Driving Trajectory Prediction Method Based on Adaboost-Markov Model Optimization

Shengnan Song[a], Yongjun Zhang[b]

State Key Laboratory of Information and Optical Communications Beijing University of Post and Telecommunications Beijing, China

[a]sn_song@126.com, [b]yjzhang@bupt.edu.cn

Keywords: Position prediction, trajectory matching rate, Adaboost, multi-order Markov, adaptive weight ratio

Abstract: The driving trajectory prediction method based on the traditional prediction algorithm model has the disadvantages of small prediction accuracy and low matching rate. This paper proposes an improved driving trajectory prediction method based on Adaboost-Markov model. The method adaptively determines the model order m, and uses the Adaboost algorithm to determine the weight coefficients to form a multi-order Markov model. The experimental results show that compared with the fixed-order Markov model, the average prediction accuracy of the Adaboost-Markov model is significantly improved, and it has lower algorithm complexity, which is suitable for vehicle driving trajectory prediction under massive data.

1. Introduction

With the popularization of smart terminal devices and the development of location technology, location-based services have been widely used. Current services are mainly focused on location queries and location sharing. The existing research methods mainly perform position prediction by establishing the historical movement model of the object and the current trajectory matching degree. The commonly used methods are Markov, Hidden Markov, Gaussian mixture model and Kalman filter.

The Markov model is one of the most widely used position prediction models because of its good time-series trajectory data representation ability. The maximum expectation algorithm in the literature [1] solves the problem of low-order Markov prediction accuracy to some extent. The literature [2] proposed that the adaptive variable order Markov model still has the problem of high matching sparsity rate. Literature [3] proposed a position prediction algorithm combining time box and Markov model. In [4], the prediction results of Markov model are modified by introducing trajectory similarity, which improves the accuracy and stability of prediction. Literature [5] proposed a Markov model based on the prefix projection database. This method has a large time cost for establishing a projection database. Literature [6] proposed the n-MMC (n-Mobility MarkovChain) model to predict user location. The literature [7] proposed the MyWay method for position prediction of mobile users, but this method did not preprocess the original trajectory, resulting in too much computation.

Aiming at the problems existing in the existing methods, this paper firstly uses the method of trajectory division and density clustering to discretize the original trajectory data into various interest regions of mobile users, and then uses Adaboost algorithm to determine the weight coefficient for multi-order fusion Markov model. On the basis of making full use of the historical trajectory sequence, the prediction accuracy is improved and the universality of the model is guaranteed.

2. Track preprocessing and model definition

2.1 Trajectory division and point of interest extraction

Definition 1 sub-trajectory: The sub-track of the track T_{Ji} is expressed as $ST_{Ji}= \{P_{s1}, P_{s2}... P_{sk}\}$ $(1\leq s_1<s_2<...<s_k\leq m)$. And the sub-tracks mentioned in this paper are all continuous sub-tracks, that is, from Ps1 to P_{sk} are continuous track sampling points.

Definition 2 angle offset: As shown in Figure 1, taking the path T_{Ji} as an example, P_1 is the initial track point and P_1P_2 is the initial moving behavior. The specific method is as follows:

$$A=\sin^2\left(\frac{Lat_j-Lat_i}{2}\right)$$

$$B=\cos\left(Lng_i\right)\cos\left(Lng_j\right)\sin^2\left(\frac{Lng_j-Lng_i}{2}\right) \tag{1}$$

$$d\left(P_i,P_j\right)=2R\arcsin\sqrt{A+B}$$

d (P_i, P_j) is the distance between the track sampling points P_i and P_j, and R is the radius of the earth.

The formula for the track angle offset is as follows:

$$\alpha_i=\arccos\frac{\left(\left|P_iP_{i+1}\right|^2+\left|P_{i+1}P_{i+2}\right|^2-\left|P_iP_{i+2}\right|^2\right)}{2\left|P_iP_{i+1}\right|^2\left|P_{i+1}P_{i+2}\right|^2} \tag{2}$$

$$\theta_i=\begin{cases}\alpha\text{-}\pi, P_iP_{i+1}*P_{i+1}P_{i+2}<0\\ \pi\text{-}\alpha, P_iP_{i+1}*P_{i+1}P_{i+2}\geq 0\end{cases} \tag{3}$$

