Modelling Spatial Patterns Using Graph Convolutional Networks

Di Zhu
Institute of Remote Sensing and Geographical Information Systems, Peking University, 5th Yiheyuan Road, Beijing, China
patrick.zhu@pku.edu.cn
https://orcid.org/0000-0002-3237-6032

Yu Liu[1]
Institute of Remote Sensing and Geographical Information Systems, Peking University, 5th Yiheyuan Road, Beijing, China
liuyu@urban.pku.edu.cn
https://orcid.org/0000-0002-0016-2902

Abstract

The understanding of geographical reality is a process of data representation and pattern discovery. Former studies mainly adopted continuous-field models to represent spatial variables and to investigate the underlying spatial continuity/heterogeneity in a regular spatial domain. In this article, we introduce a more generalized model based on graph convolutional neural networks that can capture the complex parameters of spatial patterns underlying graph-structured spatial data, which generally contain both Euclidean spatial information and non-Euclidean feature information. A trainable site-selection framework is proposed to demonstrate the feasibility of our model in geographic decision problems.

2012 ACM Subject Classification Information systems → Geographic information systems

Keywords and phrases Spatial pattern, Graph convolution, Big geo-data, Deep neural networks, Urban configuration

Digital Object Identifier 10.4230/LIPIcs.GIScience.2018.73

Category Short Paper

Funding This research was supported by the National Key Research and Development Program of China [Grant Number: 2017YFB0503602] and the National Natural Science Foundation of China [Grant Number: 41625003].

1 Introduction

The continuous-field model, which can be seen as a process of reducing the number of spatial variables required to represent reality to a finite set (a field) [6], is a fundamental perspective in modelling the complex geographical world. The variation of attributes in a field model represents the spatial pattern of certain geographical phenomenon at the conceptual level of abstraction [12, 7], as is shown in Figure 1. The analysis of spatial patterns based on field models has been studied extensively in traditional geography applications [2, 17]. Methods can be roughly divided into two types: autoregressive methods that adopt a spatial lag term

[1] Corresponding auhtor

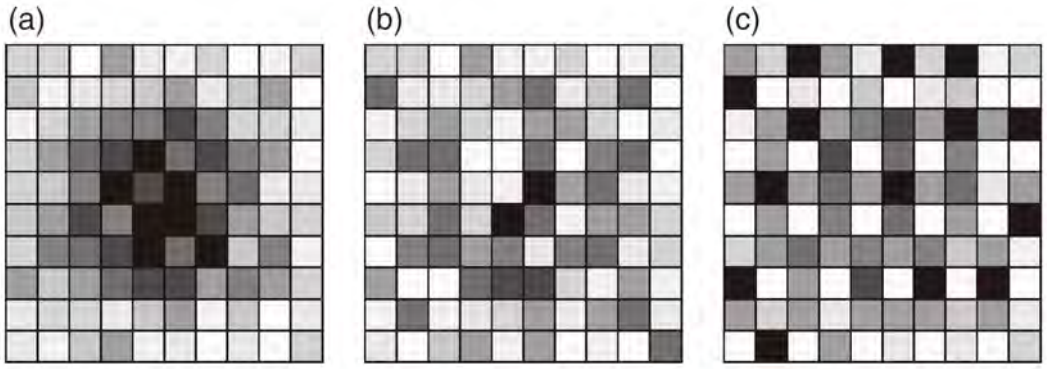

Figure 1 Spatial patterns represented in a regular grid [5]. (a) Positive spatial autocorrelation. (b) Spatial randomness. (c) Negative spatial autocorrelation.

to consider the autocorrelation of local neighborhoods [1] and geostatistical methods that use semi-variograms to characterize the spatial heterogeneity [15, 2].

To uncover the deep features of spatial patterns, convolutional neural networks (CNNs) have been introduced from computer science to investigate local stationary properties of the input data by allowing long range interactions in terms of shorter, localized interactions [11]. However, the use of CNNs becomes problematic when the data is not structured in the regular spatial domian (e.g. raster model in GIS), since the local kernel filter can no longer be defined via the Euclidean metric of the grid. Graph convolutional networks (GCNs) is a generalization of CNNs to deal with graph-structured data in the irregular spatial domain (i.e., vector model in GIS), where the input data is represented as objects and their connections. The convolutional filter in GCNs can be extended to be localized in the spectral domain of the objects' features [3, 9], thus enable us to investigate both short range interactions and long range interactions in the spatial domain. We think that GCNs are suitable for modelling the complex spatial patterns in geographical data that generally contain both Euclidean spatial information and non-Euclidean feature information [13].

