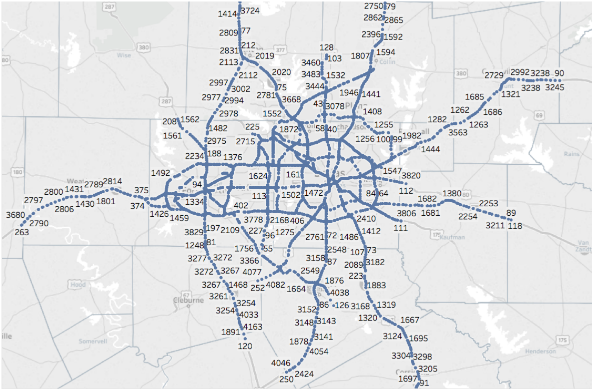
Graph Signal Processing for Traffic Prediction

Arman Hasanzadeh, Krishna Narayanan, and Nick Duffield

Department of Electrical and Computer Engineering
Texas A&M University

Dataset

- More than 10 billion data points from GPS, routing Apps, road cameras,...
- Average speed of vehicles (2min timestep) of 4700 road segments in Dallas
- Reported crashes to police dataset
- Collected by Texas A&M Transportation Institute for a year



TEXAS A&M UNIVERSITY

Problem Statement

- Real-time short-term traffic forecasting in transportation networks

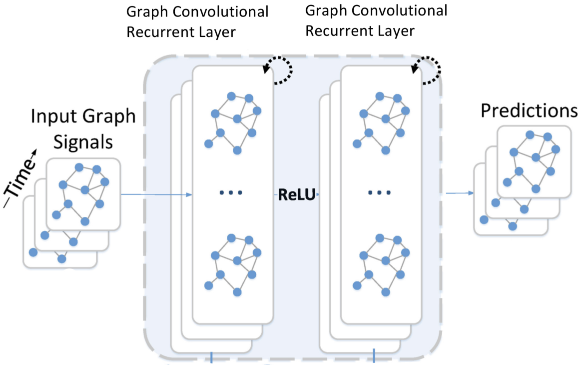
Intelligent Transportation Systems (ITS)

- Collect and process traffic data in real-time
- Car traffic delays costs $45 billion [1]
- Detecting congestion and its effect on neighboring roads
- Updating routing algorithms and traffic management strategies

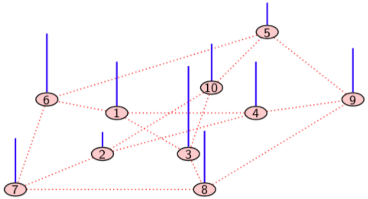
1. https ://www.citylab.com/life/2013/10/us-transportation-system-has-100-billion-worth-inefficiencies/7076/

Geometric Deep Learning (Li *et. al.*, 2018 - Cui *et. al.*, 2018)

- Traffic prediction using graph convolutional recurrent neural networks
- Computationally very expensive



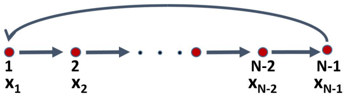
- Represent network as a graph $G = (\mathcal{V}, \mathcal{E})$
  - A is adjacency matrix
  - $D = diag(deg(v_i))$ is degree matrix
  - $L = D - A$ is Laplacian matrix
- Data defined on nodes of the graph $\longrightarrow$ graph data/signal



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{10} \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

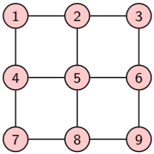
- Graph Signal Processing :
  - Leveraging graph structure for graph data analysis

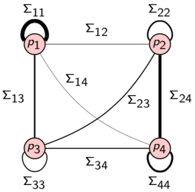
Conventional Signals as Graph Signals



- Time : unweighted and directed graphs

- Images : unweighted and undirected graphs

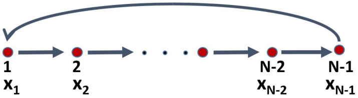




- Covariance : weighted and undirected graphs

Revisiting Discrete Signal Processing

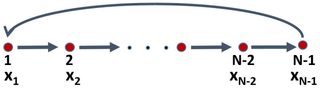
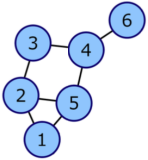
- Define the time series signal as a graph signal on ring graph



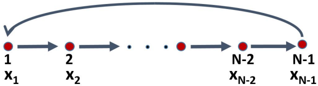
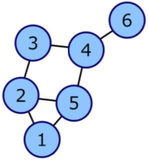
$$A = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

DSP operations can be derived from ring graph !

