32}{}^2$ →{No.1 of Zhongfu Road, Nanjing Tumor Hospital}, $S_{33}{}^2$ →{ The City River village, Yihe International Hotel}, $S_{34}{}^2$ →{ Line 9 of Metro, JinQiao Market}. In order to simplify statistics, we use the digital numbers to substitute the nearest neighbor set of coordinate points, such as $S_{11}{}^2 \to \{1\}$, $S_{12}{}^2 \to \{2\}$, $S_{13}{}^2 \to \{3\}$, $S_{14}{}^2 \to \{4\}$, $S_{15}{}^2 \to \{5\}$, $S_{21}{}^2 \to \{6\}$, $S_{22}{}^2 \to \{7\}$, $S_{23}{}^2 \to \{8\}$, $S_{24}{}^2 \to \{3\}$, $S_{25}{}^2 \to \{4\}$, $S_{26}{}^2 \to \{5\}$, $S_{27}{}^2 \to \{9\}$, $S_{31}{}^2 \to \{10\}$, $S_{32}{}^2 \to \{4\}$, $S_{33}{}^2 \to \{8\}$, $S_{34}{}^2 \to \{11\}$.

The algorithm of mining the second layer hidden states

1 Hidden state set $S^2=\phi$;
2 Foreach (T_i in TD)
3 Foreach (O_j In T_i)
4 $N_\varepsilon(O_j)$ =ε-neighbor(O_j);
5 Insert $N_\varepsilon(O_j)$ into $S_{ij}^{\ 2}$;
6 O_1 =Start point in T_i;
7 Transition point $O_j = (x_j, y_j)$;
8 Compute vector $\overrightarrow{a_i} = (x_j - x_1, y_j - y_1)$;
9 For(int m=1; m<=j; m++)
10 If($O_j \in T_m$)
11 O_1 = Start point in T_m;
12 O_k =Locate(O_i in T_m);
13 $N_\varepsilon(O_k)$ =ε-neighbor(O_k);
14 Foreach (U_k In $N_\varepsilon(O_k)$)
15 Coordinate point P_k =Locate(U_k);
16 Compute vector $\overrightarrow{a_m} = \overrightarrow{O_1P_k}$;
17 if ($\cos\langle \vec{a}_i, \vec{a}_m \rangle \le 0$ && U_k in $S_{ij}^{\ 2}$)
18 Delete U_k in Hidden $S_{ij}^{\ 2}$;

The algorithm described is as follows:

(1) Initialize the second layer of hide the states set and traverse each trajectory. (Line1-line2)

(2) Seek ε-neighbor set for each trajectory point, and insert it into the hidden state set. (line3-line5)

(3) Calculate the vector $\overrightarrow{a_i}$, which is from starting point O_1 of the i-th trajectory to the transition point O_j. (Line6-Line8)

(4) Seek the coordinate point whose observed state is O_j in traversed trajectories and find its ε-neighbor set, and then traverse it to compute $\overrightarrow{a_m} = \overrightarrow{O_1P_k}$. (Line9-Line16)

(5) If $\cos\langle \vec{a}_i, \vec{a}_j \rangle \le 0$, and U_k is in $S_{ij}^{\ 2}$, delete the U_k in $S_{ij}^{\ 2}$. Then the rest elements of its ε-neighbor set are the right nearest neighbor units. (Line17-Line18)

4.2 Compute the parameters of HMM

After determining the second layer hidden states of the training set TD, the next step is compute the other three basic parameters of HMM in the condition of the second layer hidden states, including the initial probability matrix π, state transition matrix A and confusion matrix

B. [1] elaborated in detail how to calculate the parameters of HMM in the case of the first layer hidden states. This paper only introduces how to calculate the parameters of HMM in the case of the second layer hidden states. We calculate the parameters of HMM by statistical method, and count the number of the group of coordinate points and appropriate nearest neighbor set both emerging in the training set:

The calculation of initial probability matrix π is as follows:

$$P_i = \frac{Num(S_i)}{\sum_{j=1}^{N} Num(S_j)} \quad (7)$$

where π is a one-dimensional matrix, N denotes the number of hidden states and $Num(S_i)$ denotes the number of hidden state S_i appearing in all hidden states sequences corresponding to observed states sequences. In the above training set, the frequency of each element emerging in hidden states set S is shown in Table 1.

