A Destination Prediction Algorithm using Spatial Temporal Bidirectional LSTM Networks

Zhaoxu Lu[1[0000-0002-4729-0063]], Fang Zhao[1[0000-0002-4784-5778]], Haiyong Luo[2[0000-0001-6827-4225]], Qu Wang[1[0000-0001-6551-6807]] and Wenhua Shao[1[0000-0002-2440-9981]]

[1] Beijing University of Posts and Telecommunication, Beijing 100876, China
[2] Chinese Academy of Sciences, Beijing 100190, China
yhluo@ict.ac.cn

Abstract. Destination prediction provides important support for many location-based applications, such as urban resource dispatching, targeted advertising, etc. Destination prediction using sparse and partial movement trajectories poses many challenges. The traditional destination prediction methods using Markov model suffer from the data sparsity problem. Though the recurrent neural network (RNN) based destination prediction can handle the data sparsity problem, it only focuses on the context relationship between sequences and ignores the spatial-temporal information behind trajectories. In this paper, we introduce a spatial temporal bidirectional long-short term memory (ST-BiLSTM) network to destination prediction. This proposed method not only makes advantage of LSTM to handle data sparsity and long-term dependencies, but also employs the bidirectional structure of LSTM to effectively model the beginning and end of the sequence, which have greater correlation between sequences. Furthermore, we embed the spatial-temporal factors into the gates equations to further boost the prediction accuracy of the model. Experimental results on the taxi trajectory dataset in the city of Porto, demonstrate that our proposed algorithm outperforms the standard LSTM and BiLSTM models with more than 15% and 10% accuracies, respectively.

Keywords: Destination Prediction, BiLSTM, Spatial-Temporal Embedding.

1 Introduction

With the wide application of navigation and positioning technology, it is possible to study the law of target trajectory motion. Driven by a large number of available trajectory data, destination prediction has become a hot topic. Since the Markov model can represent time series data well, it is widely used for position modeling and prediction. By dividing the corresponding position into a unified grid unit or path segment and using it as various states of the Markov process, researches[1, 2, 3] have achieved good results. Y. Di et al [4] calculated the mobile behavior similarity and then clustered the mobile behavior similarity and used the first-order Markov model for location prediction on the clustered user groups. Research [5] proposed Markov model based on Gaussian analysis to solve the rough problem of prediction results caused by equivalent division of positional time points. But the low-order Markov algorithm can only take advantage of a few recent steps, and the high-order Markov has high computational complexity and serious zero-frequency problems.

Zhang J et al.[6] uses MART (Multiple Additive Regression Trees) to predict the location of municipal shared bicycles. In particular, De [7] formulated the destination prediction as a regression problem and solved it by using a multilayer perceptron (MLP). Furthermore, with the great success of Recurrent Neural Network (RNN) in natural language processing[8], many scholars have applied it to the study of trajectory sequences. Liu Q et al.[9] firstly applied RNN to location prediction research. They used ST-RNN to extend Spatial-Temporal information into the recurrent neural network, and recommended interest points and achieved satisfactory results.

In this paper, we focus on trajectory prediction using long short-term memory network. Inspired by the work of Liu Q, we also have embedded the spatial-temporal factors into LSTM cells to make full use of data. In the sequence-based prediction process, the beginning and end of the sequence tend to be more correlated with the sequence[10], so we proposed a bidirectional network structure to enhance the learning of the two parts. The contributions of this paper are summarized as follows:

- We proposed a bidirectional network architecture to enhance the learning of the beginning and end of the trajectory sequence.
- We adopt the ST-BiLSTM model which embedding spatial-temporal information in net cells to boost the prediction accuracy.
- By conducting experiments on real taxi datasets, our algorithm outperforms the standard LSTM and BiLSTM models with more than 15% and 10% accuracies, respectively.

