Inferring Time-Varying Signals over Uncertain Graphs

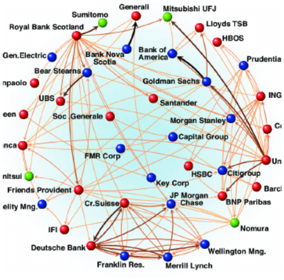
Mohammad Sabbaqi and Elvin Isufi
{m.sabbaqi, e.isufi-1}@tudelft.nl

TU Delft

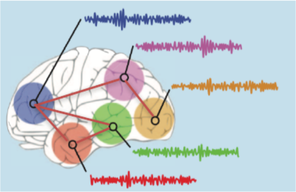
June 25, 2024

ICASSP 2024

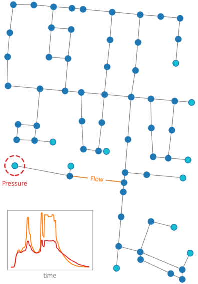
Multivariate Time Series over Networks



Financial networks



Brain recordings



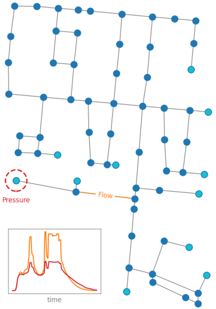
Water networks

- Data have now a spatial dependency and a temporal dependency

Delft University of Technology

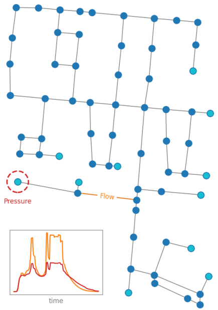
Graphs for Multivariate Time Series

- Graphs can be used to model multivariate time series
  - ⇒ The structure is a graph
  - ⇒ Time series are time-varying data assigned to the nodes



Graphs for Multivariate Time Series

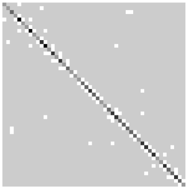
- Graphs can be used to model multivariate time series
  - ⇒ The structure is a graph
  - ⇒ Time series are time-varying data assigned to the nodes



- Nodes: junctions/sources in the city $\mathcal{V}$
- Edges: pipes connecting the junctions $\mathcal{E}$
- Data: recorded pressure in each junction varying by time

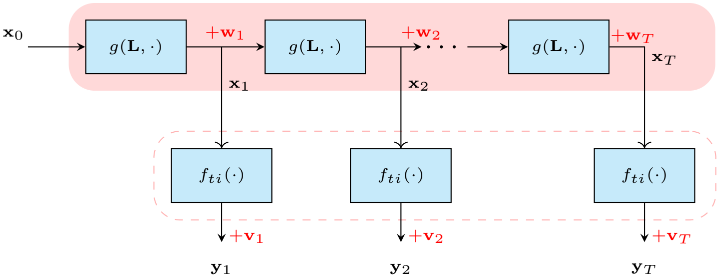
  ⇒ $\mathbf{x}_t$ over the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$

Shift op. ($\mathbf{A}$ or $\mathbf{L}$)



State Space Model for Time-Varying Graph Signals
equations

$$\begin{cases} \mathbf{x}_t = g(\mathbf{L}, \mathbf{x}_{t-1}) + \mathbf{w}_{t-1} & \rightarrow \text{state equation} \\ \mathbf{y}_t = f_t(\mathbf{L}, \mathbf{x}_t) + \mathbf{v}_t & \rightarrow \text{observation equation} \end{cases}$$

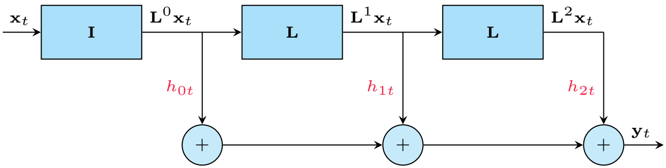


Delft University of Technology

Observation Model: Graph Filter

- We consider a graph filter at each time t for observation model
- Graph filter: shift-and-sum over the graph

$$\mathbf{y}_t = \sum_{k=0}^{K} h_{kt} \mathbf{L}^k \mathbf{x}_t = \mathbf{H}_t(\mathbf{L})\mathbf{x}_t$$



- Goal: Learn h_{kt} to have a simple evolution in the state.

