Identifying Similar Users Based on Metagraph of Check-in Trajectory Data

Rui Song, Tong Li, Xin Dong, Zhiming Ding
Faculty of Information Technology
Beijing University of Technology
songrui@emails.bjut.edu.cn, litong@bjut.edu.cn, dongxin19@foxmail.com, zmding@bjut.edu.cn

Abstract— **Identifying similar users lay the foundation in many fields, such as friend recommendation, user-based collaborative filtering, and community discovery. It is useful to analyze users' similarity based on check-in data, especially the analysis of spatiotemporal and semantic information. The existing works pursue semantic similarity of user trajectories and cannot distinguish the effects of geographical factors in a fine-grained way. This paper proposes a graph embedding approach to identify similar users based on their check-in data. We firstly identify meaningful concepts of user check-in data, based on which we design a metagraph for representing features of similar user behaviors. Then we characterize each user with a sequence of nodes that are derived through a metagraph-guided random walk strategy. Finally, the sequences are embedded to generate meaningful user vectors that are used to the similarity among users and thus identify similar users. We evaluate our proposal on two datasets, the results of which show that our proposal can outperform the baselines.**

Keywords-user similarity; metagraph; random walk; embedding; check-in data

I. INTRODUCTION

With the fast development of Location-Based Social Network (LBSN) platforms, an increasing number of users can check-in at various Point of Interests (POIs) conveniently and share their experiences, resulting in massive user check-in trajectory data. Such check-in trajectory data contains information about when and where a user visited a POI, which indicates users' behaviors and preferences. Analyzing check-in data has contributed to identifying similar users, laying the foundation in many fields, such as location prediction [1], community discovery [3], user-based collaborative filtering and location recommendation [2].

Some early user similarity studies focus only on the geographic features of the trajectory [4][5][6], which is limited by the geographic distance and ignores the semantics [7]. Recent works have measured the similarity of users by mining the semantics of GPS trajectories [7][8][9], but the geographical impacts are not considered deeply enough. For example, in Fig.1, the three different user trajectories are typical to be treated as the same, and not further distinguished in terms of their detailed geographical information [7][8][9]. However, users who live closer to each other are supposed to be more similar than others. As shown Fig.1, affected by geographical location, *trajectory1* is more similar to *trajectory2* than *trajectory3*. As such, we argue that the semantic and geographic features should all be considered when calculating user similarity.

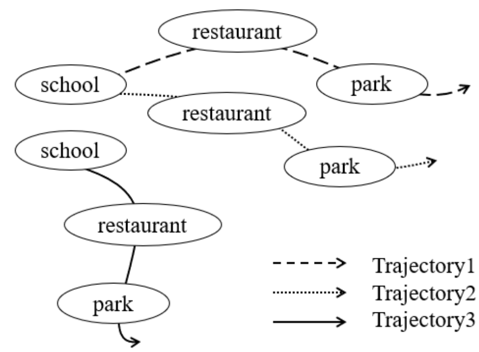


Figure 1. Examples of three users' semantic trajectories

To consider both semantic and geographic features, we leverage the graph embedding technique to measure user similarity. Graph embedding techniques have been proven as a practical approach for representing profound meanings of entities and relations. Specifically, the trajectory check-in data contains different types of nodes (user, time, POI, etc.), which establishes a graph with heterogeneous nodes. The detailed semantics of check-in data can be obtained by embedding such a graph based on metagraph, which contributes to calculating the similarity among user trajectories. Some pioneer studies have been done regarding this topic. Dong et al. [14] and Zhang et al. [15] transformed the structure into the input of the embedded model by recording the node sequences of the random walk in the graph. They use metagraph-guided random walk sequences to capture semantic information between different types of nodes and thus to improve the quality of transformation. However, they do not consider spatiotemporal features.

In this paper, we propose a metagraph embedding based approach to identify similar users using their check-in trajectory. The main contributions of this paper are summarized as follows:

1) We identify meaningful concepts of user check-in data, based on which we design a metagraph for representing features of similar user behaviors.

2) We propose an improved metagraph-guided random walk algorithm to adapt to time and location similarity, which is used to characterize each user with a sequence

DOI reference number: 10.18293/SEKE2020-118

of nodes. The sequences are embedded to generate meaningful user vectors that can be used to effectively calculate user similarity.

