Literature review - Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting

Jiawei Wang

jiawei.wang4@mail.mcgill.ca

August 23, 2019

Outline

1. Motivation
2. Overall idea & Methodology
3. Empirical study review

Motivation

Problem

$[X_0, X_2, ...X_t; \mathcal{G}] \rightarrow [X_{t+1}, ...X_{t+T}]$

Challenges

1. Complex spatial dependency
2. Non-linear temporal dynamics
3. Difficult long-term forecasting

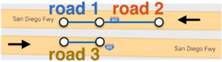

Figure: Spatial dependency

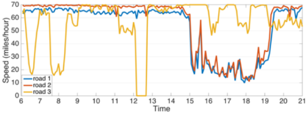

Figure: Long-term temporal variation

Highlights & Methodology

1. Complex spatial dependency Diffusion convolution
2. Non-linear temporal dynamics Recurrent neural network
3. Difficult long-term forecasting Encoder-decoder architecture

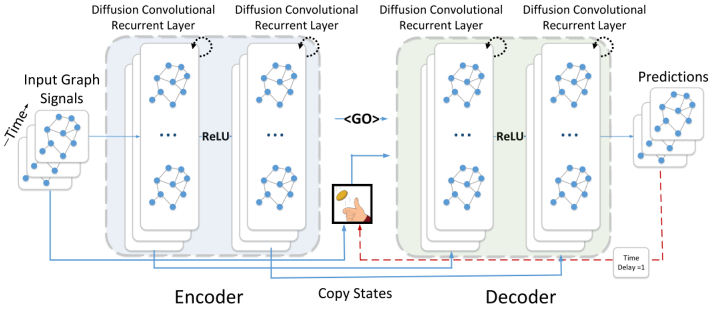

Figure: Overall structure

How to model spatial dependency through diffusion convolution?

Highlights & **Methodology**

- Diffusion process can be regarded as a Markov process considering continuous time and state.
  - Traffic flow in road network is continuous and will have dynamic impact downstream or upstream.
- Diffusion convolution

$$\boldsymbol{X}_{:,p\star\mathcal{G}}\ f_\theta = \sum_{k=0}^{K-1}\left(\theta_{k,1}(D_O^{-1}W)^k + \theta_{k,1}(D_I^{-1}W)^k\right)\boldsymbol{X}_{:,p}$$

where adjacency matrix $\boldsymbol{W}$ using threshold gaussian kernel:
$\boldsymbol{W}_{i,j} = exp(-\frac{dist(v_i, v_j)}{\sigma^2})$

Highlights & **Methodology**

- Diffusion process can be regarded as a Markov process considering continuous time and state.
  - Traffic flow in road network is continuous and will have dynamic impact downstream or upstream.
- Diffusion convolution

$$\boldsymbol{X}_{:,p\star\mathcal{G}}\ f_\theta = \sum_{k=0}^{K-1}\left(\theta_{k,1}(D_O^{-1}W)^k + \theta_{k,1}(D_I^{-1}W)^k\right)\boldsymbol{X}_{:,p}$$

where adjacency matrix $\boldsymbol{W}$ using threshold gaussian kernel:

$$\boldsymbol{W}_{i,j} = exp\left(-\frac{dist(v_i, v_j)}{\sigma^2}\right)$$

Highlights & **Methodology**

- Diffusion process can be regarded as a Markov process considering continuous time and state.
  - Traffic flow in road network is continuous and will have dynamic impact downstream or upstream.
- Diffusion convolution

$$\boldsymbol{X}_{:,p\star\mathcal{G}}\ f_\theta = \sum_{k=0}^{K-1}\left(\theta_{k,1}(D_O^{-1}W)^k + \theta_{k,1}(D_I^{-1}W)^k\right)\boldsymbol{X}_{:,p}$$

where adjacency matrix $\boldsymbol{W}$ using threshold gaussian kernel:

$$\boldsymbol{W}_{i,j} = exp\left(-\frac{dist(v_i, v_j)}{\sigma^2}\right)$$

Diffusion convolution

$$\boldsymbol{X}_{:,p\star\mathcal{G}}\ f_\theta = \sum_{k=0}^{K-1}\left(\boxed{\theta_{k,1}(D_O^{-1}W)^k} + \boxed{\theta_{k,1}(D_I^{-1}W)^k}\right)\boldsymbol{X}_{:,p}$$

- $\boldsymbol{D_O^{-1}W}$:(Out-degree)normalized state transition matrix
- $\boldsymbol{D_I^{-1}W}$:(In-degree)normalized state transition matrix
- θ: Filter parameters to train
- The power k indicates k-neighbor nodes are considered for each node in the graph(road network)

Diffusion convolution

$$\boldsymbol{X}_{:,p\star\mathcal{G}}\ f_\theta = \sum_{k=0}^{K-1}\left(\theta_{k,1}(D_O^{-1}W)^k + \theta_{k,1}(D_I^{-1}W)^k\right)\boldsymbol{X}_{:,p}$$

- $\boldsymbol{D_O^{-1}W}$:(Out-degree)normalized state transition matrix
- $\boldsymbol{D_I^{-1}W}$:(In-degree)normalized state transition matrix
- θ: Filter parameters to train
- The power k indicates k-neighbor nodes are considered for each node in the graph(road network)

