

Region2Vec: Community Detection on Spatial Networks Using Graph Embedding with Node Attributes and Spatial Interactions

Yunlei Liang

Geospatial Data Science Lab, University of Wisconsin
Madison, WI, USA
yunlei.liang@wisc.edu

Wen Ye

Geospatial Data Science Lab, University of Wisconsin
Madison, WI, USA
wey35@wisc.edu

Jiawei Zhu

School of Geosciences and Info-Physics, Central South
University
Changsha, China
jw_zhu@csu.edu.cn

Song Gao

Geospatial Data Science Lab, University of Wisconsin
Madison, WI, USA
song.gao@wisc.edu

ABSTRACT

Community Detection algorithms are used to detect densely connected components in complex networks and reveal underlying relationships among components. As a special type of networks, spatial networks are usually generated by the connections among geographic regions. Identifying the spatial network communities can help reveal the spatial interaction patterns, understand the hidden regional structures and support regional development decision-making. Given the recent development of Graph Convolutional Networks (GCN) and its powerful performance in identifying multi-scale spatial interactions, we proposed an unsupervised GCN-based community detection method *region2vec* on spatial networks. Our method first generates node embeddings for regions that share common attributes and have intense spatial interactions, and then applies clustering algorithms to detect communities based on their embedding similarity and spatial adjacency. Experimental results show that while existing methods trade off either attribute similarities or spatial interactions for one another, *region2vec* maintains a great balance between both and performs the best when one wants to maximize both attribute similarities and spatial interactions within communities.

CCS CONCEPTS

• **Computing methodologies** → *Artificial intelligence*.

KEYWORDS

spatial networks, community detection, human mobility, graph encoding, machine learning

ACM Reference Format:

Yunlei Liang, Jiawei Zhu, Wen Ye, and Song Gao. 2022. Region2Vec: Community Detection on Spatial Networks Using Graph Embedding with Node Attributes and Spatial Interactions. In *Proceedings of The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22)*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Many real world phenomena happen in the form of networks or can be represented by networks. For example, in a spatial network, nodes are usually geographic locations or regions, and edges are the spatial interactions between different places [2]. The spatial interactions can have various meanings, such as human movements or goods transportation. To extract useful knowledge from such complex networks, community detection algorithms have been widely used. A distinct characteristic of spatial networks is that the nodes may have inherent geographic relationships across different scales. Therefore, two adjacency matrices can be built for spatial networks, the first one represents the flow connections (spatial interactions), and the second one represents the geographic closeness (spatial distribution).

The convolutional network-based models provide an ideal approach to model the geographic closeness relationship. Graph Convolutional Networks (GCNs) combine both node features and edge relationships through convolution layers, and generate latent features of nodes by aggregating the neighboring relations among nodes [8]. However, there are two main issues with applying GCN to community detection. First, existing GCN models are usually supervised or semi-supervised, while community detection is essentially an unsupervised learning problem [6]. Second, GCN is not initially designed for community detection and the embedding from GCN is not community-oriented [6]. The goal of GCN learning should also be adjusted so that the characteristics of communities can be included.

To solve the abovementioned issues, we proposed a GCN-based unsupervised learning method by designing a community-oriented loss and considering both spatial interaction and geographic characteristics. Especially, we combine information from nodes, edges, neighborhoods, and multi-graphs in the GCN model to effectively learn the graph embedding. Additional clustering is then applied as a post-processing step to discover communities. We call this method *region2vec*. Although this name was first used by Xiang [16] to detect urban land use type, we would like to extend the concept of *region2vec* and use it to indicate a category of methods that generate latent feature representations based on regions' characteristics. We will demonstrate the effectiveness of

our proposed method in community detection tasks on spatial networks.

2 METHODOLOGY

2.1 Notations and Problem Definitions

Graph $G = (V, E)$ is defined via a set of nodes $V = (v_1, \dots, v_n)$, $|V| = n$ and edges E with $e_{ij} = (v_i, v_j)$. $A = [a_{ij}]_{n \times n}$ is an adjacency matrix, where $a_{ij} = 1$ if $e_{ij} \in E$, otherwise $a_{ij} = 0$. $S = [s_{ij}]_{n \times n}$ is a spatial interaction matrix, where s_{ij} represents the flow intensity between nodes v_i and v_j . An $n \times m$ attribute matrix X is used to denote the multidimensional attributes of nodes.