3) We compare our approach with the state-of-the-arts based on two datasets, the result of which show that our approach can outperform others.

The rest of this paper is organized as follows: Section II describes some related works. Section III describes the details of our method, including metagraph of check-in data, the improved metagraph-guided random walk, embedding and user similarity calculation. Section IV evaluates our approach. Section V summarizes the paper.

II. RELATED WORK

This paper identifies similar users by combining the semantic and spatio-temporal features of check-in trajectories. We leverage the metagraph-guided heterogeneous graph embedding to calculate the user's representation and then calculate the similarity. This section will introduce related works of calculating similarity based on semantic trajectory and graph embedding algorithms.

A. User Similarity Based on Semantic Trajectory

Horozov et al. [10] first construct the user activity's vectors by using the user's votes on POIs, and then calculate the Pearson similarity to represent the user similarity. Mazumdar et al. [11] compare user similarity by combining the length, support and check-in distribution of common location sequences using GPS data. However, these methods can't measure users who have similar preferences but live far away. To solve this problem, Ying et al. [7] propose a method to measure the user semantic similarity by Maximal Semantic Trajectory Pattern (MSTP). They first identify the stopping point from the GPS trajectory data, and use the landmarks collected from Google Map to form the semantic trajectory. Then the user similarity is calculated by the weighted average of the maximum semantic trajectories. Chen et al. [8] found that the similarity between two identical users is not equal to 1 in [7]'s work. To this end, Chen et al. propose a method called Maximal Trajectory Pattern (MTP) to fix the shortcomings in [7] by using the longest common semantic patterns. Later, Chen et al. [9] propose a method to calculate the user similarity according to the Common Pattern Set (CPS), they introduce the support value distribution of common patterns to solve the problem of indistinguishable pattern frequencies in literature [8]. However, these methods ignore the influence of geographical factor while studying semantics. We argue that semantic and geographic features should be considered when calculating user similarity.

B. Embedding Learning

Embedding is a way to transform discrete variables into continuous vector representations [23]. In neural networks or graphs, embedding can not only reduce the spatial dimension of nodes, but also represent the nodes in a meaningful way. Embedding learning is to learn the vector representation of nodes in the metric space through specific methods, such as LINE [12], DeepWalk [13], etc.. Deepwalk uses a neural language model (skip-gram) to embed graph. The authors first use random walks to uniformly sample the neighbors of the nodes from the graph as a path. The paths are treated as sentences, and nodes are treated as words. Then the skip-gram model is used to train the representation of the nodes. To better preserve the relationship between nodes during the embedding process, LINE proposes the concepts of the first-order similarity and second-order similarity. However, these works focus on homogeneous networks which have a single type of node or relationship. Some pioneer studies have been done regarding this topic. Dong et al. [14] propose a method (named as Metapath2vec) for embedding in heterogeneous networks. They use a random walk based on metapath to get the sequence of nodes, and then improved the skip-gram model to learn the embeddings. Zhang et al. [15] argue that metagraph has richer semantics than meta-path. They use metagraph to analyze the behaviors of authors' published papers and classify similar authors. However, there is no time and location limit for the author to publish papers, thus their work can not apply to check-in data. Based on these works, this paper leverages the metagraph to study the behavior of users visiting POIs, introduces time and location constraints in random walk algorithms, and learns user's embedded representation.

III. METHODOLOGY

The framework of our method, as shown in Fig.2, named as Metagraph-Guided Embedding (MGE). We first design the metagraph of check-in trajectory data to represent similar user behaviors. Then we characterize each user with a sequence of nodes that are derived through a metagraph-guided random walk strategy. Finally, the sequences are embedded to generate meaningful vectors that are used to calculate user similarity. Here are the basics concepts in this paper, and the details of our framework.

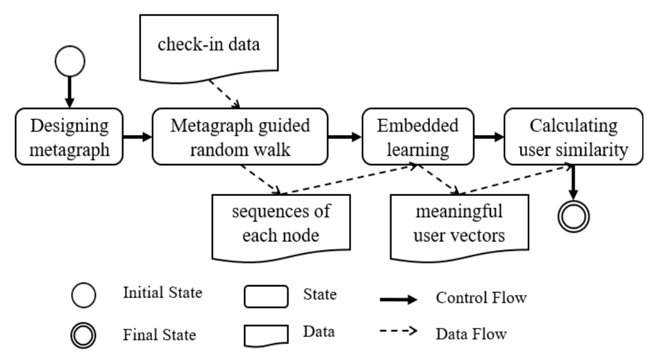


Figure 2. The framework of our method (MGE).

