Mining Trajectory Patterns with Point-of-Interest and Behavior-of-Interest

Sissi Xiaoxiao Wu†, Zixian Wu†, Weilin Zhu†, Xiaokui Yang†, Yong Li‡
† College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China
‡ Department of Electronic Engineering, Tsinghua University, Beijing, China

***Abstract*—Epidemiological investigation is one of the main means of controlling the outbreak of COVID-19. It has been proven to be effective, however, has a bottleneck that the infected person has to be questioned about his recent trajectory before any quarantine action could be taken, while sometimes trajectory information might not be timely and accurately obtained. In this paper, we propose an epidemiological investigation method which resort to artificial intelligence for extracting people's preferences and social relationship from their historical trajectory patterns. Trajectory data used in our epidemiological investigation method may include time, location, Point-of-Interest (POI), as well as Behavior-of-Interest (BOI). All of these attributes in human's trajectory are embedded into different channels in the proposed model and then fed into a classifier or a clusterer for serving different purposes. In our experiments, we applied the proposed method on a synthetic data set to conduct a classification task, and on a real data set for a clustering task. Both tasks confirm that the proposed method is effective and thus could be used to guide the preventive measures.**

***Index Terms*—COVID-19, Trajectory, location-based social networks (LBSNs), Point-of-Interest (POI), Behavior-of-Interest (BOI)**

I. Introduction

Due to the mutations of coronavirus and the lack of effective preventive measures, the whole world is in the crisis of an outbreak. To prevent the epidemic, a classic way is to cut off the transmission path based on the results of epidemiological investigation. In the epidemiological investigation, an infected person will be interviewed in person and question about his recent trajectory, such as where he has visited and who he has met with. However, this way may not be timely as interviewing an infected person is usually difficult and sometimes the interviewee cannot remember or do not want to provide accurate information. Meanwhile, one-to-one questionnaire may consume a lot of resources and labors. This motivates us to find alternative methods for epidemiological investigation.

Nowadays, many countries recommend the citizens to scan some Quick Response (QR) code to record their daily visits as a diary. A more advanced way is to recommend citizens to install some specific Apps, which are developed particularly for recording users' trajectories including the visit of time, the latitude and longitude of location, point-of-interest (POI), users' activities or any other useful attributes if allowed. For example, Hong Kong uses the "LeaveHomeSafe" mobile App and Singapore uses the "BlueTrace" App to trace the infections. It is expected that information collected by these Apps will be later desensitized and help stop the possible spreading of the virus. However, in these Apps, not every one is always willing to share his trajectory. Sometimes they may simply forget to scan the QR codes, or turn off the tracking Apps due to some privacy concern. In the meanwhile, QR-code scan and check-in Apps are inherently passive methods for epidemiological investigation, as only if one person is confirmed to be infected, we can then alert his intimate contacts. It could be more effective if we can find susceptible population and possible super-spreaders in advance based on historical trajectory data before infection happens. This provides new challenges to the researchers to develop alternative epidemiological investigation models.

In this work, we propose to utilize historical trajectory of the population to guide the latest epidemic prevention measures. Specifically, we consider an epidemiological investigation model based on trajectories with information of spatio-temporal check-ins, POI, and behavior-of-interest (BOI). Herein, BOI is defined by users' behavior patterns such as habits, amount of consumption, or residence time at some location. In our model, each node in a trajectory contains multiple attributes, i.e., time, location, POI and BOI. A multi-channel embedding algorithm is therefore provided to map each trajectory into a vector in low-dimensional space. These vectors can be further used to construct portraits for each user that generates the trajectory. By feeding those portrait vectors to a classifier or a clusterer, we can accomplish epidemiological investigation tasks for different purposes.

