Recurrent Neural Network with Attention-based Model for Predicting Urban Vehicle Trajectories

1S.k . Sahoo, 1 L Pattnaik, 1P Dash, 1S.Behera, 1N.Simran 1Subhadra Nayak

1Department of Electrical & Electronics Engineering, GITAM , BHUBANESWAR ODISHA

2 Student of Department of Electrical & Electronics Engineering GITAM , BHUBANESWAR ODISHA

Abstract

Massive amounts of spatial and temporal information data are gathered and amassed as the variety of positioning sensors and location-based devices grows. By joining the data points in a chronological order, these data—which include movement information for any moving object—are expressed as trajectory data. In particular, this study uses vehicle trajectory data from the urban traffic network to explore the prediction of urban vehicle trajectories. Recurrent neural network model for urban vehicle trajectory prediction is proposed in the earlier work. In this work, we present the Attention-based Recurrent Neural Network model for urban vehicle trajectory prediction as a means of further improving the model. The attention mechanism in this suggested model is used to incorporate network traffic state data into the trajectory prediction of urban vehicles. The Bluetooth data that was gathered in Brisbane, Australia, which incorporates private vehicle movement information, is used to assess the model. Five metrics are used to assess the model's performance: BLEU-1, BLEU-2, BLEU-3, BLEU-4, and METEOR. The outcome demonstrates that the ARNN model performs better than the RNN model.

Keywords: Vehicle Trajectory; Trajectory Prediction; Recurrent Neural Network; Attention Mechanism; Network Traffic State

1.Introduction

Trajectory data, which is a large amount of location data acquired with different location sensors and location-aware gadgets, is researched. An object's trajectory is a record of its movement across space. A location sequence arranged chronologically serves as a representation of this. We concentrate on one kind of trajectory data in our study: data on urban vehicles. One kind of trajectory data that depicts the motions of vehicles in urban networks is the urban vehicle trajectory data. This data provides options for comprehending urban traffic network movement patterns by providing a wealth of information regarding aggregate flows and disaggregate travel behaviors, including user-centric traffic experiences and system-wide mobility patterns.

This work focuses on the trajectory-based location prediction problem, one of the many uses of trajectory data mining This problem involves predicting future locations destinations and the occurrence of traffic-related events like incidents and traffic jams by analyzing a large amount of vehicle and pedestrian trajectories moving through a city. In this work, we tackle the task of forecasting the order in which the vehicle under study will visit the next locations, given the prior locations from the current trip's starting point and a historical database that depicts patterns of urban mobility.

A method for predicting a vehicle's next location using a Recurrent Neural Network

(RNN) model was presented in the previous study. Neural network models, such as RNN, are frequently employed in natural language processing. In the prior study, we elucidated the similarities between trajectory-based location prediction and text generation, and we implemented the RNN model for trajectory-based location prediction. The only input for the RNN model was the location data from previous visits. Despite having a straightforward structure, the RNN-based location prediction model yielded positive outcomes. For instance, for over 50% of all tested trajectory samples, the probability of correctly predicting the vehicle's next location was greater than 0.7, whereas the base case model only demonstrated accuracy for less than 5% of the samples.

This work proposes a novel methodology to improve location prediction accuracy using an existing RNN model. The additional input allows heterogeneous input sources to be

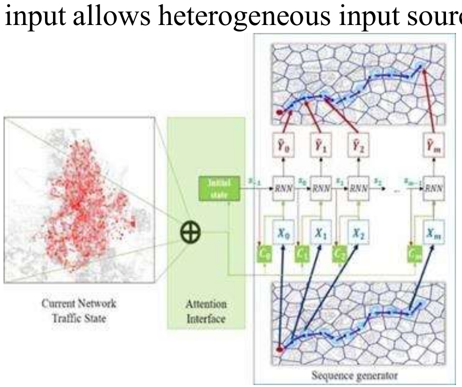

Fig. 2. Structure of the proposed Attention-based Recurrent Neural Network model (ARNN) for cell sequence prediction

To indicate the beginning and end of the journey, two virtual tokens, #start and #end, are added to the front and back of each cell sequence. To facilitate RNN training, validation, and testing, this cell sequence is divided into two input vector parts, X and Y.