The community detection aims to partition the n nodes into K communities $\{C_1, C_2, \dots, C_K\}$ and each node will have a label c_i indicating its community membership, $c_i \in \{1, 2, \dots, K\}$.

2.2 Data

The major data source used for building the spatial network (graph) is the SafeGraph business venue database¹. SafeGraph collects over 8 million points of interest (POIs) with visit patterns in the U.S. To construct the spatial flow network in this study, all the place visits are aggregated to the census tract level [7]. The census tracts are then used as the nodes, and the human movement flows are the edges with flow intensity as the weights.

The spatial adjacency matrix is built based on the geographic relationship among census tract boundaries from the TIGER/Line Shapefiles². We specifically use the Rook-type contiguity relationship, which defines neighbors by the existence of sharing edges. Only census tracts with shared borders larger than zero meters will be considered as spatially adjacent. The node attributes are collected from U.S. Census American Community Survey (ACS) 2015-2019 5-year estimates. Features including poverty population, race/ethnicity, and household income are used in the model.

2.3 Algorithm

Our goal is to identify spatial network communities, where nodes (i.e., geographic regions) within the same community satisfy the following expectations: 1) share similar attributes; 2) have intense spatial interactions; and 3) are spatially contiguous. To achieve this, we proposed a two-stage community detection algorithm using both node attributes and spatial interactions. The workflow is shown in Figure 1.

2.3.1 Stage One: Node Representation Learning. One special characteristic of spatial networks is that the nodes that are spatially adjacent tend to be similar in attributes, according to the first-law of geography (spatial dependency effect) [15]. Since the critical operation of graph convolutional neural networks is to aggregate neighbor information, making it a natural tool that fits this characteristic when given spatial adjacency matrix and node attributes as inputs.

As we define $Z^{(1)}$ and $Z^{(2)}$ as the outputs of the first and second graph convolutional layers, and $W^{(0)} \in \mathbb{R}^{m \times n_{\text{hidden}}}$ and $W^{(1)} \in \mathbb{R}^{n_{\text{hidden}} \times n_{\text{output}}}$ as the weights of two layers, the forward

propagation model can be formalized as Equation 1:

$$Z^{(1)} = \text{ReLU}(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X W_0); Z^{(2)} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} Z^{(1)} W_1 \quad (1)$$

where A and I are the spatial adjacency and identity matrices, $\tilde{A} = A + I$, and $\tilde{D}$ is the degree matrix of $\tilde{A}$.

However, GCN is a semi-supervised model and not community-oriented. So we utilize the spatial interaction flow strength and geographic distance as constraints to guide the learning process. Specifically, the nodes without flow interactions are considered as negative pairs and are pushed away in the embedding space, while those with interactions will be treated as positive pairs and drawn closer; the greater the flow intensity is, the closer we bring them together in the embedding space. Moreover, we set a threshold to push away node pairs that are spatially distant from each other to guarantee the spatial contiguity. Thus, the loss function is designed as Equation 2:

$$L_{\text{hops}} = \sum \frac{\mathbb{I}(\text{hop}_{ij} > \epsilon) d_{ij}}{\log(\text{hop}_{ij})}; \text{Loss} = \frac{\sum_{p=1}^{N_{\text{pos}}} \log(s_p) d_{\text{pos}_p} / N_{\text{pos}}}{\sum_{q=1}^{N_{\text{neg}}} d_{\text{neg}_q} / N_{\text{neg}} + L_{\text{hops}}}, \quad (2)$$

where hop_{ij} represents the hop numbers of the shortest path between v_i and v_j in the graph, and d_{ij} is the euclidean distance between the corresponding embedding representations. $\mathbb{I}(\cdot)$ is set to 1 if $\text{hop}_{ij} > \epsilon$, or 0 otherwise. Positive pairs and negative pairs of nodes are denoted by $\text{pos}_p, p \in [0, N_{\text{pos}}]$ and $\text{neg}_q, q \in [0, N_{\text{neg}}]$, respectively. Since the intensity of flow s_p has a large range of values, we adopt a log transformation so that the flow values will not get overwhelmed by the extremely large values. The pseudo code of *region2vec* is shown in Algorithm 1.

