Clustering of Human Movement Trajectories based on Distributional Representations Derived from Bi-directional LSTM Network with Geographical Coordinates

Hiroki Tanaka, Takeshi Saga, Satoshi Nakamura
Center for Advanced Intelligence Project, RIKEN
Nara Institute of Science and Technology
Ikoma, Nara, Japan
hiroki-tan@is.naist.jp, saga.takeshi.sn0@is.naist.jp, s-nakamura@is.naist.jp

***Abstract*—As the ubiquity of such wearable devices as smartphones continues to deepen its presence in modern societies, it has become possible to analyze and visualize people who are moving as part of a trajectory of big data. In this study, we cluster human movement trajectories using time-series distributional representations. For the clustering, we calculated the distance of the representation vectors derived from neural network models. Previous work leveraged the Long short-term memory (LSTM) network to train the next mesh prediction. In this study, we propose using the Bi-directional LSTM (Bi-LSTM) network and the integrated additional geographical coordinates (latitude and longitude information) in models to accurately predict the next mesh and construct user clusters. As a result, we improved the accuracy of the next mesh prediction and obtained and visualized clusters of human movement trajectories.**

***Index Terms*—transportation, trajectory, bi-directional LSTM, user clustering**

I. Introduction

These days almost every person carries their own mobile phone. Such mobile phones are also being widely applied as human movement monitoring devices for health maintenance and behavior pattern analysis. The rapid advance of location acquisition technologies has boosted the acquisition of trajectory data that track the traces of moving objects. Past works analyzed and modeled human movement trajectories [5], [7]. Our aim is to cluster human movement trajectories using time-series distributional representations to analyze and model them.

Ermagun et al. applied an econometric approach called Nested Logit, which is highly interpretable for predicting behavioral tendencies during travel, and Random Forest, which is a machine learning method that has high generalizability [3]. Hirota et al. applied Word2Vec, which has been used in natural language processing, to user route prediction by dividing maps into mesh-like sections, assigning IDs, and expressing user moving routes [6]. However, since Word2Vec only uses the IDs before and after the prediction position, there is room for improvement by exploiting the time-series information. Crivellari et al. analyzed the similarity of each landmark and each movement route by applying cosine similarity and t-Distributed Stochastic Neighbor Embedding (t-SNE) to distributed representations corresponding to each landmark acquired by Word2Vec [1]. This work described the applicability of distributed expressions by Word2Vec to cluster human movement trajectories. In a subsequent work, Crivellari et al. used Long short-term memory (LSTM) network to predict the next location from input human movement trajectories [2]. Since LSTM network propagates all the information in the input series as a hidden state, such information can be used more efficiently than Word2Vec. Lastly, Liu et al. identified a problem, arguing that the temporal and positional linearity between each sequence point was ignored in behavior sequence predictions. They dealt with this problem by constructing a prediction model by adding both position and time information as the input features of recurrent neural networks [10].

Our idea is to train recurrent neural networks and use their distributional representations to find user clusters. This paper applied a Bi-directional LSTM (Bi-LSTM) network to predict the next mesh-ID in a time series, where Bi-LSTM learns semantic similarity in both the forward and backward direction of the non-uniform length trajectory in terms of the fixed-size vectors. Additionally, mesh-ID is a code given when subdividing Japanese landscapes into rectangular sub-regions. Mesh-ID is defined by the Ministry of Internal Affairs and Communications, Japan.

As in [2], ID series were used for the modeling, their physical distance and relative position was not used properly. Here, we proposed an approach using the Bi-LSTM network and the integrated additional geographical coordinates (latitude and longitude information) in the models to accurately predict the next mesh.

This paper describes the details of the model's architecture, the training method, and the clustering result using embedded vectors after learning. We visualize the clusters obtained from the distributional representations of the trained models.

II. Proposed Method

A. Model architecture

We attempt to cluster the users who have an undefined number of visiting places. In order to cluster such data, we have to define distance. We used fixed-length embedding vectors for the distance calculation of the cluster. To estimate the user's next mesh-ID, Bi-LSTM needs to compress the information into the hidden state in the hidden layer then extract related information from it. Therefore, the hidden state vector, so-called embedding, is a compact and fixed-length information vector corresponding to the user's trajectory. The background assumption is that a precise prediction model contains better embedding vectors to be used for clustering. Hence we tried to get better prediction accuracy to get better clustering results. This approach has mostly been applied in Natural Language Processing.

