THE UNIVERSITY *of* EDINBURGH

State-space models as graphs

Víctor Elvira
School of Mathematics
University of Edinburgh

Joint work with E. Chouzenoux (INRIA Saclay, France),
B. Cox (University of Edinburgh, UK), and G. Camps-Valls
(University of Valencia, Spain)

Delft University of Technology, June 22, 2026

Forecasting tipping points (probabilistically)

- Funding call for forecasting tipping points by Advanced Research and Invention Agency (ARIA):
  - ARIA: new UK DHARPA-style agency created to tackle high-risk, high-reward societal problems.
- The programme: 27 projects invited to work together (from sensors development and deployment to models/methods/theory).

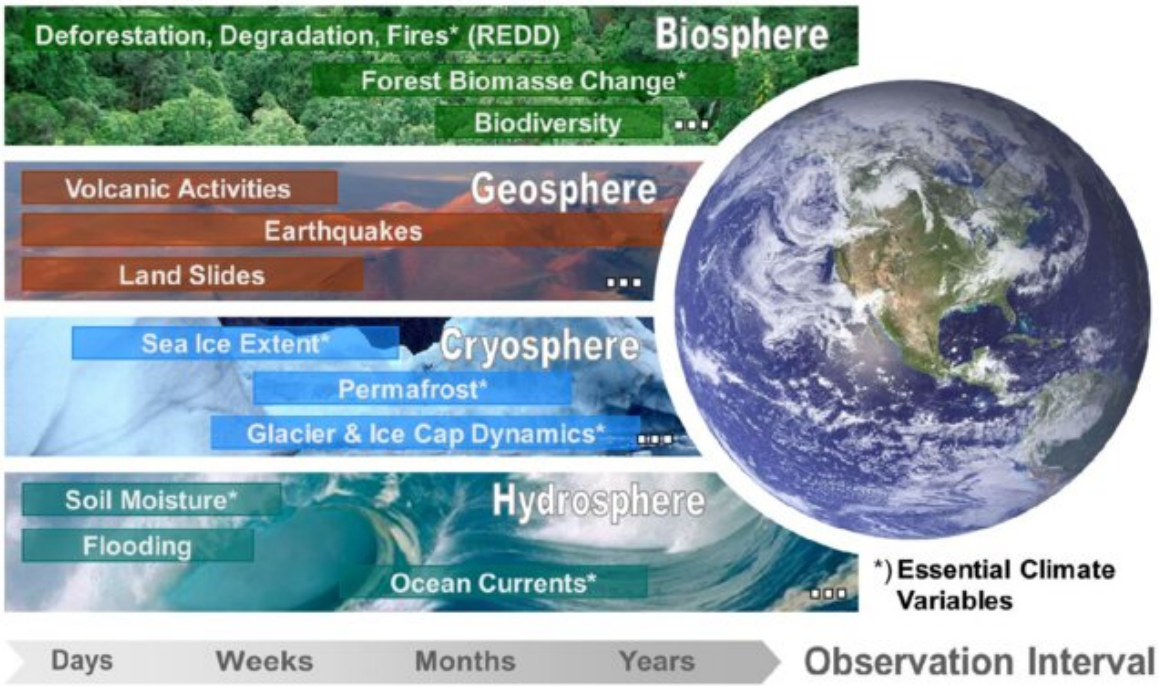
Forecasting tipping points (probabilistically)

- My project: probabilistic forecasting of tipping points
  - bring statistical/UQ/causal/signal processing tools to conceptual/box models (state dimension: 10–10^2) to global climate models (state dimension: 10^7–10^9)
- Filtering, tracking, importance sampling for rare events, prior design, and ensemble methods.
- Milestone 9: "The PI of this project is committed to engaging researchers from methodological fields (such as statistics, signal processing, machine learning, and applied mathematics) to contribute to climate science."

Outline

Dynamical systems

- **Dynamical system** is a set of elementary **units** whose evolution depends on their **local features** and **interactions over time**.[1]
  - The Earth is formed by dynamical subsystems interacting at different scales in time and space (e.g., biosphere, atmosphere, etc.)



[1] D. J. Watts and S. H. Strogatz. "Collective dynamics of small-world networks". In: *Nature* 393.6684 (1998), pp. 440–442.

Dynamical systems

- **Dynamical system** is a set of elementary **units** whose evolution depends on their **local features** and **interactions over time**.[1]
  - Omnipresent in science and engineering.
    - Earth and its geophysical systems (atmosphere, oceans)
    - biomedicine, e.g., heart electro-dynamics, brain
    - population ecology (prey-predator interactions)
    - robotics with target tracking, positioning, navigation
    - financial markets
    - wireless communications...

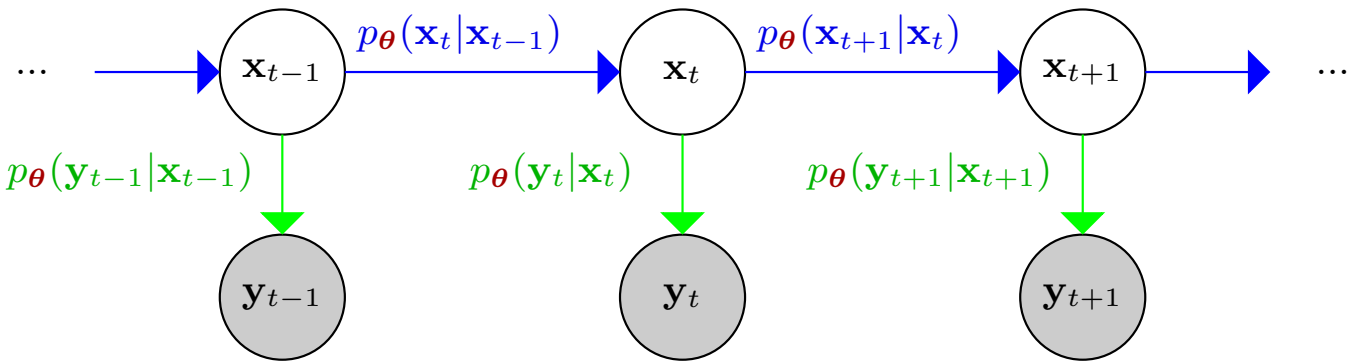
[1] D. J. Watts and S. H. Strogatz. "Collective dynamics of small-world networks". In: *Nature* 393.6684 (1998), pp. 440–442.