Algorithm 1: Region2Vec

Input: $G; A; S; X; \text{hops}_{i,j}, \forall i, j \in V$ and threshold ϵ ;

number of layers L ; weight matrices

$W^l, \forall l \in \{0, \dots, L-1\}$

Output: Node representations z_v for all $v \in V$

$Z^{(0)} \leftarrow X$;

$\tilde{A} \leftarrow A + I$;

$\text{pos}_m \leftarrow (i, j)$, for all $s_{ij} > 0$;

$\text{neg}_n \leftarrow (i, k)$, for all $s_{ik} = 0$;

for each iteration do

for $l = 0, \dots, L-1$ **do**

$Z^{(l+1)} = \text{ReLU}(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} Z^{(l)} W^l)$;

end

$d_{ij} = \|z_i - z_j\|$;

$L_{\text{hops}} = \sum \mathbb{I}(\text{hop}_{ij} > \epsilon) d_{ij} / \log(\text{hop}_{ij})$;

$\text{Loss} = \frac{\frac{1}{N_{\text{pos}}} \sum_{p=1}^{N_{\text{pos}}} \log(s_p) d_{\text{pos}_p}}{(\frac{1}{N_{\text{neg}}} \sum_{q=1}^{N_{\text{neg}}} d_{\text{neg}_q} + L_{\text{hops}})}$;

 Compute $g \leftarrow \nabla \text{Loss}$;

 Conduct Adam update using gradient estimator g

end

$z_v \leftarrow z_v^L, \forall v \in V$

¹<https://www.safegraph.com>

²<https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html>

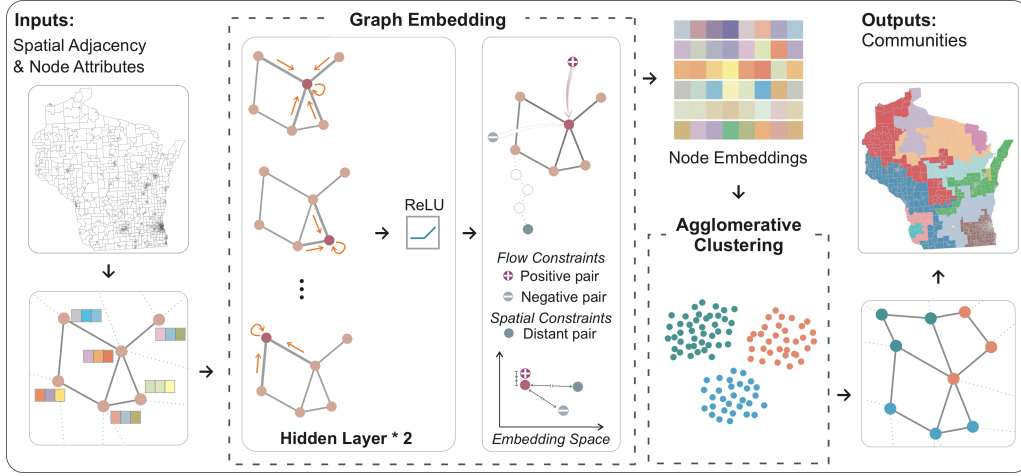


Figure 1: The workflow for community detection using the regions2vec method

2.3.2 Stage two: The Agglomerative Clustering. After obtaining the node representation, agglomerative clustering is utilized. Agglomerative clustering uses a bottom-up approach: each node is treated as a separate cluster at the beginning, and then is merged successively into groups [12]. The merge criterion is measured using the linkage type “Ward”, which minimizes the sum of squared differences within all clusters and can generate clusters with the most regular sizes compared with other types [12].

