Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting

Shengnan Guo,[1,2] **Youfang Lin,**[1,2,3] **Ning Feng,**[1,3] **Chao Song,**[1,2] **Huaiyu Wan**[1,2,3*]
[1]School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
[2]Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing, China
[3]CAAC Key Laboratory of Intelligent Passenger Service of Civil Aviation, Beijing, China
{guoshn, yflin, fengning, chaosong, hywan}@bjtu.edu.cn

Abstract

Forecasting the traffic flows is a critical issue for researchers and practitioners in the field of transportation. However, it is very challenging since the traffic flows usually show high nonlinearities and complex patterns. Most existing traffic flow prediction methods, lacking abilities of modeling the dynamic spatial-temporal correlations of traffic data, thus cannot yield satisfactory prediction results. In this paper, we propose a novel attention based spatial-temporal graph convolutional network (ASTGCN) model to solve traffic flow forecasting problem. ASTGCN mainly consists of three independent components to respectively model three temporal properties of traffic flows, i.e., recent, daily-periodic and weekly-periodic dependencies. More specifically, each component contains two major parts: 1) the spatial-temporal attention mechanism to effectively capture the dynamic spatial-temporal correlations in traffic data; 2) the spatial-temporal convolution which simultaneously employs graph convolutions to capture the spatial patterns and common standard convolutions to describe the temporal features. The output of the three components are weighted fused to generate the final prediction results. Experiments on two real-world datasets from the Caltrans Performance Measurement System (PeMS) demonstrate that the proposed ASTGCN model outperforms the state-of-the-art baselines.

Introduction

Recently, many countries are committed to vigorously develop the Intelligent Transportation System (ITS) (Zhang et al. 2011) to help for efficient traffic management. Traffic forecasting is an indispensable part of ITS, especially on the highway which has large traffic flows and fast driving speed. Since the highway is relatively closed, once a congestion occurs, it will seriously affect the traffic capacity. Traffic flow is a fundamental measurement reflecting the state of the highway. If it can be predicted accurately in advance, according to this, traffic management authorities will be able to guide vehicles more reasonably to enhance the running efficiency of the highway network.

Highway traffic flow forecasting is a typical problem of spatial-temporal data forecasting. Traffic data are recorded at fixed points in time and at fixed locations distributed in continuous space. Apparently, the observations made at neighboring locations and time stamps are not independent but dynamically correlated with each other. Therefore, the key to solve such problems is to effectively extracting the spatial-temporal correlations of data. Fig. 1 demonstrates the spatial-temporal correlations of traffic flows (also can be vehicle speed, lane occupancy, etc.). The bold line between two points represents their mutual influence strength. The darker the color of line is, the greater the influence is. In the spatial dimension (Fig. 1(a)), we can find that different locations have different impacts on A and even a same location has varying influence on A as time goes by. In the temporal dimension (Fig. 1(b)), the historical observations of different locations have varying impacts on A's traffic states at different times in the future. In conclusion, the correlations in traffic data on the highway network show strong dynamics in both the spatial dimension and temporal dimension. How to explore nonlinear and complex spatial-temporal data to discover its inherent spatial-temporal patterns and to make accurate traffic flow predictions is a very challenging issue.

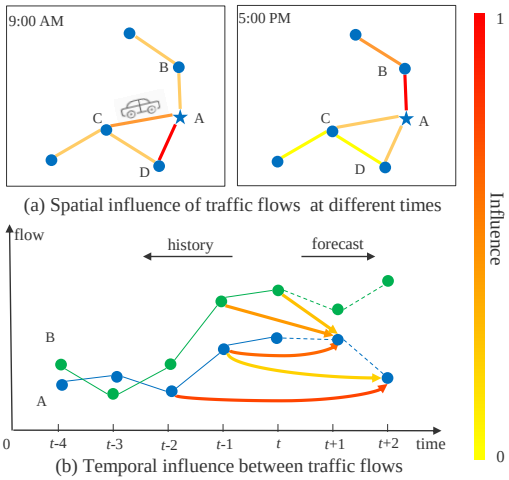


Figure 1: The spatial-temporal correlation diagram of traffic flow

Fortunately, with the development of the transportation industry, many cameras, sensors and other information collection devices have been deployed on the highway. Each

*Corresponding author: hywan@bjtu.edu.cn

device is placed at a unique geospatial location, constantly generating time series data about traffic. These devices have accumulated a large amount of rich traffic time series data with geographic information, providing a solid data foundation for traffic forecasting. Many researchers have already made great efforts to solve such problems. Early, time series analysis models are employed for traffic prediction problems. Yet, it is difficult for them to handle the unstable and nonlinear data in practice. Later, traditional machine learning methods are developed to model more complex data, but it is still difficult for them to simultaneously consider the spatial-temporal correlations of high-dimensional traffic data. Moreover, the prediction performances of this kind of methods rely heavily on feature engineering, which often requires lots of experiences from experts in the corresponding domain. In recent years, many researchers use deep learning methods to deal with high-dimensional spatial-temporal data, i.e., convolutional neural networks (CNN) are adopted to effectively extract the spatial features of grid-based data; graph convolutional neural networks (GCN) are used for describing spatial correlation of graph-based data. However, these methods still fail to simultaneously model the spatial-temporal features and dynamic correlations of traffic data.