In this article, we will introduce a way to model the spatial patterns in geographical data by constructing a graph neural network with both spatial information and feature information embedded and by designing a localized feature filter on graph that considers spatial constraints. A layer-wise neural network framework is proposed to make the model trainable. In addition, we have applied the proposed framework in a intra-urban site-selection cases based on a POI check-in dataset in Beijing, China to demonstrate the feasibility of our model.

2 Embedding spatial patterns in graphs

2.1 Graph Fourier transformation

To enable the formulation of fundamental operations such as filtering on a graph, the Graph Fourier transform is needed first, which is defined via a generalization of the Laplacian operator on the grid to the graph Laplacian [4]. In graph $G = (V, E, W)$, V is a finite set of $|V| = n$ nodes, E is a set of edges among nodes and $W \in \mathbb{R}^{n \times n}$ is a weighted adjacency matrix representing the weights of edges. An input vector $x \in \mathbb{R}^n$ is seen as a signal defined on G with x_i denotes the spectral information of node i.

▶ **Definition 1** (Graph Laplacian). Let $L = \Delta - W$ be the graph Laplacian of G, where $\Delta \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $\Delta_{ii} = \sum_j W_{ij}$, and the normalized definition is $L^s = I_n - \Delta^{-1/2} W \Delta^{-1/2}$ where I_n is the identity matrix.

As L^s is a real symmetric positive semidefinite matrix, it has a complete set of orthonormal eigenvectors $U = (u_1, \cdots, u_n)$, and their associated nonnegative eigenvalues $\lambda = (\lambda_1, \cdots, \lambda_n)$. The Laplacian is diagonalized by U such that $L^s = U\Lambda U^T$ where $\Lambda = diag([\lambda_1, \cdots, \lambda_n]) \in \mathbb{R}^{n\times n}$. The graph Fourier transform of $x \in \mathbb{R}^n$ is then defined as $\hat{x} = U^T x \in \mathbb{R}^n$.

2.2 Convolutions on graphs

▶ **Definition 2** (Graph convolutions). The convolution operators on graphs are defined as the muliplication of x with a filter $g_\theta = diag(\theta)$ parameterized by $\theta \in \mathbb{R}^n$ in the Fourier domain, i.e.:

$$g_\theta \star x = g_\theta(L^s)x = g_\theta(U\Lambda U^T)x = U g_\theta(\Lambda)U^T x. \tag{1}$$

We can understand $g_\theta(\Lambda)$ as a function of the eigenvalues of L^s, a non-parametric filter whose parameters are all free and can be trained.

However, the evaluation of Eq. 1 is computationally expensive, as the multiplication with eigenvector matrix U is $\mathbb{O}(n^2)$. To overcome this problem, [8] suggested the Chebyshev polynomials $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ up to K^{th} order to approximate $g_\theta(\Lambda)$:

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda}), \tag{2}$$

with a rescaled $\tilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I_n$, $\theta' \in \mathbb{R}^K$ is a vector of polynomial coefficients, $T_0(x) = 1$ and $T_1(x) = x$.

Furthermore, by assuming $K = 1$ and $\lambda_{max} = 2$ in Eq. 2 and some renormalization tricks, [10] proposed an expression with a single parameter $\theta = \theta'_0 = -\theta'_1$ to compute:

$$g_\theta \star x \approx \theta(I_n + \Delta^{-1/2} W \Delta^{-1/2})x = \theta\tilde{\Delta}^{-1/2}\tilde{W}\tilde{\Delta}^{-1/2}x, \tag{3}$$

where $\tilde{W} = W + I_n$ and $\tilde{\Delta}_{ii} = \sum_j \tilde{W}_{ij}$. Eq. 3 has complexity $\mathbb{O}(|E|)$ because $\tilde{W}x$ can be efficiently implemented as a product of a sparse matrix with a dense vector.

2.3 Spatial-enriched graph construction

Different from state-of-the-art graph constructions in many recognition tasks, where the adjacency matrix W are often defined by calculating the similarity among nodes, we try to enable the constructed graph to capture the relationships between the feature similarity and the spatial displacement of node pairs, i.e., to construct a spatial-enriched graph.