A. Concepts

We briefly introduce the basic concepts of check-in data and metagraph.

POI: A POI is defined as a place that has a special function or meaning to users (e.g., a school or a bank). In our method, a POI has four attributes: identifier, name, geographical location, and category.

Trajectory: A trajectory is a path that a user takes in space over a period of time. It can be regarded as a spatial point with chronological order, which records the user's geographical locations at different times. The check-in data studied in this paper has POI profiles, which is a trajectory data with location semantic.

Check-in: A check-in is a quadruple *(u, t, g, p)*, where *u* is a user, *t* is a time slot, *g* is a geographical location coordinate (including latitude and longitude), and *p* is a POI. A check-in means a user *u* visiting a POI *p* at a time slot *t* and a geographical location *g*. It is worth noting that we discretize timestamps associated with check-in records into 24-time slots based on hours, as other works have done [16].

Metagraph: A metagraph is a relational hypergraph representing multi-relational and multi-dimensional data [17]. It is a graph with its nodes denoting the entities and its edges representing the interaction between nodes. Different from the traditional graph concepts, each node in metagraph represent a set of an entity. For example, the user, POI, time slot, and geographical location can be regarded as entities. Fig.3 shows the metagraph of our paper.

B. *Designing Metagraph*

Mining check-in trajectory data can discover user behavior features. We argue that users with similar behaviors have similar check-in characteristics. For instance, users will check-in at similar time slots, geographical locations, or POIs. In order to measure the similarity of user behaviors, we design a metagraph of check-in, which can reflect the meanings that two users visit the same POI when they are at similar time slots and geographical locations. We use *U, T, G, P* to represent the set of users, time slots, geographical locations and POIs, respectively. In particular, user, time slot, geographical location, and POI are defined as entities, and serve as different types of nodes in the metagraph.

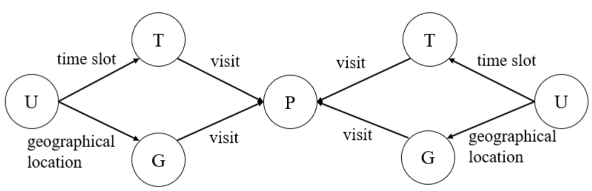


Figure 3. Metagraph of check-in data

Fig.3 shows the metagraph of check-in data, which describes that two users are relevant in check-in activity if they have similar time slots and geographical locations in the same POI. At this point, the readers only need to understand the meaning of the metagraph. The method of defining the similar time slots and similar locations will be described in detail in the next subsection.

C. *Metagraph Guided Random Walk*

Using a random walk based on metagraph, we can capture meaningful semantics from the data, which find similar user behaviors. Our metagraph of check-in data has the constraints of similar time slots and similar locations. Therefore, we improved the random walk strategy, defining similar time slots and similar locations to limit the random walk process. Here we show the definition.

Similar time slots. Given two time slots *t1* and *t2*, a time threshold τ, if $|t1\text{-}t2|\leq\tau$, then *t1* and *t2* are similar time slots.

Similarity locations. Given two geographical coordinates *g1* and *g2*, a distance threshold δ, the function *distance(a, b)* represents the geographic distance between the location *a* and *b*, if $distance(g1, g2) \leq \delta$, then *g1* and *g2* are similar locations.

To meet the constraints of time and location, we propose an improved random walk strategy and list the following four principles.

1) *Every node that random walks exists in the instantiated graph network.* Otherwise, it is meaningless.
2) *Random walk starts with user type nodes.* Because this paper studies the similarity of users, we focus on user-type nodes.
3) *Random walks are limited by the structure of the metagraph.* The metagraph is used to describe similar user behaviors. Walking in the structure of metagraph can capture the semantics of metagraph.
4) *Random walks are constrained by time and location nodes.* To ensure similar time slots, the time threshold must be met when randomly walking time-type nodes. In the same way, the distance threshold is met when randomly walking the location nodes to ensure similar locations.

