Predicting Driving Direction with Weighted Markov Model

Bo Mao[1,*], Jie Cao[1], Zhiang Wu[1], Guangyan Huang[2], and Jingjun Li[3]

[1] Jiangsu Provincial Key Laboratory of E-Business,
Nanjing University of Finance and Economics,
Nanjing, China
maoboo@gmail.com
[2] Centre for Applied Informatics,
Victoria University, Melbourne, Australia
[3] Yurun Meat Industry Company Limited, Nanjing, China

Abstract. Driving direction prediction can be useful in different applications such as driver warning and route recommendation. In this paper, a framework is proposed to predict the driving direction based on weighted Markov model. First the city POI (Point of Interesting) map is generated from trajectory data using weighted PageRank algorithm. Then, a weighted Markov model is trained for the near term driving direction prediction based on the POI map and historical trajectories. The experimental results on real-world data set indicate that the proposed method can improve the original Markov prediction model by 10% at some circumstances and 5% overall.

Keywords: driving direction prediction, trajectory mining, weighted PageRank.

1 Introduction

Along with the development of GPS and wireless communication technologies, the driving trajectory data is increased dramatically. Basically ever vehicle can be easily equipped with GPS device and becomes a sensor on the road. As the most driven type of all vehicles, taxi is excellent for road condition detection since there is not too much privacy issue compared with other cars. The trajectory data from taxi has been applied in driving route recommendation, road map generation, map matching and driving direction prediction [1]. It is important to predict the near term direction for the car drivers, which can be useful to provide information about upcoming road situation and is the basic for destination prediction.

History driving trajectories have been proved to be an efficient basis to predict the near-term driving direction [2]. Meanwhile, the city itself contains many "points of interesting (POIs)" that attract more traffic. It is useful to detect these POIs and apply them into driving direction prediction. Currently, POI is

* Corresponding author.

S. Zhou, S. Zhang, and G. Karypis (Eds.): ADMA 2012, LNAI 7713, pp. 407–418, 2012.

usually collected manually, and many places may be missed. In this paper, we generate city POI map based on taxi trajectory data, since more getting on or off events happened near the POIs. However, the identified POI is not an explicit point but an area with description of its interesting level. Then combining the created POI map and trajectory data, driving directions are predicted. We select weighted PageRank algorithm to generate the POI map and Markov Model to predict the direction.

The rest of paper is structured as follows. Section 2 lists the related work. Section 3 provides the problem statement. The proposed methodology is described in section 4. Experimental results are given in section 5 and Section 6 concludes the whole paper.

2 Related Work

Driving route prediction has been intensively studied in data mining and GIS fields. Patterson et al. [3] have used a dynamic Bayes net to detect a moving person's mode of transportation (i.e. bus, foot, car) based on GPS trajectories and predicted their route. Li et al. [4] have analyzed the periodic behaviors of animal based on their traces. In [5], trajectories of road network are classified by mining the discriminative patterns. These models and patterns can be used for direction prediction. Krumm [2] proposed a Markov model, trained from past behavior, to predict the next few road segments that a driver would take. Froehlich and Krumm [6] predicted entire routes from GPS traces by looking at which previous route a driver appeared to lock onto partway into the trip. In [7], the author developed a basic algorithm, and variations, to predict the aggregate turn behavior of drivers at intersections. Besides trajectory data, geographic features are also considered in the direction prediction [3], such as land covered by water or ice made for less popular destinations than commercial and residential areas. Also Jiang [8] suggested street-based topological representations for prediction traffic flow in GIS. Therefore, it is necessary to combine the city features or POIs into the prediction.

To detect POIs, Internet and mobile phone data are analyzed [9, 10] but it may arise the privacy concerns. Lee et al. [11] tried to turn the geo-tagged photo into list of POIs. Brouwers [12] discussed three different sensor sources (GPS, Wifi and Geolocation) and their idiosyncrasies when used for dwelling detection. Jiang [8] ranked the city space based on road network using weighted PageRank. It found that the PageRank scores are better correlated to human movement rates than the space syntax metrics. However, the geo-related data is not allows available, so trajectory based POI detection method are employed.