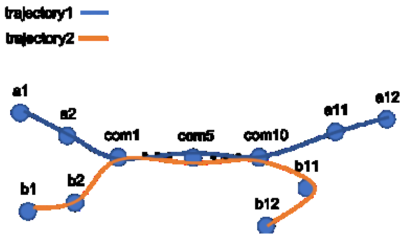
2 Spatial Temporal Bidirectional LSTM Network

In this section we will introduce the structure of our ST-BiLSTM model and then elaborate the learning and prediction process of the entire model.

2.1 Bi-LSTM

Compared with the standard RNN unit, long short-term memory [11] effectively avoid the vanishing gradient problem by introducing the gate mechanism, which is advantageous in dealing with long-term dependencies.

During the prediction process based on trajectory sequence, the most relevant parts of the prefix are its beginning and its end [8]. As is shown in the Fig. 1, trajectories T1 and T2 have a long common sequence (com1, com2, … , com10), but the start and end parts of them are significantly different. In traditional prediction tasks based on recurrent neural network, for long sequence predictions, the model's output is greatly affected by the tail, and the beginning part of the sequence may be forgotten. So these methods tend to recognize T1 and T2 as similar trajectories, which is obviously inconsistent with the facts. The Bidirectional Recurrent Neural Network can solve this problem very well [12]. In Bidirectional Recurrent Neural Network shown in Fig. 2, the first layer reads the sequence sequentially, the second layer reads it conversely, and finally the two layers of the output layer are concatenated and fed back to the output layer. Thereby it could obtain more information about the head and tail of the trajectory sequence[4].



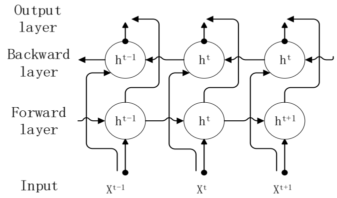


Fig. 1. Trajectories with long common sequences. **Fig. 2.** Bidirectional network architecture.

2.2 Bi-LSTM with Spatial-Temporal Information

Theoretically, the LSTM network can obtain good training and prediction effects in a rich and well-distributed data set, but the real data set is often very dense in some areas, while in other some areas are sparse. Meanwhile, as mentioned in former sections, different time spot or location have different impacts on trajectory trend.

We argue that temporal and spatial influences can work as implicit information to guide the learning of gate mechanism. The introduction of Spatial-Temporal information during the training process can also speed up the training. We directly add the spatiotemporal transfer information to each gate mechanism function of LSTM and the equations are as follows：

$$g_t = tanh(D_{xt}W_{xg} + h_{t-1}W_{hg} + F(s_{t-1}, q_{t-1}) + b_g) \tag{1}$$

$$i_t = \sigma(D_{xt}W_{xi} + h_{t-1}W_{hi} + F(s_{t-1}, q_{t-1}) + b_i) \tag{2}$$

$$f_t = \sigma(D_{xt}W_{xf} + h_{t-1}W_{hf} + F(s_{t-1}, q_{t-1}) + b_f) \tag{3}$$

where s, q∈Rd are d-dimensional vector and s0, q0 are 0 and St-1, qt-1 represent the transfer vector from lt-1 to it time interval, respectively. Finally, the function F is used to calculate the two influencing factors to obtain the overall influencing factor vector, where F chooses a simple linear addition. Msk, Mqk∈R|c|*d represent the space-time transfer matrix, respectively, c is the grid cell size.

$$F_k(s_{t-1}, q_{t-1}) = M_{sks_t} + M_{qkq_{t-1}}, k = i, f, o \tag{4}$$

We use the following formula to calculate the Spatial-Temporal transfer matrix by dividing the region and time into fixed intervals. At a given time interval ti and distance interval di, the corresponding time transfer matrix Mti and distance transfer matrix Mdi can be calculated by the following formulas, respectively.