Motivation

- Dynamical systems:
  - dynamics governed by some system laws (generally unknown)
  - observed only partially (in space and time)
- Goals:
  - **understanding** (causal) connections among complicated phenomena
  - **predicting** the future, reconstructing the past
- Methodological approach:
  1. **model** those complex systems through probabilistic, parametric models,
  2. **process** observed time-series data to **estimate** unknowns
- statistics, numerical analysis, machine learning, signal processing, ... AI?

1. Modeling: State-Space Models (SSM)

- Evolving **hidden states** $\mathbf{x}_t \in \mathbb{R}^{N_x}$, $t = 1, ..., T$.
  - it captures the state of the system
  - it allows to describe its dynamics
- **Time-series data** $\mathbf{y}_t \in \mathbb{R}^{N_y}$, $t = 1, ..., T$:
  - noisy and partial version of the system state



- Summary of Markovian SSM:
  - state model $\rightarrow$ $p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1})$
  - observation model $\rightarrow$ $p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t)$

2. Estimation/inference problems

- (Sequentially) observe data $\mathbf{y}_t$ related to the hidden state $\mathbf{x}_t$.
  - $\mathbf{y}_{1:t} \equiv \{\mathbf{y}_1, ..., \mathbf{y}_t\}$.

- Bayesian/probabilistic inference:
  - compute/approximate posterior pdfs of unknowns

- Task: predict/estimate unknowns
  - **Filtering**: $p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{y}_{1:t})$
  - **Smoothing**: $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-\tau}|\mathbf{y}_{1:t}), \qquad \tau \geq 1$
  - Prediction:
    - State prediction: $p_{\boldsymbol{\theta}}(\mathbf{x}_{t+\tau}|\mathbf{y}_{1:t}), \qquad \tau \geq 1$
    - Observation prediction: $p_{\boldsymbol{\theta}}(\mathbf{y}_{t+\tau}|\mathbf{y}_{1:t}), \qquad \tau \geq 1$
  - estimation of model parameters

The linear-Gaussian model

- The linear-Gaussian model (LG-SSM) is arguably the most relevant SSM
  - *Functional* notation:
    - Unobserved state $\rightarrow$ $\mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{q}_t$
    - Observations $\rightarrow$ $\mathbf{y}_t = \boldsymbol{H}_t \mathbf{x}_t + \mathbf{r}_t$

      where $\mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ and $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t)$.
  - *Probabilistic* notation:
    - Hidden state $\rightarrow$ $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \equiv \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t \mathbf{x}_{t-1}, \mathbf{Q}_t)$
    - Observations $\rightarrow$ $p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t \mathbf{x}_t, \mathbf{R}_t)$
- Methods (known θ):
  - **Kalman filter**: obtains the filtering pdfs $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ at each t
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
- Methods (unknown θ; build upon KF/RTS):
  - Point-wise:
    - maximum likelihood (ML)
    - maximum a posteriori (MAP)

    Optimization via:
    - gradient-based methods
    - expectation-maximization (EM)
  - fully Bayesian Monte Carlo methods[2]
    - particle Metropolis
    - particle Gibbs
    - ...

[2] C. Andrieu, A. Doucet, and R. Holenstein. "Particle markov chain monte carlo methods". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 72.3 (2010), pp. 269–342.

The linear-Gaussian model

- The linear-Gaussian model (LG-SSM) is arguably the most relevant SSM
  - *Functional* notation:
    - Unobserved state $\rightarrow$ $\mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{q}_t$
    - Observations $\rightarrow$ $\mathbf{y}_t = \boldsymbol{H}_t \mathbf{x}_t + \mathbf{r}_t$

      where $\mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ and $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t)$.
  - *Probabilistic* notation:
    - Hidden state $\rightarrow$ $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \equiv \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t \mathbf{x}_{t-1}, \mathbf{Q}_t)$
    - Observations $\rightarrow$ $p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t \mathbf{x}_t, \mathbf{R}_t)$
- Methods (known $\boldsymbol{\theta}$):
  - **Kalman filter**: obtains the filtering pdfs $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ at each t
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - maximum likelihood (ML)
    - maximum a posteriori (MAP)

    Optimization via:
    - gradient-based methods
    - expectation-maximization (EM)
  - fully Bayesian Monte Carlo methods[2]
    - particle Metropolis
    - particle Gibbs
    - ...

[2] C. Andrieu, A. Doucet, and R. Holenstein. "Particle markov chain monte carlo methods". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 72.3 (2010), pp. 269–342.

The linear-Gaussian model

- The linear-Gaussian model (LG-SSM) is arguably the most relevant SSM
  - *Functional* notation:
    - Unobserved state $\rightarrow \quad \mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{q}_t$
    - Observations $\rightarrow \quad \mathbf{y}_t = \boldsymbol{H}_t \mathbf{x}_t + \mathbf{r}_t$

      where $\mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ and $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t)$.
  - *Probabilistic* notation:
    - Hidden state $\rightarrow \quad p(\mathbf{x}_t|\mathbf{x}_{t-1}) \equiv \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t \mathbf{x}_{t-1}, \mathbf{Q}_t)$
    - Observations $\rightarrow \quad p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t \mathbf{x}_t, \mathbf{R}_t)$
- Methods (known $\boldsymbol{\theta}$):
  - **Kalman filter**: obtains the filtering pdfs $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ at each t
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - maximum likelihood (ML)
    - maximum a posteriori (MAP)

    Optimization via:
    - gradient-based methods
    - expectation-maximization (EM)
  - fully Bayesian Monte Carlo methods[2]
    - particle Metropolis
    - particle Gibbs
    - ...

[2] C. Andrieu, A. Doucet, and R. Holenstein. "Particle markov chain monte carlo methods". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 72.3 (2010), pp. 269–342.