Before we move to the main contents, some relative works are introduced in order. In the literature, location-based social networks (LBSNs) are previously proposed in [1]–[3], wherein users can check in or report their locations while sharing their social life. Typical examples of LBSNs include Foursquare, Gowalla, as well as Facebook, Weibo, and Wechat. With the wide application of the check-in services in the above social platforms, most users' historical geographic and social trajectories can be obtained, and it provides the possibility for us to better understand human movement and mobility patterns in the real world. Prior research for LBSNs can be divided into model-driven and data-driven two approaches. For

This work is supported by Shenzhen Technology Research and Development Fund JCYJ20170302145906843, and by Guangdong Science Fund 2114050003104.

model-driven, the Hawkes process (HP), which is originated from sequential and temporal data, is one of popular models for social networks analysis [4]–[6]. As two typical studies for HP, [7] first used the spatio-temporal data to reconstruct network using multivariate HP; [8] further revealed that implicit communities can be modeled by combining spatial and temporal information instead of time only. Although model-driven based methods can help explain the social relationship among users, it is difficult to handle large volumes of data, or data with multiple dimensions of attributes. In [9], the authors found that the users' mobility patterns follow a power-law distribution, which is similar to the model of language. Based on this conception, many data-driven tools for language are also applied in processing large-scale trajectory data set. For example, in [10], the authors used long short-term memory (LSTM) [11] and recurrent neural network (RNN) [12] to learn spatio-temporal trajectory patterns, so as to identify trajectories to their generating users. Except for spatio-temporal check-ins, POI information such as restaurants, retail stores, and grocery stores, is also informative in the trajectory as they represent specific physical locations which users may find common interest. In [13], the authors focused on the living patterns and utilized POI information to cluster users with similar living habits. In [14], to deal with a sparse user activity data, the authors proposed to exploit POI among different users and apply nonnegative tensor factorization to collaboratively infer users' temporal activities. Both [13] and [14] demonstrated that POI semantic can help mine users' preferences, and moreover, they both consider time-slot information as one attribute in the trajectories. Our work differs from the above works in four aspects: 1) in our model, except for spatio-temporal check-ins, we also consider attributes of location (referred them as POI information in trajectory) and attributes of users' behavior (referred them as BOI information in trajectory); 2) we map trajectory with multiple attributes into users' portrait vectors via a multi-channel embedding method; 3) we use a bidirectional encoder representations from transformers (BERT) model and a Word2vec with transformer model to pre-train the sequences in the trajectory so as to find good portraits for each user; 4) those portraits can be further fed to classfier or clusterer to mine users' preferences and relationship.

It is worth mentioning some works [15]–[17] that tried to predict infection risks prior to the spread of COVID-19 from epicenters to the city. The main approach is to simulate human mobility from known real-data of spreading records. This is beyond the scope of this work. The rest of the paper is organized as follows. In Section II, we provide the problem formulation and define the trajectory consisting of super nodes in epidemiological investigations. In Section III, a multi-channel embedding method is proposed to map each trajectory to vectors in the users' portrait space. Moreover, based on users' portraits, we provide supervised and unsupervised methods to serve different purposes in epidemiological investigations. Simulation results are illustrated in Section IV where the proposed model is verified on both synthetic data set and real data set. We conclude this work in Section V.

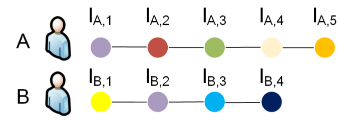


Fig. 1. A trajectory consists of locations in chronological order. Each user has his own trajectory.

II. Problem formulation

It is well known that users' mobility can be identified by (t, $\ell_{i,n}$) tuples, where t is the check-in time, $\ell_{i,n}$ is the check-in location at time t, and the number of check-ins by user i is n_i. Based on a series of spatio-temporal check-ins, a trajectory T_i can be defined by a sequence of physical locations $\ell_{i,n}$:

$$T_i = (\ell_{i,1}, \ell_{i,2}, ..., \ell_{i,n}, ..., \ell_{i,n_i}). \tag{1}$$

As shown in Fig. 1, each node in trajectory T_i denotes a physical location that user i visits and all nodes in T_i are arranged in chronological order. As each user could have its own trajectory, the length of the trajectory sequence, i.e., $|T_i|$, might be varying with user i. In the literature, there has been many works [4]–[9] that study how to extract users' latent information from spatio-temporal trajectories.