To help understand the above principles, given a graph network of check-in data in Fig.4, Table I shows the correct and incorrect walking sequences, we assume the time threshold is 3 (hours), and locations are similar between nodes. For ease of observation, the example sequences omit the time and location nodes, only leaving the user and POI nodes.

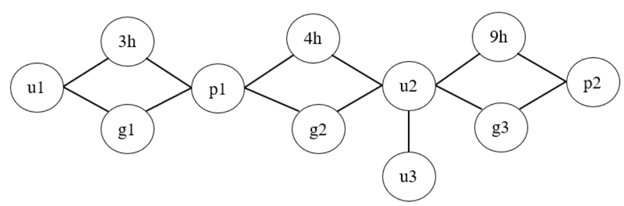


Figure 4. Example of a graph network.

TABLE I. EXAMPLES OF RANDOM WALK SEQUENCES

Example	Remark
u1, p1, u2	Correct
u1, p3, u2	Incorrect. Violate the principle 1: p3 is not in graph
p1, u2, p2	Incorrect. Violate the principle 2: the start node is not a user
u3, u2, p2	Incorrect. Violate the principle 3: does not satisfy the metagraph structure
u1, p1, u2, p2	Incorrect. Violate the principle 4: the time difference does not meet the threshold

Next, we formally describe the node transition probability of random walks. Different types of nodes (i.e. user, time, location, POI) constitute a heterogeneous information network (HIN) [18]. Given a HIN $H = (V, E)$ and the metagraph m, where V is a vertex set, E is an edge set. Equation (1) defines the node transition probability.

$$P(v_i \mid v_{i-1}; m, H) = \frac{1}{N^{\varphi_{v_{i-1}}(v_i)}} \tag{1}$$

Where $\varphi(\cdot)$ is a function of the node type, $\varphi_{v_{i-1}}(v_i)$ represents the type of v_i and the previous node is v_{i-1}, $N^{\varphi_{v_{i-1}}(v_i)}$ represents the number of nodes which of type $\varphi_{v_{i-1}}(v_i)$. If $v_i \notin V$, the probability is 0. It is worth noting that the higher frequency of a user visiting a POI or visiting at a time slot, the higher the probability of such a time slot or POI node being selected. A walk will follow the structure of the metagraph repetitively until it reaches the pre-defined length.

The pseudocode of random walk with time and location constraints algorithm is shown in Algorithm 1. In particular, given a graph network, the random walk starts at the user-type node, walk randomly according to the node transition probability and ends at a given length. Each starting node generates a given number of paths. The output is a file containing meaningful sequences of nodes.

Algorithm 1 Random Walk with Time and Location Constraints

```
Input    The HIN H = (V, E); the metagraph m; the walk length wl; the
         number of walks per node n.
Output   A path.txt that records sequences of random walks
For each u in U do
    random walk to time node t0, and location node g0
    For i ← 1 to n do
        path = [u]
        For j ← 1 to wl do
            While (1) do
                walk to the node according (1)
                If time or location similarity are satisfied
                    append the node into path; break
            End while
        write path into path.txt
        End for
    End for
End for
```

D. Embedded Learning

Through random walks under time and location constraints, we obtain sequences of each node. We aim to convert the sequences into vectors to calculate similarity. With embedded models, given a HIN $H = (V, E)$, the task is to calculate latent representations in d- dimension $X \in R^{|V|*d}$ (a.k.a. embeddings), $d \ll |V|$. Then, we choose the skip-gram model to learn the latent embeddings of nodes. This model has been validated in [14][15]. Specifically, the skip-gram learn node representation by maximizing the probability of the occurrence node v's context nodes *Context(v)* within w window size, as shown in (2).

$$\min \sum_{v \in V} \sum_{v' \in Context(v)} -\log P(v' \mid v; \theta) \tag{2}$$

Where *Context(v)* denotes v's neighborhood based on the random walks guided by metagraph, $P(v' \mid v;\theta)$ is modeled via softmax. To speed up training, like other works [19], negative sampling is used to approximate the objective function (3):

$$\log\sigma(X_{v'} \cdot X_v) + \sum_{k=1}^{K} \log\sigma(-X_{v'_k} \cdot X_v) \tag{3}$$

Where X_v is the v_{th} row of X, representing the embedding vector of node v. $\sigma(\cdot)$ is the sigmoid function, v'_k is the k_{th} negative node sampled for node v', and K is the number of negative samples.

