Australasian Transport Research Forum 2023 Proceedings
29 November – 1 December, Perth, Australia
Publication website: http://www.atrf.info

Trajectory-user linking with a deep neural network

Hua Shi, Dan He, Jiwon Kim, and Thanh Vo Tran

School of Civil Engineering, EAIT

The University of Queensland, St Lucia, Brisbane, QLD, Australia

Email for correspondence (presenting author): hua.shi@uq.edu.au

1. Introduction

In recent years, the proliferation of GPS-enabled devices has led to an explosion of data, including vast amounts of trajectory data capturing user mobility and travel behavior. One important area of research in this field is trajectory user linking, which involves analyzing patterns of behavior to identify anonymous trajectories with the users who generated them. Trajectory user linking has a wide range of applications. For instance, by linking check-in trajectories on points of interest (POI) to specific users, advertisers can better understand users' preferences and interests and deliver more targeted and personalized recommendations. Trajectory user linking can also be used to identify and detect criminal/terrorist behavior or track the transmission of pandemics by mapping suspicious trajectories to potential suspects in the database system. Furthermore, trajectory user linking can also help improve transportation planning and traffic management by providing insights of the user's mobility patterns, including their commuting routes, their favorite leisure spots, and their travel habits.

Traditional methods for trajectory user linking involve measuring the similarity between unknown trajectories and known trajectories. However, such methods are usually time-consuming and more sensitive to data quality issues. Trajectories with inconsistent sampling rates or different lengths generally induce poor linking performance. To overcome these limitations, in recent years, plenty of studies have been developed for trajectory user linking, making use of deep learning techniques that attempt to learn the nonlinear correlations of users' behaviors from the data. Most of the existing work (Gao et al., 2017; Miao et al., 2020; Zhou et al., 2018) attempt to mine the sequential transition patterns from the trajectories based on Recurrent Neural Network (RNN) models and mainly focus on the POI check-in data. Some of them consider the GPS trajectTraories. Generally, the performance of trajectory-user linking depends on several factors, such as the type of trajectory data, the length of the trajectories, and the sampling rate. For instance, POI check-in data, which typically consists of short and sparse trajectories, may perform differently than GPS data, which provides more continuous and dense trajectories. Additionally, the length of the trajectories and the sampling rate can affect the accuracy of trajectory user linking. Longer trajectories may contain more diverse spatiotemporal information but may also be more challenging to analyze due to their complexity. Similarly, trajectories with inconsistent sampling rates may introduce noise and affect the linking performance.

To address these challenges, our work proposes a deep learning model that combines the power of GNN and RNN to capture the spatiotemporal information of different types of trajectories. The GNN-based point embedding module allows us to transform each point of the trajectory into a more feasible format, such as a vector, that captures the complex relationships between different points. The RNN-based trajectory embedding module then takes these embeddings as input and learns sequential correlations to generate a trajectory representation that captures the overall behavior of the user. Finally, the user linking module maps the trajectory representation to the corresponding user, which typically works as a classification task. We conducted experiments to evaluate the effectiveness of our proposed model on three distinct categories of data: POI check-in data, Bluetooth data, and GPS data. Our results demonstrate that our model exhibits promising performance in solving the trajectory-user linking problem. Notably, our model shows an ability to handle the sparsity of POI check-in data and the noise in Bluetooth data while still capturing the richness of GPS trajectories. However, what is particularly interesting is that our experiments reveal varying performance across the different datasets. Specifically, we found that data with different characteristics exhibit different performance when applying our proposed methods with different submodules. Overall, our experiments highlight the importance of selecting appropriate submodules for different data characteristics in order to achieve optimal performance in trajectory-user linking.

2. Problem statement

In this section, we provide a formal definition on the trajectory-user linking problem.