State Equation: Stochastic PDE over Graphs

- Strict PDE in the state: $\mathbf{H}_t(\mathbf{L})$ might not exist
  - $\Rightarrow$ Hence, we allow some uncertainty in the state equation

State Equation: Stochastic PDE over Graphs

- Strict PDE in the state: $\mathbf{H}_t(\mathbf{L})$ might not exist
  - $\Rightarrow$ Hence, we allow some uncertainty in the state equation

$$d\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t dt + \mathbf{S}d\boldsymbol{\beta}_t$$

- $\boldsymbol{\beta}_t \in \mathbb{R}^F$ is a standard Wiener process (a.k.a Brownian motion)
- $\mathbf{S} \in \mathbb{R}^{N \times F}$ is the dispersion matrix

State Equation: Stochastic PDE over Graphs

- Strict PDE in the state: $\mathbf{H}_t(\mathbf{L})$ might not exist

  $\Rightarrow$ Hence, we allow some uncertainty in the state equation

$$d\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t dt + \mathbf{S} d\boldsymbol{\beta}_t$$

- $\boldsymbol{\beta}_t \in \mathbb{R}^F$ is a standard Wiener process (a.k.a Brownian motion)
- $\mathbf{S} \in \mathbb{R}^{N \times F}$ is the dispersion matrix

- Goal: Learn S $\rightarrow$ has too many parameters

  $\Rightarrow$ We parameterize it by graph structure

$$\mathbf{S} = \mathbf{B}\text{diag}(\boldsymbol{\alpha}) \longrightarrow d\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t dt + \mathbf{B}\text{diag}(\boldsymbol{\alpha}) d\boldsymbol{\beta}_t$$

- $\mathbf{B}$ is the incidence matrix

TU Delft Delft University of Technology

Problem Formulation

- Continuous-discrete state space model:

$$\begin{cases} d\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t dt + \mathbf{B}\mathrm{diag}(\boldsymbol{\alpha})d\boldsymbol{\beta}_t \\ \mathbf{y}_t = \mathbf{H}_t(\mathbf{L})\mathbf{x}_t + \mathbf{v}_t \end{cases}$$

Problem Formulation

- Continuous-discrete state space model:

$$\begin{cases} d\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t dt + \mathbf{B}\text{diag}(\boldsymbol{\alpha})d\boldsymbol{\beta}_t \\ \mathbf{y}_t = \mathbf{H}_t(\mathbf{L})\mathbf{x}_t + \mathbf{v}_t \end{cases}$$

- initial state value: $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
- noise energy: $\mathbb{E}[\mathbf{v}_k\mathbf{v}_k^\top] = \sigma^2\mathbf{I}$

TU Delft Delft University of Technology

Problem Formulation

- Continuous-discrete state space model:

$$\begin{cases} d\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t dt + \mathbf{B}\text{diag}(\boldsymbol{\alpha})d\boldsymbol{\beta}_t \\ \mathbf{y}_t = \mathbf{H}_t(\mathbf{L})\mathbf{x}_t + \mathbf{v}_t \end{cases}$$

- initial state value: $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
- noise energy: $\mathbb{E}[\mathbf{v}_k \mathbf{v}_k^\top] = \sigma^2 \mathbf{I}$

- Inference problem:
  - $\Rightarrow$ given: temporal observations $\mathbf{y}_1, \ldots, \mathbf{y}_T$
  - $\Rightarrow$ goal: estimate parameters $\boldsymbol{\alpha}$ and $\mathbf{h}_{kt}$'s

Delft University of Technology

Problem Formulation

- Continuous-discrete state space model:

$$\begin{cases} d\mathbf{x}_t = -c\mathbf{L}\mathbf{x}_t dt + \mathbf{B}\text{diag}(\boldsymbol{\alpha})d\boldsymbol{\beta}_t \\ \mathbf{y}_t = \mathbf{H}_t(\mathbf{L})\mathbf{x}_t + \mathbf{v}_t \end{cases}$$