$X = [X_0, X_1, X_2, ..., X_m] \equiv [\#start, c_1, c_2, ..., c_m]$

incorporated into the predictions. As a particular study input, we took into account the traffic conditions of the urban traffic network at the beginning of the trip. These days, drivers can use a variety of traffic information and routing services to choose their route and observe the current traffic situation in urban traffic networks. As a result, at the beginning of their trip, traffic conditions in networks are anticipated to have an impact on each vehicle's route. This study proposes an attention-based RNN model for location sequence prediction based on this concept, which allows the RNN model to take traffic conditions into account as an extra input. The Methodology section contains a thorough explanation.

2. Methodology

The representation of a vehicle trajectory, consisting of l longitude (x) and latitude (y) data points, is $Tr = [(x_1, y_1), (x_2, y_2), (x_l, y_l)]$. If the urban traffic network is divided into multiple cells (c_i, $i \in 1,, m$), the vehicle trajectory, Tr, can be represented as a series of cells ($[c_1, c_2,, c_m]$). The length of the cell sequence, m, is always less than or equal to the length of the original trajectory sequence, l, because each cell can cover multiple data points (i.e., $m \leq l$).

$Y = [Y_0, Y_1, Y_2, ..., Y_m] \equiv [c_1, c_2, ..., c_m, \#end]$

RNN model for predicting cell sequence was created and assessed in a prior study [16], however it is not an easy task to add more inputs to an RNN model. When additional input is sequential data, adding another sequence input can be done with a simple extension because the RNN model is well-suited for processing sequential data that takes dependencies across time or sequence steps into consideration. In order to combine multiple sequence inputs and compute the output, an RNN model can have multiple input layers and different hidden features. If not, processing outside of the RNN model is required. As an additional input, we will use non-sequential traffic state information. Consequently, in order to include this non-sequential traffic state information, an extra structure is required.

One solution to this problem is to use the attention mechanism. As seen in Fig. 1, the attention mechanism serves as an interface between information that has been processed externally and sequential inputs that are processed inside the RNN model. Since its introduction, this neural network mechanism has greatly improved the performance of applications like machine language translation [18] and video captioning . It was originally designed to mimic the attention mechanism found in the human brain. Specific portions of the network traffic state input can be the focus of the attention mechanism-based cell sequence generator or cell sequence prediction model, which uses the information for cell sequence generation. Setting the RNN's initial state and providing network-wide traffic state information at each cell generation step are the two tasks assigned to the attention mechanism. Typically, the RNN cell's initial state vector is set to zero because the RNN model's basic form contains no additional inputs. Nevertheless, the model ought to incorporate the extra network traffic state data that ARNN provides. Furthermore, by computing attention weights and the context vector, the attention mechanism enables the RNN to take traffic state into account at each prediction step.

Through a variety of traffic information and routing services, drivers can easily find out the current traffic situation in urban traffic networks and plan their route [17, 20]. When route B is congested, for instance, a driver is more likely to choose route A, and vice versa. Therefore, it is anticipated that the network traffic conditions at the start of each vehicle's journey will have an impact on its location sequences, or chosen routes. To improve the prediction accuracy of the RNN-based cell sequence prediction model, it is thus desirable to include network-wide traffic state information and route choice behavior based on the current traffic state.