- $x[(n-k)_N] = A^k x \implies$ circular shift
- $A = V \Lambda V^{-1}$ eigendecomposition $\Rightarrow V^{-1} =$ DFT matrix
- $\text{DFT}(x) = V^{-1}x$
- Prediction of 1-D random processes using filters

	Discrete Signal Processing	Graph Signal Processing
Graph		
Shift	$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$	?
Fourier	$\mathbf{A} = V \Lambda V^{-1}$ $\mathcal{F}(x) = V^{-1} x$	?

DSP Vs. GSP cont.

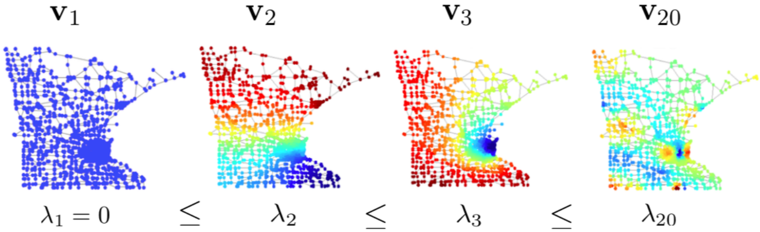
	Discrete Signal Processing	Graph Signal Processing
Graph		
Shift	$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$	$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{23} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix}$ Example : A or L
Fourier	$\mathbf{A} = V \Lambda V^{-1}$ $\mathcal{F}(x) = V^{-1}x$	$\mathbf{S} = V \Lambda V^{-1}$ $\mathrm{GFT}(x) = V^{-1}x$

Frequency Analysis of GFT

- v_k's (columns of V) are frequency atoms : $x = \sum_k \tilde{x} v_k$
- Total variation (TV) of a graph signal

$$\mathrm{TV}(x) = \sum_{(i,j)\in\mathcal{E}} A_{ij}(x_i - x_j)^2$$

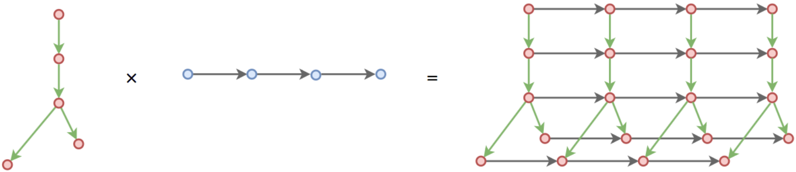
$$\lambda_k = \lambda_k v_k^T v_k = v_k^T L v_k = \sum_{(i,j)\in\mathcal{E}} A_{ij}([v_k]_i - [v_k]_j)^2 = \mathrm{TV}(v_k)$$



Eignevalues are frequencies!

Joint Fourier Transform

- L_G = graph Laplacian
- L_T = time series Laplacian
- $L_J = L_T \bigotimes \mathbb{I}_N + \mathbb{I}_T \bigotimes L_G$ = joint Laplacian
- $U_J = U_T \bigotimes U_G$ = joint Fourier transform eigenvectors
- $JFT(\mathbf{x}) = U_J^* \mathbf{x}$ where $\mathbf{x} = vec(X_{N \times T})$.



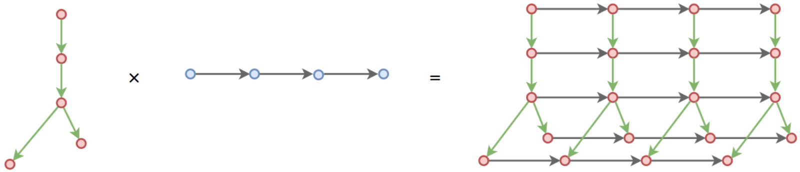
Joint Time-Vertex Wide-Sense Stationary Process

JWSS Random Processes (Loukas *et. al.*, 2016)

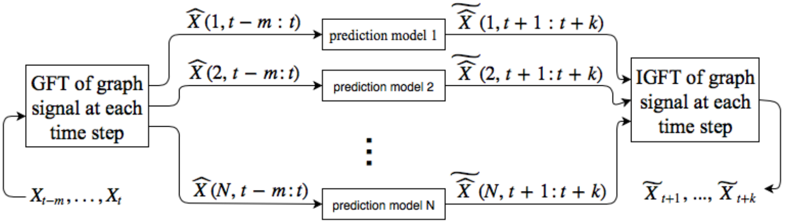
- Covariance matrix is jointly diagonizable with L_J

JWSS Random Processes (Loukas *et. al.*, 2016)

- Covariance matrix is jointly diagonizable with L_J
- $\mathbf{x} = h(L_J)\varepsilon$
  - $\varepsilon \sim \mathcal{D}(c, \mathbb{I}_{NT})$ and h is joint filter as a function of L_J
- $L_J\bar{\mathbf{x}} = \mathbf{0}_{N\times T}$ and $\Gamma(t_1, t_2) = \Gamma(1, 1 + t_2 - t_1) = \gamma_\tau(L_G)$
  - $\tau = t_2 - t_1$ and γ is a graph filter



