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 try 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 [11]. 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.

\[ H_{i,j} = \sum_{i' = 1}^{K} \sum_{j' = 1}^{K'} A_{i',j'} X_{i+i',j+j'} + c, \quad f(H_{i,j}) = \sigma(H_{i,j}) \]  \hspace{1cm} (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
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 $W$ and $t$-th column vector of $X$.

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 spatial 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 $2\pi$ 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 $X \in \mathbb{R}^{P \times T}$.

Here, we introduce notations for representing spatial information of sensor locations. We denote a similarity matrix $W \in \mathbb{R}^{P \times P}$ 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 $2\pi$ radian into $N$ partitions and put coefficient weight $a \in \mathbb{R}^N$ 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
We conduct experiments for imputing block missing observations of spatio-temporal data as illustrated in Figure 3. A bearing matrix $B \in \mathbb{R}^{P \times P}$ indicates angles from the $i$-th location to the $j$-th location that means $b_{i,j} = \text{bearing(latitude}_i, \text{longitude}_i, \text{latitude}_j, \text{longitude}_j)$ \cite{Li}. We use $B(a)$ to denote a matrix whose $(i,j)$-th element was set to $B(a)_{i,j} = a_n$ 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:

$$f(X_{i,t}) = \sum_{j \in E_i} B(a)_{i,j} W_{i,j} X_{j,t} + B(a)_{i,i} W_{i,i} X_{i,t}$$

(2)

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

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

where $\circ$ is the Hadamard product of tensors and $\sigma$ 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 \cite{Duvenaud} can be defined as:

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

(3)

where $V \in \mathbb{R}^{T \times T'}$ is a linear transformation matrix for the temporal domain, $\tilde{W} = W + I_p$ is a modified adjacent matrix with diagonal elements, $I_p$ 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. \cite{Li} 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 \cite{Behnam}. 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\textsuperscript{1} and NYC Taxi Limousine commission\textsuperscript{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 $X \in \mathbb{R}^{91 \times 8760}$. We measured the road distance among all bike stations and constructed an adjacent matrix $W$ 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 $X \in \mathbb{R}^{66 \times 8760}$. 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 \cite{Baheti}, we simply employed the Denoising Auto Encoder (DAE) \cite{Vincent} in this paper. We made a corrupted input data $X$ for DAE by setting the values of missing blocks of $X$ to 0. In training phase, we randomly set $p\%$ elements in existing 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, $1 \times 1$ 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

\textsuperscript{1}http://www.citibikenyc.com/
\textsuperscript{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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Bike sharing data set</th>
<th>Taxi data set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p = 0.3$</td>
<td>$p = 0.3$</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.764 (0.011)</td>
<td>1.607 (0.178)</td>
</tr>
<tr>
<td>GCNs</td>
<td>0.786 (0.014)</td>
<td>1.588 (0.132)</td>
</tr>
<tr>
<td>FC</td>
<td>1.027 (0.024)</td>
<td>1.634 (0.151)</td>
</tr>
<tr>
<td>Means kNNs</td>
<td>1.089 (0.019)</td>
<td>2.269 (0.161)</td>
</tr>
<tr>
<td>Means</td>
<td>1.096 (0.019)</td>
<td>2.150 (0.173)</td>
</tr>
</tbody>
</table>

To minimize the mean squared error between observations of existing locations and its estimation, 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 ($P \rightarrow P/2$) and ($P/2 \rightarrow P/4$). 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