E. Calculating User Similarity

The higher the frequency with which two users visit the same POI at similar times and locations, the higher the behavior similarity. Based on this meaning, we get the embedded representation of user nodes. In particular, the closer the user embeddings are in the vector space, the more similar the users are. In vector space, the cosine distance pays more attention to the difference from the vector direction, which helps to distinguish and measure the similarity of users. Therefore, equation (4) use the cosine distance to calculate the user's similarity and normalization to make the result in the range [0,1].

$$sim(u_a, u_b) = 0.5 + 0.5 \, \frac{u_a \cdot u_b}{|u_a||u_b|} \tag{4}$$

Where $u_a, u_b \in X$, sim is the user similarity.

IV. Evaluation

In this section, we evaluate our method (MGE) on two datasets. Specifically, we focus on two research questions (RQ) and designed experiments for each one.

- **RQ1.** Can our method (MGE) calculate user similarity effectively?

  To answer this question, we compare our method with the MTP [8] and CPS [9] that focus on the semantic trajectory similarity calculation.

- **RQ2.** Can our method (MGE) identify similar users effectively?

  To answer this question, we compared MGE with popular network representation learning methods LINE [12] and Deepwalk [13].

A. Experiment 1

1) Dataset: to observe the results of pairwise user similarity more specifically, and to accurately compare with other literature which using the same dataset, we use the synthetic dataset, which is derived from [9]. The dataset is constructed based on six users' behaviors that first five being from the literature [9], we constructed the same behavior of *u6* as *u3*, but they live far away. The dataset consists of 76 check-ins and 4

POIs from 6 users. Fig.5 shows the six users' behaviors. We use circles to indicate POIs and arrows between POIs to represent the trajectory transition direction, and thicker arrows indicate higher transition frequencies.

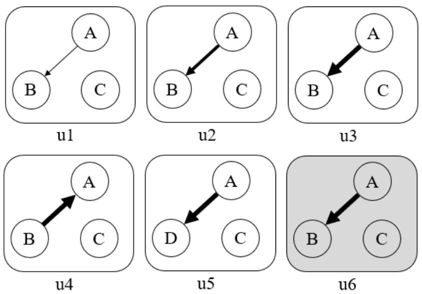


Figure 5. The six users' behaviors (The u6 living far from others).

2) Similarity metrics. the effective similarity calculation should meet the following principles, with the first four being from [9]. The last is used to verify the geographical impact.

a) sim(u, u') is in range [0,1]

b) sim(u, u') = sim(u', u)

c) sim(u, u) = 1

d) The frequency of visiting the same place affects the similarity of user behavior, i.e. sim(u2, u3) > sim(u1, u3).

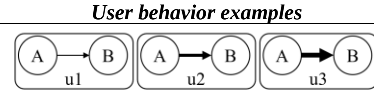
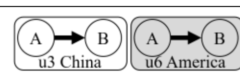
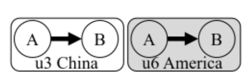
e) According to the common sense, the user's geographic location should have an impact on similarity. The similarity between geographic and semantic combination should be higher than considering only semantic similarity, i.e. sim(u3, u6) $\neq$1.

3) Comparison with Baselines: in terms of calculating user similarity, we choose MTP [8] and CPS [9] as the baselines for comparison. The reason is that after our research, most of the papers that identify similar users are based on GPS data [20][21], and the algorithms of literature [22] based on check-in data are unknown or incomplete, making it difficult to reproduce. Although [8][9] use GPS trajectory data, and their later works are to calculate user similarity by mining trajectory semantics. Since neither of them disclosed semantic trajectory datasets and codes, in order to avoid the negative impact of different datasets' types and the reproduction process on the results, we use the data from [9] and prove their weaknesses and illustrate the rationality of our method. The results of the different methods are given below.