In this work, we present a driving direction prediction framework based on just trajectory data using Weighted PageRank algorithm. In the framework, city POI map is generated from trajectories and the driving prediction is performed based on the POI map and history trajectories.

3 Problem Statement

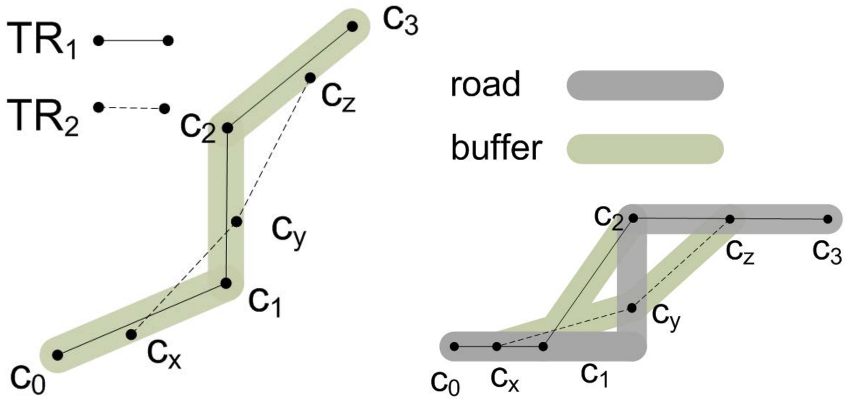
In this paper, near-term driving direction prediction is based on the city POI map and trajectory data. This section will give the definition of POI map and other related concept we use.

Definition 1. City POI map is a function f(c_P), in which $c_P = (x_P, y_P)$ is the coordinate of location P and f(c_P) returns the attraction level in location P.

Definition 2. Trajectory TR=$\{c_1, c_2, \cdots, c_n\}$ represents the coordinates in a time sequence from t_1 to t_n, in which $t_i < t_j$ if $i < j$.
A trajectory can be also viewed as a set of line segments TR=$\{c_1c_2, \cdots, c_{n-1}c_n\}$. Since no road graph is generated, we approximately define a coordinate $c_x \in TR$ (c_x belongs to TR), *iff* $\min(\text{dist}(c_x, c_ic_{i+1})|0 < i < n) < d$, which means c_x in a buffer of TR as shown in Fig. 1a. There are two trajectories: $\text{TR}_1 = \{c_0, c_1, c_2, c_3\}$ and $\text{TR}_2 = \{c_x, c_y, c_z\}$. Because the points in TR_2 are all contained by TR_1, then we define that TR1?TR2. Assume two trajectories TR_i and TR_j, if $\text{TR}_i \subset \text{TR}_j$ or $\text{TR}_j \subset \text{TR}_i$, then they are the same route or $\text{TR}_i = \text{TR}_j$.

At some circumstance, there are certain points that not belong to another trajectory even if they are along the same road e.g. point cy in Fig. 1b. To overcome the problem, we add the condition that if c_i and c_{i+1} belong to TR, then $c_i \in$TR.

In this paper, we try to predict the near term driving direction of a vehicle based on the history trajectories. Suppose $\text{TR}_c = c_{-n}, c_{-n+1}, c_{-1}, c_0$, we will find out the possible options of the next direction that contains c1 and their corresponding probabilities. The one with the biggest possibility will be assigned as the predicted direction.