Figure 1 shows a schematic diagram of our model structure, based on the LSTM network used in previous studies [2]. Since mesh-IDs were assigned, their physical distance and relative position were not leveraged. Therefore, effectively training the model might be difficult just using mesh-IDs. In this paper, we used additional latitude and longitude information for training in addition to the Bi-directional LSTM network and LSTM's baseline. We extended the past study to Bi-LSTM network and to add latitude and longitude as input vectors. Each prediction in Figure 1 is composed of fully connected layers. The output of the latitude/longitude prediction is a one-dimensional scalar value, and the output of the last mesh-ID prediction is a one-hot vector whose dimensions are the number of classes of training data (the mesh-IDs in the training data). We used python with Keras framework to train the networks. The total number of parameters representing model complexity are as follows: (1) Mesh-ID-only (LSTM network): 51,689,005, (2) Mesh-ID-only (Bi-LSTM network): 78,932,905, (3) Mesh-ID plus latitude/longitude (LSTM network): 51,689,007, (4) Mesh-ID plus latitude/longitude (Bi-LSTM network): 78,938,907.

In the model in which latitude and longitude are added to mesh-IDs, we must simultaneously learn the mesh-IDs, which are discrete values, and the latitude and longitude, which are continuous values. However, since the loss functions for the discrete values and the continuous values are different, integrated loss L_{total} was calculated by formula (1) and used as a loss function for the entire training:

$$L_{total} = \alpha \times L_{mesh} + \beta \times (L_{lat} + L_{long}) \,. \tag{1}$$

Here L_{mesh} is the cross-entropy loss for the mesh-IDs, L_{lat} and L_{long} are the mean squared error (MSE) for the latitude and longitude, and α and β are hyper-parameters that adjust the balance of the loss values. We defined formula 1 empirically to equally consider both factors: meshID and coordinates.

We used Adam with a learning rate of 0.001 as the optimization function. The number of dimensions of the hidden layer of Bi-LSTM network was set to 300 for each layer (since each is 300 dimensions in the forward and reverse directions, it becomes 600 dimensions when connected to the entire Bi-LSTM layer), and drop out was set to 0.5. For α and β, we used the combination of 1 and 1 because it had the highest accuracy for the validation data.

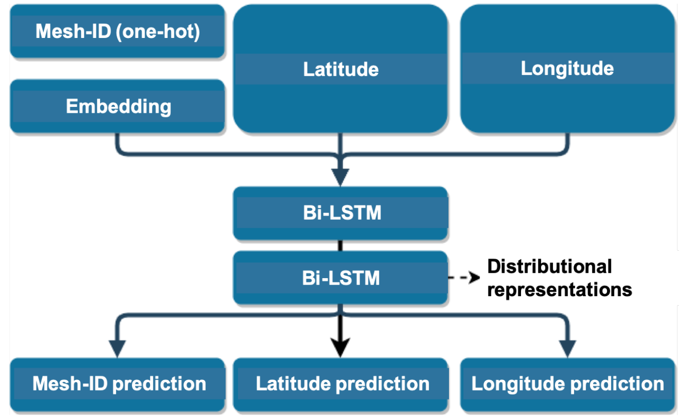

Fig. 1. Model structure of Bi-LSTM network with geographical coordinates

Since the range of the change in latitude and longitude is very small, the values are standardized so that the average value is 0 and the variance is 1, and then they are input.

B. Clustering based on distributional representations

The embedding is a compressed information vector of the user's trajectory trained to estimate the next mesh-ID. Same as Bi-LSTM application in the Natural Language Processing field [9], we tried to train this model to get a compressed meaning representation of each trajectory bi-bidirectionally.

Agglomerative hierarchical clustering was performed on the distributed representations obtained by the mesh-ID and latitude/longitude model. We assumed that we could obtain semantically similar trajectories based on the embedding. We clustered based on the 600 dimensional embedding vectors of the last layers of the Bi-LSTM. For the distributed representation for the input series, we combined the hidden states of the Bi-LSTM network in the final layer in the forward and reverse directions for a total of 600 dimensions. Agglomerative hierarchical clustering by the Ward method [11] using cosine similarity of the embedding vectors was used as the clustering. The distance between clusters was empirically set to 1.2 when the final cluster was created.