Table 2. The emerging frequency of every hidden state

S_i	1	2	3	4	5	6	7	8	9	10	11
Frequency of occurrences	1	1	2	3	2	1	1	2	1	1	1

The computation of state transition probability matrix A is as follows:

$$\mathrm{P}\left(\mathrm{S}_i \middle| \mathrm{S}_{i\text{-}1}\right) = \frac{Num\left(\mathrm{S}_{i\text{-}1}, \mathrm{S}_i\right)}{Num\left(\mathrm{S}_{i\text{-}1}\right)} \quad (8)$$

where $Num(S_{i-1}, S_i)$ denotes the count of S_i appearing after S_{i-1} in training set, and $Num(S_{i-1})$ represents the counts of S_{i-1} emerging in training set. In the above training set, the count of the hidden state S_i appearing after S_{i-1} is shown as in Table 2. (For example, the count of $\{3\}$, $\{4\}$ appearing in training set is 2):

Table 3. The count of the hidden state S_i appearing after S_{i-1}

S_{i-1} \ S_i	{1}	{2}	{3}	{4}	{5}	{6}	{7}	{8}	{9}	{10}	{11}
{1}	0	**1**	0	0	0	0	0	0	0	0	0
{2}	0	0	**1**	0	0	0	0	0	0	0	0
{3}	0	0	0	**2**	0	0	0	0	0	0	0
{4}	0	0	0	0	**2**	0	0	**1**	0	0	0
{5}	0	0	0	0	0	0	0	0	**1**	0	0
{6}	0	0	0	0	0	0	**1**	0	0	0	0
{7}	0	0	0	0	0	0	0	**1**	0	0	0
{8}	0	0	**1**	0	0	0	0	0	0	0	**1**
{9}	0	0	0	0	0	0	0	0	0	0	0

{10}	0	0	0	**1**	0	0	0	0	0	0	0
{11}	0	0	0	0	0	0	0	0	0	0	0

The calculation of Confusion matrix B is as follows:

$$P\left(O_j|S_i\right)=\frac{Num\left(S_i,O_j\right)}{Num\left(S_i\right)} \tag{9}$$

where $Num(S_i,O_j)$ denotes the count of observed state O_j occurring along with the hidden state S_i , and $Num(S_i)$ represents the count of hidden state S_i occurring in the training set. In the above training set, the count of observed state O_j occurring along with hidden state S_i is shown in **Table 3**.

Table 4. The count of observed state O_j occurring along with hidden state S_i

O_j \ S_i	(2,1)	(2,4)	(1,4)	(1,3)	(0,3)	(5,5)	(4,5)	(1,5)	(0,2)
{1}	**1**	0	0	0	0	0	0	0	0
{2}	0	**1**	0	0	0	0	0	0	0
{3}	0	0	**2**	0	0	0	0	0	0
{4}	0	0	0	**3**	0	0	0	0	0
{5}	0	0	0	0	**2**	0	0	0	0
{6}	0	0	0	0	0	**1**	0	0	0
{7}	0	0	0	0	0	0	**1**	0	0
{8}	0	0	0	0	0	0	0	**2**	0
{9}	0	0	0	0	0	0	0	0	**1**
{10}	0	0	0	0	**1**	0	0	0	0
{11}	0	0	0	0	0	0	**1**	0	0

4.3 Vehicle trajectory prediction

Based on the HMM, we use Viterbi algorithm to determine the most likely hidden state sequence of the known observation sequence (which is also called driving trajectory). The input parameters of Viterbi algorithm include hidden Markov model and the known observation sequence, while the output is the most likely hidden state sequence. And the concrete steps have been described in detail in the 3.3.2 section. Because we define the double layer hidden states, we can get two kinds of hidden states sequences in the end: the trajectory sequence and the nearest neighbor sequence. Next we propose the algorithm (DHMTP) for predicting the next k phases' the nearest neighbor set of the driving trajectory.
DHMTP Algorithm: Vehicle trajectory prediction based on HMM of double layer hidden states.

Input: Trajectory database $TD=\{Traj_1,Traj_2,...,Traj_r\}$, $HMM=\{S^1,S^2,O,A,B,\pi\}$, the driving trajectory $T=\{P_1,P_2,...,P_n\}=\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$

Output: The nearest neighbor set PN of $t_{n+1}, t_{n+2}, ..., t_{n+k}$ timestamp.