(a) (b)

Fig. 1. Example of belong to relationship

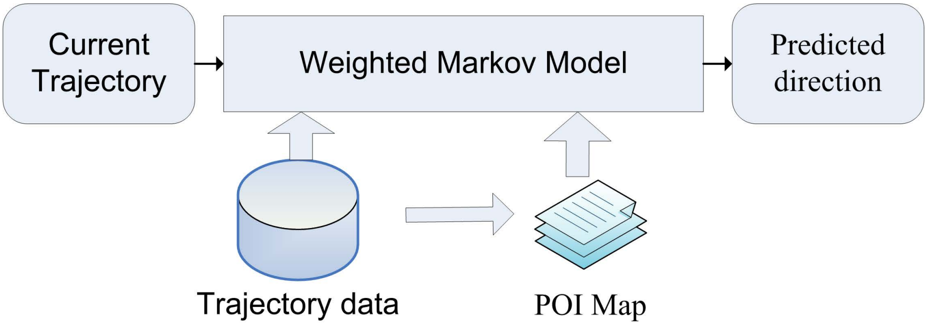


Fig. 2. Prediction Framework

4 Methodology

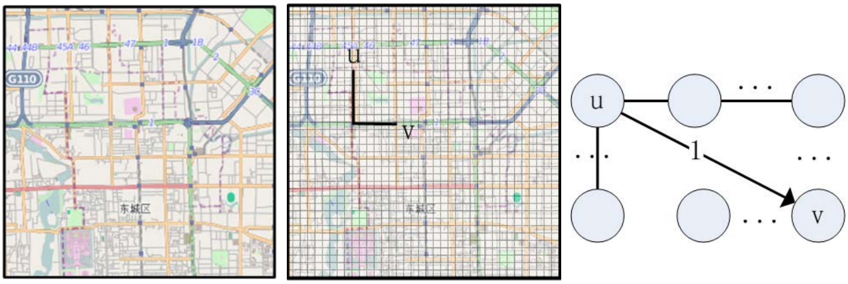
In this section, we present the framework of weighted Markov model based prediction. This framework consists of two steps: 1) POI map generation, and 2) weighted Markov prediction. We describe the framework in Section 4.1 and these two steps through Section 4.2 and 4.3 respectively.

4.1 Driving Direction Prediction Framework

The proposed prediction framework contains three main parts: trajectory dataset, POI map and weighted Markov model. Initially, trajectory data is collected from different cars with GPS tracker. The data is structured according to **Defination 2**. To separate trajectories from GPS points, we can connect the tracker with some devices such as taximeter or just apply data mining methods [13]. The POI map is created based on trajectory dataset using weighted PageRank algorithm. Each trajectory can be regarded as an edge that link two parts of the city, based on which a graph about city is generated. With weighted PageRank method, all parts of the city are ranked according to their attractions to the drivers, and a POI map of the city is created from the ranking. Based on the trajectory dataset and POI map, the prediction is implemented. When a real time trajectory TR is inputted into the system, we first test its nth-order Markov Model to find out the history trajectories that contain n latest points of TR and select the most weighted direction according to the POI map as the output result instead of most frequent direction. Rest of the section will describe the framework in details.

4.2 POI Map Generation

As defined, POI map f(c_x) is the attractive level to the drivers or passages. PageRank [14] has been proved as an efficient method to define the "attractiveness" of a web page and city morphology [15]. Therefore, we employ PageRank to calculate the POI map. To take the number of trajectories into consideration, weighted PageRank [16] is selected.



(a) (b) (c)

Fig. 3. POI map generation

The analysis in this paper is performed on the city raster map. We first divide the city (Fig. 3a) into square areas (Fig. 3b). Every square area is represented by a node in the directed city graph (Fig. 3c). For each square area or node such as u and v in Fig. 3b, if there is a trajectory connected from u to v, the number of outlinks in u and number of inlinks in are both increased by 1 as shown in Fig. 3c.

The weighted PageRank of each node in graph G is calculated according to [17]. The original PageRank algorithm [14] defined the pagerank of a node, as formula 1:

$$PR_u = 1 - d + d \sum_{v \in B(u)} \frac{PR(v)}{N_v}. \tag{1}$$

where u represents an area or node. B(u) is the set of nodes that point to u. PR(u) and PR(v) are rank scores of node u and v, respectively. N_v denotes the number of outgoing links of node v. d is a dampening factor that is usually set to 0.85.