Given the input features $X \in \mathbb{R}^{N\times C}$ of nodes V, where $N = |V|$ is the number of locations and $C \in \mathbb{R}$ is the number of features for each node, we define the adjacency matrix W according to spatial displacement of N locations. The distance matrix for locations can be considered a prior knowledge for the graph construction process and we can introduce the distance decay effect in geography to represent the spatial dependence of features in X. Derived from the gravity model, there many functions that could be used to express the spatial weighting function, such as the power function, the exponential function, and the Gaussian function [19]. Here, we consider a variant of the self-tuning Gaussian diffusion kernel [9]:

$$W_{ij} = exp^{-\frac{d(i,j)}{\sigma_i \sigma_j}}, \tag{4}$$

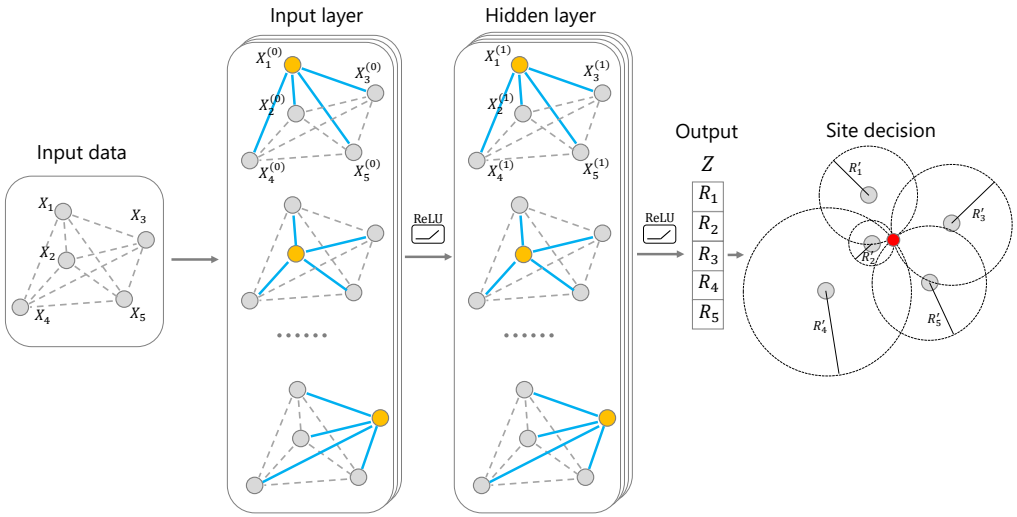

Figure 2 Illustration of the site-selection framework based on graph convolutional networks.

where $d(i,j)$ is the Euclidean distance between node i and j and σ_i is computed as the distance $d(i,i_k)$ corresponding to the k-th nearest neighbor i_k of node i. Eq. 4 gives a normalized measurement of spatial displacement in a graph whose variance is locally adapted around each location.

Compared to traditional geographical studies that choose arbitrary models to capture the effect of distance, our GCN-based model is a more universal way to model the relationship underlying spatial data. We treat the feature information and the spatial information separately, and leave the graph to learn the spatial pattern given certain training objective. The details of learned spatial pattern are restored in the layer-wise parameters of the deep graph convolutional network and can be adopted in various applications.

3 Example: site-selection tasks

One of the most common applications that implicitly consider spatial patterns is to find the best location to site a facility given the urban configurations. Traditionally, there are lots of studies that tried to solve this kind of site-selection problem through an spatial optimization model that considers some predefined spatial constraints [18]. However, if the model is simple and easy to compute, the optimization may be arbitrary to some extent; while if the model is too specific about the complex spatial relationships, the optimization are always difficult to compute.

Based on the graph convolutional model proposed in Section 2 that can learn the heterogeneity pattern underlying spatial data, we design a trainable neural network framework for the site-selection problem, illustrated in Figure 2. The site-selection framework is an example to show how our graph convolutional model can be adopted in geographic decision problems.

In Figure 2, the goal of the neural networks is to learn a complex function of spatial pattern on a graph $G = (V, E)$, which takes as input:

- A feature matrix $X \in \mathbb{R}^{N \times C}$ that contains the features x_i for every observed location i, where N is the number of given locations and C is the number of input feature types