Another advantage of using agglomerative clustering is that it supports the incorporation of connectivity constraints [12]. This characteristic is especially critical in our study as the spatial contiguity is an inherent requirement of community detection in spatial networks. The spatial adjacency matrix is used to preserve the spatial contiguity and impose local structures.

2.4 Baseline Algorithms

We use a variety of baseline algorithms to compare their performances with our proposed method *region2vec*.

Louvain: The Louvain algorithm [1] is a heuristic method based on modularity optimization. It is applied to identify communities only using flow connections.

Random walk based models: Two random walk based graph embedding models, Deepwalk [13] and Node2vec [3] are used to learn continuous feature representations. The two methods are conducted using the spatial adjacency matrix as the input, followed by the same agglomerative clustering algorithm.

LINE: Large-scale Information Network Embedding (LINE) [14] is a network embedding method suitable for arbitrary types of information networks especially with large sizes. LINE uses the spatial adjacency matrix as the input and is followed by agglomerative clustering.

K-Means: The K-Means clustering algorithm aims to group nodes based on their feature similarities [9]. The K-Means clustering algorithm is directly applied on the node multidimensional attributes but it does not consider the graph structure.

2.5 Evaluation Metrics

To compare the performance of all the methods, the following metrics are used to comprehensively evaluate the communities.

Intra/Inter Flow Ratio: The spatial interaction flow ratio is specifically designed for this study, it measures the ratio of edge weights sum within each community (intra-flow weights) when $c_i = c_j$ and the edge weights sum between different communities (inter-flow weights) when $c_i \neq c_j$, which is similar to the concept of modularity [10]. As shown in Equation 3, s_{ij} represents the flow intensity between two nodes i and j .

$$\text{Intra/Inter Flow Ratio} = \frac{\sum_{c_i=c_j} s_{ij}}{\sum_{c_i \neq c_j} s_{ij}}; c_i, c_j \in 1, 2, \dots, K \quad (3)$$

Inequality: The inequality metric was proposed by Pandey et al. [11] to measure the infrastructure inequality across multiple geographic regions (Equation 4). σ is the standard deviation and μ is the mean. A value of 1 indicates maximum inequality, and 0 indicates no inequality.

$$I = \frac{\sigma}{\sqrt{\mu(1-\mu)}}; 0 < \mu < 1. \quad (4)$$

Similarity Metrics: The cosine similarity is used to calculate the L2-normalized dot product of vectors [12].

Homogeneity Scores: The homogeneity score is used to evaluate if nodes in each community are more homogeneous and have more similar socio-economic characteristics. It is calculated based on the percentage of the population with income at or lower than 200% federal poverty level.

3 RESULTS

The performance of community detection on spatial networks are compared together for all the introduced methods. The results are based on the number of community of 14, which is the optimal community number in the Louvain algorithm.

In total four metrics are used to evaluate the performance of these methods on community detection from different perspectives

Table 1: The metrics comparison of all methods. (In bold: best; Underline: second best)

Methods	Intra/Inter Flow Ratio	Inequality	Cosine Similarity	Homogeneity
DeepWalk	2.585	0.375	0.960	0.103
K-Means	0.438	0.213	0.983	0.515
LINE	0.273	0.723	0.872	0.012
Louvain	4.864	0.373	0.964	0.080
Node2vec	2.717	0.437	0.951	0.091
Region2vec	<u>3.588</u>	<u>0.367</u>	<u>0.974</u>	<u>0.105</u>

and the results are listed in Table 1. Overall, our proposed *region2vec* method maintains a great balance between attribute similarity and spatial interaction intensity and performs the best when one wants to maximize both attribute similarities and spatial interactions simultaneously within communities.

First, the intra/inter flow ratio represents the ratio of intra-community flows and inter-community flows. The Louvain method, which takes the spatial interaction flow matrix as the only input, has the highest flow ratio value. Our proposed *region2vec* method performs the second best in the spatial interaction perspective. Following them are the two random walk based algorithms: Node2vec and Deepwalk, which have similar ratios. Lastly the K-Means method and the LINE have the lowest ratios.