A. The Super Node in Trajectory

With the development of services in social platforms, check-in information becomes more diverse. For example, the locations can be divided into specific categories, such as restaurant, school, home and supermarket. In the literature, those categories are refereed to point-of-interest (POI), which contain the semantic information or attributes of the locations. In the meanwhile, users' behaviors at each location could be also diverse. For example, why this user arrives at this location, for how long he often stays there, via what kind of transportation he takes to this location, how much money he normally spends there, whether he likes this place or not. We consider all these check-in behaviors as users' behavior-of-interest (BOI) at this location, which are also attributes of the nodes in the trajectory.

Therefore, we define a check-in in the trajectory as a super node $S_{i,n} = (\ell_{i,n}, p_{i,n}, b_{i,n})$, where $\ell_{i,n}$, $p_{i,n}$, and $b_{i,n}$ respectively denote the physical location, semantic of location and semantic of behavior conducted by user i. As is shown in Fig. 2, each user has its own trajectory consists of super nodes $S_{i,n}$. We thus re-define the trajectory T_i as

$$T_i = (S_{i,1}, S_{i,2}, ..., S_{i,n}..., S_{i,n_i}). \tag{2}$$

Herein, each super node contains multiple attributes such as location, POI, and BOI. All super nodes are arranged in chronological order to form a trajectory T_i. Apparently, the length of the trajectory, i.e., $|T_i|$, might be varying with different users.

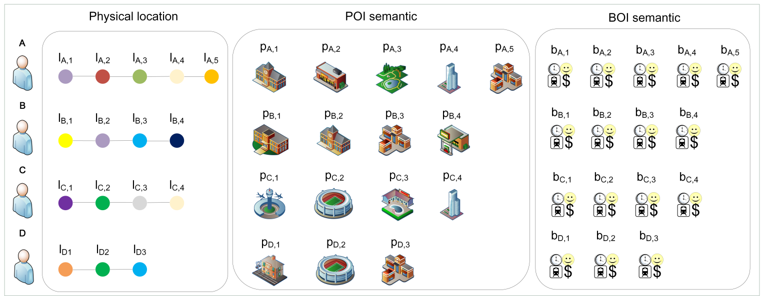


Fig. 2. Multiple attributes in trajectory. Each user will have his own trajectory. Each check-in in the trajectory has three types of attributes: geographical location, POI and BOI. POI represents the specific category of location such as park, school or work place. BOI represents users' behavior, such as transportation choice, periodic behavior, consuming amount or rating.

B. The Concatenation of Attribute Sequences

To better understand the trajectory model, we can partition attributes in each super node into different sequences, i.e., the location sequence L_i, the POI sequence P_i, and the BOI sequence B_i:

$$L_i = [\ell_{i,1}, \ell_{i,2}, ..., \ell_{i,n}, ..., \ell_{i,n_i}], \tag{3}$$

$$P_i = [p_{i,1}, p_{i,2}, ..., p_{i,n}, ..., p_{i,n_i}], \tag{4}$$

$$B_i = [b_{i,1}, b_{i,2}, ..., b_{i,n} ..., b_{i,n_i}]. \tag{5}$$

Therefore, a reshaped trajectory can be written as

$$\hat{T}_i = [L_i; P_i; B_i]. \tag{6}$$

Apparently, $\hat{T}_i$ depicts the same trajectory patterns as T_i, while we arrange attributes in different dimensions. Our next task is to find a unified model $f : \mathbb{R}^{n_i} \mapsto \mathbb{R}^d$ to map sequences in trajectory space into vectors in $\mathbb{R}^d$. That is,