In order to tackle the above challenges, we propose a novel deep learning model: ***A**ttention based **S**patial-**T**emporal **G**raph **C**onvolution **N**etwork* (ASTGCN) to collectively predict traffic flow at every location on the traffic network. This model can process the traffic data directly on the original graph-based traffic network and effectively capture the dynamic spatial-temporal features. The main contributions of this paper are summarized as follows:

- We develop a spatial-temporal attention mechanism to learn the dynamic spatial-temporal correlations of traffic data. Specifically, a spatial attention is applied to model the complex spatial correlations between different locations. A temporal attention is applied to capture the dynamic temporal correlations between different times.
- A novel spatial-temporal convolution module is designed for modeling spatial-temporal dependencies of traffic data. It consists of graph convolutions for capturing spatial features from the original graph-based traffic network structure and convolutions in the temporal dimension for describing dependencies from nearby time slices.
- Extensive experiments are carried out on real-world highway traffic datasets, which verify that our model achieves the best prediction performances compared to the existing baselines.

Related work

Traffic forecasting After years of continuous researches and practices, many achievements have been made in the studies about traffic forecasting. The statistical models used for traffic prediction include HA, ARIMA (Williams and Hoel 2003), VAR (Zivot and Wang 2006), etc. These approaches require data to satisfy some assumptions, but traffic data is too complex to satisfy these assumptions, so they usually perform poorly in practice. Machine learning methods such as KNN (Van Lint and Van Hinsbergen 2012) and SVM (Jeong et al. 2013) can model more complex data, but they need careful feature engineering. Since deep learning has brought about breakthroughs in many domains, such as speech recognition and image processing, more and more researchers apply deep learning to spatial-temporal data prediction. Zhang et al. (2018) designed a ST-ResNet model based on the residual convolution unit to predict crowd flows. Yao et al. (2018b) proposed a method to predict traffic by integrating CNN and long-short term memory (LSTM) to jointly model both spatial and temporal dependencies. Yao et al. (2018a) further proposed a Spatial-Temporal Dynamic Network for taxi demand prediction which can learn the similarity between locations dynamically. Although the spatial-temporal features of the traffic data can be extracted by these model, their limitation is that the input must be standard 2D or 3D grid data.

Convolutions on graphs The traditional convolution can effectively extract the local patterns of data, but it can only be applied for the standard grid data. Recently, the graph convolution generalizes the traditional convolution to data of graph structures. Two mainstreams of graph convolution methods are the spatial methods and the spectral methods. The spatial methods directly perform convolution filters on a graph's nodes and their neighbors. So, the core of this kind of methods is to select the neighborhood of nodes. Niepert, Ahmed, and Kutzkov (2016) proposed a heuristic linear method to select the neighborhood of every center node, which achieved good results in social network tasks. Li et al. (2018) introduced graph convolutions into human action recognition tasks. Several partitioning strategies were proposed here to divide the neighborhood of each node into different subsets and to ensure the numbers of each node's subsets are equal. The spectral methods, in which the locality of the graph convolution is considered by spectral analysis. A general graph convolution framework based on the Graph Laplacian is proposed by Bruna et al. (2014), then Defferrard, Bresson, and Vandergheynst (2016) optimized the method by using Chebyshev polynomial approximation to realize eigenvalue decomposition. Yu, Yin, and Zhu (2018) proposed a gated graph convolution network for traffic prediction based on this method, but the model does not consider the dynamic spatial-temporal correlations of traffic data.

Attention mechanism Recently, attention mechanisms have been widely used in various tasks such as natural language processing, image caption and speech recognition. The goal of the attention mechanism is to select information that is relatively critical to the current task from all input. Xu et al. (2015) proposed two attention mechanisms in the image description task and adopted a visualization method to intuitively show the effect of the attention mechanism. For classifying nodes of a graph, Velickovic et al. (2018) leveraged self-attentional layers to process graph-structured data by neural networks and achieved state-of-the-art results. To forecast the time series, Liang et al. (2018) proposed a multi-level attention network to adaptively adjust the correlations among multiple geographic sensor time series. However, it is time-consuming in practice since a separate model needs to be trained for each time series.

Motivated by the studies mentioned above, considering the graph structure of the traffic network and the dynamic spatio-temporal patterns of the traffic data, we simultaneously employ graph convolutions and the attention mechanisms to model the network-structure traffic data.

Preliminaries

Traffic Networks

In this study, we define a traffic network as an undirected graph $G = (V, E, \mathbf{A})$, as shown in Fig. 2(a), where V is a finite set of $|V| = N$ nodes; E is a set of edges, indicating the connectivity between the nodes; $\mathbf{A} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix of graph G . Each node on the traffic network G detects F measurements with the same sampling frequency, that is, each node generates a feature vector of length F at each time slice, as shown by the solid lines in Fig. 2(b).

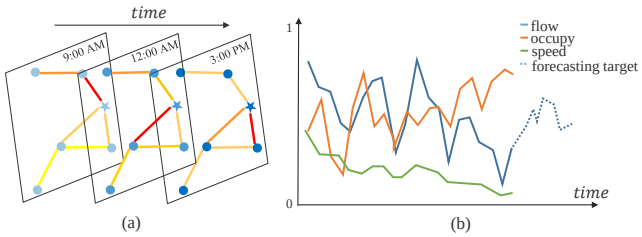


Figure 2: (a) The spatial-temporal structure of traffic data, where the data at each time slice forms a graph; (b) Three measurements are detected on a node and the future traffic flow is the forecasting target. Here, all measurements are normalized to [0,1].