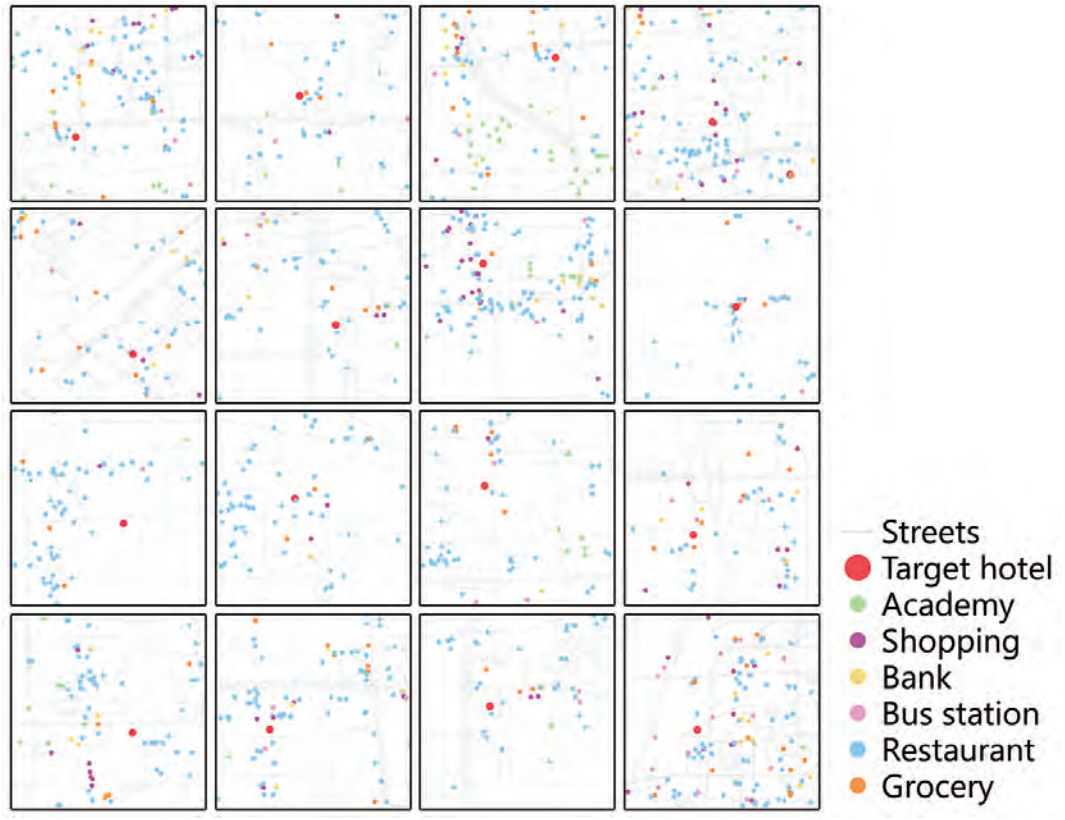

Figure 3 Illustration of some input training samples with only six POI types visualized. There are actually 242 POI types in total, and the multi-channel features contained in our dataset are not shown in this figure, such as the check-in number of each facility, the area of each facility, the number of photos took at each location.

- A fully-connected spatial distance matrix $W \in \mathbb{R}^{N \times N}$ summarized using Eq. 4 that represents the spatial structure of observed locations

and outputs a decision vector $Z = [R_1, \cdots, R_N] \in \mathbb{R}^N$ that contains the distances between the optimal site and all given locations. By calculating the virtual decision vector $Z' = [R'_1, \cdots, R'_N] \in \mathbb{R}^N$ for all potential locations in the area, we can find an optimal site that minimize $||Z - Z'||$ or we can reject a proposal of site-selection given a distance threshold.

For simplicity, we display a simple two-layer GCN to capture the spatial dependence among urban locations and make prediction. Recalling the convolutional filter introduced in Eq. 3, let $\widehat{W} = \tilde{\Delta}^{-1/2}\tilde{W}\tilde{\Delta}^{-1/2}$, the forward propagation then takes the simple form:

$$Z = ReLU\left(\widehat{W} ReLU\left(\widehat{W} X \Theta^{(0)}\right)\Theta^{(1)}\right), \tag{5}$$

where $\Theta^{(0)} \in \mathbb{R}^{C \times H}$ is the input-to-hidden parameters for a hidden layer with H feature maps. $\Theta^{(1)} \in \mathbb{R}^{H \times 1}$ is the hidden-to-output parameters for an output desicion vector Z.

Assuming all the existed facilities in urban areas are successful samples of site-selection given their circumstances, we then backpropagate the model with the mean square error loss function (MSELoss) between the output decision vector Z and the real location vector Z^*. Computational skills such as stochasitic gradient descent, batch normalization and activation functions are all adopted in our work to train the model.

We utilized a dataset collected from Sina Weibo in 2014 that contains 868 million check-in records for 143,576 points of interest (POIs) in Beijing [14]. The dataset contains multiple features to form the multi-channel enriched feature matrix X as our model input. By randomly capturing 28,000 snapshots ($3km \times 3km$) that contain at least one built-up hotel as our input training samples and 7000 snapshots with the same settings as our validation set, we can adopt the framework in Figure 3 to train a network that tries to learn the function of spatial configurations between hotels and their complex urban environment. The original input training samples are shown in Figure 3. The trained network can thus be used to evaluate the built-up environment and decide where to build a potential hotel. In practice, methods of patch extraction and normalization are applied to make the input training samples comparable and combinable [16]. We formalized the comparable training graphs into minibatches without the information of target hotels, but record the ground truth decision vectors Z^* of each input sample for the calculation of MSELoss.