Definition 3 Distance offset: The distance offset is the vertical distance from the trajectory sample point to the line where the initial movement behavior is. As shown in Figure 1, the vertical distances d_1 and d_2 from the sampling points P_3 and P_4 to the initial movement behavior P_1P_2 extension are the distance offset. Its calculation formula is as follows:

$$d_i=\left|P_2P_{i+2}\right|\sin\angle P_1P_2P_{i+2} \tag{4}$$

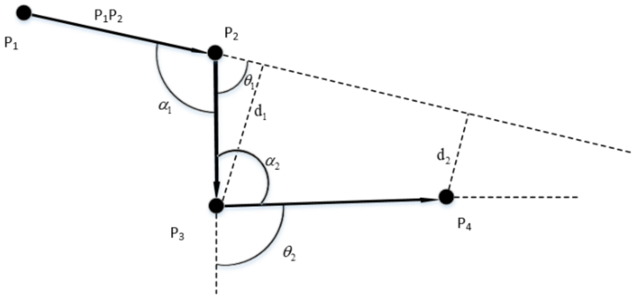

Figure 1. Example of angle and distance offset.

For the trajectory T_{Ji}, first, starting from the sampling point P_3, calculate the corner offset θ_1 and the distance offset degree d1. If the absolute value of θ_1 does not exceed the corner offset threshold θ_{th}, and d_1 does not exceed the distance offset threshold d_{th}, then continue to calculate the offset of the sampling point P_4, and so on. Then, this paper uses the improved density peak clustering algorithm [7] to cluster the user feature points. All the feature points in a cluster represent an interest region of the user, and use the centroid of the cluster as the region of interest representative:

$$c_i=\frac{1}{\left|C_i\right|}\sum_{P\in C_i}\left(Lat_j,Lng_j\right) \tag{5}$$

c_i represents the centroid point of class cluster i, $|C_i|$ represents the number of points of interest contained in class cluster i, P_j represents a point of interest in class cluster i, and Lat_j and Lng_j represent the latitude and longitude of the point of interest respectively.

2.2 Model state definition

Definition 4 Markov chain: Assume that there is a stochastic process $\{X_n, n \in T\}$, and there are finite states $i_0, i_1, ..., in \in I$, if $P\{X_{n+1}=i_{n+1}| X_0=i_0, X_1=i_1, ..., X_n=i_n\} = P\{X_{n+1}=i_{n+1}|X_{n=in}\}$, then $\{X_n, n \in T\}$ is the Markov chain, i_e the next object The state is only related to the state at this time, also known as the 1st order Markov chain.

Definition 5 Trajectory sequence: The user's trajectory sequence is a discretized representation of its original trajectory after trajectory preprocessing and clustering, and consists of various regions of interest extracted by clustering.

Definition 6 Prefix trajectory sequence: Suppose a given trajectory sequence $T_{Si}=\{C_{k1}, C_{k2}, ..., C_{kj}\}(1\leq i\leq n)$, and there is $1<l < j$, to predict which C_{kl} is The region of interest, called $\{C_{k1}, C_{k2}, ..., C_k)\}$ is its corresponding prefix trajectory sequence, and C_k is called the user's current interest region.

Definition 7 Trajectory Markov chain: The state space of the trajectory Markov chain is defined by the user's region of interest $C_1, C_2, ..., C_n$, if the user's next region of interest depends on the current region of interest and the past k-1 regions of interest, then the Markov chain is called the k-order trajectory Cove chain, namely:

$$C_p = \arg_{C_{n+1}} \max\{p(C_{n+1}|C_{n-k+1}, C_{n-k+2}, ..., C_n)\} \tag{6}$$

Among them, C_p represents the prediction result of the user's next interest area, and Cn represents the current interest area of the user.

When the sequence is matched, it may cause the matching to fail or only match a very small number of historical trajectories, that is, the matching is sparse, which may cause the prediction performance of the model to decrease. As shown in Figure 2, taking a user's historical trajectory sequence pattern tree as an example, a third-order Markov model is assumed, the prefix trajectory sequence is $C_2 \rightarrow C_4 \rightarrow C_5$, and the user's next region of interest is predicted based on the mobile pattern tree. . However, there is no historical trajectory with $C_2 \rightarrow C_4 \rightarrow C_5$ as the prefix sequence in this pattern tree, and the third-order Markov model cannot make predictions.

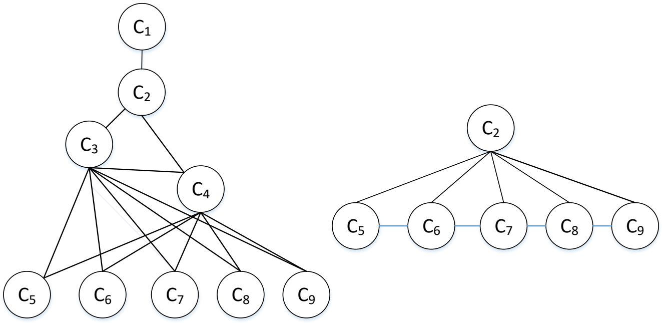

Figure 2. User mobile history mode tree.

