LEARNING LARGE-SCALE LOCATION EMBEDDING FROM HUMAN MOBILITY TRAJECTORIES WITH GRAPHS

A PREPRINT

Chenyu Tian
Tsinghua University
Shenzhen, China
tiancy19@mails.tsinghua.edu.cn

Yuchun Zhang *
Tencent Inc
Shenzhen, China
yuchunzhang@tencent.com

Zefeng Weng
Tencent Inc
Shenzhen, China
mangozfmeng@tencent.com

March 2, 2021

ABSTRACT

GPS coordinates and other location indicators are fine-grained location indicators that are difficult to be effectively utilized by machine learning models in Geo-aware applications. Previous location embedding methods are mostly tailored for specific problems that are taken place within areas of interest. When it comes to the scale of the entire cities, existing approaches always suffer from extensive computational cost and significant information loss. Increasing amount of location based service (LBS) data are being accumulated and released to public and enables us to study the urban dynamics and human mobility. This study learns vector representations for locations using the large-scale LBS data. Different from existing studies, we propose to consider both spatial connection and human mobility, and jointly learn the representations from a flow graph and a spatial graph through a GCN aided skip-gram model named GCN-L2V. This model embeds context information in human mobility and spatial information. By doing so, GCN-L2V is able to capture relationships among locations, and provide a better notion of semantic similarity in a spatial environment. Across quantitative experiments and case study, we empirically demonstrate that the representations learned by GCN-L2V are effective. GCN-L2V can be applied in a complementary manner to other place embedding methods and down-streaming Geo-aware applications.

Keywords Spatio-temporal data · Representation learning · Location embedding

1 Introduction

In modern society, location-based service (LBS) systems have brought great convenience to people's daily life with different specifications such as navigators, store recommendation, journey recording, pervasive games and social network. At the same time, LBS service generates a large amount of data every day. These data contain abundant spatial information. The mining of LBS data brings great potential for many industrial and commercial applications, such as traffic flow analysis, travel path recommendation, and location-based social networks. Location indicators like GPS coordinates are widely used in Geo-aware applications, but they are difficult to be effectively utilized by machine learning models, because location indicators only contain the information of unique locations and physical distance. More than that, each location has rich information like land use types, which are of vital importance to a wide range of Geo-related applications. In order to obtain these meaningful and accurate information, a key step is to extract effective feature representations from the raw location indicator records.

Recently, unsupervised text encoding algorithms such as Word2Vec [1] have been effectively utilized in many Natural Language Processing (NLP) tasks, and graph encoding algorithms such as node2vec [2] have been widely applied in graph learning tasks. Word embedding and graph embedding greatly boosted related studies. The intuition behind that is training deep models which encode words or nodes into vector space representations based on their positions and their context. Similar idea can be borrowed to the field of Geographic Information Science (GIS). Finding the

*Corresponding author

latent representations of different locations can achieve the embedding from high-dimension sparse location patterns (like the one-hot encoding) to low-dimension dense patterns, which can serve as the geographical "Word2Vec" to help spatial related studies. If there are models that can mapping each location into an meaningful embedding vector, they can benefit nearly all studies that need spatial information as inputs, such as similar position querying and place recommendations.

However, there are few available large-scale location embedding products or algorithms. Existing location embedding methods are mostly tailored for specific problems that are taken place within areas of interest (AOI) and Point of Interest (POI) [3–6]. However, in reality, the quantity of locations without specific utility (like suburban areas without POIs) are much larger than the quantity of AOI, but they are ignored in most studies. These limitations constrain the utility and flexibility of these models. Also, when it comes to the scale of large cities, countries, or even the whole earth, existing approaches may suffer from extensive computational cost and significant information loss.