The linear-Gaussian model

- The linear-Gaussian model (LG-SSM) is arguably the most relevant SSM
  - *Functional* notation:
    - Unobserved state $\rightarrow \quad \mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{q}_t$
    - Observations $\rightarrow \quad \mathbf{y}_t = \boldsymbol{H}_t \mathbf{x}_t + \mathbf{r}_t$

      where $\mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ and $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t)$.
  - *Probabilistic* notation:
    - Hidden state $\rightarrow \quad p(\mathbf{x}_t|\mathbf{x}_{t-1}) \equiv \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t \mathbf{x}_{t-1}, \mathbf{Q}_t)$
    - Observations $\rightarrow \quad p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t \mathbf{x}_t, \mathbf{R}_t)$
- Methods (known $\boldsymbol{\theta}$):
  - **Kalman filter**: obtains the filtering pdfs $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ at each t
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - maximum likelihood (ML)
    - maximum a posteriori (MAP)

    Optimization via:
    - gradient-based methods
    - expectation-maximization (EM)
  - fully Bayesian Monte Carlo methods[2]
    - particle Metropolis
    - particle Gibbs
    - ...

[2] C. Andrieu, A. Doucet, and R. Holenstein. "Particle markov chain monte carlo methods". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 72.3 (2010), pp. 269–342.

The linear-Gaussian model

- The linear-Gaussian model (LG-SSM) is arguably the most relevant SSM
  - *Functional* notation:
    - Unobserved state $\rightarrow \quad \mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{q}_t$
    - Observations $\rightarrow \quad \mathbf{y}_t = \boldsymbol{H}_t \mathbf{x}_t + \mathbf{r}_t$

      where $\mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ and $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t)$.
  - *Probabilistic* notation:
    - Hidden state $\rightarrow \quad p(\mathbf{x}_t|\mathbf{x}_{t-1}) \equiv \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t \mathbf{x}_{t-1}, \mathbf{Q}_t)$
    - Observations $\rightarrow \quad p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t \mathbf{x}_t, \mathbf{R}_t)$
- Methods (known $\boldsymbol{\theta}$):
  - **Kalman filter**: obtains the filtering pdfs $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ at each t
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - maximum likelihood (ML)
    - maximum a posteriori (MAP)

    Optimization via:
    - gradient-based methods
    - expectation-maximization (EM)
  - fully Bayesian Monte Carlo methods[2]
    - particle Metropolis
    - particle Gibbs
    - ...

[2] C. Andrieu, A. Doucet, and R. Holenstein. "Particle markov chain monte carlo methods". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 72.3 (2010), pp. 269–342.

Kalman Filter: prediction step

Goal is to go from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ and $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ in two steps:

1. **Prediction** step (marginalization of Gaussian):

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}$$

- Suppose that filtered distribution at $t-1$ is Gaussian $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \equiv \mathcal{N}(\mathbf{m}_{t-1}, \mathbf{P}_{t-1})$.
- Predictive distribution is also Gaussian $p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \equiv \mathcal{N}(\mathbf{m}_t^-, \mathbf{P}_t^-)$
  - Mean: $\mathbf{m}_t^- = \mathbf{A}_t\mathbf{m}_{t-1}$
  - Variance: $\mathbf{P}_t^- = \mathbf{A}_t\mathbf{P}_{t-1}\mathbf{A}_t^T + \mathbf{Q}_t$

- Interpretation:
  - The mean is projected through matrix $\mathbf{A}_t$
  - The **uncertainty** is propagated too through $\mathbf{A}_t$, plus the variance of the process noise

Kalman Filter: update step

2. **Update** step (product of Gaussians):

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}$$

- The filtered distribution at time t is also Gaussian $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \equiv \mathcal{N}(\mathbf{m}_t, \mathbf{P}_t)$
  - Mean: $\mathbf{m}_t = \mathbf{m}_t^- + \boldsymbol{K}_t \left(\mathbf{y}_t - \boldsymbol{H}_t \mathbf{m}_t^-\right)$
  - Variance: $\mathbf{P}_t = (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t)\, \mathbf{P}_t^-$

  where $\boldsymbol{K}_t = \mathbf{P}_t^- \boldsymbol{H}_t^T \left(\boldsymbol{H}_t \mathbf{P}_t^- \boldsymbol{H}_t^T + \mathbf{R}_t\right)^{-1}$ is the optimal Kalman gain.

- Interpretation:
  - The mean is corrected w.r.t. the predictive in the direction of the residual/error.
  - The variance is propagated by $\boldsymbol{H}_t$ and divided by the covariance of the residual/error.

Kalman summary and RTS smoother

- Hidden state $\rightarrow$ $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \equiv \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t\mathbf{x}_{t-1}, \mathbf{Q}_t)$
- Observations $\rightarrow$ $p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t\mathbf{x}_t, \mathbf{R}_t)$

Kalman filter

- Initialize: $\mathbf{m}_0$, $\mathbf{P}_0$
- For $t = 1, \ldots, T$

  Predict stage:

$$\mathbf{x}_t^- = \mathbf{A}_t\mathbf{m}_{t-1}$$
$$\mathbf{P}_t^- = \mathbf{A}_t\mathbf{P}_{t-1}\mathbf{A}_t^\top + \mathbf{Q}_t$$

  Update stage:

$$\mathbf{z}_t = \mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^-$$
$$\mathbf{S}_t = \mathbf{H}\mathbf{P}_t^-\mathbf{H}_t^\top + \mathbf{R}_t$$
$$\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}_t^\top\mathbf{S}_t^{-1}$$
$$\mathbf{m}_t = \mathbf{x}_t^- + \mathbf{K}_t\mathbf{z}_t$$
$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^\top$$