Definition 1 (Trajectory): *A trajectory T_u generated by the user u is denoted by a sequence of spatiotemporal points $T_u = \{p_1, p_2, \dots, p_{|T_u|}\}$ organized chronologically, where each point $p_i = (l_i, t_i)$ consists of a geographic coordinates l_i (e.g., longitude and latitude) and a timestamp t_i.*

An unlinked trajectory $\tilde{T}$ is anonymous whose corresponding user is unknown yet. And the trajectory-user linking problem tries to map each unknown trajectory to its corresponding user from a candidate set of users. Thus, we assume that all the unlinked trajectories are generated by one of user in the pre-defined candidate set. Cooorespondingly, we define the trajectory-user linking problem as follow.

Definition 2 (Trajectory-user Linking): *Given a set of unlinked trajectories $\tilde{\Gamma}$ and a set of candidate users U, we aim to learn a mapping function $f: \tilde{\Gamma} \rightarrow U$ that links every unlinked trajectory $\tilde{T} \in \tilde{\Gamma}$ to its corresponding user $u \in U$.*

3. Methodology

In this section, we give an overview of our model, which contains three components as shown in Figure 1. In the following, we present the details of each sub-module individually.

Point-level representation learning: Given a sequence of trajectory, the first component aims to generate a low-dimensional vector for each point of the trajectory, as the input for the sequential neural network to learn the correlation between trajectories and users. The goal of this process is not only to mitigate the problem of the curse of dimensionality, but also to incorporate rich information from different types of features for further process. For instance, TULER (Gao et al., 2017) and TULVAE (Zhou et al., 2018) apply a technique usually used in natural language process (NLP) for word embedding, name Word2Vec (Mikolov et al., 2013), which is a semi-supervised method for point embedding. In general, a trajectory contains location, timestamp, and semantic information such as POI category. The initial representations for different features are usually one-hot encoding. Rather than directly taking this format as input, most of the existing work tend to convert the one-hot format into a dense vector with lower dimension. A straightforward way is to augment one layer of perceptron before the main network. Specifically, we develop a transformation matrix $E \in \mathbb{R}^{|f| \times d}$ such that each category of features can be represented by a vector format, i.e., $v = f \cdot E$, where f indicate the feature vector and d is the targeted dimension of the output representation. In implementation, this can be easily achieved by adding a linear layer where E is made up by the parameter matrix. Then we apply a graph neural network on top of the linear layer for feature embedding. Recently, many research studies working on spatiotemporal data explore the trend of applying GNNs for representation learning, which we refer to as graph-based embedding (Wu et al., 2019). The intuition of applying GNN for trajectory-user linking is that the location points with similar features ideally should have similar representation, which can be achieve by message sharing among neighbors in the GNNs. In this case, the construction of the underlying conceptual graph is critical since the information propagation from node to node is based on the topological structure such that the correlations of features from not only this node, but also other nearby nodes are captured. In our work, we build the graph structure based on two categories of connections: spatial connection and mobility connection. We construct a spatial graph based the location information, where an edge is added to two nodes, if the distance of their locations is within a certain threshold. In terms of the mobility connection, a visit graph is built according to the sequential moving pattern, where each edge represents a visiting/traveling behavior from one point to another by one user. Afterwards, two graphs are consolidated into one global graph and for the point-level representation learning. By considering the mobility connections between two points, our GNN-based point embedding module can identify and amplify meaningful patterns while dampening the influence of noise.