$$\mathrm{M_{ti}} = \frac{\left[M_{L(t_i)}(U(t_i) - t_i) + M_{U(t_i)}(t_i - L(t_i))\right]}{\left[(U(t_i) - t_i) + (t_i - L(t_i))\right]} \tag{5}$$

$$\mathrm{M_{di}} = \frac{\left[M_{L(d_i)}(U(d_i) - d_i) + M_{U(d_i)}(d_i - L(d_i))\right]}{\left[(U(d_i) - d_i) + (d_i - L(d_i))\right]} \tag{6}$$

where U(ti), L(ti) represent the upper and lower bounds of the time period ti, U(di) and L(di) represent the upper and lower bounds of the distance segment di. In this way, the transition probability matrix can be obtained from a continuous data set, and it also provides a better solution for modelling continuous spatial-temporal factors.

3 Destination Prediction

3.1 Encoding with Fixed Window

Given a trajectory sequence Ti = {p1, p2, p3, ..., pn-1, pn}, we first model the sequence {p1, p2, p3, ..., pn-1} as input using ST-BiLSTM and get two vector Hifn-1, Hib1. Hifn-1, Hib1 are the last-time outputs of the forward and backward LSTM layer respectively. And then, Hifn-1 and Hib1 are combined into a vector Vi which is used as the encoded part of the network to represent the sequence Ti.
Please note that the first paragraph of a section or subsection is not indented. The first paragraphs that follows a table, figure, equation etc. does not have an indent, either.

In order to better enable the model to recognize these short-term dependencies, the idea of sliding windows is introduced during the training process. We propose the input of the model is not a single position but a window of k successive GPS points of the trajectory. So that a sliding window with K is trained down the trajectory sequence instead of the previous one. The encoding process is shown in Table 1, where LSTMf and LSTMb are the forward and backward recurrent neural layer, respectively.

Table 1. Encoding Procedure.

	Encoding Procedure
1.	$H^i_{fi} = 0$, $H^i_{bn-1-i}=0$, i=0
2.	for T^i_j in T^i:
3.	$T^i_j = Wi \cdot T^i_j$
4.	$H^i_{fj} = LSTM^f(H^i_{fj-1}, T^i_j)$
5.	for T^i_j in T^r_i:
6.	$T^i_j = Wi \cdot T^i_j$
7.	$H^i_{bj} = LSTM^b(H^i_{bj-1}, T^i_j)$
8.	return concat(H^i_{fn-1}, H^i_{b1})

3.2 Destination Prediction by Decoding

We use the three-layer fully connected network as the decoding part to obtain the output of the model, and then we will describe the detail of this decoding structure. Given a trajectory sequence Ti={p1, p2, p3, ..., pn-1, pn}, we use the coding structure of the previous layer to encode the vector Vi, and the layer structure uses Vi. As an input to the network. Since the meta-data contained in the trajectory also plays a key role in predicting destination, they also should be added to the input of decoding structure.

Inspired by the word vectors generator in NLP, we trained each part of the metadata of these trajectory to a word vector table. These word vectors are combined with the vector Vi obtained by the Bi-LSTM layer to serve as input for decoding the partially connected network. The final model architecture is shown in Fig. 3.

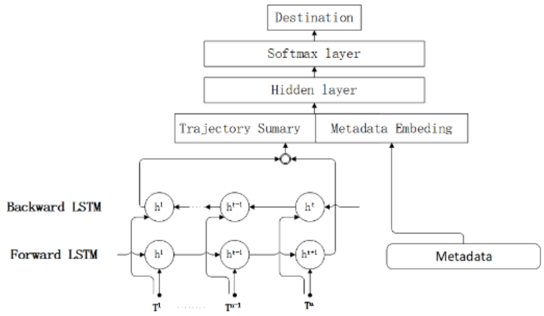


Fig. 3. The architecture of ST-BiLSTM

The final position will be calculated by the following equation. Where Pd is obtained by the softmax layer which represents the probability distribution of each grid. And the cell i whose probability is Pdi is the final prediction result.

$$P^d{}_i = \max(P^d) \tag{7}$$

4 Experiments and Results Analysis

4.1 Data and Experiment Settings

Dataset. In this paper we use a taxi trajectory dataset in the city of Porto. It's also published by Kaggle 2015 and contains more than 1.7 million track information of 442 taxis from 2013-07-01 to 2014-06-30. The trajectory information is composed of a series of GPS point sequences. we used the first 70% for training and the remaining 30% for test.