The original PageRank is proposed for web pages, but in city transportation scenario the number of trajectory from one area to another is critical. For example, the train station trend to be an important POI since it is the destination of many trajectories, which can be reflected well in the weighted PageRank algorithm. The Weighted PageRank Algorithm assigns larger rank values to more important (popular) nodes instead of dividing its rank value evenly among its outlink nodes. The weight from the number of inlinks and outlinks is recorded as W_{in} and W_{out}(v,u), defined as follows respectively.

$$W^{in}_{(v,u)} = \frac{I_u}{\sum_{p \in R(v)} I_p}. \tag{2}$$

$$W^{out}_{(v,u)} = \frac{O_u}{\sum_{p \in R(v)} O_p}. \tag{3}$$

Where I_u/O_u and I_p/O_p is the number of inlinks/outlinks of node u and p. R(v) denotes all nodes that are pointed from v. Then the weighted PageRank is defined in formula (4)

$$PR(u) = (1-d) + d \sum_{v \in B(u)} PR(v) W^{in}_{(v,u)} W^{out}_{(v,u)}. \quad (4)$$

To calculate the Weighted PageRank, PR(u) is set to 1 for all nodes. Then iteratively compute the new PR(u) based on formula (4) until the difference between two rounds are smaller than a threshold (e.g. ¡10$_{-5}$). Previous experiments [14, 17] showed that the PageRank gets converged to a reasonable tolerance in the roughly logarithmic (logn), which is also applicable for Weighted PageRank.

4.3 Weighted Markov Model

According to [2], the prediction of near-term driving direction can be based on its past route. Therefore, Markov model is employed. A trajectory can be defined as $\{\cdots, c_{-2}, c_{-1}, c_0, c_1, c_2, \cdots\}$, where c_0 is the current position that can be updated, and $c_1, c_2, \cdots$ are the unknown future position of the vehicle. We try to predict the direction that contains both c_0 and c_1 as defined section 2. At anytime, we have the information $c_0, c_{-1}, \cdots$ back to the beginning of the trip. The Markov model is built on these known positions, and gives a probabilistic prediction over the future directions. The first order Markov model is only based on current position c_0, and the second order Markov model is based on c_0 and c_{-1}. In general, n-th order Markov is based on $c_0, \cdots, c_{-n+1}$ as shown in formula (5). Note that P$[c_1]$ is not the probability of the vehicle appearing at c_1 but its direction to c_1.

$$P[c_1] = P[c_1 | c_0, c_{-1}, \cdots, c_{-n+1}]. \quad (5)$$

In original Markov model, the probability of each direction ci is calculated as formula (6), in which N$[c_p \cdots c_q]$ is the number of trajectories containing position $c_p \cdots c_q$.

$$P[c_i | c_0, c_{-1}, \cdots, c_{-n+1}] = \frac{N[c_i, c_0, c_{-1}, \cdots, c_{-n+1}]}{N[c_0, c_{-1}, \cdots, c_{-n+1}]}. \quad (6)$$

Based on formula (6), we can select the direction with biggest probability as the next possible direction.

The original Markov model only takes the number of trajectories into consideration, but the POI distribution along trajectories could also affect the driving direction as people trend to go to these "interesting spots". Therefore, in this paper we improved the original Markov model with the POI map and proposed the weighted Markov model in formula (7):

$$P[c_i | c_0, c_{-1}, \cdots, c_{-n+1}] = \frac{\sum f(c_k \ c_{k_e})}{\sum f(c_l \ c_{l_e})}. \quad (7)$$

In formula (7) f(c_p c_q) is the maximum value of POI map along the route from c_p to c_q. k_e and l_e are the end point of k and l respectively. If POI map is a