Traffic Flow Forecasting

Suppose the f-th time series recorded on each node in the traffic network G is the traffic flow sequence, and $f \in (1, ..., F)$. We use $x_t^{c,i} \in \mathbb{R}$ to denote the value of the c-th feature of node i at time t , and $\mathbf{x}_t^i \in \mathbb{R}^F$ denotes the values of all the features of node i at time t. $\mathbf{X}_t = (\mathbf{x}_t^1, \mathbf{x}_t^2, ..., \mathbf{x}_t^N)^T \in \mathbb{R}^{N \times F}$ denotes the values of all the features of all the nodes at time t. $\boldsymbol{\mathcal{X}} = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_\tau)^T \in \mathbb{R}^{N \times F \times \tau}$ denotes the value of all the features of all the nodes over τ time slices. In addition, we set $y_t^i = x_t^{f,i} \in \mathbb{R}$ to represent the traffic flow of node i at time t in the future.

Problem. Given $\boldsymbol{\mathcal{X}}$, all kinds of the historical measurements of all the nodes on the traffic network over past τ time slices, predict future traffic flow sequences $\mathbf{Y} = (\mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^N)^T \in \mathbb{R}^{N \times T_p}$ of all the nodes on the whole traffic network over the next T_p time slices, where $\mathbf{y}^i = (y_{\tau+1}^i, y_{\tau+2}^i, ..., y_{\tau+T_p}^i) \in \mathbb{R}^{T_p}$ denotes the future traffic flow of node i from $\tau + 1$.

Attention Based Spatial-Temporal Graph Convolutional Networks

Fig. 3 presents the overall framework of the ASTGCN model proposed in this paper. It consists of three independent components with the same structure, which are designed to respectively model the recent, daily-periodic and weekly-periodic dependencies of the historical data.

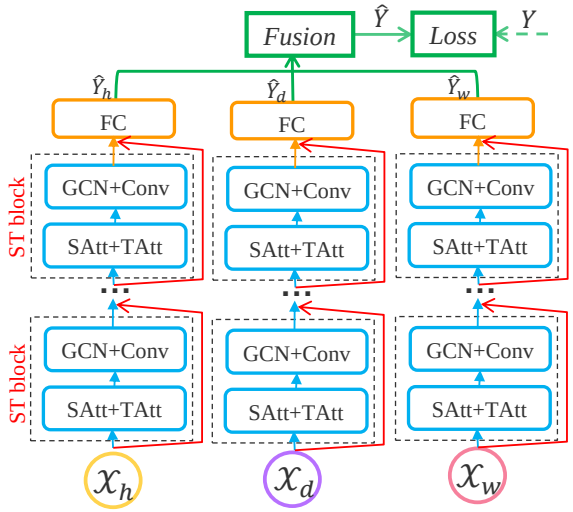


Figure 3: The framework of ASTGCN. SAtt: Spatial Attention; TAtt: Temporal Attention GCN: Graph Convolution; Conv: Convolution; FC: Fully-connected; ST block: Spatial-Temporal block.

Suppose the sampling frequency is q times per day. Assume that the current time is t_0 and the size of predicting window is T_p. As shown in Fig. 4, we intercept three time series segments of length T_h, T_d and T_w along the time axis as the input of the recent, daily-period and weekly-period component respectively, where T_h, T_d and T_w are all integer multiples of T_p. Details about the three time series segments are as follows:

(1) The *recent* segment:
$\boldsymbol{\mathcal{X}}_h = (\mathbf{X}_{t_0-T_h+1}, \mathbf{X}_{t_0-T_h+2}, ..., \mathbf{X}_{t_0}) \in \mathbb{R}^{N \times F \times T_h}$, a segment of historical time series directly adjacent to the predicting period, as shown by the green part of Fig. 4. Intuitively, the formation and dispersion of traffic congestions are gradual. So, the just past traffic flows inevitably have influence on the future traffic flows.

(2) The *daily-periodic* segment:
$\boldsymbol{\mathcal{X}}_d = (\mathbf{X}_{t_0-(T_d/T_p)*q+1}, ..., \mathbf{X}_{t_0-(T_d/T_p)*q+T_p},$
$\mathbf{X}_{t_0-(T_d/T_p-1)*q+1}, ..., \mathbf{X}_{t_0-(T_d/T_p-1)*q+T_p}, ...,$
$\mathbf{X}_{t_0-q+1}, ..., \mathbf{X}_{t_0-q+T_p}) \in \mathbb{R}^{N \times F \times T_d}$ consists of the segments on the past few days at the same time period as the predicting period, as shown by the red part of Fig. 4. Due to the regular daily routine of people, traffic data may show repeated patterns, such as the daily morning peaks. The purpose of the daily-period component is to model the daily periodicity of traffic data.

(3) The *weekly-periodic* segment:
$\boldsymbol{\mathcal{X}}_w = (\mathbf{X}_{t_0-7*(T_w/T_p)*q+1}, ..., \mathbf{X}_{t_0-7*(T_w/T_p)*q+T_p},$
$\mathbf{X}_{t_0-7*(T_w/T_p-1)*q+1}, ..., \mathbf{X}_{t_0-7*(T_w/T_p-1)*q+T_p}, ...,$
$\mathbf{X}_{t_0-7*q+1}, ..., \mathbf{X}_{t_0-7*q+T_p}) \in \mathbb{R}^{F \times N \times T_w}$ is composed of the segments on last few weeks, which have the same week attributes and time intervals as the forecasting period, as

shown by the blue part of Fig. 4. Usually, the traffic patterns on Monday have a certain similarity with the traffic patterns on Mondays in history, but may be greatly different from those on weekends. Thus, the weekly-period component is designed to capture the weekly periodic features in traffic data.