$$\boldsymbol{X}_{i,1} = f(L_i), \quad \boldsymbol{X}_{i,2} = f(P_i), \quad \boldsymbol{X}_{i,3} = f(B_i), \tag{7}$$

where $\boldsymbol{X}_{i,1}, \boldsymbol{X}_{i,2}, \boldsymbol{X}_{i,3} \in \mathbb{R}^d$. Note that, the length of the sequence in the trajectory space is n_i. After this mapping, we can concatenate the above vectors into a long vector $\boldsymbol{X}_i \in \mathbb{R}^{3d}$:

$$\boldsymbol{X}_i = (\boldsymbol{X}_{i,1}; \boldsymbol{X}_{i,2}; \boldsymbol{X}_{i,3}), \tag{8}$$

which could be considered as the portrait of user i. In the next section, we will introduce how to obtain $\boldsymbol{X}_i$ from trajectories and how to utilize it to serve different purposes.

III. Processing the Trajectory

A. The Pre-training Process

In the real world, trajectory sequences are often high-dimensional, sparse and not uniformly sampled. Therefore, it is usually difficult to directly extract preferences and users' relationship from the trajectory patterns. For each sequence of attributes, i.e., L_i, P_i, and B_i, we aim to find a unified model to map sequences into vectors in $\mathbb{R}^{3d}$. We call this process as pre-training. In language processing, some unsupervised feature-based methods named embedding, such as Word2vec [18], Doc2vec [19], or BERT [20], can map a word or a document into a low-dimensional and dense representation. This motivates us follow the same way to map tokens (values of attributes) in trajectories into the same representation.

Once we have got dense representation for each user, the next step is to use the portrait as features to guide unsupervised or supervised tasks. Recently, in many tasks such as text classification, self-attention mechanism [21] has made significant improvements by overcoming dependency of long distance in sequences. In the next, we will propose fusion methods based on self-attention mechanism.

B. An Unsupervised Feature-based Model

In Fig. 3, we introduce an unsupervised feature-based model. In this model, the location sequence L_i, the POI sequence P_i, and the BOI sequence B_i are independently embedded by a BERT block. BERT is a novel pre-training model which is originally devised for language processing. Previously, most language processing models such as LSTM and RNN are unidirectional. That is, the pre-training object is completed by predicting the next word based on the previous word. BERT, however, aims to alleviate this limitation of unidirectionality by using a "masked language model", which randomly masks some of the tokens from the input and then predict the original token id of the masked one based only on its context, so that part of the information can be merged into the token representation. To find a good token contextual representations, BERT resorts to the so-called transformer [21], which is a machine translation framework based on self-attention mechanism. The attention mechanism typically maps a query vector and a set of key-value vector pairs to an output vector. The output of the transformer is defined as a weighted sum of the values, wherein the weight assigned to each value is computed following a compatibility function in terms of the query and the corresponding key. The key of the transformer design is to set the distance between any two tokens to 1, which is very effective in solving the long-term dependency problem in sequence processing. To capture sequential sequences, the transformer uses feature of position embedding when encoding token vectors. Specifically, position coding adds the position information of the token in the sequence, so that the transformer can distinguish tokens in different positions.