The Attention-based Recurrent Neural Network (ARNN) model for cell sequence prediction is depicted in Fig. 2. This model takes two types of input data: network traffic state data, which is the first type, and the second is the vehicle trajectory data represented in cell sequence. The model computes the RNN unit's initial state ($s-1$) after processing the state of the network traffic as of right now. Next, using the previous state vector as a basis, the attention interface computes the context vector (Ci). The ith RNN unit receives the context vector (Ci) as input and updates the current state vector (si) with the corresponding input vector element (Xi). Based on the context vector and previous state vector (α i, j) = f (C, s i 1)), the attention weight α(i, j) is computed. The likelihood of attending to the jth cell at the ith sequence is represented by the attention weight. Consequently, 1 ($\forall$ j α(i, j) = 1) is the sum of α(i, j) at each sequence.

To represent the hidden characteristics of the cells, the word-embedding method is applied to the processing of the input cell sequence (X). During the training phase, each RNN unit uses input vector X as a direct input to determine the output vector (Yˆi). Only the front n cell sequence elements are utilized directly in the testing phase, though. Next, we use a random sampling based on the multinomial distribution with probability Yˆi to extract the next cell, which is also used as the next element of the input vector, since the output vector indicates the likelihood of each cell being visited.

A basic Long Short Term Memory (LSTM) cell is used as RNN cell. And the model also uses the Adam optimizer to update the model parameters.

1. Model Performance Evaluation

Data

Urban Vehicle Trajectory Data

The Queensland Department of Transport and Main Roads (TMR) and Brisbane City Council (BCC) provided the Bluetooth sensors in Brisbane, Australia, which collected the vehicle trajectory data used in this study. The state-controlled roads and intersections within Brisbane City have Bluetooth sensors installed. These sensors are designed to identify and time the passage of Bluetooth devices, such as mobile phones and in-car navigation systems. Vehicle trajectories of individual vehicles can be constructed by joining data points that have the same Bluetooth device identifier (MAC ID). Every vehicle trajectory shows the locations of Bluetooth sensors that a subject vehicle passes in a time-ordered sequence. The corresponding vehicle's trip is deemed to have ended if it is stationary for more than an hour. We used the vehicle trajectory data that was gathered in March 2016 for this case study. A day's worth of trajectories equals about 276,000, and in March 2016, 8,556,767 vehicle trajectories were gathered. 200,000 vehicle trajectories were sampled at random for the training dataset, 10,000 for the validation dataset (used in hyper-parameter searching), and 200,000 for the testing dataset.

In order to apply the vehicle trajectory clustering and cell partitioning method suggested in earlier research, the Brisbane urban traffic network is divided into cells [23, 24]. The cells are intended to have a radius of 300 meters. Consequently, 5,712 cells in total are produced. Out of them, 2,746 cells are regarded as active since, according to historical data on vehicle trajectories, no vehicle has visited the remaining cells. After processing, the vehicle trajectory data are converted into cell sequence data.

Network Traflc State Data

The network traffic state can be represented in a number of ways, including by average speed and density. In this study, the network traffic state is represented by vehicle accumulation, which is defined as the cell density. By counting the number of vehicles within a cell at any one time, the vehicle accumulation for that cell can be approximated. We computed the vehicle accumulation of every cell at every minute by processing the vehicle trajectory data. By dividing the vehicle accumulation by the historical maximum number of vehicle accumulation in each cell, the vehicle accumulation data are normalized.

2. Conclusion

Building on earlier research, we suggested a novel method for integrating network traffic state data into urban vehicle trajectory prediction models. The network traffic state data was utilized for vehicle trajectory prediction using the attention mechanism, and the outcomes of the ARNN model were contrasted with those of the RNN model that was previously in use. Consequently, the ARNN model outperformed the RNN model. It has been verified that the attention mechanism, which establishes a structural connection between the network traffic state input and the RNN model, yields better vehicle path prediction. Notably, because ARNN took into account both the cells to be visited and the alignment of the cells in the sequence, it demonstrated a significant performance improvement in the METEOR score. As the original cell sequence lengthens, the performance improvement rates tend to fall and converge to 1. This issue needs to be investigated in order to maintain consistent performance gains for the ARNN model.