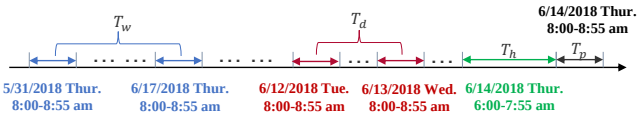


Figure 4: An example of constructing the input of time series segments (suppose the size of predicting window is 1 hour, and T_h, T_d and T_w are twice of T_p.

The three components share the same network structure and each of them consists of several spatial-temporal blocks and a fully-connected layer. There are a spatial-temporal attention module and a spatial-temporal convolution module in each spatial-temporal block. In order to optimize the training efficiency, we adopted a residual learning framework (He et al. 2016) in each component. In the end, the outputs of the three components are further merged based on a parameter matrix to obtain the final prediction result. The overall network structure is elaborately designed to describe the dynamic spatial-temporal correlations of traffic flows.

Spatial-Temporal Attention

A novel spatial-temporal attention mechanism is proposed in our model to capture the dynamic spatial and temporal correlations on the traffic network (as described in Fig. 1). It contains two kinds of attentions, i.e., spatial attention and temporal attention.

Spatial attention In the spatial dimension, the traffic conditions of different locations have influence among each other and the mutual influence is highly dynamic. Here, we use an attention mechanism (Feng et al. 2017) to adaptively capture the dynamic correlations between nodes in the spatial dimension.

Take the spatial attention in the recent component as an example:

$$\mathbf{S} = \mathbf{V}_s \cdot \sigma((\boldsymbol{\mathcal{X}}_h^{(r-1)}\mathbf{W}_1)\mathbf{W}_2(\mathbf{W}_3\boldsymbol{\mathcal{X}}_h^{(r-1)})^T + \mathbf{b}_s) \quad (1)$$

$$\mathbf{S}'_{i,j} = \frac{\exp(\mathbf{S}_{\mathrm{i,j}})}{\sum_{j=1}^{N}\exp(\mathbf{S}_{\mathrm{i,j}})} \quad (2)$$

where $\boldsymbol{\mathcal{X}}_h^{(r-1)} = (\mathbf{X}_1, \mathbf{X}_2, ...\mathbf{X}_{T_{r-1}}) \in \mathbb{R}^{N\times C_{r-1}\times T_{r-1}}$ is the input of the r^{th} spatial-temporal block. C_{r-1} is the number of channels of the input data in the r^{th} layer. When $r = 1$, $C_0 = F$. T_{r-1} is the length of the temporal dimension in the r^{th} layer. When $r = 1$, in the recent component $T_0 = T_h$ (in the daily-period component $T_0 = T_d$ and in the weekly-period component $T_0 = T_w$). $\mathbf{V}_s, \mathbf{b}_s \in \mathbb{R}^{N\times N}$, $\mathbf{W}_1 \in \mathbb{R}^{T_{r-1}}$, $\mathbf{W}_2 \in \mathbb{R}^{C_{r-1}\times T_{r-1}}$, $\mathbf{W}_3 \in \mathbb{R}^{C_{r-1}}$ are learnable parameters and sigmoid σ is used as the activation function. The attention matrix $\mathbf{S}$ is dynamically computed according to the current input of this layer. The value of an element $\mathbf{S}_{i,j}$ in $\mathbf{S}$ semantically represents the correlation strength between node i and node j. Then a softmax function is used to ensure the attention weights of a node sum to one. When performing the graph convolutions, we will accompany the adjacency matrix $\mathbf{A}$ with the spatial attention matrix $\mathbf{S}' \in \mathbb{R}^{N\times N}$ to dynamic adjust the impacting weights between nodes.

Temporal attention In the temporal dimension, there exist correlations between the traffic conditions in different time slices, and the correlations are also varying under different situations. Likewise, we use an attention mechanism to adaptively attach different importance to data:

$$\mathbf{E} = \mathbf{V}_e \cdot \sigma(((\boldsymbol{\mathcal{X}}_h^{(r-1)})^T\mathbf{U}_1)\mathbf{U}_2(\mathbf{U}_3\boldsymbol{\mathcal{X}}_h^{(r-1)}) + \mathbf{b}_e) \quad (3)$$

$$\mathbf{E}'_{i,j} = \frac{\exp(\mathbf{E}_{\mathrm{i,j}})}{\sum_{j=1}^{T_{r-1}}\exp(\mathbf{E}_{\mathrm{i,j}})} \quad (4)$$

where $\mathbf{V}_e, \mathbf{b}_e \in \mathbb{R}^{T_{r-1}\times T_{r-1}}$, $\mathbf{U}_1 \in \mathbb{R}^{N}$, $\mathbf{U}_2 \in \mathbb{R}^{C_{r-1}\times N}$, $\mathbf{U}_3 \in \mathbb{R}^{C_{r-1}}$ are learnable parameters. The temporal correlation matrix $\mathbf{E}$ is determined by the varying inputs. The value of an element $\mathbf{E}_{i,j}$ in $\mathbf{E}$ semantically indicates the strength of dependencies between time i and j. At last, $\mathbf{E}$ is normalized by the softmax function. We directly apply the normalized temporal attention matrix to the input and get $\hat{\boldsymbol{\mathcal{X}}}_h^{(r-1)} = (\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, ..., \hat{\mathbf{X}}_{T_{r-1}}) = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_{T_{r-1}})\mathbf{E}' \in \mathbb{R}^{N\times C_{r-1}\times T_{r-1}}$ to dynamically adjust the input by merging relevant information.