The motivation of this study is to learn meaningful location embedding using LBS data. Because these data contains the records of human mobility, which reveals an important association information between any locations through the dynamic human mobility flow. In addition, human mobility data generally have a better coverage of unpopular areas and time periods. It means analyzing human mobility has a potential to learn urban functions through location embedding. To find latent representations of Geo indicators. The key challenge lies in how to generate a meaningful context for locations using the mobility flow data, in a similar way to the sentence contents for word embedding or the neighbor context for graph embedding. Another challenge is the data sparsity because the data usually follow a long-tail distribution with regard to locations [6]. This study leverage the massive amount of raw LBS records to train location embedding. We address this issue by proposing a method that generates fine grained location embedding, which leverages spatial information and human mobility trajectories. In experiment, we implement two quantitative experiments and case study and empirically demonstrate that the representations learned by the proposed model are effective. In summary, the key contributions of this work are:

1. We propose an graph aided representation learning model called GCN-L2V for location embedding. We also integrate flow graph and spatial graph to solve information loss issue by including both context information in human mobility and spatial information of locations.

2. To the best of our knowledge, this is the first location representation learning model that combines both graph learning and word embedding in NLP, which complement each other by giving an overall relationship and learn the vector representations of detailed locations in urban cities.

3. Two quantitative experiments and case study provide a better notion of semantic similarity in a spatial environment and empirically demonstrate that the representations learned by GCN-L2V are effective.

2 Related Works

Learning the embedding of places and locations has been a popular research topic in the urban computing field. In recent years, several efforts have been made to encode GPS coordinates or POIs at the feature level. In many supervised learning studies, location embeddings are indirectly trained as an auxiliary module to import the spatial information into the model to help classification and regression tasks. Feng *et al.* [4] proposed DeepMove to predict human mobility using a multi-modal embedding recurrent neural network and location embedding is correspondingly generated as a byproduct of the original prediction-based model. Gao *et al.* [7] identified and linked trajectories to users. The model represents each check-in with a low-dimensional vector to mitigate the problem of the curse of dimensionality. The intuition of location embedding is also applied in various tasks including next location prediction [3], place-of-interest recommendation [8], and trajectory similarity computation [9]. However, these methods are mostly tailored for specific problems that are taken place within areas of interest, while information of less-visited location has been hardly cared. When it comes to the scale of large cities, countries, or even the whole earth, existing approaches always suffer from extensive computational cost and significant information loss.

There are also some representation studies that directly learn location embeddings. Yin *et al.* [10] trained a neural network in each Universal Transverse Mercator (UTM) zone to learn the semantic embeddings from the initial GPS encoding. The training labels are derived from large-scale geotagged documents such as tweets, check-ins, and images that are available from social sharing platforms. Mai *et al.* [11] proposed Space2vec using an encoder-decoder framework to encode the absolute positions and spatial relationships of places based on POI information. But for large-scale location embedding studies, labels for all locations are expensive to get and the attributes are difficult to define because some locations may have multiple labels and unpopular locations have no label. Also, the labels of the locations may change over time.

Figure 1: Split the area into locations using Google S2. Then project the LBS data records into location trajectories.

Compared with labeled location data like POI types, mobility trajectories are more easily accessible. Recent studies have applied the embedding methods to mobility data with more spatio-temporal details. Models have applied ideas similar to word embeddings in NLP using social media check-in data, where the analogy of locations = words, trajectories = sentences was made [5, 6, 12, 13]. The drawback of these models is that the local context window approach and negative sample sampling ignore the overall relationship, which is more likely to cause high-exposure words to get too much weight. Shimizu *et al.* [14] generated fine grained place embeddings using human mobility trajectories, which leverages spatial hierarchical information according to the local density of observed data points. ZE-Mob created embeddings of places using the New York Taxi GPS dataset [15]. Wang and Li [16] considered both temporal dynamics and multi-hop transitions in learning the dynamic region representations and proposed to jointly learn the representations from a flow graph and a spatial graph. However, they are mostly tailored for specific problems that are taken place within a small amount of AOI and POI, but many unpopular locations are ignored.

3 Method

3.1 Problem Formulation

3.1.1 Space Discretization

In practice, it is difficult to embed infinite continuous location indicator. A more practical way is to split continuous geo indicators into discrete cells. Instead of splitting locations into grids as [10, 14], this study uses Google S2 geocoding, which is a public domain geocoding algorithm that uses a Hilbert space-filling curve to split the earth surface into hierarchical cells. Google S2 geocoding offers properties like arbitrary precision and the possibility of gradually removing characters from the end of the code to reduce its size. In this way, we encode the continuous geographical coordinates into detailed discretized cell representations, referred to as "locations" in this paper (as the cells shown in the Fig.1).