DHMTP Algorithm

1. $\varepsilon - NS = \varnothing$;
2. $RT(PT_2, PT_2, ..., PT_n)$ =Viterbi(T, S^1);
3. $N_\varepsilon(N_\varepsilon(P_1), N_\varepsilon(P_2), ..., N_\varepsilon(P_n))$ =Viterbi(T, S^2);
4. Set PT_n =Extract(RT , t_n);
5. Set $N_\varepsilon(P_n)$ = Extract(N_ε, t_n);
6. Foreach ($U_k \in N_\varepsilon(P_n)$)
7.   If($\exists$ $Traj$ and its ε-neighbor contain U_k and $Traj$ in PT_n)
8.   Insert U_k in $\varepsilon - NS$;
9. PN =LKNU($\varepsilon - NS$, k-Markov) ;

The above algorithm is described as follows:

(1) Initialize the most likely second layer hidden states set $\varepsilon - NS = \varnothing$ corresponding to observation point PT_n at t_n timestamp. (Line1)

(2) Using Viterbi algorithm to determine the first layer hidden state sequence and the second layer hidden state sequence corresponding to the observed state sequence. PT_i represents the most likely hidden state of the first layer corresponding to the i-th trajectory point, namely the possible trajectories set [1], for example, $PT_i = \{traj_1, traj_3\}$. $N_\varepsilon(P_i)$ represents the most likely hidden state set of the second layer corresponding to the i-th trajectory point. (Line2-Line3)

(3) Extract two hidden state sets PT_n and $N_\varepsilon(P_n)$. (Line4-Line5)

(4) Further determine $\varepsilon - NS$ at t_n according to the first layer hidden state set, and specific practices are: traverse each element U_k of $N_\varepsilon(P_n)$, and seek the trajectory in trajectory database whose ε-neighbor set contains U_k. If the trajectory is in PT_n, insert U_k into $\varepsilon - NS$. (Line6-Line8)

(5) According to the second layer hidden state set $\varepsilon - NS$ and k-step transition matrix, adopt the LKNU algorithm to predict the nearest neighbor units of the next k stages. (Line9)

The algorithm of LKNU is as follows:

Input: k-Markov, $\varepsilon - NS$;

Output: The nearest neighbor units of the next k stages.

LKNU Algorithm

---

1. $K\_Neighbour$ Set $PN = \varnothing$;
2. $X_n = \varepsilon - NS(1)$;
3. For (int m=1;m<=k;m++)
4.   Foreach(S_j^2 in S^2)
5.   Compute $P(X_{n+m} = S_j^2 \mid X_n = S_i^2) = P_{ij}^m$
6. $PN(m) = \{S_j^2 \mid \max_{j \in N}(P_{ij}^m)\}$

The description of LKNU algorithm is as follows:

(1) Initialize $K\_Neighbour$ set $PN = \varnothing$; (Line1)

(2) Determine X_n according to $\varepsilon - NS$, and traverse the second layer hidden states. Then calculate the probability of possible neighbor unit P_{ij}^{m} in the future m-th timestamp, and take the neighbor unit corresponding to the maximum P_{ij}^{m} as the predicted result of future m-th timestamp, and insert the neighbor unit into PN. (Line2 - Line8)

5. Experiment and Analysis

To test performance of the proposed trajectory prediction method in this paper, we design and implement the trajectory prediction DHMTP algorithm based on HMM. Experiments are carried out to verify the effectiveness and efficiency of the algorithm. The effectiveness of the algorithm is mainly to investigate the accuracy of it, while the efficiency of the algorithm is mainly to investigate the time of the query. According to the similarity of the algorithm design background and the comparability of the network model, the accuracy and time performance of the TPMO algorithm, Naive algorithm and DHMTP algorithm are compared in this paper. The experimental platform is R2012a MATLAB, and experimental data from taxis' GPS location data of Nanjing at this time from weekday morning to evening. In this paper, a part of the trajectories are selected randomly from the database as testing trajectories, and the rest are as training data of HMM.

The simulation parameters and the value range in this paper are listed as follows:

k: predicted stage length, and $k \in [1, 4]$.