Spatial-Temporal Convolution

The spatial-temporal attention module let the network automatically pay relatively more attention on valuable information. The input adjusted by the attention mechanism is fed into the spatial-temporal convolution module, whose structure is presented in Fig. 5. The spatial-temporal convolution module proposed here consists of a graph convolution in the spatial dimension, capturing spatial dependencies from neighborhood and a convolution along the temporal dimension, exploiting temporal dependencies from nearby times.

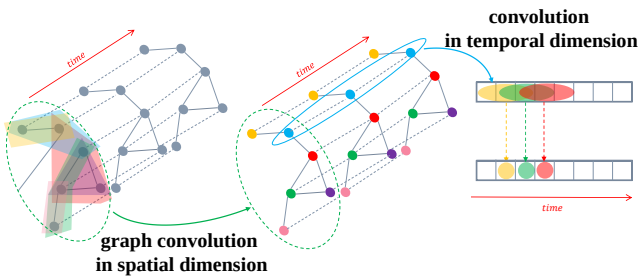


Figure 5: The architecture of spatial-temporal convolutions of ASTGCN.

Graph convolution in spatial dimension The spectral graph theory generalizes the convolution operation from grid-based data to graph structure data. In this study, the traffic network is a graph structure in nature, and the features of

each node can be regarded as the signals on the graph (Shuman et al. 2013). Hence, in order to make full use of the topological properties of the traffic network, at each time slice we adopt graph convolutions based on the spectral graph theory to directly process the signals, exploiting signal correlations on the traffic network in the spatial dimension. The spectral method transforms a graph into an algebraic form to analyze the topological attributes of graph, such as the connectivity in the graph structure.

In spectral graph analysis, a graph is represented by its corresponding Laplacian matrix. The properties of the graph structure can be obtained by analyzing Laplacian matrix and its eigenvalues. Laplacian matrix of a graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, and its normalized form is $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{N\times N}$, where $\mathbf{A}$ is the adjacent matrix, $\mathbf{I}_N$ is a unit matrix, and the degree matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix, consisting of node degrees, $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. The eigenvalue decomposition of the Laplacian matrix is $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{\Lambda} = diag([\lambda_0, ..., \lambda_{N-1}]) \in \mathbb{R}^{N\times N}$ is a diagonal matrix, and $\mathbf{U}$ is Fourier basis. Taking the traffic flow at time t as an example, the signal all over the graph is $x = \mathbf{x}_t^f \in \mathbb{R}^N$, and the graph Fourier transform of the signal is defined as $\hat{x} = \mathbf{U}^T x$. According to the properties of the Laplacian matrix, $\mathbf{U}$ is an orthogonal matrix, so the corresponding inverse Fourier transform is $x = \mathbf{U}\hat{x}$. The graph convolution is a convolution operation implemented by using linear operators that diagonalize in the Fourier domain to replace the classical convolution operator (Henaff, Bruna, and LeCun 2015). Based on this, the signal x on the graph G is filtered by a kernel g_θ:

$$g_\theta *_G x = g_\theta(\mathbf{L})x = g_\theta(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T)x = \mathbf{U}g_\theta(\mathbf{\Lambda})\mathbf{U}^T x \quad (5)$$

where $*_G$ denotes a graph convolution operation. Since the convolution operation of the graph signal is equal to the product of these signals which have been transformed into the spectral domain by graph Fourier transform (Simonovsky and Komodakis 2017), the above formula can be understood as Fourier transforming g_θ and x respectively into the spectral domain, then multiplying their transformed results, and doing the inverse Fourier transform to get the final result of the convolution operation. However, it is expensive to directly perform the eigenvalue decomposition on the Laplacian matrix when the scale of the graph is large. Therefore, Chebyshev polynomials are adopted in this paper to solve this problem approximately but efficiently (Simonovsky and Komodakis 2017):

$$g_\theta *_G x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathbf{L}})x \quad (6)$$

where the parameter $\theta \in \mathbb{R}^K$ is a vector of polynomial coefficients. $\tilde{\mathbf{L}} = \frac{2}{\lambda_{max}}\mathbf{L} - \mathbf{I}_N$, λ_{max} is the maximum eigenvalue of the Laplacian matrix. The recursive definition of the Chebyshev polynomial is $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, where $T_0(x) = 1$, $T_1(x) = x$. Using approximate expansion of Chebyshev polynomial to solve this formulation corresponds to extracting information of the surrounding 0 to $(K-1)^{th}$-order neighbors centered on each node in the graph by the convolution kernel g_θ. The graph convolution module uses the Rectified Linear Unit (ReLU) as the final activation function, i.e., ReLU($g_\theta *_G x$).

In order to dynamically adjust the correlations between nodes, for each term of Chebyshev polynomial, we accompany $T_k(\widetilde{\mathbf{L}})$ with the spatial attention matrix $\mathbf{S}' \in \mathbb{R}^{N\times N}$, then obtain $T_k(\widetilde{\mathbf{L}}) \odot \mathbf{S}'$, where $\odot$ is the Hadamard product. Therefore, the above graph convolution formula changes to $g_\theta *_G x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k(T_k(\tilde{\mathbf{L}}) \odot \mathbf{S}')x$.