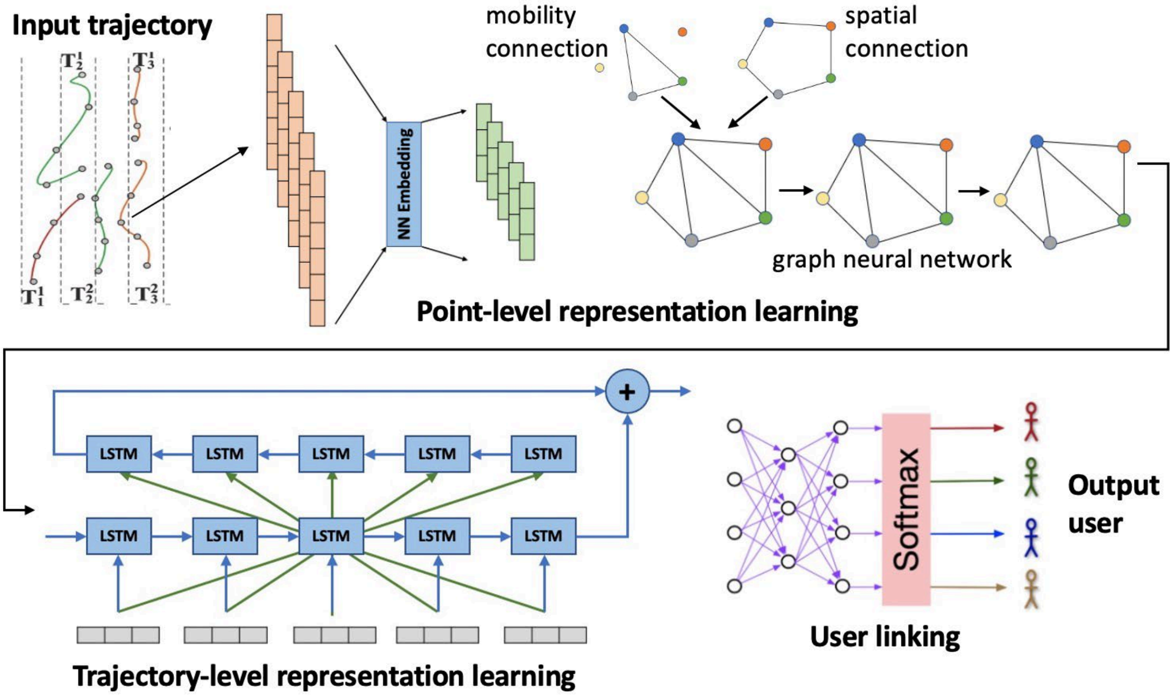

Figure 1: Framework of trajectory-user linking model.

Trajectory-level representation learning: The trajectory-level representation learning module is a crucial component in the trajectory-user linking model. Its main objective is to transform the sequence of point-level embeddings obtained from the previous module into a fixed-length representation for the entire trajectory. This representation should capture not only the features of individual points but also the sequential correlations between them. In the field of deep learning, RNN and its variants such as LSTM and GRU have been widely used for learning patterns in sequential data (Miao et al., 2020; Yu et al., 2019). To provide more detail, the bidirectional LSTM model takes as input a sequence of trajectory $T_u = \{p_1, p_2, \dots, p_{|T_u|}\}$

where each point p_i is represented by a vector. The model contains a sequence of LSTM units, each of which takes a point vector as input. The hidden state of the previous unit is used as the input to the consecutive unit for the next time step, which enables the model to capture longer dependencies from the trajectory. In a bidirectional LSTM, two sequences of LSTM units are used, one processing the input sequence in its original order and the other processing it in reverse order. This allows the model to capture not only the past context of each point but also its future context. The outputs of the two LSTM sequences are then combined to form the final representation of the trajectory. Figure 1 shows the architecture of our trajectory-level representation learning module. As illustrated, the bidirectional LSTM model takes a sequence of point-level embeddings as input and outputs a fixed-length trajectory representation, which is used for the subsequent trajectory-user linking task.

User linking: After the trajectory-level representation learning, the final step is to link the trajectory to its corresponding user. This is usually done by a classification task, where the model predicts the user label of the trajectory based on its representation. We use a SoftMax function to compute the probability distribution over all users, and the predicted user is the one with the highest probability, as shown in Figure 1.

4. Experimental study

In this section, we present an experimental study for the proposed model on three different data sets. We first provide a description of our data sets, followed by the experiment setting and performance results. Finally, we discuss the insights that we observe based on the performance results and point out the potential future direction of this research topic.

