Angle-based Convolution Networks for Extracting Local Spatial Features

Koh Takeuchi
NTT
Kyoto University
koh.t@acm.org

Hisashi Kashima
Kyoto University
RIKEN Center for AIP
kashima@i.kyoto-u.ac.jp

Naonori Ueda
NTT
RIKEN Center for AIP
naonori.ueda@lab.ntt.co.jp

Abstract

Convolutional Neural Networks (CNNs) have been a powerful feature extracting method for various machine learning problems, but it is limited to a regular grid shape spatial locations, e.g., pixels of images or GPS grids. However, due to physical or privacy preserving problems, spatio-temporal data in the real world often consist of irregular spatial locations. To overcome this limitation, we propose Angle-based Convolutional Networks (ACNs) that leverages the local feature extraction function of CNNs and the graph formulation of Graph Convolutional Neural Networks (GCNs). Our method considers angles among spatial locations and introduces coefficient weight for angle partitions that enables us to extract local features appears in spatio-temporal data without regular grids.

1 Introduction

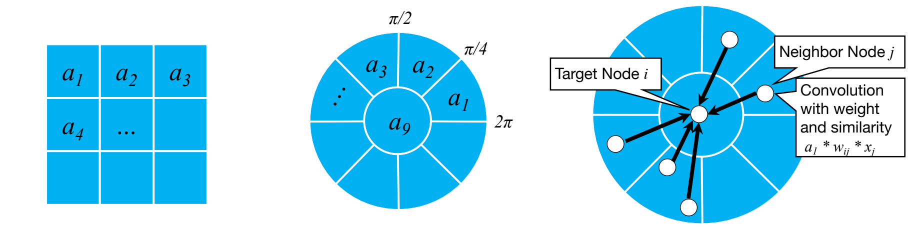
Spatio-temporal data analysis has attracted considerable interest from a wide range of applications in which researchers tries to understand human activities in cities [8, 21], climate conditions [2, 23, 13], public health [16, 14] and animal behavior in natural environments [10]. In these analyses, we assume that spatio-temporal data is a multivariate time series that provides where and when each data was obtained. Although spatio-temporal data inherit plenty of information about space and temporal domains, its spatial location often has an irregular grid shape. For example, when we attempt to deploy a sensor network system in a field, it is almost unfeasible to put sensors regularly due to physical and governmental problems. The granularity of spatial locations is often limited to a certain range, such as an administrative area, for protecting the anonymity of individuals [1]. This property makes spatio-temporal data analysis more complicated than other applications assuming a regular grid shape such as pixels in images [4].

Convolutional Neural Networks (CNNs) [7] is a workhorse in machine learning and has intensively contributed to current improvements in spatio-temporal data analysis with regular spatial locations, such as GPS grids [17, 20, 13]. Traditionally, they have utilized 2-dimensional convolution layers of CNNs to extract local spatial features. Intuitively, it consists of weight corresponding to specific locations of input data and learns local features by taking an inner product between its weight and data patches. However, CNNs are limited to a regular grid location [19] and cannot apply to irregular spatial locations that almost always appears in spatio-temporal data analysis.

$$\boldsymbol{H}_{i,j} = \sum_{i'=1}^{K}\sum_{j'=1}^{K} \boldsymbol{A}_{i',j'}\boldsymbol{X}_{i+i',j+j'} + c, \quad f(\boldsymbol{H}_{i,j}) = \sigma(\boldsymbol{H}_{i,j}) \tag{1}$$

Graph Convolutional Networks (GCNs) [6] is a recently proposed method to consider a graph structure among data features. GCNs can extract features from input data by taking an inner product

Workshop on Modeling and Decision-Making in the Spatiotemporal Domain, 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.