3.1.2 Location Embedding

We denote the embedding of each location l as u_l, which is a d-dimensional vector. Similar to word embedding, location embedding are vector representations of the characteristics of the location learned from human mobility trajectories. The location embedding should be able to reflect the similarity among locations with highly-related locations close with each other on the vector space.

Thus, in this study, the input dataset contains huge amount of LBS records, each record has the format as $(userID, time, location)$ and the location is represented by GPS coordinates. Human mobility trajectories are defined as sequences of locations. As shown in the Figure 1, for each user, each stay-point is firstly converted into location IDs by space discretization. To extract continuous human mobility trajectories, then the sequences are generated by partition the locations records with maximum time window between consecutive records to ensure good correlations among these location records. After that, we get the real-world trajectories of human mobility, which can serve as the context information for training location embedding.

Given the human mobility trajectories, we aim to learn a vector representation $u_l \in \mathbb{R}^d$ in the d-dimensional embedding space.

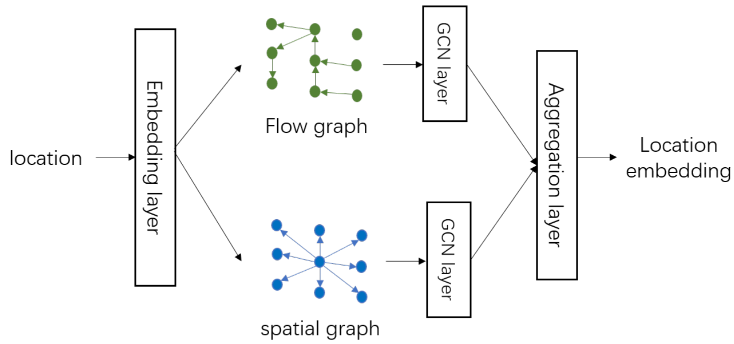


Figure 2: The framework of the GCN-L2V.

3.2 Graph Construction

There are two kinds of relations we want to capture, mobility flow relationship and spatial adjacency. The first type of relation is derived from the mobility flow among the locations, which is formulated into the flow graph G_f. The second one is the spatial adjacency defined as the spatial graph G_s. The intuitions and definitions of these two graphs will be introduced in detail in the followings.

3.2.1 Flow Graph

If two locations are frequently co-occurred in human trajectories, they are more likely to have correlations. For example, people usually travel among different subway stations. So we define the flow graph as a directed graph $G_f = (V, E_f)$. The vertices V is the set of locations and E_f is the set of edges. The edge e_{ij} on E_f represents the frequency of co-occurrence from location i to location j in consecutive trajectories, and the weights of the edge are monotonically increasing with the number of co-occurrence trips.

3.2.2 Spatial Graph

Each path on flow graph consists of a sequence of locations, and the flow graph models the mobility pattern of crowds. However, there are some issues with the flow graph. During learning the representation of locations, the flow graph suffers information loss issue. It means that if there is no mobility flow going in or out certain locations during the studied time span, the information of this region may lost. Also, the flow graph cannot recognize the same or nearby locations. Thus, the spatial graph is introduced.

The basic assumption of the spatial graph is that human mobility is bounded by space. Typically, adjacent locations are more likely to have similar characteristics (like locations in same natural parks and residential neighborhoods). When there is no location transition observed, the probability that people appeared at different locations is inversely correlated to the distance they need to travel. The spatial graph $G_s = (V, E_s)$ shares the same structure and exactly the same vertices with the flow graph. The only difference is the edge construction mechanism. The edge weight e_{ij} on E_s refers to the adjacency between location i and location j, which is defined as

$$e_{ij} = \exp\left(-\frac{\operatorname{dist}\left(v_i, v_j\right)}{\Delta}\right), \text{ if dist } \left(v_i, v_j\right) \leq \Delta, \tag{1}$$

where $\operatorname{dist}(v_i, v_j)$ is the spatial distance between the centers of the two locations, and Δ is the maximum distance threshold. The design of the spatial graph naturally incorporates the spatial adjacency.