4.1 Data description

Foursquare includes 227,428 Foursquare check-ins with 38,333 distinct POIs generated by 1083 users in New York City (NYC) over 5 months in 2012. Each check-in contains a few attributes, including time stamp, GPS coordinates, and semantic meaning (represented by venue ID, venue categories, and venue category ID). The overall sampling rate of this dataset is quite low (e.g., hourly).

Bluetooth (Xu et al., 2020) data are captured by road-side Bluetooth Media Access Control (MAC) Scanners (BMSs) around the city of Brisbane, Australia. It contains 192 millions of BMS readings for 683, 000 distinct objects captured by 1,028 BMSs over a period of one month. The sampling rates for the Bluetooth are various from seconds to hours depending on the locations among the BMSs.

Geolife (Zheng et al., 2009) GPS trajectory dataset was collected in (Microsoft Research Asia) Geolife project by 182 users in a period of over five years (from April 2007 to October 2012). This dataset recorded a broad range of users' outdoor movements with different travel modes. A large proportion of trajectories are logged in a dense representation with a high sampling rate (every 1-5 seconds). We resample the GPS points with a sampling rate of 5 minutes to reduce the redundancy.

4.2 Experiment settings

When analyzing user mobility and travel behavior using trajectory data, grouping spatiotemporal points of a user into one trajectory can result in a long sequence containing multiple trips with different purposes. However, in real-world scenarios, linking a user based on one purposeful trip is more meaningful than a long sequence with historical points. Additionally, training a neural network model to map long sequences of trajectories to a specific user can be computationally expensive due to the required neurons. Therefore, it is necessary to divide a long trajectory sequence into shorter segments. We conducted preliminary studies on different trajectory segmentation methods and found that a straightforward, yet efficient way is to split the trajectory into segments with equal lengths of time span. Specifically, each sub-trajectory has a fixed duration of, for example, 6 hours. To test the performance of our trajectory-user linking model, we extracted data from three datasets, each containing 112 users with 13180, 3620, and 11050 trajectories for Foursquare, Bluetooth, and Geolife, respectively. We split each dataset into 80% for training and 20% for testing, and optimized our model with the Adam optimizer, minimizing the cross-entropy between the ground truth user label and the linked one. To evaluate the model's performance, we predict the top-k candidate users for each testing trajectory and use accuracy at k (ACC@K) and macro-F1 as performance metrics, which are defined as follows, where $P^*$and R^* are precision and recall averaged across all classes.

$$\mathrm{ACC@K} = \frac{\#\ \text{correctly identified trajectories @ K}}{\#\ \text{trajectories}}, \qquad \mathrm{macro-F1} = \frac{2 \times \mathrm{P}^* \times \mathrm{R}^*}{\mathrm{P}^* + \mathrm{R}^*}$$

4.3 Performance study

We conducted an experimental study to evaluate the effectiveness of our model on three different datasets, and compared its performance with models that remove certain components. As the RNN-based trajectory-level embedding is a crucial submodule for trajectory-user linking, we kept it as the base and examined the impact of different point-level embedding methods. Specifically, we investigated the effectiveness of the GNN model on point-level embedding by comparing it to a linear layer only. Furthermore, the underlying graph structure in the GNN module was built using two different connectivity approaches, and we evaluated the performance of the model by removing the spatial connectivity and considering only the visiting connectivity. It should be noted that for check-in data, spatial distances between two consecutive POIs can be quite large, so we cannot rely solely on spatial connectivity.

Table 1: Performance results for different data sets.