(a) 2-dimensional Convolution whose kernel size is three (k=3)

(b) Angle-based Convolution whose section size is eight (N=8)

(c) Example of Angle-based Convolution whose section size is eight (N=8)

Figure 1: (a) Convolutional layer. (b) Angle-based Convolutional layer. (c) An example of angle-based convolutional layers with five nearest neighbors. This operation corresponds to take an inner product of i-th row of $\boldsymbol{W}$ and t-th column vector of $\boldsymbol{X}$.

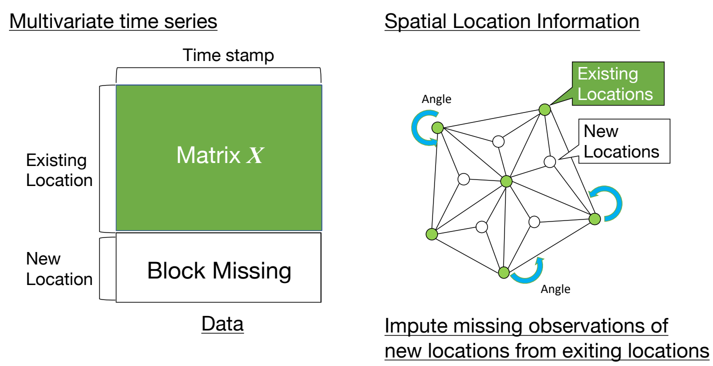


Figure 2: Spatio-temporal data analysis for block missing observation imputations.

among an adjacent graph matrix and a linear transformation matrix. Although GCNs have been successfully applied to various kinds of spatio-temporal data analysis [9, 18], it has no function to extract local features from the graph structure as CNNs does on grids. Because it does not consider an angle between a spatial location and another location.

In this paper, we propose an Angle-based Convolutional Networks (ACNs) that leverages both the virtue of CNNs and GCNs to extract local features from spatio-temporal data with irregular spacial locations. By extending the convolutional layer of CNNs to consider both angles and distances among spatial locations, we introduce an angle-based convolutional layer that divides 2π radian into discrete partitions and assigns coefficient weight w for specific partitions. Then, from GPS information of spatial locations, we obtain nearest neighbors of a spatial location and extract local features by taking an inner product between input data, weights, and similarities among locations. We illustrate our convolutional layer in Figures 1. We conduct experiments for imputing block missing observations of real-world spatio-temporal data illustrated in 2 and show that denoising autoencoder [15] with our method improves the performances from that with GCNs and other baselines.

2 Angle-based Convolution Networks (ACNs)

In this section, we introduce an angle-based convolutional networks for analyzing spatio-temporal data with irregular spatial locations. Let us consider a case where we have spatio-temporal data observed from P locations with T timestamps. We represent this multivariate time series as a matrix $\boldsymbol{X} \in \mathbb{R}^{P \times T}$.

Here, we introduce notations for representing spatial information of sensor locations. We denote a similarity matrix $\boldsymbol{W} \in \mathbb{R}^{P \times P}$ between spatial locations whose elements $w_{i,j} \geq 0$ set to a positive value if a pair of locations (i, j) has a relationship, or set to 0 otherwise. We split 2π radian into N partitions and put coefficient weight $\boldsymbol{a} \in \mathbb{R}^N$ whose n-th element a_n is a weight for the n-th angle partition of $[2\pi(n-1)/N, 2\pi n/N]$, see Figure 1 (c). We additionally put a weight for a self-loop as

a_{N+1}. A bearing matrix $\boldsymbol{B} \in \mathbb{R}^{P\times P}$ indicates angles from the i-th location to the j-th location that means $b_{i,j} = \mathrm{bearing}(\mathrm{latitude}_i, \mathrm{longtitude}_i, \mathrm{latitude}_j, \mathrm{longtitude}_j)$ [3]. We use $\boldsymbol{B}(\boldsymbol{a})$ to denote a matrix whose (i, j)-th element was set to $\boldsymbol{B}(\boldsymbol{a})_{i,j} = a_n$ if $b_{i,j} \in [2\pi(n-1)/N, 2\pi n/N]$. Its diagonal elements were set to a_{N+1}. We define our angle-based convolutional layer as:

$$f(\boldsymbol{X}_{i,t}) = \sum_{j\in E_i} \boldsymbol{B}(\boldsymbol{a})_{i,j}\boldsymbol{W}_{i,j}\boldsymbol{X}_{j,t} + \boldsymbol{B}(\boldsymbol{a})_{i,i}\boldsymbol{W}_{i,i}\boldsymbol{X}_{i,t} \tag{2}$$

where E_j is a set of neighbors of the i-th location. We illustrate an example of an our convolutional layer in Figure 1 (c). With a matrix representation, our convolutional layer can be represented as :

$$\boldsymbol{H} = (\boldsymbol{W} \circ \boldsymbol{B}(\boldsymbol{a}))\boldsymbol{X}, \quad f(\boldsymbol{X}; \boldsymbol{a}) = \sigma(\boldsymbol{H})$$

where $\circ$ is the Hadamard product of tensors and σ is a rectification function, such as the sigmoid function or the ReLU functions. For simplicity, the above definition only considers spatio-temporal data with a single mode. It is straightforward to extend to any number of modalities.

3 Related works

With the above formulation, a convolutional layer of GCNs [6] can be defined as:

$$\boldsymbol{H} = (\tilde{D}^{-\frac{1}{2}}\tilde{\boldsymbol{W}}\tilde{D}^{-\frac{1}{2}})\boldsymbol{X}\boldsymbol{V}, \quad f(\boldsymbol{X}) = \sigma(\boldsymbol{H}) \tag{3}$$

where $\boldsymbol{V} \in \mathbb{R}^{T\times T'}$ is a linear transformation matrix for the temporal domain, $\tilde{\boldsymbol{W}} = \boldsymbol{W} + \boldsymbol{I}_P$ is a modified adjacent matrix with diagonal elements, $\boldsymbol{I}_P$ is the identity matrix, and $D_{i,j} = \sum_{i,j} W_{i,j}$. Different from Eq. (3), it does not put any learnable weights for the adjacent graph matrix. Therefore, this network is not able to extract local spatial features but just obtain a weighted mean of neighbors. Li et al. [9] proposed the diffusion convolution layer to consider spatial dependencies by using bidirectional random walks. However, they have not consider an angle based coefficients and thus their goal is different from ours. In CV, a convolution-like operation on graph signals was proposed for point cloud classifications [12]. Their application did not include spatio-temporal data analysis. Thus the focus is different from ours.

4 Experiments

We conduct experiments for imputing block missing observations of spatio-temporal data as illustrated in Figure 2. We employ trip record data sets provided by a bike sharing system [1] and NYC Taxi Limousine commission [2] in New York City. From the bike sharing data set, we selected $P = 91$ bike stations in Manhattan and counted the number of bikes returned at bike stations every hour from Jan. 2017 to Dec. 2017 ($T = 8760$) that means $\boldsymbol{X} \in \mathbb{R}^{91\times 8569}$. We measured the road distance among all bike stations and constructed an adjacent matrix $\boldsymbol{W}$ by k-nearest-neighbors with $k = 5$ where weights were set to $w_{i,j} = 1$ for simplicity. From the taxi data set, we obtained the number of yellow and green cabs dropped their passenger at a taxi zone every hour from Jan. 2017 to Dec. 2017 and selected $P = 66$ zones in Manhattan. Thus, we constructed a matrix $\boldsymbol{X} \in \mathbb{R}^{66\times 8760}$. We checked a road map of NYC and set $w_{i,j} = 1$ if the i-th and j-th zones are connected by a road or $w_{i,j} = 0$ otherwise. We randomly picked $p\%$ ($p = 0.3, 0.5$) of bike stations or taxi zones and used them as new locations meaning missing blocks. We treated the rest as existing locations.