RTS smoother

- For $t = T, \ldots, 1$

  Smoothing stage:

$$\mathbf{x}_{t+1}^- = \mathbf{A}_t\mathbf{m}_t$$
$$\mathbf{P}_{t+1}^- = \mathbf{A}_t\mathbf{P}_t\mathbf{A}_t^\top + \mathbf{Q}_t$$
$$\mathbf{G}_t = \mathbf{P}_t\mathbf{A}_t^\top(\mathbf{P}_{t+1}^-)^{-1}$$
$$\mathbf{m}_t^s = \mathbf{m}_t + \mathbf{G}_t(\mathbf{m}_{t+1}^s - \mathbf{x}_{t+1}^-)$$
$$\mathbf{P}_t^s = \mathbf{P}_t + \mathbf{G}_t(\mathbf{P}_{t+1}^s - \mathbf{P}_{t+1}^-)\mathbf{G}_t^\top$$

- ✓ Filtering distribution: $p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{P}_t)$
- ✓ Smoothing distribution: $p(\mathbf{x}_t|\mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_t^s, \mathbf{P}_t^s)$
- ✗ How to proceed if model parameters are unknown ?
  - we consider:
    - known $\mathbf{H}_t$ and $\mathbf{R}_t$
    - constant and unknown $\mathbf{A}_t = \mathbf{A}$ and $\mathbf{Q}_t = \mathbf{Q}$ $\Rightarrow$ estimate $\theta = [\mathbf{A}; \mathbf{Q}]$

Kalman summary and RTS smoother

- Hidden state $\rightarrow$ $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \equiv \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t\mathbf{x}_{t-1}, \mathbf{Q}_t)$
- Observations $\rightarrow$ $p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t\mathbf{x}_t, \mathbf{R}_t)$

Kalman filter

- Initialize: $\mathbf{m}_0$, $\mathbf{P}_0$
- For $t = 1, \ldots, T$

  Predict stage:

$$\mathbf{x}_t^- = \mathbf{A}_t\mathbf{m}_{t-1}$$
$$\mathbf{P}_t^- = \mathbf{A}_t\mathbf{P}_{t-1}\mathbf{A}_t^\top + \mathbf{Q}_t$$

  Update stage:

$$\mathbf{z}_t = \mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^-$$
$$\mathbf{S}_t = \mathbf{H}\mathbf{P}_t^-\mathbf{H}_t^\top + \mathbf{R}_t$$
$$\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}_t^\top\mathbf{S}_t^{-1}$$
$$\mathbf{m}_t = \mathbf{x}_t^- + \mathbf{K}_t\mathbf{z}_t$$
$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^\top$$

RTS smoother

- For $t = T, \ldots, 1$

  Smoothing stage:

$$\mathbf{x}_{t+1}^- = \mathbf{A}_t\mathbf{m}_t$$
$$\mathbf{P}_{t+1}^- = \mathbf{A}_t\mathbf{P}_t\mathbf{A}_t^\top + \mathbf{Q}_t$$
$$\mathbf{G}_t = \mathbf{P}_t\mathbf{A}_t^\top(\mathbf{P}_{t+1}^-)^{-1}$$
$$\mathbf{m}_t^s = \mathbf{m}_t + \mathbf{G}_t(\mathbf{m}_{t+1}^s - \mathbf{x}_{t+1}^-)$$
$$\mathbf{P}_t^s = \mathbf{P}_t + \mathbf{G}_t(\mathbf{P}_{t+1}^s - \mathbf{P}_{t+1}^-)\mathbf{G}_t^\top$$

- ✓ Filtering distribution: $p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{P}_t)$
- ✓ Smoothing distribution: $p(\mathbf{x}_t|\mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_t^s, \mathbf{P}_t^s)$
- ✗ How to proceed if model parameters are unknown ?
  - we consider:
    - known $\mathbf{H}_t$ and $\mathbf{R}_t$
    - constant and unknown $\mathbf{A}_t = \mathbf{A}$ and $\mathbf{Q}_t = \mathbf{Q}$ $\Rightarrow$ estimate $\theta = [\mathbf{A}; \mathbf{Q}]$

Kalman summary and RTS smoother

- Hidden state $\rightarrow$ $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \equiv \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t\mathbf{x}_{t-1}, \mathbf{Q}_t)$
- Observations $\rightarrow$ $p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t\mathbf{x}_t, \mathbf{R}_t)$

Kalman filter

- Initialize: $\mathbf{m}_0$, $\mathbf{P}_0$
- For $t = 1, \ldots, T$

  Predict stage:

$$\mathbf{x}_t^- = \mathbf{A}_t\mathbf{m}_{t-1}$$
$$\mathbf{P}_t^- = \mathbf{A}_t\mathbf{P}_{t-1}\mathbf{A}_t^\top + \mathbf{Q}_t$$

  Update stage:

$$\mathbf{z}_t = \mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^-$$
$$\mathbf{S}_t = \mathbf{H}\mathbf{P}_t^-\mathbf{H}_t^\top + \mathbf{R}_t$$
$$\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}_t^\top\mathbf{S}_t^{-1}$$
$$\mathbf{m}_t = \mathbf{x}_t^- + \mathbf{K}_t\mathbf{z}_t$$
$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^\top$$

RTS smoother

- For $t = T, \ldots, 1$

  Smoothing stage:

$$\mathbf{x}_{t+1}^- = \mathbf{A}_t\mathbf{m}_t$$
$$\mathbf{P}_{t+1}^- = \mathbf{A}_t\mathbf{P}_t\mathbf{A}_t^\top + \mathbf{Q}_t$$
$$\mathbf{G}_t = \mathbf{P}_t\mathbf{A}_t^\top(\mathbf{P}_{t+1}^-)^{-1}$$
$$\mathbf{m}_t^s = \mathbf{m}_t + \mathbf{G}_t(\mathbf{m}_{t+1}^s - \mathbf{x}_{t+1}^-)$$
$$\mathbf{P}_t^s = \mathbf{P}_t + \mathbf{G}_t(\mathbf{P}_{t+1}^s - \mathbf{P}_{t+1}^-)\mathbf{G}_t^\top$$