- initial state value: $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
- noise energy: $\mathbb{E}[\mathbf{v}_k \mathbf{v}_k^\top] = \sigma^2 \mathbf{I}$

- Inference problem:
  - ⇒ given: temporal observations $\mathbf{y}_1, \ldots, \mathbf{y}_T$
  - ⇒ goal: estimate parameters $\boldsymbol{\alpha}$ and $\mathbf{h}_{kt}$'s

- Approach:

1. Discretization of state
2. Recovering the state
3. Estimating model parameters

TU Delft Delft University of Technology

Step1: Discretization of Continuous State

- Using the transition matrix of an LTI-SPDE:

$$\mathbf{x}_{t+\triangle t} = \tilde{\mathbf{L}}\mathbf{x}_t + \mathbf{q}_t$$

with:

$$\mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$
$$\tilde{\mathbf{L}} = \exp(-c\triangle t\mathbf{L})$$
$$\mathbf{Q} = \int_0^{\triangle t} e^{-c\mathbf{L}(\triangle t - s)}\mathbf{B}\mathrm{diag}^2(\boldsymbol{\alpha})\mathbf{B}^\top e^{-c\mathbf{L}(\triangle t - s)} ds$$

Step1: Discretization of Continuous State

- Using the transition matrix of an LTI-SPDE:

$$\mathbf{x}_{t+\triangle t} = \tilde{\mathbf{L}}\mathbf{x}_t + \mathbf{q}_t$$

with:

$$\mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$
$$\tilde{\mathbf{L}} = \exp(-c\triangle t\mathbf{L})$$
$$\mathbf{Q} = \int_0^{\triangle t} e^{-c\mathbf{L}(\triangle t - s)}\mathbf{B}\mathrm{diag}^2(\boldsymbol{\alpha})\mathbf{B}^\top e^{-c\mathbf{L}(\triangle t - s)} ds$$

- With first-order Taylor approximation:

$$\tilde{\mathbf{L}} \approx \mathbf{I} - c\triangle t\mathbf{L}$$
$$\mathbf{Q} \approx \triangle t\mathbf{B}\mathrm{diag}^2(\boldsymbol{\alpha})\mathbf{B}^\top$$

Step2: Recovering the State

Kalman-Bucy filtering/smoothing

- Given model parameters $\boldsymbol{\alpha}$ and $\mathbf{H}_t$: state recovery via Kalman filtering

  $\Rightarrow$ optimum Bayesian solution (recursive closed form $\rightarrow$ linear complexity in time)

Step2: Recovering the State
Kalman-Bucy filtering/smoothing

- Given model parameters $\boldsymbol{\alpha}$ and $\mathbf{H}_t$: state recovery via Kalman filtering

  $\Rightarrow$ optimum Bayesian solution (recursive closed form $\rightarrow$ linear complexity in time)

prediction step:

$$\mathbf{x}_t^{t-1} = \tilde{\mathbf{L}}\mathbf{x}_{t-1}^{t-1};$$
$$\mathbf{P}_t^{t-1} = \tilde{\mathbf{L}}\mathbf{P}_{t-1}^{t-1}\tilde{\mathbf{L}} + \mathbf{B}\mathrm{diag}^2(\boldsymbol{\alpha})\mathbf{B}^\top;$$

correction step:

$$\mathbf{K}_t = \mathbf{P}_t^{t-1}\mathbf{H}_t{}^\top(\mathbf{H}_t\mathbf{P}_t^{t-1}\mathbf{H}_t{}^\top + \sigma^2\mathbf{I})^{-1}$$
$$\mathbf{x}_t^t = \mathbf{x}_t^{t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^{t-1});$$
$$\mathbf{P}_t^t = \mathbf{P}_t^{t-1} - \mathbf{K}_t\mathbf{P}_t^{t-1}\mathbf{K}_t^\top;$$

- $\mathbf{K}_t$ is the Kalman gain
- $(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^{t-1})$ is the prediction error
- $\mathbf{P}_t$ is the covariance matrix of the state at time t

TU Delft Delft University of Technology

Step3: Estimating Model Parameters

a maximum a priori approach

- Having the state via Kalman filter → likelihood on the observation
  ⇒ can be even computed recursively!