3.3 GCN aided Embedding Layer

Graph Convolution Network (GCN) has achieved extraordinary performance on several different types of tasks with graph structure, such as node classification and network representation. GCN manipulates the spectral domain with graph Fourier transforms to apply convolutions in spectral domains [17]. Following study makes graph convolution more promising by reducing the computational complexity from $O(n^2)$ to linear [18].

In order to include two graphs mentioned above, we propose to use the GCN-L2V to learn the representation of each location as shown as Figure 2. First, there is an embedding layer, which maps every location l to a vector representation

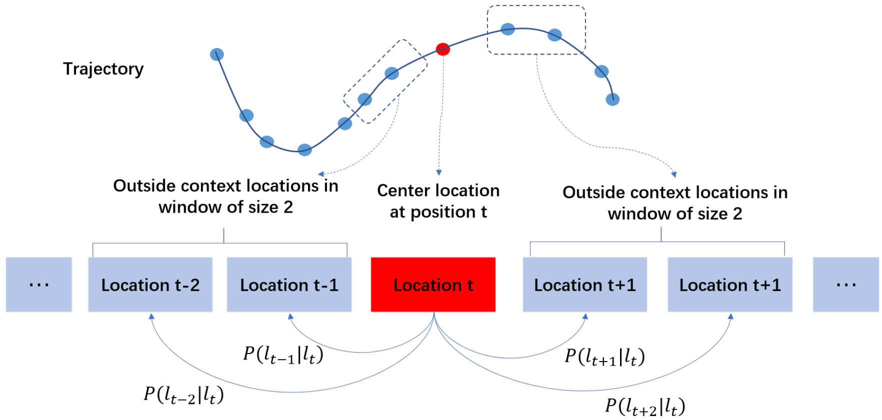


Figure 3: Skip-gram model for learning location embedding.

u_l^0. After that, parallel GCNs are applied to aggregate information from neighbor nodes on the flow graph and spatial graph. Without loss of generality, mathematically the computation of GCN on flow graph follows this formula:

$$U^f = \sigma\left(\tilde{D}_f^{-\frac{1}{2}} \tilde{A}_f \tilde{D}_f^{-\frac{1}{2}} U^0 W_f^{(l)}\right), \tag{2}$$

where σ is the non-linear activation function, and W_f is the weight matrix of the GCN layer. $\tilde{D}_f$ and $\tilde{A}_f$ represent normalized degree matrix and adjacency matrix of the flow graph.

Following, an aggregate layer is applied to derive the final embedding as

$$U = Agg(U^f, U^s), \tag{3}$$

where $Agg(\cdot)$ can be aggregation functions like mean/max pooling.

3.4 Representation Learning

We adopt the skip-gram model [19] to learn the location embedding. As shown in the Figure 3, first, randomly initialize the embedding for each location. After, for every trajectory τ as $(l_1, l_2, ..., l_m)$, we will go through each position in the trajectory and define the center location as l_c and its context locations as l_o. To identify the context locations of the t-th position, define a window of size m which means the model will look at locations from $t-m$ to $t+m$ as the context. We try to maximize the likelihood of the context words given the center word. This likelihood can be represented using the formula,

$$L(\theta) = \prod_{t=1}^{m} \prod_{\substack{-m \le j \le m \\ j \ne 0}} P\left(l_{t+j} \mid l_t; \theta\right), \tag{4}$$

where θ is the parameters of the model, and

$$P(l_o \mid l_c; \theta) = \frac{\exp\left(\boldsymbol{u}_{l_o}^{\top} \boldsymbol{u}_{l_c}\right)}{\sum_{l \in \text{Locations}} \exp\left(\boldsymbol{u}_l^{\top} \boldsymbol{u}_{l_c}\right)}. \tag{5}$$

To put this equation in a form such that it is easy to take derivatives, take the log of the equation. Also, negative sampling is used with K is the number of negative samples by selecting a few locations that are not in the context window. The likelihood function can be formulated as

$$L(\theta) = \sum_{t=1}^{m} \Big(\sum_{\substack{m \le j \le m \\ j \ne 0}} \log\left(\sigma\left(\boldsymbol{u}_{l_{t+j}}{}^{\top} \boldsymbol{u}_{l_t}\right)\right) - \sum_{k=1}^{K} \log\left(\sigma\left(\boldsymbol{u}_k^{\top} \boldsymbol{u}_{l_t}\right)\right)\Big) \tag{6}$$

Finally, we learn the embedding by maximizing the probability of real context locations and minimizing the probability of random locations appearing around the center location.