We can generalize this definition to the graph signal with multiple channels. For example, in the recent component, the input is $\hat{\boldsymbol{\mathcal{X}}}_h^{(r-1)} = (\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, ..., \hat{\mathbf{X}}_{T_{r-1}}) \in \mathbb{R}^{N\times C_{r-1}\times T_{r-1}}$, where the feature of each node has C_{r-1} channels. For each time slice t, performing C_r filters on the graph $\hat{\mathbf{X}}_t$, we get $g_\theta *_G \hat{\mathbf{X}}_t$, where $\Theta = (\Theta_1, \Theta_2, ..., \Theta_{C_r}) \in \mathbb{R}^{K\times C_{r-1}\times C_r}$ is the convolution kernel parameter (Kipf and Welling 2017). Therefore, each node is updated by the information of the 0∼K-1 neighbors of the node.

Convolution in temporal dimension After the graph convolution operations having captured neighboring information for each node on the graph in the spatial dimension, a standard convolution layer in the temporal dimension is further stacked to update the signal of a node by merging the information at the neighboring time slice, as shown by the right part in Fig. 5. Also take the operation on the r^{th} layer in the recent component as an example:

$$\boldsymbol{\mathcal{X}}_h^{(r)} = ReLU(\Phi * (ReLU(g_\theta *_G \hat{\boldsymbol{\mathcal{X}}}_h^{(r-1)}))) \in \mathbb{R}^{C_r\times N\times T_r} \quad (7)$$

where $*$ denotes a standard convolution operation, Φ is the parameters of the temporal dimension convolution kernel, and the activation function is ReLU.

In conclusion, a spatial-temporal convolution module is able to well capture the temporal and spatial features of traffic data. A spatial-temporal attention module and a spatial-temporal convolution module forms a spatial-temporal block. Multiple spatial-temporal blocks are stacked to further extract larger range of dynamic spatial-temporal correlations. Finally, a fully connected layer is appended to make sure the output of each component has the same dimension and shape with the forecasting target. The final fully connected layer uses ReLU as the activation function.

Multi-Component Fusion

In this section, we will discuss how to integrate the outputs of the three components. Take forecasting the traffic flow on the whole traffic network at 8:00 am on Friday as an example. It can be observed that the traffic flows at some areas have obvious peak periods in the morning, so the outputs of the daily-period and weekly-period components are more crucial. However, there are no distinct traffic cycle patterns in some other places, thus the daily-period and weekly-period components may be helpless. Consequently, when the outputs of different components are fused, the impacting weights of the three components for each node are different, and they should be learned from the historical data. So the

final prediction result after the fusion is:

$$\hat{\mathbf{Y}} = \mathbf{W}_h \odot \hat{\mathbf{Y}}_h + \mathbf{W}_d \odot \hat{\mathbf{Y}}_d + \mathbf{W}_w \odot \hat{\mathbf{Y}}_w \quad (8)$$

where $\odot$ is the Hadamard product. $\mathbf{W}_h$, $\mathbf{W}_d$ and $\mathbf{W}_w$ are learning parameters, reflecting the influence degrees of the three temporal-dimensional components on the forecasting target.

Experiments

In order to evaluate the performance of our model, we carried out comparative experiments on two real-world highway traffic datasets.

Datasets

We validate our model on two highway traffic datasets PeMSD4 and PeMSD8 from California. The datasets are collected by the Caltrans Performance Measurement System (PeMS) (Chen et al. 2001) in real time every 30 seconds. The traffic data are aggregated into every 5-minute interval from the raw data. The system has more than 39,000 detectors deployed on the highway in the major metropolitan areas in California. Geographic information about the sensor stations are recorded in the datasets. There are three kinds of traffic measurements considered in our experiments, including total flow, average speed, and average occupancy.

PeMSD4 It refers to the traffic data in San Francisco Bay Area, containing 3848 detectors on 29 roads. The time span of this dataset is from January to February in 2018. We choose data on the first 50 days as the training set, and the remains as the test set.

PeMSD8 It is the traffic data in San Bernardino from July to August in 2016, which contains 1979 detectors on 8 roads. The data on the first 50 days are used as the training set and the data on the last 12 days are the test set.

Preprocessing

We remove some redundant detectors to ensure the distance between any adjacent detectors is longer than 3.5 miles. Finally, there are 307 detectors in the PeMSD4 and 170 detectors in the PeMSD8. The traffic data are aggregated every 5 minutes, so each detector contains 288 data points per day. The missing values are filled by the linear interpolation. In addition, the data are transformed by zero-mean normalization $x' = x - mean(x)$ to let the average be 0.

Settings

We implemented the ASTGCN model based on the MXNet[1] framework. According to Kipf and Welling (2017), we test the number of the terms of Chebyshev polynomial $K \in \{1, 2, 3\}$. As K becomes larger, the forecasting performance improves slightly. So does the kernel size in the temporal dimension. Considering the computing efficiency and the degree of improvement of the forecasting performance, we set $K = 3$ and the kernel size along the temporal dimension to 3. In our model, all the graph convolution layers use 64 convolution kernels. All the temporal convolution layers use 64 convolution kernels and the time span of the data is adjusted by controlling the step size of the temporal convolutions. For the lengths of the three segments, we set them as: $T_h = 24$, $T_d = 12$, $T_w = 24$. The size of the predicting window $T_p = 12$, that is to say, we aim at predicting the traffic flow over one hour in the future. In this paper, the mean square error (MSE) between the estimator and the ground truth are used as the loss function and minimized by back-propagation. During the training phase, the batch size is 64 and the learning rate is 0.0001. In addition, in order to verify the impact of the spatio-temporal attention mechanism proposed here, we also design a degraded version of ASTGCN, named Multi-Component Spatial-Temporal Graph Convolution Networks (MSTGCN), which gets rid of the spatial-temporal attention. The settings of MSTGCN are the same as those of ASTGCN, except no spatial-temporal attention.