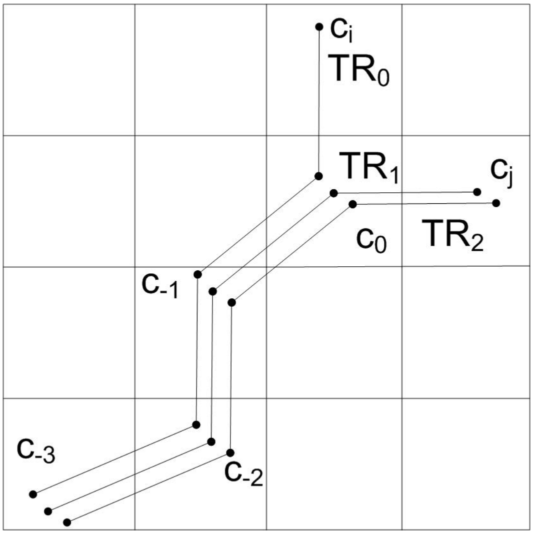


Fig. 4. Weighted Markov model

constant value, then formula (7) will be the same as (6). Fig. 4 illustrates an example of original and weighted Markov model.

In Fig. 4, three history trajectories are listed. For trajectory $[c_0, c_{-1}, c_{-2}, c_{-3}]$, we can predict its next direction will be c_j with probability 2/3. However, suppose the attractive value in area ci is 3 and attractive value in area c_j is 1, then the predicted next direction will be ci with probability 3/5.

5 Performance Evaluation

In this section, we evaluate the performance of the near-term driver direction prediction based on weighted Markov model. We describe the experimental data and environment in Section 5.1. We demonstrate the POI map generated from trajectories in Section 5.2 and report the prediction results comparing with original Markov model in Section 5.3.

5.1 Experimental Setting

All experiments are conducted on an Intel Core2 Duo 2.4GHz PC with 3.25GB of RAM, running on Windows XP SP3. Eclipse 3.5.2 is selected as the IDE. All programs are written in Java.

The test dataset came from a taxi company in San Francisco area. The collected trajectories contain 536 taxis driving in three months from April to July, 2008. Each trajectory is consisted of several records that have latitude, longitude, flag bit (1 for with passenger and 0 for not), and the current time. The

(a) (b)

Fig. 5. Snapshots of the start/end point distribution of training trajectories

Table 1. Generative models of shilling attacks

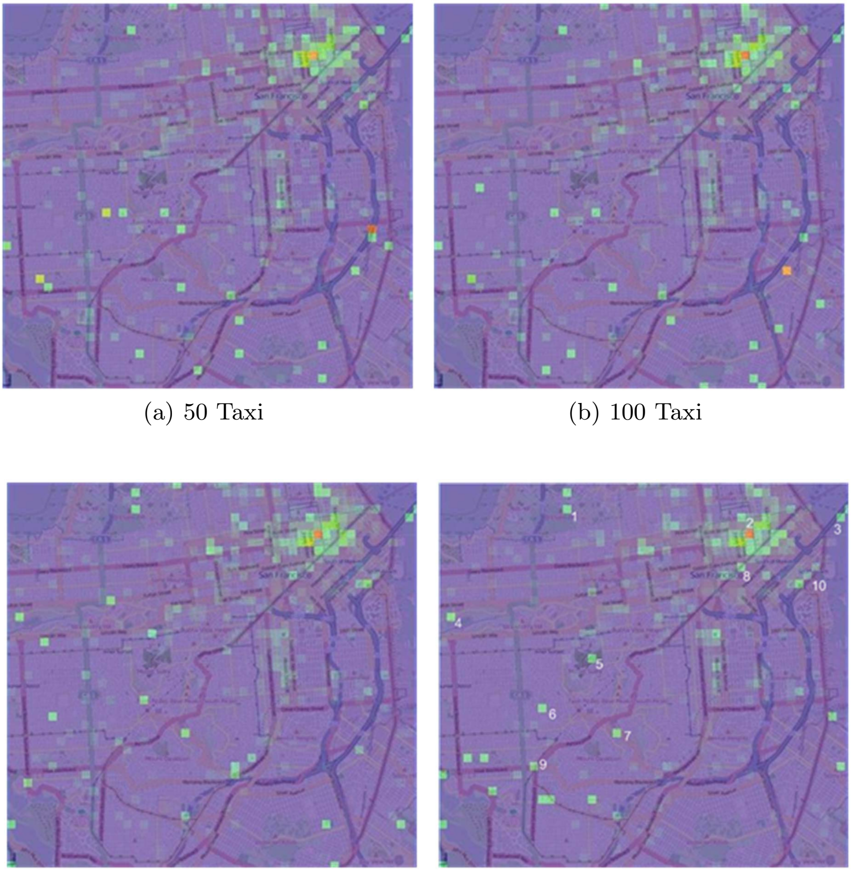
Label	Location	Label	Location
1	San Francisco National Cemetery	6	Golden Gate Heights Park
2	Downtown, Nob Hill	7	School of the Arts
3	Golden Gate Bridge	8	Center BART station
4	Golden Gate Park	9	West Portal
5	University of San Francisco	10	Train Station and AT&T Park