Table 1: The metrics of different methods on binary classification

Models	Accuracy	Precision	Recall	F1 score	AUC
raw GPS	0.5483	0.5350	0.5841	0.5545	0.5632
Word2Vec	0.8959	0.9215	0.8626	0.8903	0.9238
Node2Vec	0.8896	0.8915	0.8884	0.8865	0.9442
GCN-L2V	**0.9251**	**0.9269**	**0.9211**	**0.9229**	**0.9682**
GCN-L2V (flow w/o)	0.8895	0.8939	0.8819	0.8860	0.9472
GCN-L2V (spatial w/o)	0.8973	0.9036	0.9003	0.9014	0.9575

4 Experiment

In the following section, we describe the data and the strategies used to evaluate the proposed method. We perform two quantitative evaluations to compare with the embedding baselines. We then perform a case study to illustrate the effectiveness of the learned embedding.

4.1 Dataset Description

We examine the performance of the proposed model on a real-world dataset of Guangzhou, which is one of the most populous cities with an area of 7434 km^2 in China. The data are collected from 1st December to 31st December, 2019. The overall number of LBS records is 75,132,685 collected from 1,763,585 users. In the prepossessing part, we mapping the GPS coordinates into 109,455 locations, with the Google S2 level in 16 (the average area of each location is about 19,800 m^2).

4.2 Model Evaluation

We name our embedding method as GCN-L2V (GCN aided Location2Vec). To demonstrate the effectiveness of the proposed model, We compare it with the following baselines:

1. **Word2Vec** [1] is an architecture that can be used to learn embeddings from large datasets. Embeddings learned through Word2Vec have proven to be successful on a variety of downstream natural language processing tasks. It is adapted to location representation learning task by treating locations as words and trajectories as contexts.

2. **Node2vec** [2] is an algorithmic framework for representation learning on graphs. Given any graph, it can learn continuous feature representations for the nodes, which can then be used for various downstream machine learning tasks. And the embedding is learned using the flow graph and spatial graph.

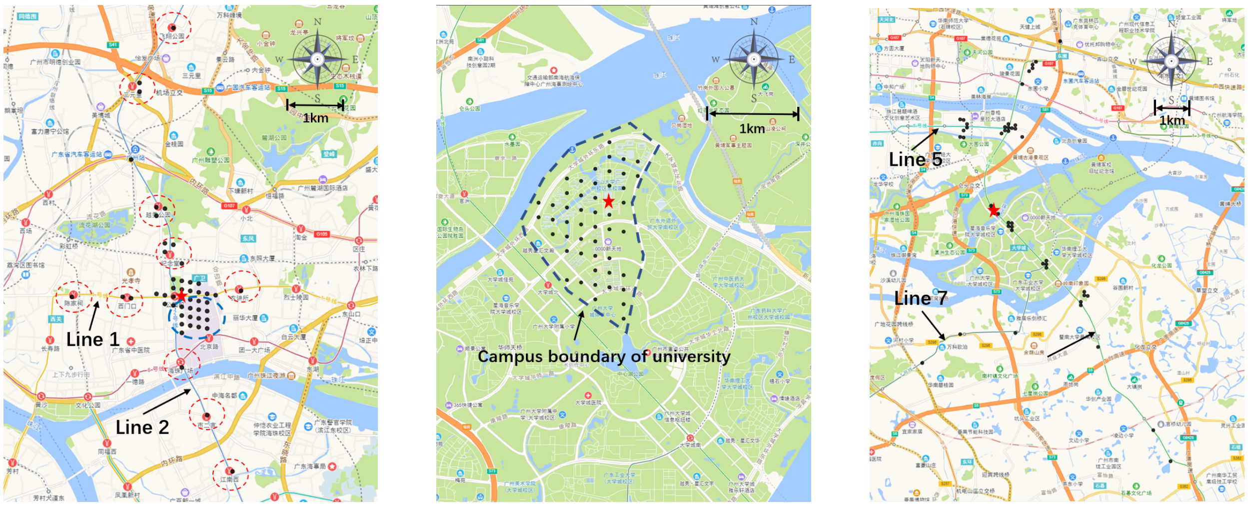
As ablation experiments, GCN-L2V (flow w/o) is the GCN-L2V without using the flow graph, and GCN-L2V (spatial w/o) is without using the spatial graph. A well-designed location embedding should: (1) preserves a certain amount of semantic information and a certain amount of spatial information of the location; (2) encodes similar (dissimilar) locations to similar (dissimilar) embedding vectors, where the similarity between two vectors can be characterized by some metrics, such as cosine similarity score. To test the effectiveness of the proposed algorithm, we design the two following downstream tasks to evaluate the ability of containing semantic and spatial information of these embeddings seperately. For all mentioned approaches, we set the embedding dimension as 8.