[1] https://mxnet.apache.org/

Baselines

We compare our model with the following eight baselines:

- HA: Historical Average method. Here, we use the average value of the last 12 time slices to predict the next value.
- ARIMA (Williams and Hoel 2003): Auto-Regressive Integrated Moving Average method is a well-known time series analysis method for predicting the future values.
- VAR (Zivot and Wang 2006): Vector Auto-Regressive is a more advanced time series model, which can capture the pairwise relationships among all traffic flow series.
- LSTM (Hochreiter and Schmidhuber 1997): Long Short-Term Memory network, a special RNN model.
- GRU (Chung et al. 2014): Gated Recurrent Unit network, a special RNN model.
- STGCN (Li et al. 2018): A spatial-temporal graph convolution model based on the spatial method.
- GLU-STGCN (Yu, Yin, and Zhu 2018): A graph convolution network with a gating mechanism, which is specially designed for traffic forecasting.
- GeoMAN (Liang et al. 2018): A multi-level attention-based recurrent neural network model proposed for the geo-sensory time series prediction problem.

Root mean square error (RMSE) and mean absolute error (MAE) are used as the evaluation metrics.

Comparison and Result Analysis

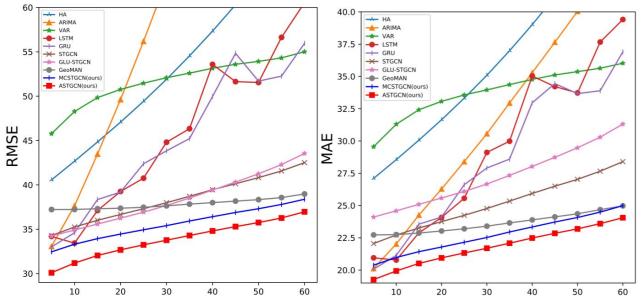
We compare our models with the eight baseline methods on PeMSD4 and PeMSD8. Table 1 shows the average results of traffic flow prediction performance over the next one hour.

It can be seen from Table 1 that our ASTGCN achieves the best performance in both two datasets in terms of all evaluation metrics. We can observe that the prediction results of the traditional time series analysis methods are usually not ideal, demonstrating those methods' limited abilities of modeling nonlinear and complex traffic data. By comparison, the methods based on deep learning generally obtain

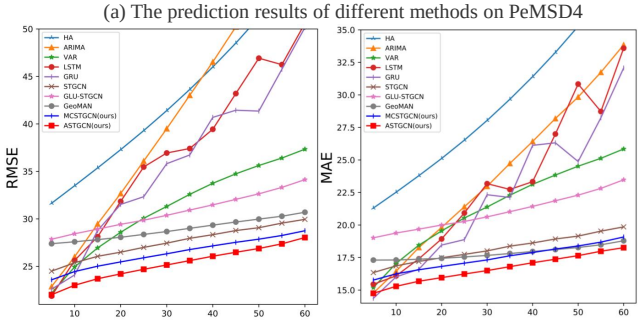
Model	PeMSD4		PeMSD8	
	RMSE	MAE	RMSE	MAE
HA	54.14	36.76	44.03	29.52
ARIMA	68.13	32.11	43.30	24.04
VAR	51.73	33.76	31.21	21.41
LSTM	45.82	29.45	36.96	23.18
GRU	45.11	28.65	35.95	22.20
STGCN	38.29	25.15	27.87	18.88
GLU-STGCN	38.41	27.28	30.78	20.99
GeoMAN	37.84	23.64	28.91	17.84
MSTGCN (ours)	**35.64**	**22.73**	**26.47**	**17.47**
ASTGCN (ours)	**32.82**	**21.80**	**25.27**	**16.63**

Table 1: Average performance comparison of different approaches on PeMSD4 and PeMSD8.

better prediction results than the traditional time series analysis methods. Among them, the models which simultaneously take both the temporal and spatial correlations into account, including STGCN, GLU-STGCN, GeoMAN and two versions of our model, are superior to the traditional deep learning models such as LSTM and GRU. Besides, GeoMAN performs better than STGCN and GLU-STGCN, indicating the multi-level attention mechanisms applied in GeoMAN are efficient in capturing the dynamic changings of traffic data. Our MSTGCN, without any attention mechanisms, achieve better results than the previous state-of-the-art models, proving the advantages of our model in describing spatial-temporal features of the highway traffic data. Then combined with the spatial-temporal attention mechanisms, our ASTGCN further reduces the forecasting errors.



(a) The prediction results of different methods on PeMSD4



(b) The prediction results of different methods on PeMSD8

Figure 6: Performance changes of different methods as the forecasting interval increases.

Fig. 6 shows the changes of prediction performance of various methods as the prediction interval increases. Overall, as the prediction interval becomes longer, the corresponding difficulty of prediction is getting greater, hence the prediction errors also increase. As can be seen from the figure, the methods only taking the temporal correlation into account can achieve good results in the short-term prediction, such as HA, ARIMA, LSTM and GRU. However, with the increase of the prediction interval, their prediction accuracy drops dramatically. By comparison, the performance of VAR drops slower than those methods. This is mainly because VAR can simultaneously consider the spatial-temporal correlations which are more important in the long-term prediction. However, when the scale of the traffic network becomes larger, i.e., there are more time series considered in the model, the prediction error of VAR increases, as shown in Fig.6, its performance on PeMSD4 is worse than that on PeMSD8. The errors of deep learning methods increase slowly with prediction interval increases, and their overall performance is good. Our ASTGCN model achieves the best prediction performance almost all the time. Especially in the long-term prediction, the differences between ASTGCN and other baselines are more significant, showing that the strategy of combining attention mechanism with graph convolution can better mine the dynamic spatial-temporal patterns of traffic data.