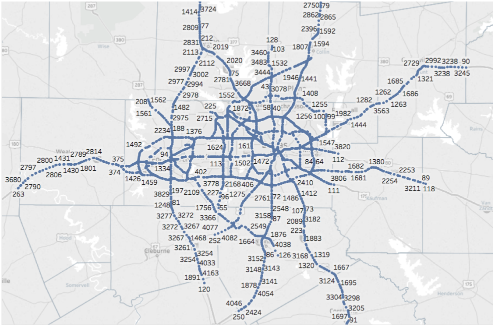
- Signal in frequency domain uncorrelated in each frequency
- GFT of signal at each time step - uncorrelated time series in frequency
- Independent prediction models for time series at each frequency



Peeking At The Dataset

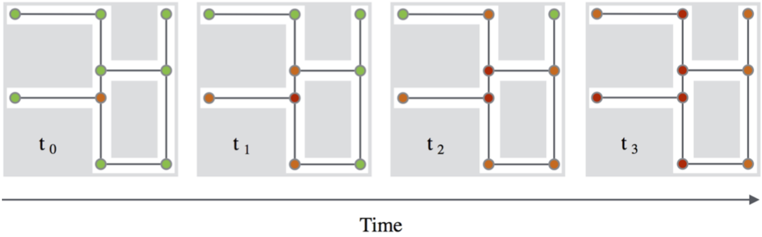
TEXAS A&M UNIVERSITY

- Line graph of network topology $\Rightarrow$ consistent with GSP



- Graph is too big ! $\Rightarrow$ not stationary
  - Idea : Cluster the graph into smaller stationary subgraphs $\Rightarrow$ how ?
  - Separate predictive models for each cluster

- Principal patterns on graph : spreading patterns of congestion in transportation network - spatial relation

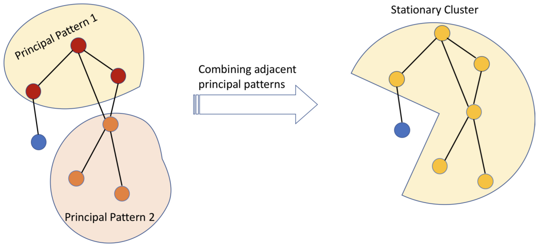


- principal patterns are almost stationary subgraphs
- Idea : use principal patterns to cluster the graph

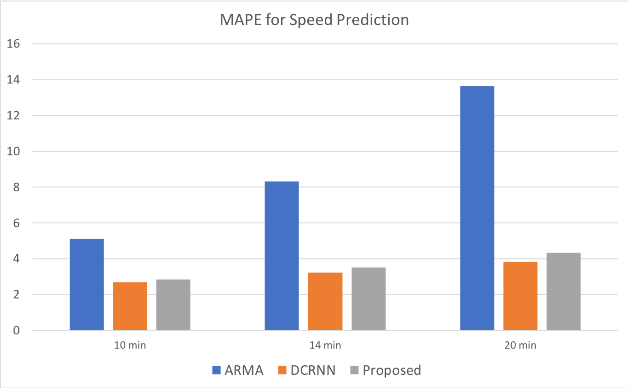
Extracting Principal Patterns from Historical Data

- Principal patterns : Weakly connected components of joint graph when non-congested nodes are removed
- A road is congested if Travel Time Index (TTI) goes beyond a threshold

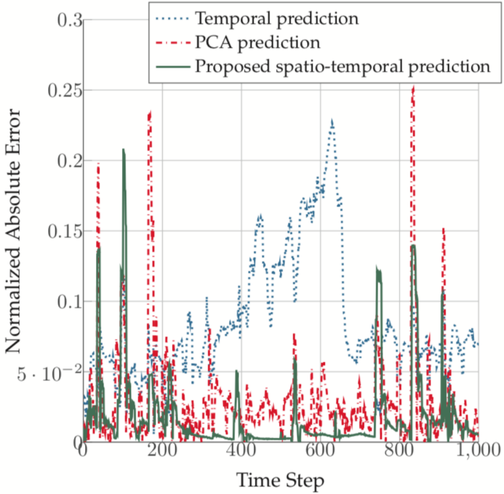
$$\text{Travel Time Index} = \frac{\text{Current Travel Time of the Road}}{\text{Free Flow Travel Time of the Road}}$$



Numerical Results



Numerical Results, cont.



- Static graph embedding using variational graph autoencoders
- Dynamic link prediction using variational recurrent graph autoencoders

Thanks !