- ✓ Filtering distribution: $p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{P}_t)$
- ✓ Smoothing distribution: $p(\mathbf{x}_t|\mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_t^s, \mathbf{P}_t^s)$
- ✗ How to proceed if model parameters are unknown ?
  - we consider:
    - known $\mathbf{H}_t$ and $\mathbf{R}_t$
    - constant and unknown $\mathbf{A}_t = \mathbf{A}$ and $\mathbf{Q}_t = \mathbf{Q}$ $\Rightarrow$ estimate $\boldsymbol{\theta} = [\mathbf{A}; \mathbf{Q}]$

Outline

Goal of the talk

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{q}_t, \qquad \mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q})$$

This talk: modeling and inference approaches

- **Sparse graphical model** to represent (i) the (Granger) **causal dependencies** among the states, and (ii) the **correlation** among the state noises.
- **Algorithms** to estimate $\mathbf{A}$ and $\mathbf{Q}$

A graphical perspective on $\mathbf{A}$

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{q}_t, \qquad \mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q})$$

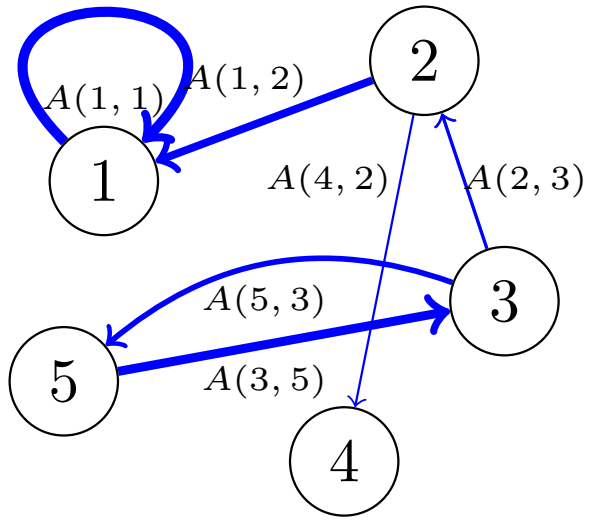
- $\mathbf{A}$ interpreted as a **sparse directed graph**

$$\mathbf{A} = \begin{pmatrix} 0.9 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & -0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 \\ 0 & -0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \end{pmatrix}$$

- $A(i,j)$ is the linear effect from node j at time $t-1$ to node i at time t:

$$x_{t,i} = \sum_{j=1}^{N_x} A(i,j) x_{t-1,j} + q_{t,i}$$

- $A(i,j) \neq 0 \Rightarrow x_{t-1,j}$ conditionally Granger-causes $x_{t,i}$.

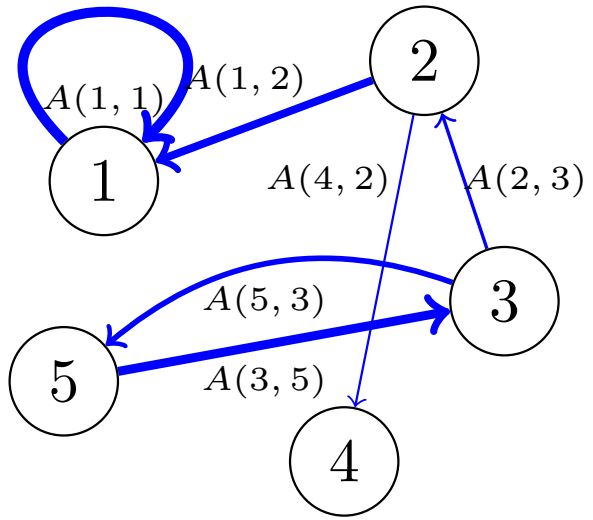


A graphical perspective on $\mathbf{A}$

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{q}_t, \qquad \mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q})$$

- $\mathbf{A}$ interpreted as a **sparse directed graph**

$$\mathbf{A} = \begin{pmatrix} 0.9 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & -0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 \\ 0 & -0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \end{pmatrix}$$



- $A(i,j)$ is the linear effect from node j at time $t-1$ to node i at time t:

$$x_{t,i} = \sum_{j=1}^{N_x} A(i,j) x_{t-1,j} + q_{t,i}$$

- $A(i,j) \neq 0 \Rightarrow x_{t-1,j}$ conditionally Granger-causes $x_{t,i}$.

Granger causality

Disclaimer: Granger causality is a statistical test to determine if one time series is useful to predict another one (controversial type of causality!)

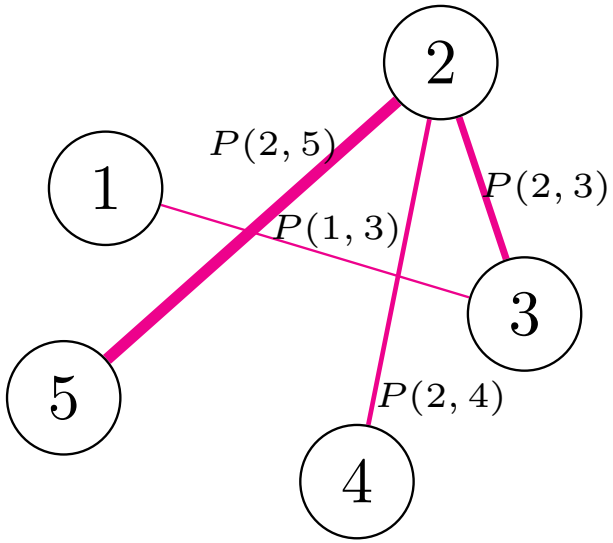
A graphical modeling $\mathbf{P} = \mathbf{Q}^{-1}$

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{q}_t, \qquad \mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q})$$