Highlights & **Methodology**

Diffusion convolution

$$\boldsymbol{X}_{:,p\star\mathcal{G}}\ f_\theta = \sum_{k=0}^{K-1}\left(\theta_{k,1}(D_O^{-1}W)^k + \theta_{k,1}(D_I^{-1}W)^k\right)\boldsymbol{X}_{:,p}$$

- $\boldsymbol{D_O^{-1}W}$:(Out-degree)normalized state transition matrix
- $\boldsymbol{D_I^{-1}W}$:(In-degree)normalized state transition matrix
- $\boldsymbol{\theta}$: Filter parameters to train
- The power k indicates k-neighbor nodes are considered for each node in the graph(road network)

Highlights & **Methodology**

Diffusion convolution

$$\boldsymbol{X}_{:,p\star\mathcal{G}}\ f_\theta = \sum_{k=0}^{K-1}\left(\boxed{\theta_{k,1}(D_O^{-1}W)^k} + \boxed{\theta_{k,1}(D_I^{-1}W)^k} \right)\boldsymbol{X}_{:,p}$$

- $\boldsymbol{D_O^{-1}W}$:(Out-degree)normalized state transition matrix
- $\boldsymbol{D_I^{-1}W}$:(In-degree)normalized state transition matrix
- $\boldsymbol{\theta}$: Filter parameters to train
- The power k indicates k-neighbor nodes are considered for each node in the graph(road network)

Highlights & **Methodology**

What is the relation between diffusion convolution and spectral graph convolution?

- Bruna,et al(2013). Spectral Networks and Locally Connected Networks on Graphs.
- MichaÃńl Defferrard, et al(2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering.
- Cui, Z.,et al(2018). High-Order Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting.

Highlights & **Methodology**

What is the relation between diffusion convolution and spectral graph convolution?

- Bruna,et al(2013). Spectral Networks and Locally Connected Networks on Graphs.
- MichaÃńl Defferrard, et al(2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering.
- Cui, Z.,et al(2018). High-Order Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting.

Diffusion convolution takes into account both two directions respectively among a graph, while spectral graph convolution focus on undirected graph.

Empirical study review

- Comprehensive model validation:
  1. Three evaluation metrics:MAE, MAPE, RMSE
  2. Two real-world datasets
  3. Six baselines: HA, ARIMA, FNN, SVR, RNN-based
  4. Two specific validation: model without diffusion convolution component or RNN structure.
- Interesting interpretation:

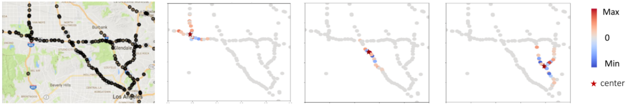

Figure: Visualization of $\theta_{k,1}(D_O^{-1}W)^k + \theta_{k,1}(D_I^{-1}W)^k$

Thank you for attention !

Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). **Spectral Networks and Locally Connected Networks on Graphs**, 1–14. Retrieved from http://arxiv.org/abs/1312.6203

两信号的图卷积

$$(f * h)_G = U(U^T h \odot U^T f)$$

卷积核的推导

$$U^T h = [\hat{h}(\lambda_1), \hat{h}(\lambda_2) \cdots \hat{h}(\lambda_n)]$$

$$U^T h \odot U^T f = \boldsymbol{diag}(\boldsymbol{\hat{h}}(\boldsymbol{\lambda}))U^T f$$

应用

$$\boldsymbol{y} = U\boldsymbol{\mathcal{G}_\theta}(\boldsymbol{\Lambda})U^T \boldsymbol{x}$$

$$\mathcal{G}_\theta(\Lambda) = diag(\theta), \theta \in \mathbb{R}^n$$

p个输入channel，q个输出channel

$$\boldsymbol{y}_j = U \sum_{i=0}^{p} \boldsymbol{\mathcal{G}_\theta}(\boldsymbol{\Lambda})_j U^T \boldsymbol{x}_i, j = 1,2 \ldots q$$

Michaël Defferrard Xavier Bresson Pierre Vandergheynst. (2016). **Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering**. *Nips*, (59), 395–398. https://doi.org/10.1016/j.commatsci.2018.05.018

令 $\mathcal{G}_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k$ 　得 $U\mathcal{G}_\theta(\Lambda)U^T = \sum_{k=0}^{K-1} \theta_k U \Lambda^k U^T = \sum_{k=0}^{K-1} \theta_k \prod_{i=1}^{k} U\Lambda U^T = \sum_{k=0}^{K-1} \theta_k L^k = \mathcal{G}_\theta(L)$

$$y = \mathcal{G}_\theta(L)x$$

巧妙之处：
1. 避免了Graph $Laplacian$的特征分解
2. L^k可以进行$k - loclaized$的筛选
3. 压缩参数量$n \to K$

$chebyshev\ polyminal\ function\ approximation$:

$$f(x) \approx \sum_{k=0}^{K} c_k T_k(x), x \in [-1,1]$$

$$T_0(x) = 1, T_1(x) = x, T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$

$\mathcal{G}_\theta(L) \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{L})$, 放缩 $\tilde{L} = \frac{2L}{\lambda_{max}} - I$

$$y \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{L})\, x$$

巧妙之处：
1. chebyshev approximation的系数直接通过神经网络训练计算
2. 避免幂运算