Currenly, we are still optimizing the experiment for this site-selection task. After more than 200 epochs of training, the average prediction accuracy on the validation set (7,000 samples) can reach around 50 meters, but the result is not very stable due to the abnormally complex POI configurations in Beijing, China. However, we believe the simple framework proposed in this section casts light on the applications of graph convolutions in geographic decision systems.

4 Conclusion and Discussion

In this article, we introduced a generalized model that can capture the spatial pattern in geographical data using graph convolutional networks. By embedding the feature information and the spatial information separately into the graph network, and designing a feature-based localized filter on the graph, our model can learn both short and long range interactions among space and approximate the high-dimensional parameters of spatial patterns according to certain training objectives. Based upon that, we proposed a trainable site-selection framework using spatial-enriched graph convolutional neural networks to demonstrate the feasibility of our model to be adopted in various geographic problems.

Important open questions remain: How about universality of the graph convolutional networks, how could it be transferred to other applications directly? How to evaluate the model's parameters in a way that is both quantitative, interpretable and intuitive for geographical analysis? How to incorporate more understanding of spatial interactions into the graph-based model except for the distance decay? In addition, this initial work has only focused on the multi-features in a single dataset; a promising area is to integrate the features of multi-sourced geo-data such as street networks, remote sensing spectra and other social sensing datasets. An improved version of our model is needed to characterize and explain the intertwined spatial variation pattern in our complex geographic world. We plan to address these questions in on-going works.

References

1 Luc Anselin. Spatial data analysis with gis: An introduction to application in the social sciences. *Ncgia Technical Reports*, 1992.

2 Noel Cressie. The origins of kriging. *Mathematical geology*, 22:239–252, 1990.

3 Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint*, page arXiv:1606.09375, 2016.

4 R. K. Chung Fan. *Spectral graph theory*. American Mathematical Society, 1997.

5 A Stewart Fotheringham and Peter A. Rogerson. *The SAGE handbook of spatial analysis*. SAGE, 2008.

6 Michael F. Goodchild. Geographical data modeling. *Computers & Geosciences*, 18(4):401–408, 1992.

7 Michael F. Goodchild, May Yuan, and Thomas J. Cova. Towards a general theory of geographic representation in gis. *International Journal of Geographical Information Science*, 21(3):239–260, 2007.

8 David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied & Computational Harmonic Analysis*, 30(2):129–150, 2009.

9 Mikael Henaff, Joan Bruna, and Yann Lecun. Deep convolutional networks on graph-structured data. *arXiv preprint*, page arXiv:1506.05163, 2015.

10 Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint*, page arXiv:1609.02907, 2017.

11 Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

12 Yu Liu, Michael. F Goodchild, Qinghua Guo, Yuan Tian, and Lun Wu. Towards a general field model and its order in gis. *International Journal of Geographical Information Science*, 22(6):623–643, 2008.

13 Yu Liu, Xi Liu, Song Gao, Li Gong, Chaogui Kang, Ye Zhi, Guanghua Chi, and Li Shi. Social sensing: A new approach to understanding our socioeconomic environments. *Annals of the Association of American Geographers*, 105(3):512–530, 2015.

14 Ying Long and Xingjian Liu. How mixed is beijing, china? a visual exploration of mixed land use. *Environment & Planning A*, 45(12):2797–2798, 2013.

15 Georges Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.

16 Giannis Nikolentzos, Polykarpos Meladianos, Jean Pierre Tixier, Konstantinos Skianis, and Michalis Vazirgiannis. Kernel graph convolutional neural networks. *arXiv preprint*, page arXiv:1710.10689, 2017.

17 J. K. Ord and Arthur Getis. Local spatial autocorrelation statistics: Distributional issues and an application. *Geographical Analysis*, 27(4):286–306, 1995.

18 Daoqin Tong and Alan T. Murray. Spatial optimization in geography. *Annals of the Association of American Geographers*, 102(6):1290–1309, 2012.

19 Di Zhu, Zhou Huang, Li Shi, Lun Wu, and Yu Liu. Inferring spatial interaction patterns from sequential snapshots of spatial distributions. *International Journal of Geographical Information Science*, 32(4):783–805, 2018.