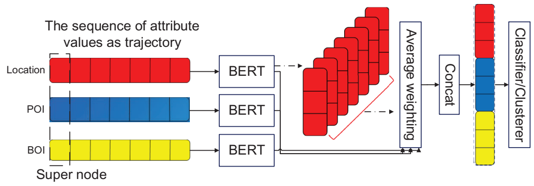


Fig. 3. Fuse multiple attributes by a BERT framework

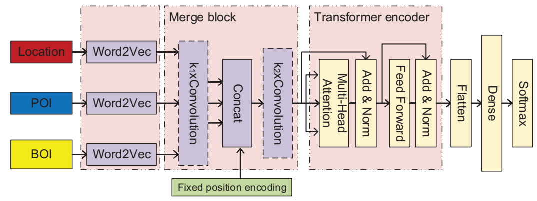


Fig. 4. A framework based on a transformer

As shown in Fig. 3, after encoding sequences of attributes by BERT, each token in L_i, P_i, or B_i, i.e., $\ell_{i,n}, p_{i,n}, b_{i,n}$ will have its own representation $\boldsymbol{x}_{i,1}^n, \boldsymbol{x}_{i,2}^n, \boldsymbol{x}_{i,3}^n \in \mathbb{R}^d$. To map a sequence into a vector, we just average weighting on these representations: e.g., $\boldsymbol{X}_{i,1} = \frac{1}{n_i}\sum_{n=1}^{n_i} \boldsymbol{x}_{i,1}^n$. By concatenating vectors of three sequences, i.e., (8), users' portrait can be encoded and later fed into a classifier or a clusterer.

C. A Supervised Feature-based Model

In the above BERT and Doc2vec frameworks, the pre-training process is unsupervised. However, as we have labels in this data set, we can actually adopt the framework in Fig. 4 to perform a supervised pre-training. Inspired by the combination of contextual representation and attention mechanism in BERT, we propose a supervised learning scheme based on Word2vec, a merge block, and transformer. To proceed, the location sequence L_i, the POI sequence P_i, and the BOI sequence B_i are first independently embedded by a Word2vec model [18]. The reason why we choose Word2vec here is to the preserve feature extraction efficiency and convergence on large volume data. As described above, the contextual information is significantly important, so we try to solve this problem by the attention mechanism. Since it might be coarse by concatenating encoded tokens for transformer's inputs, we employ a merge block to process embedded vectors before the input of transformer. The detailed structure is shown in Fig. 4, wherein a fixed position encoder provides a code pos which represents the order of the current token in the sequence. If we adopt one-hot encoding, then pos will be with length n_i. A 1-D convolution layer with k_1 filters is then applied on each encoded token to extract features. After this convolution layer and a concatenation block, we have the vectors $\boldsymbol{x}^n \in \mathbb{R}^{3\times k_1+n_i} = (\boldsymbol{x}_{i,1}^n; \boldsymbol{x}_{i,2}^n; \boldsymbol{x}_{i,3}^n; pos)$. To avoid processing a high-dimensional input in the transformer, another convolution layer with k_2 filters is used to set the input dimension of the transformer as $\boldsymbol{x}^n \in \mathbb{R}^{k_2}$. Note that, this pre-training process is supervised in a sense that we optimize the weights in the merge block and the transformer based on the labels. In the end, we feed $[\boldsymbol{x}^1; \boldsymbol{x}^2; ...; \boldsymbol{x}^{n_i}]$ into the dense layer with a softmax function to conduct the classification task.

IV. Numerical Results

In this section, we test the proposed frameworks on both a synthetic data set and a real data set. As the synthetic data has labels, we thus embed the trajectory and feed the feature vectors into a classifier to accomplish a classification task. The real data set has no labels and thus we utilize the feature vectors in a clustering task.

A. The Synthetic Data

The synthetic data we used in this experiment is generated from a real advertising system. In the scenario of an advertising system, as users will continuously browse numerous advertisements, which is quite similar to the check-in process in the trajectory model. Therefore, we may consider user's click sequence as the trajectory. Each advertisement may contain different attributes, which are analog to POI and BOI information in the trajectory model. We therefore generate our synthetic data from the data set coming from 2020 Tencent advertising algorithm competition [22] (c.f., *https://algo.qq.com/*). The detail of this data set is shown in Table I. It contains approximately 30 million users' advertisement browse history and each has 6 attributes. According to the data properties in the advertising system, we choose "creative id", "ad id", and "advertiser id" three attributes as our location, POI, and BOI attributes to form a super node. The user's age and gender information are considered as Label A and Label B for each user in our trajectory model. Herein, Label A is a binary-classification label and Label B is a multi-classification label.