duration between two records is one minute, which reduces the accuracy of the prediction but still preserve the repre-sentativeness of the methods comparison.

357 taxis out of total 536 are selected as the training data (2/3) and others (1/3) are test data. Only the trajectories that are sending passengers are taken into consideration, and there are totally 250,438 trajectories used as history data. The POI map will also be generated from these train trajectories. Fig. 5 gives the snapshots of the distribution of training trajectories. Fig. 5a shows the start pint (red) and end point (blue), and Fig. 5b illustration their density map.

5.2 Generated POI Map

The generated POI map based on different number of trajectory data is given in Fig. 6. The test area is divided into 50*50 sub areas. The weighted PageRank algorithm identifies many local attractive centers (POIs) compared with the density distribution map in Fig. 5b. We can see that the detected POIs are gathered into a stable location along with the increasing of taxi number. These POIs are distributed in highway junc-tion, residential area and some park areas. The experimental result indicates the effectiveness of the proposed method to detect the POIs in the city. Based on the POI map, we will predict the near term driving direction of vehicle.



(a) 50 Taxi

(b) 100 Taxi

(c) 200 Taxi

(d) 375 Taxi

Fig. 6. Generated POI map from different number of Taxis data

Finally, 2/3 of the whole dataset are select to generate the POI map used for pre-diction as Fig. 6d. We identify some of the POIs according to the local map and shown in Table 1. It is shown that the weighted PageRank is effective in POI detec-tion. For example, the most weighted parts are located in the downtown area (Lable 2); some other detected POI areas are scenic spots such as Golden Gate Bridge (Lable 3) and Golden Gate Park (Lable 4); Universities like University of San Francisco (Lable 5) and School of the Arts (Lable 6) are also identified as POI area; Finally, the transportation centers e.g. Center BART station, West Portal and Train Station and AT&T Park (Lable 8,9,10 respectively) are correctly detected as POIs.