TABLE II. PAIRWISE USER SIMILARITY

	u1			**u2**			**u3**			**u4**			**u5**			**u6**		
	MTP	***CPS***	***MGE***	***MTP***	***CPS***	***MGE***	***MTP***	***CPS***	***MGE***	***MTP***	***CPS***	***MGE***	***MTP***	***CPS***	***MGE***	***MTP***	***CPS***	***MGE***
u1	1.0	1.0	1.0	1.0	0.96	0.88	1.0	0.93	0.83	0.83	0.76	0.81	0.83	0.50	0.62	1.0	0.93	0.66
u2	1.0	0.96	0.88	1.0	1.0	1.0	1.0	0.97	0.89	0.58	0.71	0.80	0.58	0.47	0.63	1.0	0.97	0.67
u3	1.0	0.93	0.83	1.0	0.97	0.89	1.0	1.0	1.0	0.79	0.67	0.77	0.79	0.44	0.57	1.0	1.0	0.79
u4	0.83	0.76	0.81	0.58	0.71	0.80	0.79	0.67	0.77	1.0	1.0	1.0	0.79	0.44	0.58	0.79	0.67	0.59
u5	0.83	0.50	0.62	0.58	0.47	0.63	0.79	0.44	0.57	0.79	0.44	0.58	1.0	1.0	1.0	0.79	0.44	0.57
u6	1.0	0.93	0.66	1.0	0.97	0.67	1.0	1.0	0.79	0.79	0.67	0.59	0.79	0.44	0.57	1.0	1.0	1.0

TABLE III. COMPARING THE METRICS OF METHODS ACCORDING TO THE VALUES IN TABLE II

	Similarity metrics					**Result**		
Methods	***a)***	***b)***	***c)***	***d)***	***e)***	***Description***	***User behavior examples***	***Example values in Table II***
MTP	√	√	√	×	×	Violate d): The similarity of *u1*, *u2* and *u3* should not be the same.		sim(u1,u2) = sim(u1,u3) = sim(u2,u3)
						Violate e): The similarity of *u3* and *u6* should not be the same.		sim(u3,u6) = 1
CPS	√	√	√	√	×	Violate e): The similarity of *u3* and *u6* should not be the same.		sim(u3,u6) = 1
MGE	√	√	√	√	√	Meet all metrics		

Results and analysis. In Table II, and we describe the results of different methods separately. For the sake of observation, we list the measurement results of each method in Table III, which satisfy with " √ " and do not satisfy with "×", and give detail description and examples.

For MTP, the first three users *u1*, *u2* and *u3*, are not distinguished, which violates the metric *d)*. MTP's weakness has been proven in [9]. For CPS, the similarity between *u3* and *u6* is 1, which violates metric *e)* and cannot find the difference in geographical location.

MGE can distinguish *u3* and *u6* and satisfy all metrics. The reason is that we randomly walked nodes with similar geographical locations while considering semantics. Randomness can reduce but not eliminate the similarities of faraway users.

B. Experiment 2

1) *Dataset:* we use the Foursquare datasets that is one of the most popular online location-based social networks. This dataset consists of 372,387 check-ins and 90,089 POIs from 4,144 users over four years (December 2009 to July 2013). We construct the graph with users, check-in time and location, and POI as different types of nodes. We use 80% data as the training dataset and the rest as the test dataset.

2) *The experiment design:* the node classification is used to evaluate MGE. We leverage third-party labels to determine the class of each node. Foursquare offers ten categories of POI (including Arts & Entertainment, College & University, Event, Food, Nightlife Spot, Outdoors & Recreation, Professional & Other Places, Residence, Shop & Service and Travel & Transport), which can be used as labels for POI node. Like literature [15], the user's label is assigned to the category of the user's most visited POIs. In this paper, the node embeddings are as the input to the logistic regression classifier. **F1-measure**, **Precision** and **Recall** are applied to evaluate the performance [24].

3) *Parameter selection:* in random walks and embeddings, there are several parameters, such as the number of walks per node, walk length, the vector dimension and location threshold δ. We perform an analysis of these parameters in MGE and select appropriate parameters by observing the F1-measure. Fig. 6 shows the results using the control variable method. It can be seen from Fig. 6(a) and Fig. 6(b) that a larger value of walk length and numbers does not mean that the effect is better. F1 peaks at 20 and 50 respectively, but overall the parameters have little effect on the result, and the extreme value of the result is around 0.03. For the location threshold in Fig. 6(c), 10km works best. This shows that users within 10km of their current location are more likely to visit the same place, so their similarity is high. In Fig. 6(d), the dimension of the vector peaked at 96. As the dimension increased, the result did not change much.