Although there are various kinds of NNs for missing value imputation problems [11], we simply employed the Denoising Auto Encoder (DAE) [15] in this paper. We made a corrupted input data $\tilde{\boldsymbol{X}}$ for DAE by setting the values of missing blocks of $\boldsymbol{X}$ to 0. In training phase, we randomly set $p\%$ elements in exiting locations to 0. We constructed DAE models with our layer, GCN layer and Fully Connected layer (FC). As a proposed method, we employ a simple network architecture that consists of angle convolutional layers with 16 filters, 1×1 convolutional layers, and a skip connection illustrated in Figure 3. Since the weight of our proposed layer depends on only angle partitions, we trained our method with observed data from existing locations. Then used the learned weight for imputing missing blocks where. For DAE with other layers, we trained each model

[1]http://www.citibikenyc.com/

[2]http://www.nyc.gov

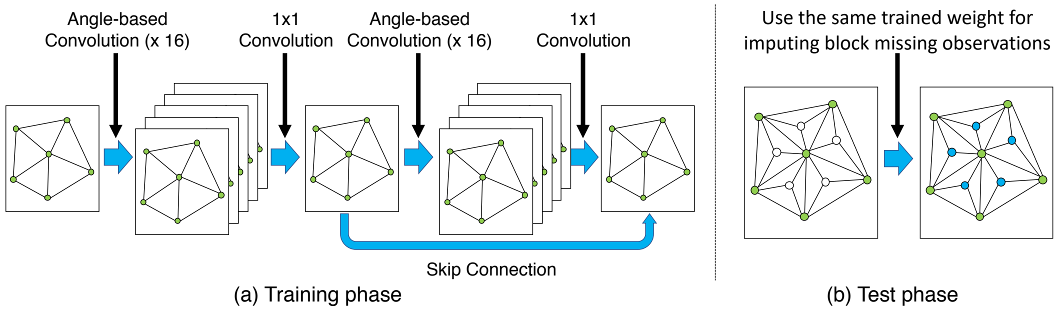


Figure 3: Network architecture for our proposed method.

Table 1: Result of the bike sharing and the taxi data set. We showed averages and standard deviations of RMSE from five runs with randomly generated missing locations. Bold face corresponds the minimum averages.

	Bike sharing data set		Taxi data set	
Method	$p = 0.3$	$p = 0.5$	$p = 0.3$	$p = 0.5$
Proposed	$\mathbf{0.764}$ $(\mathbf{0.011})$	$\mathbf{0.781}$ $(\mathbf{0.008})$	$\mathbf{1.607}$ $(\mathbf{0.178})$	$\mathbf{1.588}$ $(\mathbf{0.132})$
GCNs	0.786 (0.014)	0.833 (0.009)	1.634 (0.151)	1.748 (0.189)
FC	1.027 (0.024)	1.036 (0.019)	2.269 (0.161)	2.149 (0.146)
Means kNNs	1.089 (0.019)	1.091 (0.022)	2.150 (0.173)	1.937 (0.094)
Means	1.096 (0.019)	1.104 (0.019)	2.228 (0.179)	1.937 (0.094)

to minimize the mean squared error between observations of existing locations and its estimation $\min_\Theta \sum_{p \in E_P} \sum_{t=1}^{T} (\boldsymbol{X}_{p,t} - g(\tilde{\boldsymbol{X}}_{p,t}))^2)$, where E_P is a set of exiting spatial locations, Θ is a parameter, and g is a function of DAE.

We used the ReLU function as the rectification function in experiments. We optimized all models with Adam [5] where the learning rate is set to 0.01 and the size of mini batch was set to $168 = 24 * 7$. The number of epoch was set to 100. For our method, the number of partitions was set to $N = 4$. For DAE with GCNs (GCNs), we used a network architecture consists of two GCN layers where $T' = 4$ and $T' = 168$. For FC, we used two fully connected layer that project P from $(P \to P/2)$ and $(P/2 \to P/4)$. We also employed means of all observed values (Means) and means of observed values of kNNs (Means kNNs) as baselines. We run experiments five times and measured Root Mean Squared Error (RMSE) for new locations. We show the results in Table 1.