Num: the number of training trajectories, and $Num \in [100, 2000]$.

ε: neighbor threshold, and $\varepsilon \in [100, 500]$.

5.1 Prediction accuracy analysis

In order to verify the performance of the proposed method, the average value of prediction accuracy under different number of testing trajectories is taken to evaluate the prediction performance. As shown in **Fig. 6** (a), when the predicted stage length k=1, with the increase of the number of test trajectories, the prediction accuracy of DHMTP is the highest compared to the other two algorithms. We find the accuracy of DHMTP is about 18.3% greater than that of TPMO, and is about 23.1% greater than that of Native. TPMO is sensitive to historical data, and the more concentrated the distribution of trajectories is, the more obvious the effect of the hot spot region mining algorithm is, but if the distribution of the trajectory is looser, the effect of the hot spot region mining algorithm is not more obvious and the performance of prediction is worse. Naive algorithm only considers the first-order probability intensity matrix, while TPMO takes into the n-order complex intensity matrix consideration, and Naive did not analyze the different directions of trajectories, so its prediction error rate was the highest. Compared with TPMO and Native, the generality of HMM is better.

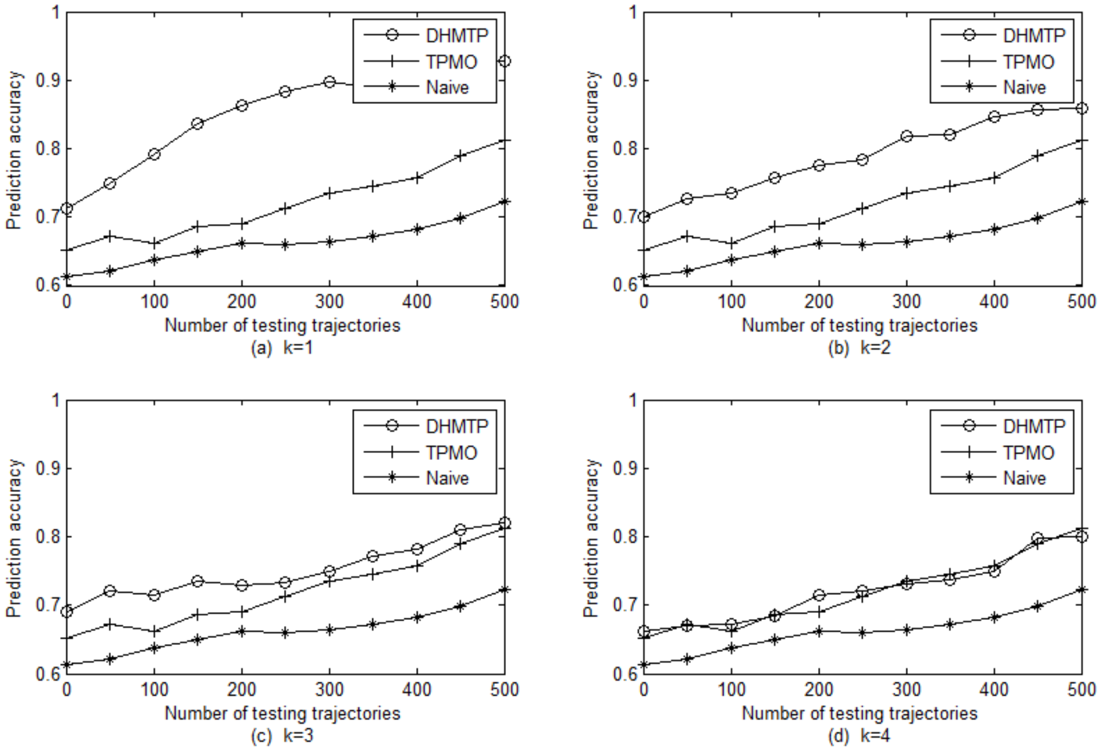

Fig. 6. Prediction accuracy comparison among DHMTP, TPMO and Naive under distinct k

In contrast to **Fig. 6** (a-d), we can find that with the increase of the number of prediction length k, the accuracy of DHMTP algorithm is generally decreased, and when k=1, the accuracy is the greatest, and when k=4, the accuracy of DHMTP algorithm is very close to that of TPMO algorithm. This is because what we inputted are these observation points from timestamp t_1 to timestamp t_n, and what we mined are these hidden sequences corresponding to these observation points from timestamp t_1 to timestamp t_n, and the parameters of k-step transition probability matrix are not more imprecise with the increase of predicted length k. Therefore, the future timestamp is closer to t_n, the accurate rate of the predicted trajectory neighbor point is higher.