3. Adaboost-Markov modeling

This paper proposes a mobile user location prediction method based on Adaboost-Markov model. As a representative lifting method, Adaboost algorithm can combine a plurality of weak classifiers to generate a strong classifier by changing the probability distribution of training data and the weighting coefficient of weak classifier [8]. In this paper, the model order k is adaptively determined, and the 1st order to kth order Markov model is used as k weak predictors. The probability distribution of the user trajectory data and the weight coefficients of each order Markov model are changed by the Adaboost algorithm, and finally a multi-order fusion Markov is generated. The model is used to predict user location.

The order k is determined by the maximum matching step length of the user prefix trajectory sequence and the historical trajectory sequence. Taking Figure 3 as an example, the user's original trajectory data is preprocessed to establish a historical trajectory sequence library. The length of the prefix trajectory sequence that participates in matching is usually determined by two methods. One is to artificially specify the length of the prefix track sequence to participate in the matching. The user determines the number of prefix track sequence elements participating in the matching and inputs the sequence of the predicted prefix track to be corresponding length, and continue to find a match in the historical track sequence library and repeat the above steps until the match is successful.

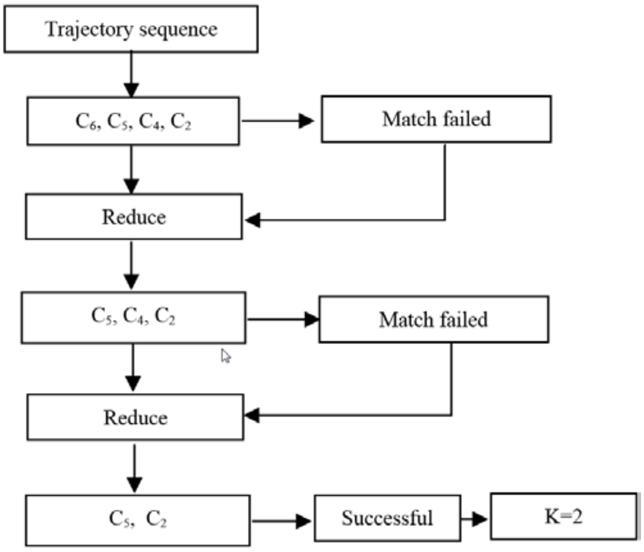

Figure 3. Example of adaptive determination of order k.

After the model order k is determined, the first-order to k-order Markov model is merged, that is, the Adaboost-Markov model is composed of m-order (m=1, 2,..., k) Markov models, and the weight coefficients α_m of each order model are The prediction error size of the model is negatively correlated: the larger the prediction error, the smaller the weight coefficient of the model, and vice versa. For trajectory training samples with N elements, the prediction error e_m of the m-order (m=1, 2... k) Markov model is calculated as follows:

$$e_m = \sum_{i=1}^{N} w_{mi} I_m^i \tag{7}$$

Among them, w_{mi} is the weight of the training samples corresponding to the m-order (m=1, 2... k) Markov model, and there is usually 1/N when the accident occurs. It is the predictor of the training result i of the m-order Markov model, and its formula is as shown in (8):

$$I_m^i = \begin{cases} 1, & \text{error prediction} \\ 0, & \text{correct prediction} \end{cases} \tag{8}$$

After calculating the prediction error e_m, the model weight coefficient α_m can be calculated, and the calculation method is as shown in formula (9):

$$\alpha_m = \frac{1}{2}\log\frac{1\text{-}e_m}{e_m} \tag{9}$$

Then, according to the weight coefficient of the m-th model obtained by the above method, the weight of the training sample is updated, and the weight w_{m+1} of the training sample corresponding to the m+1 order (m=1, 2... k-1) Markov model is calculated. The calculation method is as shown in (10) and (11):

$$w_{m+1,\ i} = \frac{w_{mi}}{Z_m}\exp\left(\alpha_m I_m^i\right) \tag{10}$$

$$Z_m = \sum_{i=1}^{N} w_{mi}\exp\left(\alpha_m I_m^i\right) \tag{11}$$

Finally, the weighting coefficients of the various models are normalized:

$$\beta_{\mathrm{m}}=\frac{\alpha_{\mathrm{m}}}{\sum_{\mathrm{m}=1}^{\mathrm{k}}\alpha_{\mathrm{m}}} \tag{12}$$

The Adaboost-Markov model can be expressed as follows:

$$\mathrm{G}(\mathrm{x})=\sum_{\mathrm{m}=1}^{\mathrm{k}}\beta_{\mathrm{m}}\mathrm{G}_{\mathrm{m}}(\mathrm{x}) \tag{13}$$

4. Test analysis

The trajectory data set used in the experiment was from the “Beijing Water Affairs Bureau Drainage Center Information System—Sludge Flow Monitoring System”, and the data for one year of 2017-2018 trial operation. Contains sludge trajectory data of 25 reclaimed water plants and 19 sludge disposal sites in Beijing.