Experiment Settings. We divided the research area into the specification 60*80 grid cells, so each cell represents 0.71*0.74 km2. And we also set other size 70*90 and 40*60 as the comparison. During the training process, the batch_size is set to 75, the learning rate learning_rate is 0.01, and the stochastic gradient descent (SGD) is used to minimize the objective function value.

4.2 Comparison Methods

In order to further verify the effectiveness of the ST-BiLSTM algorithm introduced in this paper, we compare it with the following algorithms for prediction results：

• MLP [7]. Based on the fully connected neural network and combined with the idea of regression, it predicts the taxi end point with latitude, longitude directly and achieved remarkable precision. This is also the baseline method in our paper;

• LSTM [11]. LSTM has more advantages in the learning of long sequences than RNN, it is natural to consider using it for the prediction tasks of this paper, and as a comparison;

• BiLSTM. The bidirectional structure can better learn the begin and end parts of the sequence while maintaining the advantages of the LSTM itself;

• ST-RNN [9]. ST-RNN makes recommendation of interest points by combining time-space transfer matrix in RNN, and achieves better precision.

4.3 Evaluation

We adopt the mean haversine distance to evaluation our model prediction accuracy, which is defined as follows:

$$d_{haver\sin e}(x, y) = 2R\arctan\left(\sqrt{\frac{a(x, y)}{a(x, y)-1}}\right) \tag{8}$$

Where R is the radius of the earth, and a(x, y) is defined as follows (lon_x is the longitude of point x, and lat_y is its longitude):

$$a(x,y) = \sin^2\left(\frac{lat_y - lat_x}{2}\right) + \cos(lat_y)\cos(lat_x)\sin^2\left(\frac{lon_y - lon_x}{2}\right) \tag{9}$$

In order to further verify the effectiveness of the proposed algorithm, based on the semi-positive distance, the dis@k distance is used as the evaluation standard. The dis@k is defined as follows：

$$dis@k = \min(d(x_1, y), d(x_2, y), ..., d(x_k, y)) \tag{10}$$

Where d is the mean haversine distance, k = 1, 2, 3 in this paper. Obviously, when k is 1, dis@k is same as the original mean haversine distance evaluation.

4.4 Results and Analysis

The Influence of Network Architecture. From the comparison of RNN and LSTM models, LSTM converges earlier than RNN, and the prediction accuracy is more prominent. In terms of network structure, in Fig. 4 and Table 2, ST-BiLSTM and Bi-LSTM with the bidirectional structure are superior in accuracy to their corresponding single layer models ST-LSTM and LSTM. These prove that LSTM is superior to the RNN unit in the research content of this paper, and the bidirectional structure is indeed superior to the single layer structure in terms of prediction accuracy.

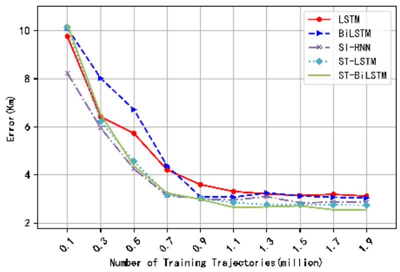


Fig. 4. Prediction accuracy under different methods.

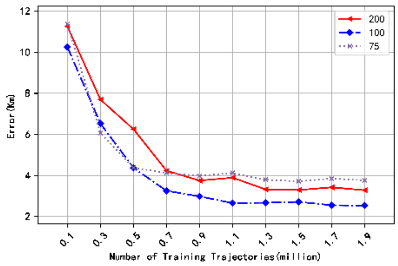


Fig 5. Prediction accuracy under different grid size.