$$\mathcal{L}_t(\boldsymbol{\alpha}, \mathbf{h}_t) = \mathcal{L}_{t-1}(\boldsymbol{\alpha}, \mathbf{h}_t) + \frac{1}{2}\log|\mathbf{S}_t| + \frac{1}{2}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^{t-1})^\top \mathbf{S}_t^{-1}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^{t-1})$$

where: $\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_t^{t-1}\mathbf{H}_t{}^\top + \sigma^2\mathbf{I}$

Delft University of Technology

Step3: Estimating Model Parameters

a maximum a priori approach

- Having the state via Kalman filter → likelihood on the observation
  - ⇒ can be even computed recursively!

$$\mathcal{L}_t(\boldsymbol{\alpha}, \mathbf{h}_t) = \mathcal{L}_{t-1}(\boldsymbol{\alpha}, \mathbf{h}_t) + \frac{1}{2}\log|\mathbf{S}_t| + \frac{1}{2}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^{t-1})^\top \mathbf{S}_t^{-1}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^{t-1})$$

where: $\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_t^{t-1}\mathbf{H}_t{}^\top + \sigma^2\mathbf{I}$

- error matters more when confidence is high!
- no l_2-norm risk anymore → covariance-based norm replaced
- gradient descent and inference is over!

Experiments

- **Datasets:**
  - ⇒ Synthetic data: based on presented state space model (200 nodes, 10000 samples)
  - ⇒ Weather data: NOAA (109 nodes, ∼8500 samples)
  - ⇒ Traffic data: METR-LA (207 nodes, ∼28000 samples)

Experiments

- **Datasets:**
  - ⇒ Synthetic data: based on presented state space model (200 nodes, 10000 samples)
  - ⇒ Weather data: NOAA (109 nodes, $\sim$8500 samples)
  - ⇒ Traffic data: METR-LA (207 nodes, $\sim$28000 samples)

- **Tasks:**
  - ⇒ Interpolation: randomly removing data over nodes at each time
  - ⇒ Extrapolation: forecasting time series, using state equation

Interpolation
data imputation

Table: Interpolation task performance for both synthetic and weather temperature dataset. The experiments are performed based on different portions of unobserved data.

rNMSE	Synthetic			Weather			Traffic		
	10%	20%	30%	10%	20%	30%	10%	20%	30%
LMS	0.40	0.46	0.46	0.42	0.43	0.49	0.41	0.45	0.48
StarGP	0.31	0.31	0.36	0.25	0.24	0.31	0.21	0.25	0.29
G-SPDE	**0.12**	**0.14**	**0.16**	**0.13**	**0.14**	**0.17**	**0.16**	**0.15**	**0.27**
No-$\boldsymbol{\alpha}$	0.24	0.27	0.30	0.23	0.28	0.31	0.27	0.33	0.37
Fixed-$\boldsymbol{\alpha}$	0.24	0.26	0.29	0.22	0.26	0.29	0.24	0.27	0.36
Learn-$\mathbf{S}$	0.23	0.20	0.20	0.21	0.21	0.27	0.19	0.22	0.34

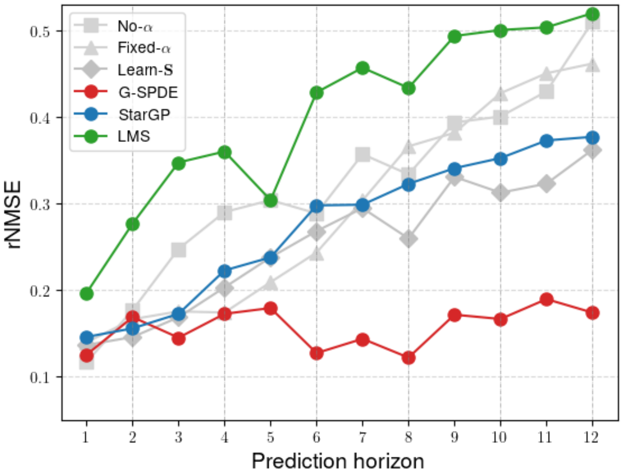


Figure. Traffic forecasting performance in rNMSE for proposed models with different prediction horizons.

Thanks for your attention!