C. Experimental evaluation

The data, provided by Agoop Corp, include the trajectory obtained from smartphone apps. It is a chronologically ordered set of lat/long pairs. It contains not only start and end but also every sampling point that is labeled as departure or arrival. The sequence of mesh-ID is denoted as follows: $\{start, \{departure :, arrival :\}, \{departure :, arrival :\}, ..., end\}$. Departure is tagged when departing after staying in a certain place for a certain period of time in smartphone users. Since our sampling is based on the event that staying at a place for a certain period of time, we don't consider the data while moving. However, it would be better to consider the trip type for better prediction. For training the models, we used the daily time-series history of smartphone users from January to March 2021.

To perform an accurate analysis, we filtered the data of users who have 30 or more series with position error of 100 m or less with respect to the original data. To reduce the amount of calculation, we only analyzed the data of iPhone users tagged as departures or arrivals. After the above filtering, the total number of target users was 35,888. These data were respectively divided into 8:1:1 ratios for training, validation, and testing. Finally, we calculated the accuracy of the number of correct predictions divided by the number of mesh-IDs in the testing data.

For the clustering, if we use all the data for the mesh-ID estimation, visualizing and interpreting the result becomes too complicated. Thus, we filtered the data with a series length of 200 or more, and the number of target users became 36. We calculated the cophenetic correlation coefficient to show how faithfully a dendrogram preserves the pairwise distances among the original, unmodeled data points [4].

III. Results

A. Last mesh prediction

Table I compares the last mesh ID prediction by the Mesh-ID-only model (LSTM network and Bi-LSTM network) and the Mesh-ID plus latitude/longitude model. The accuracy was the higher when the Bi-LSTM network and latitude/longitude information was added than the accuracy when training only with Mesh-ID with LSTM network (baseline [2]). Although for this last mesh prediction task, the accuracy was the highest when the uni-directional and latitude/longitude information was applied, bi-directional LSTM might be better for clustering since it learns semantic similarity in both the forward and backward direction of the non-uniform length trajectory in terms of the fixed-size vectors. We think this accuracy is meaningful although it is not very high. Because the total number of mesh-IDs in Tokyo prefecture is 21939, for example, the random prediction would obtain about 0.004% of accuracy. We had another experiment in data of the Kyoto area where the total number of mesh-IDs is around 8000. In such data, we obtained more than 50% of accuracy [8].

A training curve was analyzed to investigate the effect of adding latitude and longitude. The learning transitions for the mesh-IDs are shown in Figures 2.

We confirmed that although the loss value for the mesh-IDs gradually declined, the loss value for the latitude did not improve after a large decrease in the early stages. This shows that the latitude and longitude information was used at the early stage of training and suggests that the mesh-ID information might be used for training as it progressed.

Furthermore, comparing the transition of the cross-entropy loss of the mesh-ID predictions for the models with latitude/longitude (Figure 2) and without them (Figure 3), we confirmed that the addition of latitude/longitude reduced the loss faster in the early stage. The addition of the latitude and longitude information, which explicitly indicates the relative positional relationship between the mesh-IDs, fueled faster decrease in the loss value in the early learning stages. To confirm the validity, the training was conducted three times, and the same tendency was confirmed in all three trials.