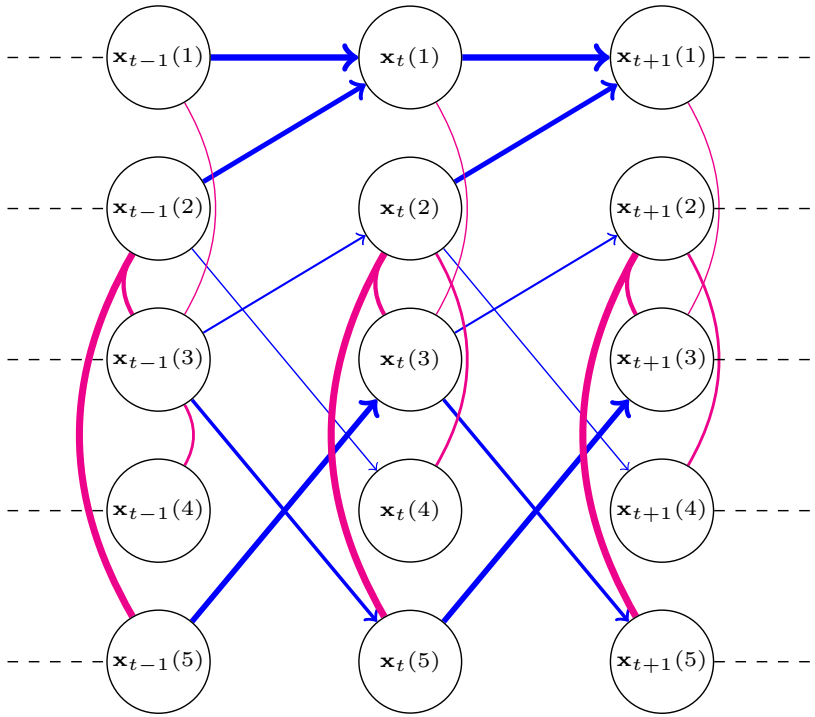
- $\mathbf{P} = \mathbf{Q}^{-1}$ interpreted as **sparse undirected graph** (Gaussian graphical models).

$$\mathbf{q}_t(n) \perp\!\!\!\perp \mathbf{q}_t(\ell) | \{\mathbf{q}_t(j), j \in 1, \ldots, N_x \backslash \{n, \ell\}\} \Longleftrightarrow P(n, \ell) = P(\ell, n) = 0.$$

$$\mathbf{P} = \mathbf{Q}^{-1} = \begin{pmatrix} 2 & 0 & -0.1 & 0 & 0 \\ 0 & 0.9 & 0.3 & -0.2 & 0.5 \\ -0.1 & 0.3 & 0.8 & 0 & 0 \\ 0 & -0.2 & 0 & 2 & 0 \\ 0 & 0.5 & 0 & 0 & 1.5 \end{pmatrix}$$



Summary of the graphical interpretation



Summary representation of the graphical model, for the example graphs $\mathbf{A}$ *and* $\mathbf{P}$ *from the two previous slides.*

DGLASSO (dynamic graphical lasso) algorithm: maximum a posteriori (MAP) estimator of $\mathbf{A}$ and $\mathbf{P}$ under **lasso sparsity regularization** on both matrices, given the observed sequence $\mathbf{y}_{1:T}$.

Outline

Proposed penalized formulation

Goal. MAP estimate of $\mathbf{A}$ and $\mathbf{P}$ ($\mathbf{P} = \mathbf{Q}^{-1}$):

$$\mathbf{A}^*, \mathbf{P}^* = \underset{\mathbf{A},\mathbf{P}}{\operatorname{argmax}} \; p(\mathbf{A}, \mathbf{P}|\mathbf{y}_{1:T}) = \underset{\mathbf{A}}{\operatorname{argmax}} \; p(\mathbf{A}, \mathbf{P}) p(\mathbf{y}_{1:T}|\mathbf{A}, \mathbf{P})$$

$$= \underset{\mathbf{A},\mathbf{P}}{\operatorname{argmin}} \; \underbrace{-\log p(\mathbf{A}, \mathbf{P})}_{\mathcal{L}_0(\mathbf{A},\mathbf{P})} \underbrace{-\log p(\mathbf{y}_{1:T}|\mathbf{A}, \mathbf{P})}_{\mathcal{L}_{1:T}(\mathbf{A},\mathbf{P})} = \mathcal{L}(\mathbf{A}, \mathbf{P})$$

1. Lasso penalty (prior): we promote **sparse matrices** $(\mathbf{A}, \mathbf{P})$ for **graph interpretability**:

$$\mathcal{L}_0(\mathbf{A}, \mathbf{P}) = \lambda_A \|\mathbf{A}\|_1 + \lambda_P \|\mathbf{P}\|_1,$$

2. log likelihood:

$$\mathcal{L}_{1:T}(\mathbf{A}, \mathbf{P}) = \sum_{t=1}^{T} \frac{1}{2} \log |2\pi \mathbf{S}_t(\mathbf{A}, \mathbf{P})| + \frac{1}{2} \mathbf{z}_t(\mathbf{A}, \mathbf{P})^\top \mathbf{S}_t(\mathbf{A}, \mathbf{P})^{-1} \mathbf{z}_t(\mathbf{A}, \mathbf{P}).$$

   - evaluation running KF with $(\mathbf{A}, \mathbf{P})$

Challenges:

- **Joint** minimization with **non-smooth** and **non-convex** loss.
- gradient-based solutions are challenging (unrolling KF recursion) and numerically unstable

Proposed penalized formulation

Goal. MAP estimate of $\mathbf{A}$ and $\mathbf{P}$ ($\mathbf{P} = \mathbf{Q}^{-1}$):

$$\mathbf{A}^*, \mathbf{P}^* = \underset{\mathbf{A},\mathbf{P}}{\operatorname{argmax}} \; p(\mathbf{A}, \mathbf{P}|\mathbf{y}_{1:T}) = \underset{\mathbf{A}}{\operatorname{argmax}} \; p(\mathbf{A}, \mathbf{P}) p(\mathbf{y}_{1:T}|\mathbf{A}, \mathbf{P})$$
$$= \underset{\mathbf{A},\mathbf{P}}{\operatorname{argmin}} \; \underbrace{-\log p(\mathbf{A}, \mathbf{P})}_{\mathcal{L}_0(\mathbf{A},\mathbf{P})} \underbrace{-\log p(\mathbf{y}_{1:T}|\mathbf{A}, \mathbf{P})}_{\mathcal{L}_{1:T}(\mathbf{A},\mathbf{P})} = \mathcal{L}(\mathbf{A}, \mathbf{P})$$