TABLE I
Attributes of advertisement data

Our attributes	Ad. attributes	No. of value
physical location	creative id	2,481,135
POI	ad id	2,264,190
BOI	advertiser id	52,090
user	user	900,000
gender	Label A	2
age	Label B	10

We first run the BERT framework in Fig. 3 on this synthetic data. The BERT architecture we used is based on the RobertaModel [23]. In the pre-training process, 20% tokens are randomly masked or replaced by other tokens, so that the transformer can keep a distributed representation of each token. To accelerate the convergence of BERT, we first embed sequences of attributes into a 128-dimensional vector by a Word2vec block for an initialization of BERT's lookup table. Then, we follow the parameters setting in [20] to set the number of layers as 12, the hidden size as 512, and the number of attention heads as 16 in the BERT block and eventually BERT will output a 512-dimensional vector for each tokens in trajectory sequences. We consider that each token' contribution is the same and thus the average weights of all tokens in a sequence can generate a 512-dimensional vector for representing each attribute sequence, i.e., $\boldsymbol{X}_{i,1}$ for L_i, $\boldsymbol{X}_{i,2}$ for P_i, and $\boldsymbol{X}_{i,3}$ for B_i; see equation (7). After this mapping, a fusion process concatenates $\boldsymbol{X}_i = (\boldsymbol{X}_{i,1}; \boldsymbol{X}_{i,2}; \boldsymbol{X}_{i,3}) \in \mathbb{R}^{1536}$ as shown in (8).

Moreover, we also replace the unsupervised BERT pre-training process in Fig. 3 with a Doc2vec embedding block for a comparison. Herein the Doc2vec block is implemented using the gensim library [24]. The embedding dimension size is set as $d = 150$ and the window size as 5. We remove the tokens with a frequency of less than 5 and run both the skip-gram model and the CBOW model [19] to obtain two dense vectors. Each attribute sequence (L_i, P_i, or B_i) is represented by concatenating two dense vectors into a 300-dimensional vector; see equation (7). After this mapping, a fusion process is applied to obtain concatenated vectors $\boldsymbol{X}_i \in \mathbb{R}^{900}$.

After a unsupervised pre-training, users' portrait ($\boldsymbol{X}_i$) can be fed into a classifier, such as a lightgbm classifier [25] or a DNN classifier. To conduct the classification task, we split 90% of the data as the training data set and the rest as the testing data set. As shown in Table II, for the binary-classification task, the fusion vector $\boldsymbol{X}_i$ can provide a slightly better performance than the individual vector $\boldsymbol{X}_{i,1}$, $\boldsymbol{X}_{i,2}$ or $\boldsymbol{X}_{i,3}$, while for the multi-classification task, the fusion vector $\boldsymbol{X}_i$ can significantly improve the accuracy. This implies that jointly using location, POI and BOI information can improve the performance of the trajectory model. In Fig. 5(a) and Fig. 5(b), we show the convergence of the training process. In each epoch, the proposed model is trained on a complete training set. From the figures, we see that after several epochs, the performance of classifiers becomes stable. A DNN classifier converges much faster than tree-based ensemble learning classifier. The fusion of different attributes can indeed improve the performance, especially in the multi-classification task.

TABLE II
THE PERFORMANCE OF METHODS ON SYNTHETIC DATA

Method	Attribute \ Accuracy	Age	Gender
Doc2vec+lightgbm	physical location	38.7%	93.0%
	POI	38.9%	93.1%
	BOI	37.1%	89.7%
	3 attributes fusion	**41.4%**	**93.5%**
BERT+DNN	physical location	39.4%	92.9%
	POI	39.1%	93.0%
	BOI	37.1%	89.8%
	3 attributes fusion	**42.1%**	**93.5%**
Word2vec+Transformer	physical location	42.3%	93.8%
	POI	42.3%	93.8%
	BOI	39.9%	90.4%
	3 attributes fusion	**47.2%**	**94.3%**