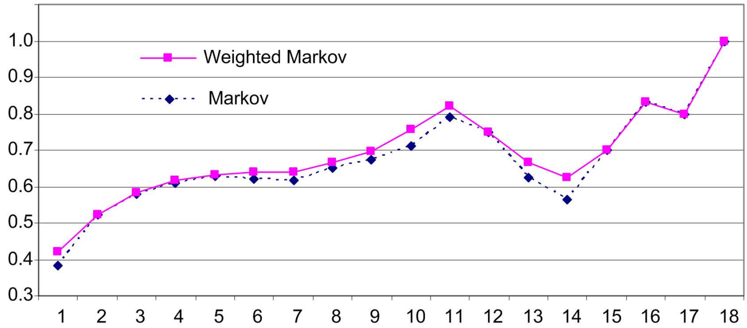


Fig. 7. Prediction accuracy of the original and Weighted Markov Models

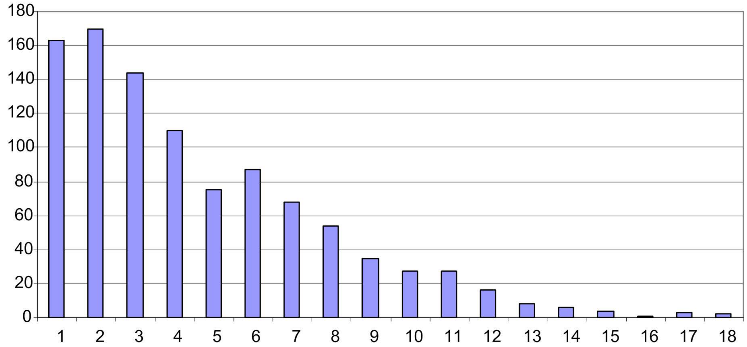


Fig. 8. Number of matched points

5.3 Results of Weighted Markov Prediction

In the prediction experiment, 250,438 trajectories from 357 taxis are used as train data. Set P which contains1000 trajectories from remain taxis is used as test data. We first generate the POI map of the area as shown in Section 5.2. For each trajectory TR_i in P, we randomly set a position c_0^i as the current location, and we will predict the next driving direction $c_0^i c_1^i$ with the proposed weighted Markov method.

The next direction is determined by $HT(k)$ the historical trajectories which have the same direction with TR_i in the last k points as defined in Section 4.3. Therefore, we can calculate the accuracy of prediction by testing if HT(k,1), the historical trajectories which have the same direction with TR_i in the last k points and the next 1 point, have the majority weight (over 50%) of $HT(k)$. For example, in Fig. 4, $HT(4)$ $=TR_0, TR_1, TR_2$, and $HT(4,1)$ $=TR_0$ if $c_1 = c_i$ or $HT(4,1)$ $=TR_1, TR_2$ if $c_1 = c_j$. For the original Markov model, if $c_1 = c_j$ the prediction will be correct since TR_1, TR_2 takes $2/3$ of $HT(4)$; for the weighted Markov model, if $c_1 = c_i$, the prediction will be correct, because TR_0 takes $3/5$ of $HT(4)$; otherwise the prediction will be wrong.

In the prediction, we calculate the prediction accuracy of original Markov model and the proposed weighted Markov model and the results are given in Fig 7. The proposed method increases the prediction accuracy more than 10% for the trajectories with only one point match. The overall accuracy is increased around 5% by applying the weighted Markov model.

The accuracy improvement is mainly gained from the POI map included by weighted Markov model, which actually reflect the attractiveness of the potential destinations according to the experimental results.

We also notice that the prediction accuracy is increasing along the number of matched points. However, as the total number of matched trajectories decreases, the random factor increases. Therefore, we can see the accuracy drop when number of matched points over 12. Fig. 8 demonstrates the number of matched points of the 1000 trajectories (set P) for prediction. For example, we can see that around 160 trajectories in P only have one matched point with the base dataset. Meanwhile there are fewer than 20 trajectories when number of matched points is 12, which is the reason for the drop of prediction accuracy shown in Fig. 7.

6 Conclusions

In this paper, we have presented a framework of weighted Markov model based driving direction prediction with trajectory data. We have performed the analysis about the driving behavior, which leads to the conclusion that POI map can improve the predic-tion accuracy.

We have conducted experiments by comparing prediction accuracy of original Markov model and the proposed weighted Markov model. The experimental results show that weighted PageRank algorithm is effective to compute the city POI map and the POI map can improve the driving direction prediction accuracy.

Overall, we believe that we have provided a method to analysis distribution of POIs in city area and a framework to better predict the near term driving direction. In future research, we will focus on the time related POI map considering different time people may go to different places.