Table 2. The prediction error comparison at dis@k evaluation

Method	dis@1	dis@3	dis@5
MLP	2.81	-	-
LSTM	3.12	3.10	3.07
BiLSTM	3.04	2.97	2.89
ST-RNN	2.86	2.19	2.72
ST-LSTM	2.74	2.70	2.63
ST-BiLSTM	2.53	2.40	2.44

The Influence of Spatial-Temporal Embedding. To fully consider the influence of space-time factors on the overall prediction environment, we add time and space factors to RNN, LSTM, and BiLSTM respectively. The experimental results are shown in Fig. 4 and Table 2. The prediction accuracy of each algorithm after embedding time and space factors has improved. Moreover, ST-RNN, ST-LSTM, ST-BiLSTM prediction accuracy is higher than MLP under top5 prediction point.

The Influence of Grid Size. Theoretically, the smaller the grid cell partition is, the higher the prediction accuracy is. However, the smaller cell division in the actual training process means higher training cost of the network which will lead to the lower accuracy. As shown in Fig. 5, the final prediction accuracy of 60*80's granularity is higher than 40*60, indicating that the prediction accuracy can be improved by improving the grid cell partition specification to some extent. On the other hand, if the grid specification is too intensive, the prediction accuracy will be reduced. In the Figure 7, the prediction accuracy under the 70*90 specification is not as good as the 60*80 specification.

5 Conclusion

We propose a ST-BiLSTM model based on LSTM to adopt a bidirectional structure and embed spatial-temporal factors. On the one hand, by introducing spatial-temporal factors to overcome the influence of data sparseness in partial regions, on the other hand, the introduction of bidirectional structure also enables the start and ending parts of long trajectories to be better studied. The model of this paper is tested on the taxi trajectory dataset int the city of Porto, and the experimental results also fully proves the validity of the model. In the future, we should collect other datasets as much as possible for verification, and further accelerate the training process.

6 Acknowledgment

This work was supported in part by the National Key Research and Development Program (2018YFB0505200), the National Natural Science Foundation of China (61872046 and 61671077), and the Open Project of the Beijing Key Laboratory of Mobile Computing and Pervasive Device.

References

1. S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, “Next place prediction using mobility Markov chains,” 2012.
2. J. A. Alvarez-Garcia, J. A. Ortega, L. Gonzalez-Abril, and F. Velasco, “Trip destination prediction based on past GPS log using a Hidden Markov Model,” Expert Syst. Appl., 2010.
3. R. Simmons, B. Browning, Yilu Zhang, and V. Sadekar, “Learning to Predict Driver Route and Destination Intent,” 2008.
4. A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures.,” Neural Netw., vol. 18, no. 5–6, pp. 602–10, 2005.
5. Z. D. Qiao Yan-lei, DU Yong-ping, “A Location Prediction Method of Markov Based on Gaussian Analysis,” Comput. Technol. Dev., vol. 28, no. 1, pp. 41–44, 2018.
6. J. Zhang, X. Pan, M. Li, and P. S. Yu, “Bicycle-sharing system analysis and trip prediction,” in Proceedings - IEEE International Conference on Mobile Data Management, 2016.
7. A. De Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio, “Artificial neural networks applied to taxi destination prediction,” in CEUR Workshop Proceedings, 2015.
8. T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, “Rnnlm-recurrent neural network language modeling toolkit,” 2011 ASRU Work., 2011.
9. Q. Liu, S. Wu, L. Wang, and T. Tan, “Predicting the Next Location : A Recurrent Model with Spatial and Temporal Contexts,” AAAI, 2017.
10. A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures.,” Neural Netw., vol. 18, no. 5–6, pp. 602–10, 2005.
11. S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., 1997.
12. M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., 1997.