4) *Comparison with Baselines:* in terms of embedding, we compare two popular network representation learning methods, LINE [12], DeepWalk [13]. After parameter selection, we use the same parameters in Table IV for all embedding methods.

Results and analysis. From Fig. 7, the results of LINE and DeepWalk are similar, with the F1-measure of about 0.54. MGE is higher than the baselines, with the F1-measure of about 0.63, indicating that considering similar time slots and similar locations can identify users in the same category better. Overall, our method has better performance than all baselines in F1-measure, Precision and Recall.

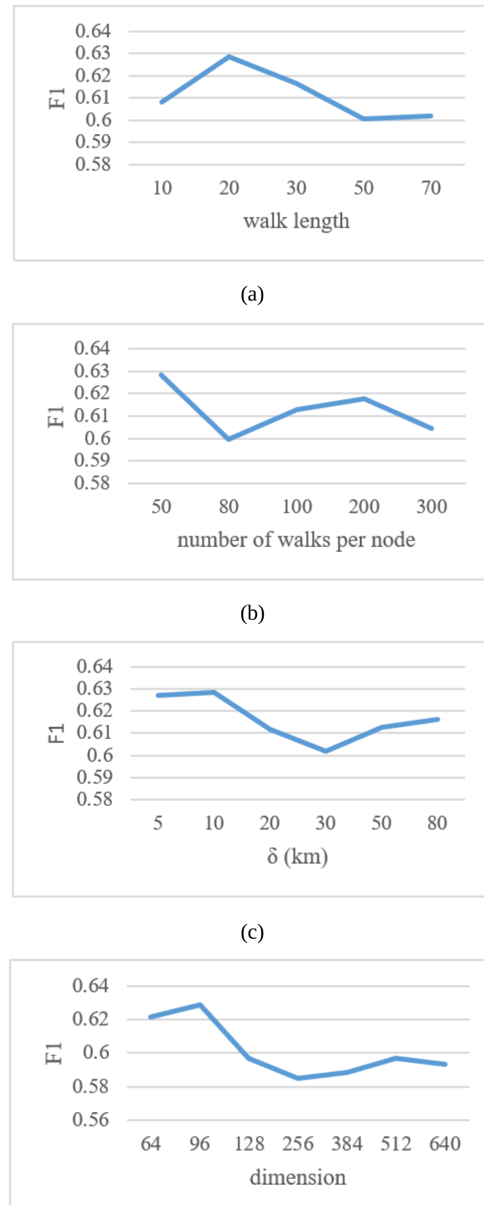


Figure 6. Comparison of different parameters

TABLE IV. PARAMETER SETTINGS

Parameter	**Number of walks per node**	**Walk length**	**Vector dimension**	***τ***	***δ***
value	50	20	96	3h	10km

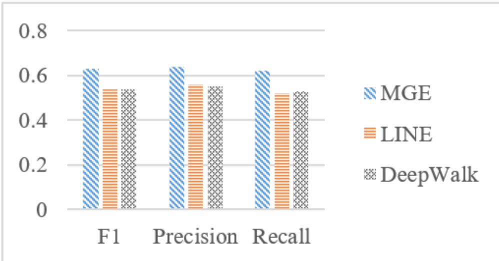


Figure 7. Comparison of different methods.

V. CONCLUSION

This paper proposes a systematic method for identifying similar users based on user check-in trajectory data. In particular, we base our approach on metagraph embeddings, in which we first designed a metagraph to represent the check-in behavior of similar users. Then, we apply a customized metagraph-guided random walk algorithm to integrate semantic and geographic similarity into our analysis. Finally, the heterogeneous skip-gram model is used to graph embedding so that we can calculate representation vectors of users, and calculate user similarity. We have designed and conducted a series of experiments which have shown the effectiveness of our methods over existing approaches.

ACKNOWLEDGMENT

This work is supported by National Key R&D Program of China (No. 2017YFC0803300, 2017YFC0803307) and the National Natural Science of Foundation of China (No. 61902010, 91546111, 91646201).

REFERENCES

[1] R. Wu, G. Luo, Q. Yang, and J. Shao, “Learning individual moving preference and social interaction for location prediction,” IEEE Access, vol. 6, pp. 10675-10687, 2018.