Fig. 7 compares the prediction accuracy under the different threshold ε with different number of training trajectories (Num). We can see from the graph, the more the training trajectories, the higher the prediction accuracy. In different number of training trajectories, when the training trajectories are more and the threshold ε is smaller, the prediction accuracy is higher. Because the HMM contains more historical trajectory data, so we can obtain more nearest neighbor sets and the nearest neighbor unit is more accurate, the prediction accuracy is also improved.

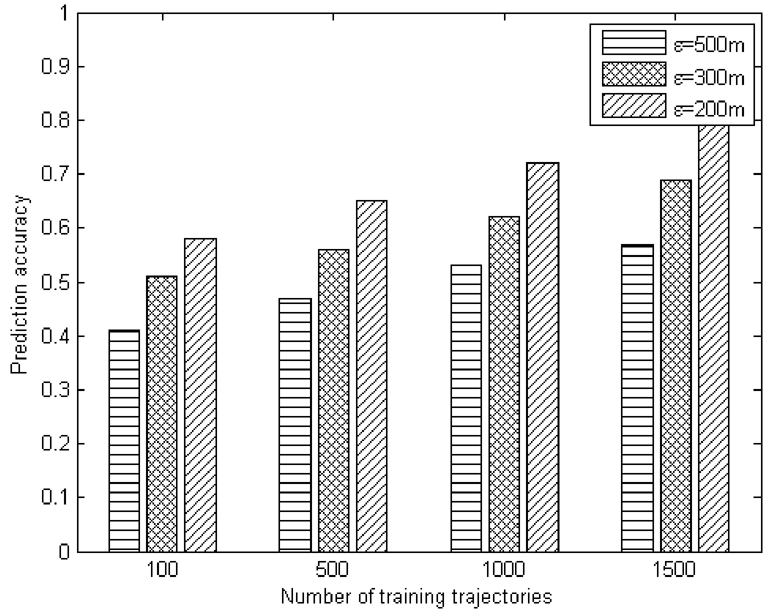

Fig. 7. Prediction accuracy comparison among different training sets

5.2 Prediction time Analysis

Fig. 8 shows the prediction results of the three algorithms under different number of testing trajectories. We can see from the figure, the cost of TPMO algorithm is significantly more than DHMTP algorithm and Naive algorithm. Because the TPMO algorithm not only needs to mine hot spot areas and cluster trajectories, but also construct Bayesian network of continuous time, which extremely consumes time. The time of DHTMP algorithm is stable, and it is always close to Naive algorithm, regardless of whether the number of testing trajectories increases, it is only increased by 30ms compared with Naïve algorithm. This is because the DHMTP algorithm needs to use the HMM to extract hidden states and observed states.