Acknowledgments. This research is supported by National Natural Science Foundation of China under Grants No. 41201486, 61103229 and 71072172, Jiangsu Provincial Colleges and Universities Outstanding S&T Innovation Team Fund under Grants No. 2011013, Key Project of Natural Science Research in Jiangsu Provincial Colleges and Universities under Grants No. 12KJA520001, Key Technologies R&D Program of China under Grants No. SQ2013BAJY4130, National Key Technologies R&D sub Program in 12th five-year-plan under Grants No. SQ2011GX07E03990, the Natural Science Foundation of Jiangsu Province of China under Grant BK2012863, International S&T Cooperation Program of China under Grants No. 2011DFA12910, and Transformation Fund for Agricultural Science and Technology Achievements under Grants No. 2011GB2C100024. Jiangsu Province demonstration project of Internet of things (Yurun Group).

References

[1] Zheng, Y., Zhou, X.: Tri-Training: Exploiting Unlabeled Data Using Three Classifiers. Spinger (2011)
[2] Krumm, J.: A markov model for driver turn prediction. In: World Congress on Society of Automotive Engineers (SAE), Detroit, MI, USA (2008)
[3] Patterson, D.J., Liao, L., Fox, D., Kautz, H.: Inferring High-Level Behavior from Low-Level Sensors. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, pp. 73–89. Springer, Heidelberg (2003)
[4] Li, Z., Han, J., Ding, B., Kays, R.: Mining periodic behaviors of object movements for animal and biological sustainability studies. Data Min. Knowl. Discov. 24(2), 355–386 (2012)
[5] Lee, J.G., Han, J., Li, X., Cheng, H.: Mining discriminative patterns for classifying trajectories on road networks. IEEE Trans. Knowl. Data Eng. 23(5), 713–726 (2011)
[6] Froehlich, J., Krumm, J.: Route prediction from trip observations. In: 2008 World Congress on Society of Automotive Engineers (SAE) (2008)
[7] Krumm, J.: Where will they turn: predicting turn proportions at intersections. Personal Ubiquitous Comput. 14(7), 591–599 (2010)
[8] Jiang, B.: Ranking spaces for predicting human movement in an urban environment. Int. J. Geogr. Inf. Sci. 23(7), 823–837 (2009)
[9] Schonland, A., Williams, P.: Using the internet for travel and tourism survey research: Experiences from the net traveler survey. Journal of Travel Research 35(2), 81–83 (1996)
[10] Ahas, R., Aasa, A., Roose, A., Ülar, M., Silm, S.: Evaluating passive mobile positioning data for tourism surveys: An estonian case study. Tourism Management 29(3), 469–486 (2008)
[11] Lee, C., Greene, D., Cunningham, P.: Detecting grand tours of europe with geotags. In: 2nd Workshop on Computational Social Science and the Wisdom of Crowds at NIPS 2011 (2011)
[12] Brouwers, N., Woehrle, M.: Detecting dwelling in urban environments using gps, wifi, and geolocation measurements. In: Proc. 2nd Int'l Workshop on Sensing Applications on Mobile Phones, pp. 1–5 (November 2011)
[13] Zhu, Y., Zheng, Y., Zhang, L., Santani, D., Xie, X., Yang, Q.: Inferring taxi status using gps trajectories. CoRR abs/1205.4378 (2012)
[14] Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (November 1999); Previous number = SIDL-WP-1999-0120
[15] Jiang, B., Liu, C.: Street-based topological representations and analyses for predicting traffic flow in gis. International Journal of Geographical Information Science 23(9), 1119–1137 (2009)
[16] Xing, W., Ghorbani, A.: Weighted pagerank algorithm. In: CNSR, pp. 305–314. IEEE Computer Society (2004)
[17] Ridings, C., Shishigin, M.: PageRank Convered. Technical report (2006)