In the supervised pre-training experiment, referring to the documentation of Word2vec, we adopt a skip-gram Word2vec model with typical embedding dimension size and window size as 128 and 8, respectively. To feed the input of transformer with a fixed length, we truncate or pad all sequences to a length of 64. Thus, the dimension of the fixed position encoding block in the merge block is 64×64 and one-hot encoding is adopted here. We set the first convolution layer with $k_1 = 64$ filters and the second convolution layer is set with $k_2 = 128$ filters. In the transformer block, 8 parallel self-attention layers are employed in the multi-head attention module and the convolution layer with 32 filters is in the feed-forward network. All convolution layers in the framework are with kernel size 1 and followed by ReLU activation function. We only use one encoder in the transformer to avoid overfitting. As shown in Table II, the accuracy of transformer in age and gender are 47.2% and 94.3% respectively, which are both better than other methods. The convergence comparison is also displayed in Fig. 5(c).

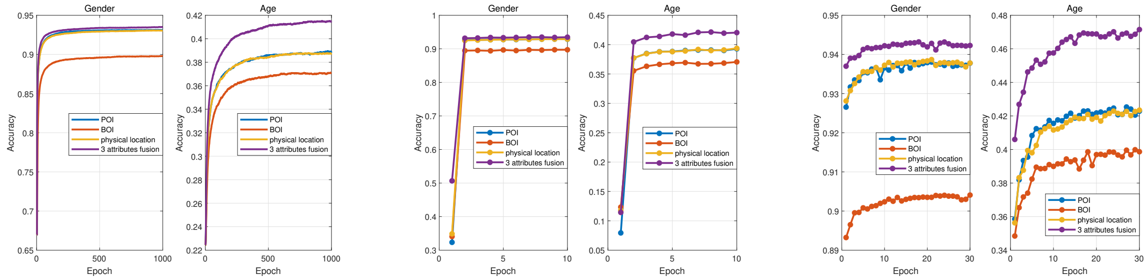
This series of experiments show that the multi-attribute trajectory can represent users' preferences and relationships. Therefore, in the application of epidemiological investigation, one can label users according to different purposes and make further prediction.

B. The Real Data

To set up the experiment on real data, we adopt the Foursquare data set in [14]. This data set contains check-ins in Tokyo collected for about 10 month (from 12 April 2012 to 16 February 2013). There are in total $573,703$ check-ins, which contain about the specific location where the user visited, the timestamp when the user visited this location, as well as POI, i.e., the concrete category of location. We use the timestamp to construct the BOI attribute by summarizing users' timeslots where users frequently appear at this location. Specifically, we divide a week into 168 hours, and users' behavior will be represented as any one of 168 hours according to their timestamp. As we don't have proper labels in this data set, a clusterer is applied to analyze users' portraits. We adopt the unsupervised pre-training model in Fig. 3 to find each user's portrait vector $\boldsymbol{X}_i$. Then, we applied a k-means algorithm and visualize the results by a t-SNE [26] method as shown in Fig. 6. Note that herein the number of clusters is estimated by an Eblow method [27]. Fig. 6 can help us to explore the similarities and differences of trajectory patterns among users. For example, in this figure, if a group of people in the same cluster are infected, we can find the individuals that are more suspicious to be affected by using their trajectory patterns, which are nodes with the same color in Fig. 6. In this way, we can identify key users for supervision and devise more advanced strategies to control the pandemic.

V. Conclusion

In this paper, we provided two frameworks to extract users' preferences and relationships from trajectories with multi-attributes. Each attribute sequence is embedded with unsupervised or supervised pre-training blocks, then fed to a classifier or a clusterer for different tasks. By evaluating the proposed frameworks on synthetic and real data sets, we prove that fusing multiple attributes can help improve the accuracy in the binary-classification or multi-classification tasks. This provides us a feasible way to better mine trajectory for controlling the epidemic.