1. Lasso penalty (prior): we promote **sparse matrices** $(\mathbf{A}, \mathbf{P})$ for **graph interpretability**:
$$\mathcal{L}_0(\mathbf{A}, \mathbf{P}) = \lambda_A \|\mathbf{A}\|_1 + \lambda_P \|\mathbf{P}\|_1,$$

2. log likelihood:
$$\mathcal{L}_{1:T}(\mathbf{A}, \mathbf{P}) = \sum_{t=1}^{T} \frac{1}{2} \log |2\pi \mathbf{S}_t(\mathbf{A}, \mathbf{P})| + \frac{1}{2} \mathbf{z}_t(\mathbf{A}, \mathbf{P})^\top \mathbf{S}_t(\mathbf{A}, \mathbf{P})^{-1} \mathbf{z}_t(\mathbf{A}, \mathbf{P}).$$
   - evaluation running KF with $(\mathbf{A}, \mathbf{P})$

Challenges:

- **Joint** minimization with non-smooth and non-convex loss.
- gradient-based solutions are challenging (unrolling KF recursion) and numerically unstable

Proposed penalized formulation

Goal. MAP estimate of $\mathbf{A}$ and $\mathbf{P}$ ($\mathbf{P} = \mathbf{Q}^{-1}$):

$$\mathbf{A}^*, \mathbf{P}^* = \underset{\mathbf{A},\mathbf{P}}{\operatorname{argmax}} \; p(\mathbf{A}, \mathbf{P}|\mathbf{y}_{1:T}) = \underset{\mathbf{A}}{\operatorname{argmax}} \; p(\mathbf{A}, \mathbf{P}) p(\mathbf{y}_{1:T}|\mathbf{A}, \mathbf{P})$$
$$= \underset{\mathbf{A},\mathbf{P}}{\operatorname{argmin}} \; \underbrace{-\log p(\mathbf{A}, \mathbf{P})}_{\mathcal{L}_0(\mathbf{A},\mathbf{P})} \underbrace{-\log p(\mathbf{y}_{1:T}|\mathbf{A}, \mathbf{P})}_{\mathcal{L}_{1:T}(\mathbf{A},\mathbf{P})} = \mathcal{L}(\mathbf{A}, \mathbf{P})$$

1. Lasso penalty (prior): we promote **sparse matrices** $(\mathbf{A}, \mathbf{P})$ for **graph interpretability**:
$$\mathcal{L}_0(\mathbf{A}, \mathbf{P}) = \lambda_A \|\mathbf{A}\|_1 + \lambda_P \|\mathbf{P}\|_1,$$

2. log likelihood:
$$\mathcal{L}_{1:T}(\mathbf{A}, \mathbf{P}) = \sum_{t=1}^{T} \tfrac{1}{2} \log |2\pi \mathbf{S}_t(\mathbf{A}, \mathbf{P})| + \frac{1}{2} \mathbf{z}_t(\mathbf{A}, \mathbf{P})^\top \mathbf{S}_t(\mathbf{A}, \mathbf{P})^{-1} \mathbf{z}_t(\mathbf{A}, \mathbf{P}).$$
   - evaluation running KF with $(\mathbf{A}, \mathbf{P})$

Challenges:

- **Joint** minimization with non-smooth and non-convex loss.
- gradient-based solutions are challenging (unrolling KF recursion) and numerically unstable

Proposed penalized formulation

Goal. MAP estimate of $\mathbf{A}$ and $\mathbf{P}$ ($\mathbf{P} = \mathbf{Q}^{-1}$):

$$\mathbf{A}^*, \mathbf{P}^* = \underset{\mathbf{A},\mathbf{P}}{\operatorname{argmax}} \; p(\mathbf{A}, \mathbf{P}|\mathbf{y}_{1:T}) = \underset{\mathbf{A}}{\operatorname{argmax}} \; p(\mathbf{A}, \mathbf{P}) p(\mathbf{y}_{1:T}|\mathbf{A}, \mathbf{P})$$
$$= \underset{\mathbf{A},\mathbf{P}}{\operatorname{argmin}} \; \underbrace{-\log p(\mathbf{A}, \mathbf{P})}_{\mathcal{L}_0(\mathbf{A},\mathbf{P})} \underbrace{-\log p(\mathbf{y}_{1:T}|\mathbf{A}, \mathbf{P})}_{\mathcal{L}_{1:T}(\mathbf{A},\mathbf{P})} = \mathcal{L}(\mathbf{A}, \mathbf{P})$$

1. Lasso penalty (prior): we promote **sparse matrices** $(\mathbf{A}, \mathbf{P})$ for **graph interpretability**:
$$\mathcal{L}_0(\mathbf{A}, \mathbf{P}) = \lambda_A \|\mathbf{A}\|_1 + \lambda_P \|\mathbf{P}\|_1,$$

2. log likelihood:
$$\mathcal{L}_{1:T}(\mathbf{A}, \mathbf{P}) = \sum_{t=1}^{T} \tfrac{1}{2} \log |2\pi \mathbf{S}_t(\mathbf{A}, \mathbf{P})| + \frac{1}{2} \mathbf{z}_t(\mathbf{A}, \mathbf{P})^\top \mathbf{S}_t(\mathbf{A}, \mathbf{P})^{-1} \mathbf{z}_t(\mathbf{A}, \mathbf{P}).$$
   - evaluation running KF with $(\mathbf{A}, \mathbf{P})$

Challenges:

- **Joint** minimization with **non-smooth** and **non-convex** loss.
- gradient-based solutions are challenging (unrolling KF recursion) and numerically unstable

DGLASSO algorithm

- **DGLASSO:** A block alternating majorization-minimization algorithm:[3] Initialize $(\mathbf{A}^{(0)}, \mathbf{P}^{(0)})$, and at each iteration $i \in \mathbb{N}$,
  (a) Run RTS to build function $\mathcal{Q}(\mathbf{A}, \mathbf{P}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)})$ (E-step)
  (b) Update transition matrix (M-step):

$$\mathbf{A}^{(i+1)} = \underset{\mathbf{A}}{\operatorname{argmin}} \; \mathcal{Q}(\mathbf{A}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \lambda_A \|\mathbf{A}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A} - \mathbf{A}^{(i)}\|_F^2$$