For the inequality, a lower value represents that the nodes within communities are more similar and have lower variations. We use the median inequality to represent each method in Table 1. The K-Means clustering method has the lowest median inequality as it clusters nodes purely based on their attribute similarity; the nodes in the same cluster have more similar features and therefore, are more equal. The proposed *region2vec* method has the second lowest inequality, meaning that it also has a good performance for grouping nodes with similar features. The remaining four methods have higher inequality as they do not consider feature information.

For the cosine similarity, as expected, K-Means clustering that uses only node attributes in the process performs the best and it has the highest cosine similarity according to Table 1. The *region2vec* method, again, is rated the second best.

Last but not least, as the homogeneity score is a metric evaluating how homogeneous the clusters are in terms of lower-income population percentage, K-Means clustering has the highest score. The *region2vec* has the second highest score, meaning that it is able to better group homogeneous nodes than the other four baselines that are purely based on edge information in graphs.

4 CONCLUSIONS

This study proposed an unsupervised community detection method called *region2vec* on spatial networks. Using a GCN-based model, *region2vec* considers the spatial adjacency, spatial interaction flows, and the node attributes. Through a community-oriented loss function, this method first generates embedding for nodes based on attribute similarity and flow interactions. Communities are further identified through agglomerative clustering with the spatial adjacency constraint. The *region2vec* method has been compared with the most commonly used community detection methods and shown a great performance when considering both node attributes and spatial interactions. Our future work will apply the proposed

method to the regionalization problems such as the rational service area development in public health and the redistricting problem in political science.

This research demonstrates the good potential of graph embedding and GCN in the community detection on spatial networks, as well as the integration of geospatial constraints in deep learning models, which can contribute to the increasing interests on GeoAI development in the SIGSPATIAL community [4, 5].

ACKNOWLEDGMENTS

We acknowledge the funding support from the County Health Rankings and Roadmaps program of the University of Wisconsin Population Health Institute, Wisconsin Department of Health Services, and the National Science Foundation funded AI institute [Grant No. 2112606] for Intelligent Cyberinfrastructure with Computational Learning in the Environment (ICICLE). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funders.

REFERENCES

- [1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.
- [2] Song Gao, Yu Liu, Yaoli Wang, and Xijun Ma. 2013. Discovering spatial interaction communities from mobile phone data. *Transactions in GIS* 17, 3 (2013), 463–481.
- [3] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.
- [4] Yingjie Hu, Song Gao, Dalton Lunga, Wenwen Li, Shawn Newsam, and Budhendra Bhaduri. 2019. GeoAI at ACM SIGSPATIAL: progress, challenges, and future directions. *Sigspatial Special* 11, 2 (2019), 5–15.
- [5] Krzysztof Janowicz, Song Gao, Grant McKenzie, Yingjie Hu, and Budhendra Bhaduri. 2020. GeoAI: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. , 625–636 pages.
- [6] Di Jin, Bingyi Li, Pengfei Jiao, Dongxiao He, and Hongyu Shan. 2019. Community Detection via Joint Graph Convolutional Network Embedding in Attribute Network. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11731 LNCS (2019), 594–606. https://doi.org/10.1007/978-3-030-30493-5_55
- [7] Yuhao Kang, Song Gao, Yunlei Liang, Mingxiao Li, Jinmeng Rao, and Jake Kruse. 2020. Multiscale dynamic human mobility flow dataset in the US during the COVID-19 epidemic. *Scientific Data* 7, 1 (2020), 1–13.
- [8] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (2017), 1–14. arXiv:1609.02907
- [9] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.
- [10] Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103, 23 (2006), 8577–8582.
- [11] Bhartendu Pandey, Christa Brelsford, and Karen C Seto. 2022. Infrastructure inequality is a characteristic of urbanization. *Proceedings of the National Academy of Sciences* 119, 15 (2022), e2119890119.
- [12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
- [13] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.
- [14] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.
- [15] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46, sup1 (1970), 234–240.
- [16] Mingjun Xiang. 2020. Region2vec: An Approach for Urban Land Use Detection by Fusing Multiple Features. In *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*. 13–18.