(a) Doc2vec + Lightgbm (b) BERT + DNN (c) Word2vec + Transformer

Fig. 5. The convergence of three methods

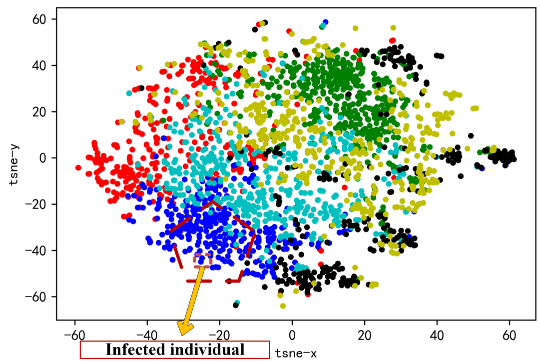


Fig. 6. The visualization of user's portrait

References

[1] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1082–1090.

[2] M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010, pp. 458–461.

[3] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo, "Socio-spatial properties of online location-based social networks," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, no. 1, 2011.

[4] K. Zhou, H. Zha, and L. Song, "Learning Social Infectivity in Sparse Low-rank Networks Using Multi-dimensional Hawkes Processes," in *AISTATS*, 2013.

[5] L. Q. Tran, M. Farajtabar, L. Song, and H. Zha, "NetCodec: Community Detection from Individual Activities," in *SDM*, 2015.

[6] A. C. Türkmen, G. Çapan, and A. T. Cemgil, "Clustering Event Streams With Low Rank Hawkes Processes," *IEEE Signal Processing Letters*, vol. 27, pp. 1575–1579, 2020.

[7] B. Yuan, H. Li, A. L. Bertozzi, P. J. Brantingham, and M. A. Porter, "Multivariate Spatiotemporal Hawkes Processes and Network Reconstruction," *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 2, pp. 356–382, 2019.

[8] A. Likhyani, V. Gupta, S. P.K., D. P., and S. Bedathur, "Modeling Implicit Communities from Geo-tagged Event Traces using Spatio-Temporal Point Processes (to appear)," in *Proc. of the 21st International Conference on Web Information Systems Engineering (WISE 2020)*, October 2020.

[9] H. Gao, J. Tang, and H. Liu, "Exploring Social-Historical Ties on Location-Based Social Networks," *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, 11 2012.

[10] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, and F. Zhang, "Identifying Human Mobility via Trajectory Embeddings," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1689–1695.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *CoRR*, vol. abs/1412.3555, 2014.

[13] H. Cao, F. Xu, J. Sankaranarayanan, Y. Li, and H. Samet, "Habit2vec: Trajectory Semantic Embedding for Living Pattern Recognition in Population," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1096–1108, 2020.

[14] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2015.

[15] Q. Hao, L. Chen, F. Xu, and Y. Li, "Understanding the Urban Pandemic Spreading of COVID-19 with Real World Mobility Data," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '20, 2020, pp. 3485–3492.

[16] J. Feng, Z. Yang, F. Xu, H. Yu, M. Wang, and Y. Li, "Learning to Simulate Human Mobility," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '20, 2020, pp. 3426–3433.

[17] C. Xiao, J. Zhou, J. Huang, A. Zhuo, J. Liu, H. Xiong, and D. Dou, "C-Watcher: A Framework for Early Detection of High-Risk Neighborhoods Ahead of COVID-19 Outbreak," 2020.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[19] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, no. 2, 2014, pp. 1188–1196.

[20] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *CoRR*, vol. abs/1810.04805, 2018.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *CoRR*, vol. abs/1706.03762, 2017.

[22] H. Wang, https://github.com/guoday/Tencent2020_Rank1st, note = Accessed: 2021-01-20.

[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[24] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, May 2010, pp. 45–50.

[25] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 3146–3154.

[26] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[27] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.