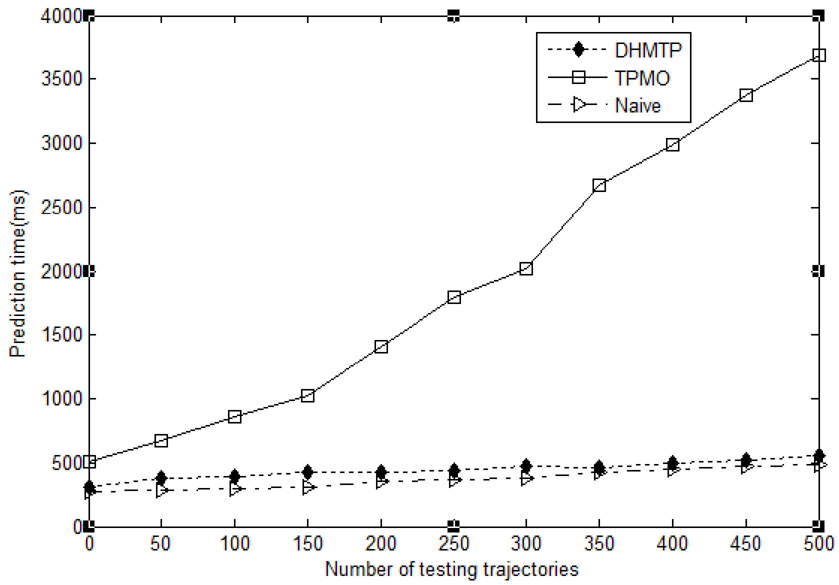

Fig. 8. Prediction time comparison among DHMTP, TPMO and Naive

6. Conclusion

Vehicle trajectory prediction is a challenging research topic. In this paper, DHMTP algorithm is proposed based on HMM for the low prediction accuracy of the existed algorithms. The algorithm firstly mimes the double layers hidden states of trajectory points, and then uses HMM to model and calculate relative parameters of HMM. Secondly, the most likely double layers hidden sequences are solved by the Viterbi algorithm. Lastly, k phases of the near neighbor units of vehicle driving trajectory are predicted. As a direction of the future work, the improvement will be from two points: (1) take the influence of the objective factors on location prediction into consideration, such as the traffic lights and traffic jams; (2) investigate to enhance the anti-disturbance of the environmental factors to suit HMM better for trajectory predictions.

References

[1] Ning Ye, Zhong-qin Wang, Reza Malekian, Qiaomin Lin and Ru-chuan Wang, "A Method for Driving Route Predictions Based on Hidden Markov Model," *Mathematical Problems in Engineering*, vol. 15, pp.1-12, January, 2015. Article (CrossRef Link)

[2] Dong, Wook Lee, S. H. Baek and H. Y. Bae, "aCN-RB-tree: Constrained Network-Based Index for Spatio-Temporal Aggregation of Moving Object Trajectory," *Ksii Transactions on Internet & Information Systems*, vol. 3, no. 5, pp. 527-547, October, 2009. Article (CrossRef Link)

[3] D. Phan, I. S. Na and S. H. Kim, "Triangulation Based Skeletonization and Trajectory Recovery for Handwritten Character Patterns," *Ksii Transactions on Internet & Information Systems*, vol. 9, no. 1, pp. 358-377, January, 2015. Article (CrossRef Link)

[4] X. Yin, B. Wang and W. Li, "Background Subtraction for Moving Cameras based on trajectory-controlled segmentation and Label Inference," *Ksii Transactions on Internet & Information Systems*, vol. 9, no. 1, pp. 4092-4107, October, 2015. Article (CrossRef Link)

[5] Xuezhi Wen, Ling Shao, Yu Xue and Wei Fang, "A rapid learning algorithm for vehicle classification," *Information Sciences*, vol. 295, no. 1, pp. 395-406, October, 2015. Article (CrossRef Link)

[6] Ning Ye, Zhong-qin Wang, Reza Malekian, Ying-ya Zhang and Ru-chuan Wang, "A Method of Vehicle Route Prediction Based on Social Network Analysis," *Journal of Sensors*, vol. 15, no. 1, pp.1-10, Apirl, 2015. Article (CrossRef Link)

[7] P. Liu, A. Kurt and U. ÖZgüNer, "Trajectory prediction of a lane changing vehicle based on driver behavior estimation and classification," in *Proc. of Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on. IEEE*, pp. 942-947, November 7-12, 2014. Article (CrossRef Link)

[8] A. Houenou, P. Bonnifait and V. Cherfaoui, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *Proc. of Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE*, pp. 4363-4369, November 6-9, 2013. Article (CrossRef Link)

[9] Eddy and R. Sean, "Hidden Markov models," *Current Opinion in Structural Biology*, vol. 6, no. 6, pp. 361-365, June, 1996. Article (CrossRef Link)

[10] K. Hongfa, "Trajectory Prediction Algorithm of Moving Object Based on MGM(1,N)," *Geomatics & Information Science of Wuhan University*, vol. 37, no. 6, pp. 662-666, June, 2012. Article (CrossRef Link)

[11] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao and D. Cheung, "Mining, indexing, and querying historical spatiotemporal data," in *Proc. of the 2004 ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press*, pp. 236-245, January 21-25, 2004. Article (CrossRef Link)