Table.1. Table Type Styles

Data category	***Quantity***
All transfer track list	10896
Violation order	768
Total distance of the track	329021
Total track duration	9431

In order to verify the validity of the model, this paper compares the four models: the first-order Markov model, the second-order Markov model, the multi-order fusion Markov model with weighted coefficient averaging, and the Adaboost-Markov model proposed in this paper. Among them, the difference between the multi-order fusion Markov model of the weight coefficient and the Adaboost-Markov model is that the former assigns the same weight coefficient to the 1~k order Markov model, while the latter uses the Adaboost algorithm to adjust the corresponding weight according to the prediction error of each order model. Coefficient. A random extraction of 90% from the sludge vehicle trajectory data set is used to train the above model, and the remaining 10% of the trajectory data is used to detect the prediction performance.

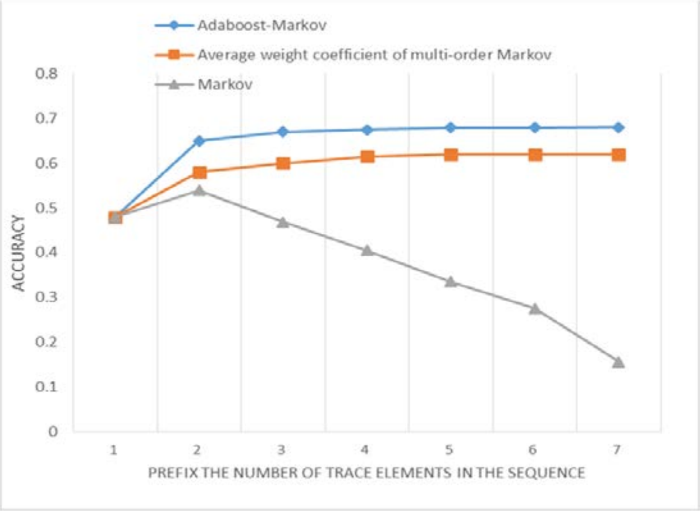

Figure 4. Relationship between number of prefix trajectory sequence elements and predictionaccuracy.

As shown in Fig. 4, the horizontal axis represents the number of elements of the prefix track sequence, and the vertical axis represents the prediction accuracy. For the common Markov model, the number of prefix elements is equivalent to the model order, and the Adaboost-Markov model and the weighted coefficient average multi-order fusion Markov model adopts the adaptive method to determine the model order k. Therefore, with the increase of the number of prefix elements, the prediction accuracy of the ordinary Markov model firstly shows an increasing trend, reaching the maximum value at the 2nd order, and then the matching thinning rate is gradually increased due to the increase of the order, resulting in accurate prediction. The rate is gradually decreasing.

Then, the above four models are used for single-step prediction, that is, the user's next region of interest is predicted. Figures 5 and 6 respectively show the prediction effects of the four models on the small-scale trajectory data set and the large-scale trajectory data set.

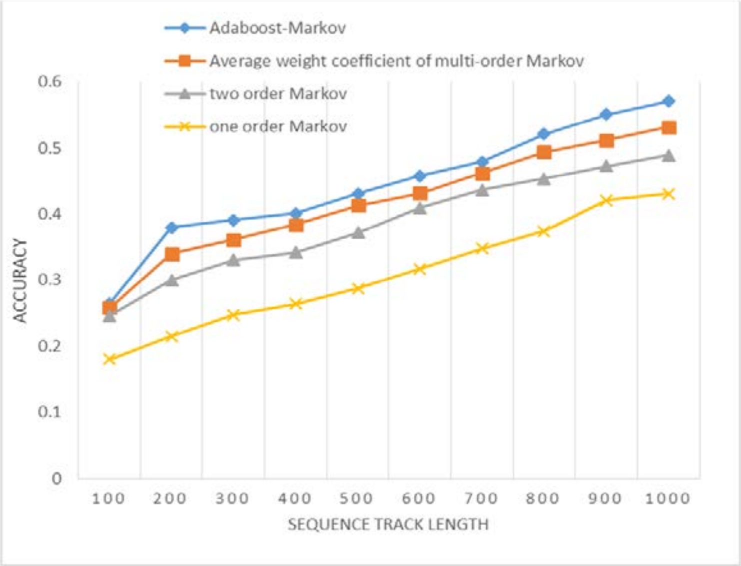

Figure 5. Prediction accuracy comparison under small-scale trajectory data set.