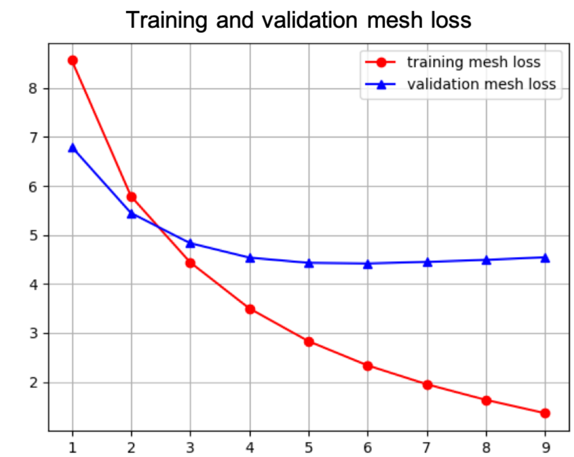

Fig. 2. Mesh-ID cross entropy loss in a model of Mesh-ID plus latitude/longitude

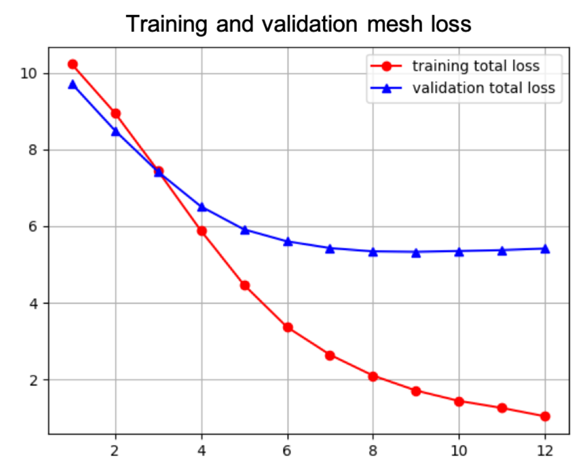

Fig. 3. Mesh-ID cross entropy loss in a model of Mesh-ID-only

TABLE I
ACCURACIES IN TESTING DATA FOR EACH MODEL

Model	Accuracy %
Mesh-ID-only (LSTM network)	21.5
Mesh-ID-only (Bi-LSTM network)	29.0
Mesh-ID plus latitude/longitude (LSTM network)	33.8
Mesh-ID plus latitude/longitude (Bi-LSTM network)	30.8

B. Clustering

Figure 4 shows the clustering results based on the distributed representations obtained by the Bi-LSTM network of the

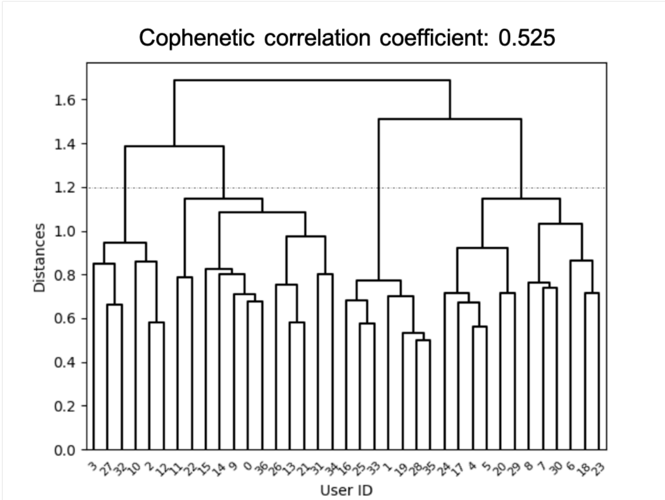

Fig. 4. Clustering in a model of Mesh-ID plus latitude/longitude: four clusters

Mesh-ID plus the latitude/longitude model. The values of the cophenetic correlation coefficient (0.525) are shown at the top of the figure. When the clusters are divided at the distance between clusters of 1.2 (the dotted line) for the model with latitude and longitude added to the mesh-ID, they are divided into four clusters.

IV. Visualizations

We developed the system to process the daily data and train the prediction model every day. The system visualizes based on pre-computed data containing clustering numbers and K-centroid users, which is calculated every day. We calculated the centroid vectors by averaging the distributional representations of each cluster. Then we calculated the nearest K users from the centroid vectors based on cosine similarity. Users can interactively change and filter some parameters (e.g., time, the index of the cluster) on the system. Based on the obtained clusters, we can see the actual route of a representative users or all the users who belong to the clusters. The demonstration visualized an example of the actual route and different characteristics of obtained clusters. Different users are represented by different colors. The system works with python on Windows/Mac/Ubuntu.

Figure 5 shows a map of the top three nearest users to the centroid vectors in cluster number 2. These users moved around Itabashi-Ku and route 264. Figure 6 shows a map of the top three nearest users to the centroid vectors in cluster number 3. These users moved around Hachioji and Nishi-Hachioji station.

V. Conclusion

We proposed Bi-LSTM network and integrated additional geographical coordinates (latitude and longitude information) into models to accurately predict the last mesh and constructed clusters based on distributional representations.