[2] J. Zhang and C. Chow, “TICRec: a probabilistic framework to utilize temporal influence correlations for time-aware location recommendations,” IEEE Transactions on Services Computing, vol. 9, no. 4, pp. 633-646, 2016.

[3] C. Xu, L. Zhu, Y. Liu, J. Guan and S. Yu, “Dp-ltod: differential privacy latent trajectory community discovering services over location-based social networks,” IEEE Transactions on Services Computing, pp.1-1, 2018.

[4] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W. Ma, “Recommending friends and locations based on individual location history,” ACM Transactions on The Web, vol. 5, no. 1, pp. 109-152, 2011.

[5] E. H. Lu and V. S. Tseng, “Mining cluster-based mobile sequential patterns in location-based service environments,” Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, pp. 273-278, 2009.

[6] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W. Y. Ma, “Mining user similarity based on location history,” Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 34.1-34.10, 2008.

[7] J. C. Ying, H. C. Lu, W. C. Lee, T. C. Weng, and V. S. Tseng, “Mining user similarity from semantic trajectories,” Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, ACM, pp. 19-26, 2010.

[8] X. Chen, J. Pang, and R. Xue, “Constructing and comparing user mobility profiles for location-based services,” Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 261-266, 2013.

[9] X. Chen, R. Lu, X. Ma, and J. Pang, “Measuring user similarity with trajectory patterns: principles and new metrics,” Web Technologies and Applications, pp. 437-448, 2014.

[10] T. Horozov, N. Narasimhan, and V. Vasudevan, “Using location for personalized POI recommendations in mobile environments,” International Symposium on Applications and the Internet (SAINT'06), pp. 6-129, 2006.

[11] P. Mazumdar, B. K. Patra, R. Lock, and S. B. Korra, “An approach to compute user similarity for gps applications,” Knowledge-Based Systems, vol. 113, pp. 125-142, 2016.

[12] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: large-scale information network embedding,” Proceedings of the 24th International Conference on World Wide Web, pp. 1067-1077, 2015.

[13] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: online learning of social representations,” Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.701-710, 2014.

[14] Y. Dong, N. V. Chawla, and A. Swami, “Metapath2vec: scalable representation learning for heterogeneous networks,” Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 135-144, 2017.

[15] D. Zhang, J. Yin, X. Zhu, and C. Zhang, “MetaGraph2Vec: complex semantic path augmented heterogeneous network embedding,” Advances in Knowledge Discovery and Data Mining, Springer, Cham, pp.196-208, 2018.

[16] T. Qian, B. Liu, Q. V. H. Nguyen, and H. Yin, “Spatiotemporal representation learning for translation-based poi recommendation,” ACM Transactions on Information Systems, vol. 37, no. 2, pp. 18.1-18.24, 2019.

[17] Z. Huang, Y. Zheng, R. Cheng, Y. Sun, N. Mamoulis, and X. Li, “Meta structure: computing relevance in large heterogeneous information networks,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1595-1604, 2016.

[18] Y. R. Lin, J. Sun, P. Castro, R. B. Konuru, H. Sundaram, and A. Kelliher, “MetaFac: community discovery via relational hypergraph factorization,” Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.527-536, 2009.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 26, pp. 3111-3119, 2013.

[20] L. Guo, X. Gao, B. Wu, H. Guo, H. Y. Xu, and Y. Wei, “Discovering common behavior using staying duration on semantic trajectory,” Journal of Computer Research and Development, vol. 54, no. 1 pp. 111-122, 2017.

[21] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi, “Trajectory clustering via deep representation learning,” 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 3880-3887, 2017.

[22] W. You, Z. Chenghu, and P. Tao, “Semantic-geographic trajectory pattern mining based on a new similarity measurement,” ISPRS International Journal of Geo-Information, vol. 6, no. 7, pp. 212-, 2017.

[23] H. Y. Cai, V. W. Zheng, and K. C. Chang, “A comprehensive survey of graph embedding: problems, techniques, and applications,” IEEE Transactions on Knowledge & Data Engineering, vol. 30, no. 9, pp.1616-1637, 2018.

[24] X. W. Meng, R. C. Li, Y. J. Zhang, and W. Y. Ji, “Survey on mobile recommender systems based on user trajectory data,” Journal of Software, vol. 29, pp. 3111-3133, 2018.