References:

[1] J. Zimmerman, A. Tomasic, C. Garrod, D. Yoo, C. Hiruncharoenvate, R. Aziz, N. R. Thiruvengadam, Y. Huang, A. Steinfeld, Field trial of tiramisu: crowd-sourcing bus arrival times to spur co-design, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2011, pp. 1677–1686.

[2] A. Noulas, S. Scellato, N. Lathia, C. Mascolo, Mining user mobility features for next place prediction in location-based services, in: Data mining (ICDM), 2012 IEEE 12th international conference on, IEEE, 2012, pp. 1038–1043.

[3] S. Gambs, M.-O. Killijian, M. N. del Prado Cortez, Next place prediction using mobility markov chains, in: Proceedings of the First Workshop on Measurement, Privacy, and Mobility, ACM, 2012, p. 3.

[4] W. Mathew, R. Raposo, B. Martins, Predicting future locations with hidden Markov models, in: Proceedings of the 2012 ACM conference on ubiquitous computing, ACM, 2012, pp. 911–918.

[5] J. Krumm, E. Horvitz, Predestination: Inferring destinations from partial trajectories, in: International Conference on Ubiquitous Computing, Springer, 2006, pp. 243–260.

[6] J. Krumm, E. Horvitz, Predestination: Where do you want to go today?, Computer 40 (4).

[7] E. Horvitz, J. Krumm, Some help on the way: Opportunistic routing under uncertainty, in: Proceedings of the 2012 ACM conference on Ubiquitous Computing, ACM, 2012, pp. 371–380.

[8] A. Y. Xue, J. Qi, X. Xie, R. Zhang, J. Huang, Y. Li, Solving the data sparsity problem in destination prediction, The VLDB JournalThe International Journal on Very Large Data Bases 24 (2) (2015) 219–243.

[9] B. D. Ziebart, A. L. Maas, A. K. Dey, J. A. Bagnell, Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior, in: Proceedings of the 10th international conference on Ubiquitous computing, ACM, 2008, pp. 322–331.

[10] G. Wang, J. Kim, The Prediction of Traffic Congestion and Incident on Urban Road Networks Using Naive Bayes Classifier, in: Australasian Transport Research Forum (ATRF), 38th, 2016, Melbourne, Victoria, Australia, 2016.

[11] J. D. Mazimpaka, S. Timpf, Trajectory data mining: A review of methods and applications, Journal of Spatial Information Science 2016 (13) (2016) 61–99.

[12] S. Choi, H. Yeo, J. Kim, Network-Wide Vehicle Trajectory Prediction in Urban Traffic Networks using Deep Learning, Transportation Research Record (2018) 0361198118794735.

[13] J. L. Adler, Investigating the learning effects of route guidance and traffic advisories on route choice behavior, Transportation Research Part C: Emerging Technologies 9 (1) (2001) 1–14.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[15] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: International conference on machine learning, 2015, pp. 2048–2057.

[16] T. Cabannes, M. A. S. Vincentelli, A. Sundt, H. Signargout, E. Porter, V. Fighiera, J. Ugirumurera, A. M. Bayen, The impact of GPS-enabled shortest path routing on mobility: a game theoretic approach 2, Tech. rep. (2018).

[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[18] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[19] J. Kim, K. Zheng, S. Ahn, M. Papamanolis, P. Chao, Graph-based analysis of city-wide traffic dynamics using time-evolving graphs of trajectory data, in: Australasian Transport Research Forum (ATRF), 38th, 2016.

[20] J. Kim, K. Zheng, J. Corcoran, S. Ahn, M. Papamanolis, Trajectory Flow Map: Graph-based Approach to Analysing Temporal Evolution of Aggregated Traffic Flows in Large-scale Urban Networks, Tech. rep. (2017).

[21] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics, 2002, pp. 311–318.

[22] S. Banerjee, A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, 2005, pp. 65–72.