4.2.1 Urban Subway Station Classification

As mentioned, location embedding should contain semantic information of locations. Thus, we firstly use the learned location embeddings directly as features in a classification model to classify subway stations. In practice, we use the logistic regression as the classifier with 5-fold cross-validation. In terms of the experimental settings, we sample equal number of subway station locations and non-subway locations. Then the goal is to learn a binary classification model to determine whether the given location is a subway station. The measurement metrics are accuracy, precision, recall, F1 score, and Area Under Curve (AUC). As shown in the Table.1, the embedding learned by GCN-L2V results with best performance. Moreover, GCN-L2V (spatial w/o) also perform better than GCN-L2V (flow w/o), which means flow graph is more important in learning context information compared with spatial graph. Complementing with each other, GCN-L2V achieves satisfactory results and learns the common patterns of subway stations well.

Table 2: The metrics of different methods on region analysis

Models	Accuracy@5	Accuracy@10	Accuracy@20
Word2Vec	0.5615	0.4523	0.3635
Node2Vec	0.6077	0.5254	0.4373
GCN-L2V	**0.8754**	**0.8586**	**0.6681**
GCN-L2V (flow w/o)	0.8646	0.8015	0.6604
GCN-L2V (spatial w/o)	0.7077	0.7314	0.4893



(a) Gongyuanqian subway station (an interchange station of subway lines).

(b) University campus in the high education mege center.

(c) Guanzhou subway station (an important transportation hub).

Figure 4: Representative locations and their related locations on the embedding vector space.

4.2.2 Region Analysis

Except from semantic information, spatial information should also be kept. Here, we define regions as larger areas with specific land use types, for example, schools, hospitals, transport hubs, and residential communities. There are usually many locations in one region. Intuitively, the embeddings of locations in the same region should be similar with each other if they contain useful spatial information. In this task, we use the boundary data of regions in Guangzhou with irregular shapes. We first sample one location in each of these regions. Then, we find its most close K locations of the sampled location in terms of cosine similarity as Equation 7.

$$\text{Similarity} < \boldsymbol{u}_i, \boldsymbol{u}_j >= \frac{\boldsymbol{u}_i \cdot \boldsymbol{u}_j}{\|\boldsymbol{u}_i\| \cdot \|\boldsymbol{u}_j\|} \tag{7}$$

We hope that these locations still in the same region as the sampled location. Therefore, we use the top K accuracy (Accuracy@K) as the evaluation metrics, which calculate the percentage of the top K close locations that are still in the same region. As shown in the Table 2, the embedding learned by GCN-L2V results with best performance. From the comparisons, it can be found that adding spatial graph can better help to improve the performance than adding flow graph, because spatial graph contains overall distance information between locations. In summary, the embedding learned by GCN-L2V achieves good balance between keep the context information and spatial information.