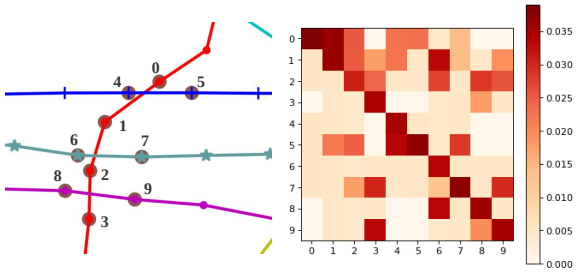


Figure 7: The attention matrix obtained from the spatial attention mechanism.

In order to investigate the role of attention mechanisms in our model intuitively, we perform a case study: picking out a sub-graph with 10 detectors from the PeMSD8 and showing the average spatial attention matrix among detectors in the training set. As shown on the right side of Fig. 7, in the spatial attention matrix, the i-th row represents the correlation strength between each detector and the i-th detector. For instance, look at the last row, we can know traffic flows on the 9th detector is closely related to those on the 3th and 8th detectors. This is reasonable since these three detectors are close in space on the real traffic network, as shown on the left side of Fig. 7. Hence, our model not only achieves a best forecasting performance but also shows an interpretability advantage.

Conclusion and Future Work

In this paper, a novel attention based spatial-temporal graph convolution model called ASTGCN is proposed and successfully applied to forecasting traffic flow. The model

combines the spatial-temporal attention mechanism and the spatial-temporal convolution, including graph convolutions in the spatial dimension and standard convolutions in the temporal dimension, to simultaneously capture the dynamic spatial-temporal characteristics of traffic data. Experiments on two real-world datasets show that the forecasting accuracy of the proposed model is superior to existing models. The code has been released at: `https://github.com/wanhuaiyu/ASTGCN`.

Actually, the highway traffic flow is affected by many external factors, like weather and social events. In the future, we will take some external influencing factors into account to further improve the forecasting accuracy. Since the ASTGCN is a general spatial-temporal forecasting framework for the graph structure data, we can also apply it to other pragmatic applications, such as estimating time of arrival.

Acknowledgments

This work was supported by the Natural Science Foundation of China (No. 61603028).

References

Bruna, J.; Zaremba, W.; Szlam, A.; and Lecun, Y. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*.

Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P.; and Jia, Z. 2001. Freeway performance measurement system: mining loop detector data. *Transportation Research Record: Journal of the Transportation Research Board* (1748):96–102.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *NIPS 2014 Workshop on Deep Learning*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, 3844–3852.

Feng, X.; Guo, J.; Qin, B.; Liu, T.; and Liu, Y. 2017. Effective deep memory networks for distant supervised relation extraction. In *International Joint Conference on Artificial Intelligence*, 19–25.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Henaff, M.; Bruna, J.; and LeCun, Y. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Jeong, Y.-S.; Byon, Y.-J.; Castro-Neto, M. M.; and Easa, S. M. 2013. Supervised weighting-online learning algorithm for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems* 14(4):1700–1707.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*.

Li, C.; Cui, Z.; Zheng, W.; Xu, C.; and Yang, J. 2018. Spatio-Temporal Graph Convolution for Skeleton Based Action Recognition. In *AAAI Conference on Artificial Intelligence*, 3482–3489.

Liang, Y.; Ke, S.; Zhang, J.; Yi, X.; and Zheng, Y. 2018. GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction. In *International Joint Conference on Artificial Intelligence*, 3428–3434.

Niepert, M.; Ahmed, M.; and Kutzkov, K. 2016. Learning convolutional neural networks for graphs. In *International conference on machine learning*, 2014–2023.

Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* 30(3):83–98.

Simonovsky, M., and Komodakis, N. 2017. Dynamic edgeconditioned filters in convolutional neural networks on graphs. In *Computer Vision and Pattern Recognition*, 3693–3702.

Van Lint, J., and Van Hinsbergen, C. 2012. Short-term traffic and travel time prediction models. *Artificial Intelligence Applications to Critical Transportation Issues* 22(1):22–41.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Williams, B. M., and Hoel, L. A. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering* 129(6):664–672.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057.

Yao, H.; Tang, X.; Wei, H.; Zheng, G.; Yu, Y.; and Li, Z. 2018a. Modeling spatial-temporal dynamics for traffic prediction. *arXiv preprint arXiv:1803.01254*.

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; and Ye, J. 2018b. Deep multi-view spatial-temporal network for taxi demand prediction. In *AAAI Conference on Artificial Intelligence*, 2588–2595.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *International Joint Conference on Artificial Intelligence*, 3634–3640.

Zhang, J.; Wang, F.-Y.; Wang, K.; Lin, W.-H.; Xu, X.; and Chen, C. 2011. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems* 12(4):1624–1639.

Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X.; and Li, T. 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence* 259:147–166.

Zivot, E., and Wang, J. 2006. Vector autoregressive models for multivariate time series. *Modeling Financial Time Series with S-PLUS®* 385–429.