Since the current method must conduct a heuristic analysis while comparing the map information and the action sequence, researchers need to observe it, which is time-consuming and expensive. To solve this problem, an objective analysis method must be studied that incorporates such landmarks as points of interest (e.g., popular stores) and geographical features that are likely to be useful for predicting human movement tendencies. In the future, transportation types such as walking, driving, etc., should be considered. We will explore the comparison with other approaches in terms of modeling and clustering.

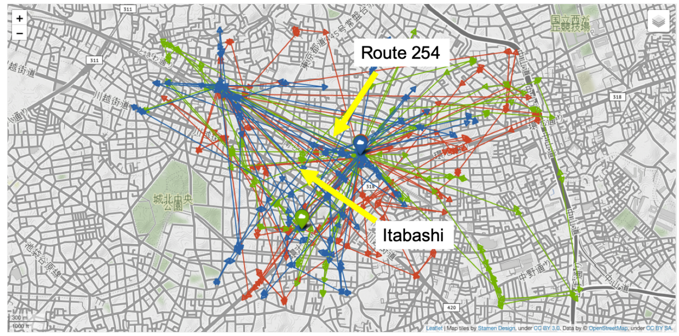

Fig. 5. Clusters number 2 with top three nearest

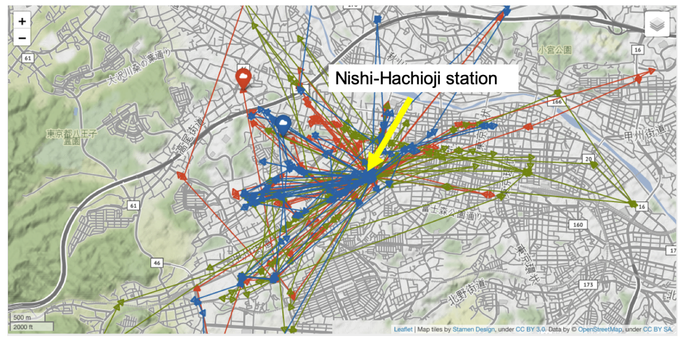

Fig. 6. Clusters number 3 with top three nearest

References

[1] Alessandro Crivellari and Euro Beinat. From Motion Activity to GeoEmbeddings: Generating and Exploring Vector Representations of Locations, Traces and Visitors through Large-Scale Mobility Data. ISPRS International Journal of Geo-Information 8, 3, 2019.

[2] Alessandro Crivellari and Euro Beinat. LSTM-Based Deep Learning Model for Predicting Individual Mobility Traces of Short-Term Foreign Tourists. Sustainability 12, 1, 2020.

[3] Rashidi TH Ermagun A and Lari ZA. Mode Choice for School Trips: Long-Term Planning and Impact of Modal Specification on Policy Assessments. Transportation Research Record, 2513, 1, 97–105, 2015.

[4] James Farris. On the Cophenetic Correlation Coefficient. Systematic Zoology, 18, 279, 1969.

[5] Zhenni Feng and Yanmin Zhu. A Survey on Trajectory Data Mining: Techniques and Applications. IEEE Access, 2056–2067, 2016.

[6] Masaharu Hirota, Tetsuya Oda, Masaki Endo, and Hiroshi Ishikawa. Generating Distributed Representation of User Movement for Extracting Detour Spots. In Proceedings of the 11th International Conference on Management of Digital EcoSystems, 250–255, 2019.

[7] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human Trajectory Forecasting in Crowds: A Deep Learning Perspective. arXiv, 2020.

[8] Motoi Kubo, Hiroki Tanaka, and Satoshi Nakamura. User Clustering based on Distributed Representations for Understanding Tourist Behaviors. Proceedings of the Annual Conference of JSAI, 2020. (in Japanese)

[9] Takeshi Saga, Hiroki Tanaka, Hidemi Iwasaka, and Satoshi Nakamura, Multimodal Prediction of Social Responsiveness Score with BERT-based Text Features, IEICE Trans. Information and Systems, Vol. E105-D, 3, 2022.

[10] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next Location: A Recurrent Model with Spatial and Temporal Contexts. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 194–200, 2016.

[11] Fionn Murtagh and Pierre Legendre. Ward's Hierarchical Clustering Method: Clustering Criterion and Agglomerative Algorithm. arXiv, 11, 2011.