4.3 Case Study

For better illustration of the location embedding's ability in capturing locations' relationships via human mobility. Some representative locations (red stars) and their top most related locations (black point) in terms of cosine similarity on the vector space is presented.

In Figure 4a, we select the location of the Gongyuanqian subway station as the center location. The Gongyuanqian subway station is an interchange station of Line 1 (east-west direction) and Line 2 (north-south direction) of the Guangzhou Metro. Overall, the distribution has a cross shape. It is observed that upstream and downstream subway station on these two lines have strong relationship with the center location, which are labeled in red dotted circles, indicating many people have trips between these stations and pass through Gongyuanqian station. Also, its nearby locations are also highly related. Especially the area labeled in blue dotted circle, which is a popular commercial street called Beijing Road with many people moving around. In Figure 4b, we select the location of a college campus in the High Education Mega Center. It shows that the most related locations are clustering within the campus. It is reasonable because most students and teachers in the campus spend their time in the surrounding area and there are relatively fewer trips to the outside. In Figure 4c, we select the location of Guanzhou Station, which is one of the transportation hubs in Guangzhou. The related locations are extended along transportation lines and most of the related locations are in remote places. Compared with above two cases, there are fewer related locations nearby, because people usually do not spend much time in this location as a transfer subway, and there are few POIs in this location. All these cases show that the model encodes related locations to similar embedding vectors in a meaningful way. The results are reasonable and different types of locations show different distribution patterns.

5 Conclusion

In this paper, we tackle the location embedding problem and propose a general-purpose space representation model in an unsupervised manner. For learning representations of discrete spatial regions, our method jointly embeds context information in human mobility and spatial information and mitigates the data sparsity problem especially in less populated areas. By doing so, we are able to capture relationships among locations, and provide a better notion of semantic similarity in a spatial environment. Across two quantitative experiments and case study, we empirically demonstrate that the representations learned by GCN-L2V are effective. Our proposed method can be applied in a complementary manner to other place embedding methods and down-streaming GIS related tasks. For future works, temporal information should also be considered in this study because the human mobility is highly related with time period.

References

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[2] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016.

[3] Di Yao, Chao Zhang, Jianhui Huang, and Jingping Bi. Serm: A recurrent model for next location prediction in semantic trajectories. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2411–2414, 2017.

[4] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, pages 1459–1468, 2018.

[5] Xin Liu, Yong Liu, and Xiaoli Li. Exploring the context of locations for personalized location recommendations. In *IJCAI*, pages 1188–1194, 2016.

[6] Bo Yan, Krzysztof Janowicz, Gengchen Mai, and Song Gao. From itdl to place2vec: Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '17, New York, NY, USA, 2017. Association for Computing Machinery.

[7] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. Identifying human mobility via trajectory embeddings. In *IJCAI*, volume 17, pages 1689–1695, 2017.

[8] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. Content-aware hierarchical point-of-interest embedding model for successive poi recommendation. In *IJCAI*, pages 3301–3307, 2018.

[9] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S Jensen, and Wei Wei. Deep representation learning for trajectory similarity computation. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 617–628. IEEE, 2018.

[10] Yifang Yin, Zhenguang Liu, Ying Zhang, Sheng Wang, Rajiv Ratn Shah, and Roger Zimmermann. Gps2vec: Towards generating worldwide gps embeddings. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 416–419, 2019.

[11] Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. Multi-scale representation learning for spatial feature distributions using grid cells. *arXiv preprint arXiv:2003.00824*, 2020.

[12] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. Poi2vec: Geographical latent representation for predicting future visitors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[13] Y. Zhou and Y. Huang. Deepmove: Learning place representations through large scale movement data. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2403–2412, 2018.

[14] Toru Shimizu, Takahiro Yabe, and Kota Tsubouchi. Learning fine grained place embeddings with spatial hierarchy from human mobility trajectories. *arXiv preprint arXiv:2002.02058*, 2020.

[15] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. Representing urban functions through zone embedding with human mobility patterns. In *IJCAI*, pages 3919–3925, 2018.

[16] Hongjian Wang and Zhenhui Li. Region representation learning via mobility flow. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 237–246, 2017.

[17] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26:3111–3119, 2013.