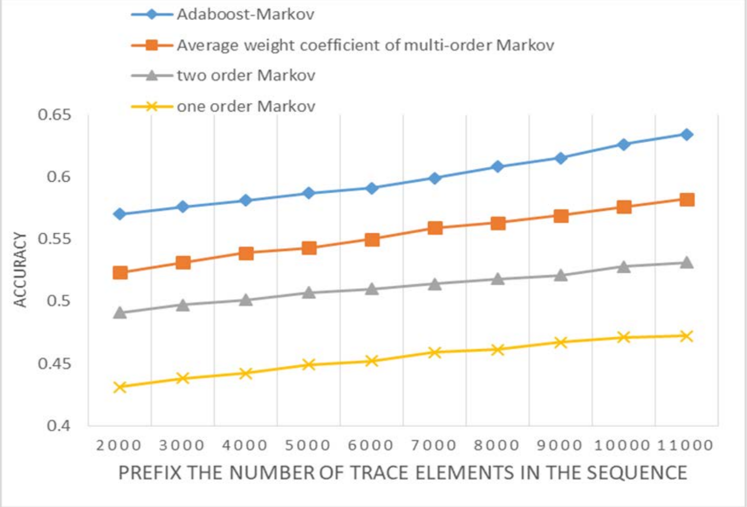

Figure 6. Prediction accuracy comparison under large-scale trajectory data set.

It can be found that the accuracy of the Adaboost-Markov model proposed in this paper is significantly higher than that of the other three models: the experiment of small-scale trajectory dataset, the first-order Markov model, the second-order Markov model, and the weight. Compared with the multi-order fusion Markov model with coefficient average, the average prediction accuracy of Adaboost-Markov model increased by 39.73%, 18.3% and 9.12%, respectively. In the experiment of large-scale trajectory dataset, the prediction accuracy increased by 20.83%. 11.3% and 5.38%.

5. Conclusion

In this paper, the position prediction of Markov model has the disadvantages of low prediction accuracy and matching sparseness. The combination of trajectory division and density clustering is used as the trajectory data preprocessing method. The combination of Adaboost algorithm and multi-order fusion Markov model is proposed. The prediction method makes full use of the historical trajectory sequence and improves the prediction accuracy. It has high prediction accuracy and good universality. In the future work, the above model will be further optimized to study the spatio-temporal and group characteristics of trajectory data in more depth, taking into account factors such as weather, time (working days and rest days) and user-related social data, with a view to further Improve the prediction accuracy of the model. The text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

Acknowledgments

The communication author of this paper is Yongjun Zhang (yjzhang@bupt.edu.cn).

References

[1] WIDHALM P, NITSCHE P, BRANDIE N. Transport mode detection with realistic smartphone sensor data [C] // Proceedings of the 21stInternational Conference on Pattern Recognition. Piscataway, NJ: IEEE, 2012: 573-576.

[2] SHIH D H, SHIH M H, YEN D C, et al. Personal mobility pattern mining and anomaly detection in the GPS era [J]. American Cartographer, 2016, 43 (1): 55-67.

[3] GUNDUZ S, YAVANOGLU U, SAGIROGLU S. Predicting next location of twitter users for surveillance [C] // Proceedings of the 12thInternational Conference on Machine Learning and Applications.Washington, DC: IEEE Computer Society, 2013: 267-273.

[4] BOGOMOLOV A, LEPRI B, STAIANO J, et al. Once upon a crime: towards crime prediction from demographics and mobile data [J]. Eprint Arxiv, 2014: 427-434.

[5] QIAO S, HAN N, and ZHU W, et al. TraPlan: an effective three-in-one trajectory-prediction model in transportation networks [J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16 (3): 1188-1198.

[6] YU X G, LIU Y H, WEI Det al. Hybrid Markov model for mobile path prediction [J]. Journal on communications 2006, 27 (12): 61-69.

[7] LV M Q, CHENL, CHEN G C. Position prediction based on adaptive multi-order Markov model [J]. Journal of Computer Research and Development, 2010, 47 (10): 1764-1770.

[8] CHEN M, YU X, LIU Y. Mining moving patterns for predicting next location [J]. Information Systems, 2015, 54 (C): 156-168.