Models	Metrics	Data set Foursquare	 Bluetooth	 Geolife
GNN (V+S)	ACC@1	57.69%	62.66%	**49.61%**
	ACC@5	63.91%	79.23%	**73.22%**
	Macro-F1	53.80%	53.39%	**25.92%**
GNN (V)	ACC@1	**67.75%**	63.36%	48.19%
	ACC@5	**81.71%**	78.18%	72.17%
	Macro-F1	**65.81%**	54.55%	24.24%
NN-Embed	ACC@1	58.36%	**69.50%**	47.19%
	ACC@5	65.18%	**85.18%**	71.65%
	Macro-F1	55.28%	**61.35%**	23.52%

The results of our experimental study, presented in Table 1, highlight the effectiveness of different point-level embedding methods for trajectory-user linking. Note that we conducted each experiment three times and calculated the average of the results. As shown in the table, Bluetooth data consistently outperforms the other two datasets, with the ACC@1 and ACC@5 metrics achieving values of around 70% and 85%, respectively, in the best case. On the other hand, the Geolife dataset shows the worst performance across all methods, likely due to the complexity of GPS trajectories, which often contain a larger number of points with more diverse and frequent location updates. Importantly, our study also revealed that different point-level embedding methods are more suited for different types of data. For example, check-in data from Foursquare performs better with the GNN module that considers only visiting connectivity. This is because different users might visit nearby POIs in different orders, making it challenging to distinguish between them when considering spatial connectivity. In contrast, Bluetooth data appears to perform better with the NN-embedding method, without the GNN module. Lastly, our results show that GPS trajectories exhibit the best performance with the GNN module and the worst performance with NN-embedding only for point-level embedding. Overall, these findings highlight the importance of carefully selecting the appropriate point-level embedding method based on the characteristics of different datasets, as well as the potential benefits of using GNN-based models for GPS trajectory data.

5. Conclusion

In this paper, we proposed a novel deep learning model that leverages the strengths of both GNN and RNN to address the challenging task of trajectory-user linking. Our model was evaluated on three different datasets: POI check-in data, Bluetooth data, and GPS data, and the results demonstrated promising performance in accurately linking trajectories to their corresponding users. However, we observed that the model's performance varied across the datasets, which highlights the importance of selecting appropriate submodules for different data characteristics to achieve optimal performance. Moving forward, our future work will focus on investigating more in-depth impact on the performance of various methods. Future research could involve detailed investigations into the interplay between data characteristics and the algorithmic approaches, as well as systematic comparisons of their capabilities across a range of datasets. Such analysis will be invaluable for advancing our understanding of trajectory-user linking and refining our algorithmic choices.

5. References

Gao, Q., Zhou, F., Zhang, K., Trajcevski, G., Luo, X., Zhang, F., 2017. Identifying Human Mobility via Trajectory Embeddings 1689–1695.

Miao, C., Wang, J., Yu, H., Zhang, W., Qi, Y., 2020. Trajectory-User Linking with Attentive Recurrent Network, in: AAMAS. https://doi.org/10.5555/3398761.3398864

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space. https://doi.org/10.48550/arXiv.1301.3781

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S., 2019. A Comprehensive Survey on Graph Neural Networks. https://doi.org/10.1109/TNNLS.2020.2978386

Xu, Y., He, D., Chao, P., Kim, J., Hua, W., Zhou, X., 2020. Route Reconstruction Using Low-Quality Bluetooth Readings, in: Proceedings of the 28th International Conference on Advances in Geographic Information Systems, SIGSPATIAL ’20. Association for Computing Machinery, New York, NY, USA, pp. 179–182.

Yu, Y., Si, X., Hu, C., Zhang, J., 2019. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Comput. 31, 1235–1270. https://doi.org/10.1162/neco_a_01199

Zheng, Y., Zhang, L., Xie, X., Ma, W.-Y., 2009. Mining interesting locations and travel sequences from GPS trajectories. Presented at the Proceedings of WWW 2009, pp. 791–800. https://doi.org/10.1145/1526709.1526816

Zhou, F., Gao, Q., Trajcevski, G., Zhang, K., Zhong, T., Zhang, F., 2018. Trajectory-User Linking via Variational AutoEncoder. pp. 3212–3218. https://doi.org/10.24963/ijcai.2018/446