[12] M. Morzy, "Mining frequent trajectories of moving objects for location prediction," in *Proc. of the 5th Int'l Conf. on Machine Learning and Data Mining in Pattern Recognition. LNCS 4571, Heidelberg: Springer-Verlag*, pp.667-680, July 12-16, 2007. Article (CrossRef Link)

[13] J. C. Ying, W. C. Lee, T. C. Weng and S. Tseng, "Semantic trajectory mining for location prediction," in *Proc. of the 19th ACM SIGSPATIAL Int'l Conf. on Advances in Geographic Information Systems. New York: ACM Press*, pp.34-43, January 13-17, 2011. Article (CrossRef Link)

[14] S. J. Qiao, K. Jin, N. Han and C. J. Tang, "Trajectory prediction algorithm based on Gaussian mixture model," *Journal of Software*, vol. 26, no. 5, pp.21-32, May, 2015. Article (CrossRef Link)

[15] S. J. Qiao, D. Shen and X. Wang, "A Self-Adaptive Parameter Selection Trajectory Prediction Approach via Hidden Markov Models," *Intelligent Transportation Systems IEEE Transactions on*, vol. 16, no. 1, pp.284-296, February, 2015. Article (CrossRef Link)

[16] A. Asahara, A. Sato, K. Maruyama and K. Seto, "Pedestrian-movement prediction based on mixed Markov-chain model," in *Proc. of 19th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst*, pp. 25-33, January 1-5, 2011. Article (CrossRef Link)

[17] S. Gambs, M. Killijian, D. P. Cortez and N. Miguel, "Next place prediction using mobility Markov chains," in *Proc. of 1st Workshop Meas., Privacy*, Mobility, pp. 1-6, April 10-15, 2012. Article (CrossRef Link)

[18] L. Wangao, X. Zhao and D. Sun, "Prediction of trajectory based on modified Bayesian inference," *Journal of Computer Applications*, vol. 33, no. 7, pp.1960-1963, April, 2013. Article (CrossRef Link)

[19] S. J. Qiao, J. Peng, T. R. Li, Y. Zhu and L. X. Liu, "Uncertain trajectory prediction of moving objects based on CTBN," *Journal of University of Electronic Science and Technology of China*, vol. 41, no. 5, pp.759−763, November, 2012. Article (CrossRef Link)

[20] I. Kaparias and M. Bell, "A reliability-based dynamic re-routing algorithm for in-vehicle navigation," in *Proc. of Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. IEEE*, pp. 974-979, October 11-15, 2010. Article (CrossRef Link)

[21] N. S. Pai, H. J. Kuang and T. Y. Chang, "Implementation of a Tour Guide Robot System Using RFID Technology and Viterbi Algorithm-Based HMM for Speech Recognition," *Mathematical Problems in Engineering*, vol. 22, no. 1, pp. 12-25, March, 2014. Article (CrossRef Link)

Ning Ye received the BS degree in Computer Science from Nanjing University, China, in 1994, the MS degree in School of Computer & Engineering from Southeast University, China, in 2004, and the Ph.D. degree in Institute of Computer Science from Nanjing University of Post and Telecommunications, China, in 2009 and is currently a Professor there. In 2010, Ning Ye worked as a Visiting Scholar and Research Assistant in the Department of Computer Science, University of Victoria, Canada. Her research interests include information processing in wireless networks and Internet of Things. She is a senior member of Chinese Computer Federation (CCF).

Yingya Zhang received the BS degree in Computer Science and Technology from Jiangsu University of Science and Technology, China in 2014. She is currently pursuing the MS degree in Computer Software and Theory at Nanjing University of Posts and Telecommunications. Her research interest includes information processing in wireless networks and Internet of Things.

Ruchuan Wang was born in Anhui Province, China. He researched on graphic processing at University of Bremen and program design theory Ludwig Maximilian Muenchen Unitversitaet from 1984 to 1992. He is a professor and tutor of Ph.D. candidate in Nanjing University of Posts and Telecommunications since 1992. Major research interests include wireless sensor networks, information security.

Dr. Reza Malekian is currently a Senior Lecturer with the Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, Pretoria, South Africa. His current research interests include advanced sensor networks, Internet of Things, and mobile communications. Dr. Malekian is also a Chartered Engineer and a Professional Member of the British Computer Society. He is an associate editor for the IEEE Internet of Things Journal.