From the result, we confirmed that our proposed method showed the best performances on $p = 0.3$ and $p = 0.5$ of both data sets. GCNs placed second in all settings. This result indicates the benefits of utilizing a graph structure for spatio-temporal data analysis as reported in existing papers and suggests a benefit of our proposed layer. It worked better than just getting a weighted means of neighbor locations by considering angles among spatial locations.

5 Discussion

We introduced angle-based convolutional networks that enable us to extract local features from spatio-temporal data with irregular spatial locations. We defined an angle-based convolutional layer as an extension of CNNs and GCNs that leverages angles and similarities of spatial locations. We demonstrated preliminary experiments for block missing value imputation problems with two spatio-temporal data sets. We fixed various hyperparameters in experiments. Further evaluations with carefully tuned hyperparameters should be done to confirm the difference between ours and exiting methods firmly. Other graph-based missing value imputation, such as Kernelized Probabilistic Matrix Factorization methods [22] should be employed as additional baselines.
Acknowledgements: This work was supported by JST CREST Grant Number JPMJPR18UG, Japan.

References

[1] Ellen K Cromley and Sara L McLafferty. *GIS and public health*. 2011.

[2] James B Elsner, Robert E Hodges, and Thomas H Jagger. Spatial grids for hurricane climate research. *Climate dynamics*, 39(1-2):21–36, 2012.

[3] Bernard J Hamrock and William J Anderson. Rolling-element bearings. 1983.

[4] Minh X Hoang, Yu Zheng, and Ambuj K Singh. FCCF: forecasting citywide crowd flows based on big data. In *Proceedings of SIGSPATIAL*, 2016.

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[6] Thomas N Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*, 2016.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of NIPS*, 2012.

[8] John Krumm and Eric Horvitz. Risk-Aware Planning: Methods and Case Study for Safer Driving Routes. In *Proceedings of AAAI*, 2017.

[9] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *Proceedings of ICLR*, 2018.

[10] Toby A Patterson, Len Thomas, Chris Wilcox, Otso Ovaskainen, and Jason Matthiopoulos. State–space models of individual animal movement. *Trends in ecology & evolution*, 23(2):87–94, 2008.

[11] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In *Proceedings of ICML*, 2011.

[12] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of CVPR*, 2017.

[13] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution. In *Proceedings of SIGKDD*.

[14] Jay M Ver Hoef, Erin Peterson, and David Theobald. Spatial statistical models that use flow and stream distance. *Environmental and Ecological statistics*, 13(4):449–464, 2006.

[15] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of ICML*, 2008.

[16] Lance A Waller and Carol A Gotway. *Applied spatial statistics for public health data*, volume 368. John Wiley & Sons, 2004.

[17] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. Deep Distributed Fusion Network for Air Quality Prediction. In *Proceedings of SIGKDD*, 2018.

[18] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of IJCAI*, 2018.

[19] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *Proceedings of AAAI*, 2017.

[20] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. DNN-Based Prediction Model for Spatial-Temporal Data. In *Proceedings of SIGSPATIAL*, 2016.

[21] Yu Zheng. Trajectory data mining: an overview. *ACM TIST*, 6(3):29, 2015.

[22] Tinghui Zhou, Hanhuai Shan, Arindam Banerjee, and Guillermo Sapiro. Kernelized Probabilistic Matrix Factorization: Exploiting Graphs and Side Information. In *Proceedings of SDM*, 2012.

[23] Xun Zhou, Shashi Shekhar, and Reem Y Ali. Spatiotemporal change footprint pattern discovery: an inter-disciplinary survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(1):1–23, 2014.