  (c) Run RTS to build function $\mathcal{Q}(\mathbf{A}, \mathbf{P}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)})$ (E-step)
  (d) Update precision matrix (M-step):

$$\mathbf{P}^{(i+1)} = \underset{\mathbf{P}}{\operatorname{argmin}} \; \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \lambda_P \|\mathbf{P}\|_1 + \frac{1}{2\theta_P} \|\mathbf{P} - \mathbf{P}^{(i)}\|_F^2$$

- Proximal terms, with stepsizes $(\theta_A, \theta_P) > 0$ guarantee convergence of iterates to a critical point of $\mathcal{L}(\mathbf{A}, \mathbf{P})$.
- Convenient **bi-convex** structure of $\mathcal{Q}(\cdot, \cdot; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}})$:
  - step (b) is a lasso-like regression problem
  - step (d) is a GLASSO-like problem
  - both optimization steps (b) and (d) require modern optmisation algorithms, e.g., Dykstra proximal splitting solver[4]

[3] E. Chouzenoux and V. Elvira. "Sparse graphical linear dynamical systems". In: *Journal of Machine Learning Research* 25.223 (2024), pp. 1–53.

[4] H. H. Bauschke and P. L. Combettes. "A Dykstra-like algorithm for two monotone operators". In: *Pacific Journal of Optimization* 4.3 (2008), pp. 383–391.

Convergence theorem

Assuming exact resolution of both inner steps (b) and (d), the sequence $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i\in\mathbb{N}}$ produced by DGLASSO algorithm:

- satisfies

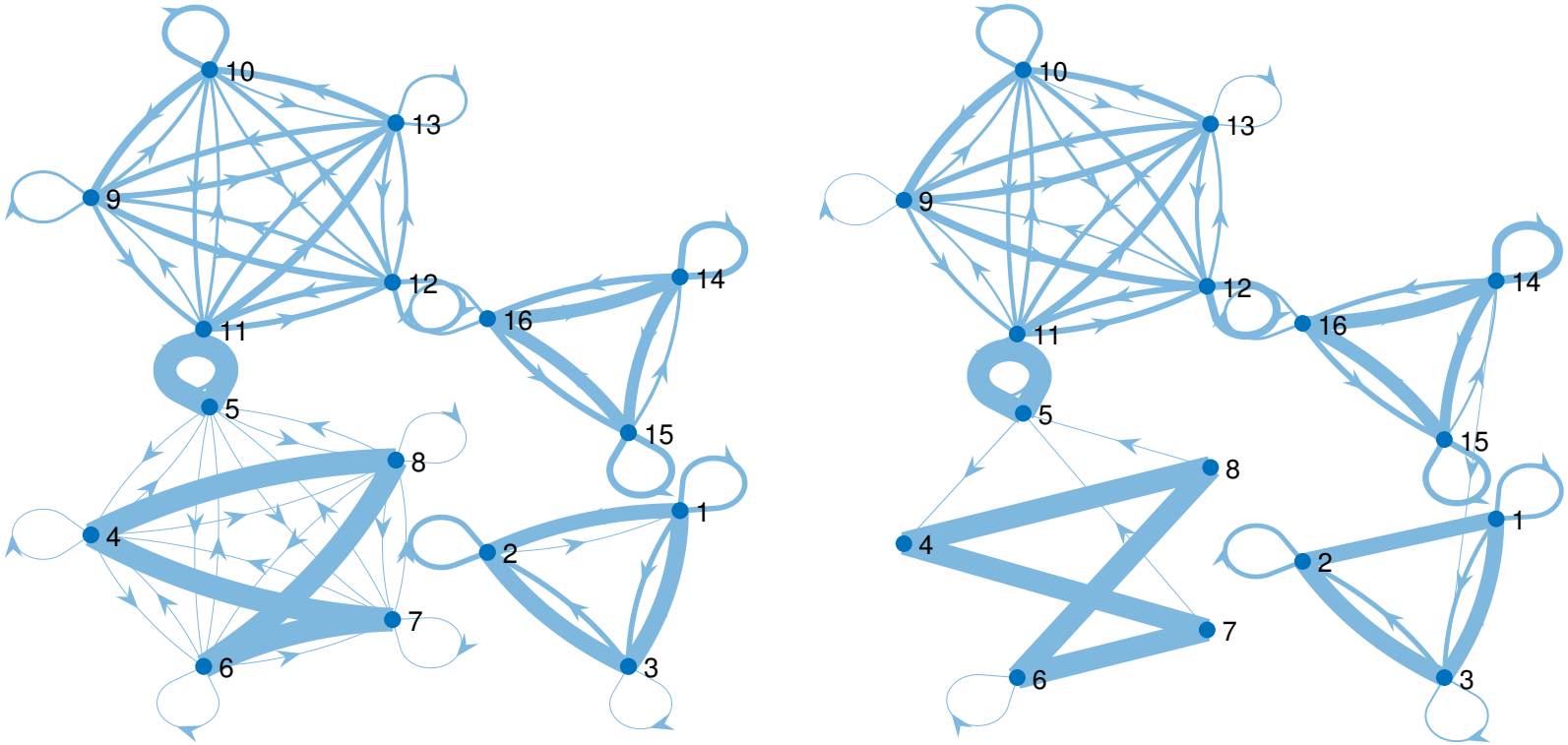
$$(\forall i \in \mathbb{N}) \quad \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) \leq \mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}), \text{ and}$$

- converges to a critical point of $\mathcal{L}(\mathbf{A}, \mathbf{P})$.

- Proof based on the recent work.[5]

[5] D. N. Phan, N. Gillis, et al. "An inertial block majorization minimization framework for nonsmooth nonconvex optimization". In: *Journal of Machine Learning Research* 24.18 (2023), pp. 1–41.

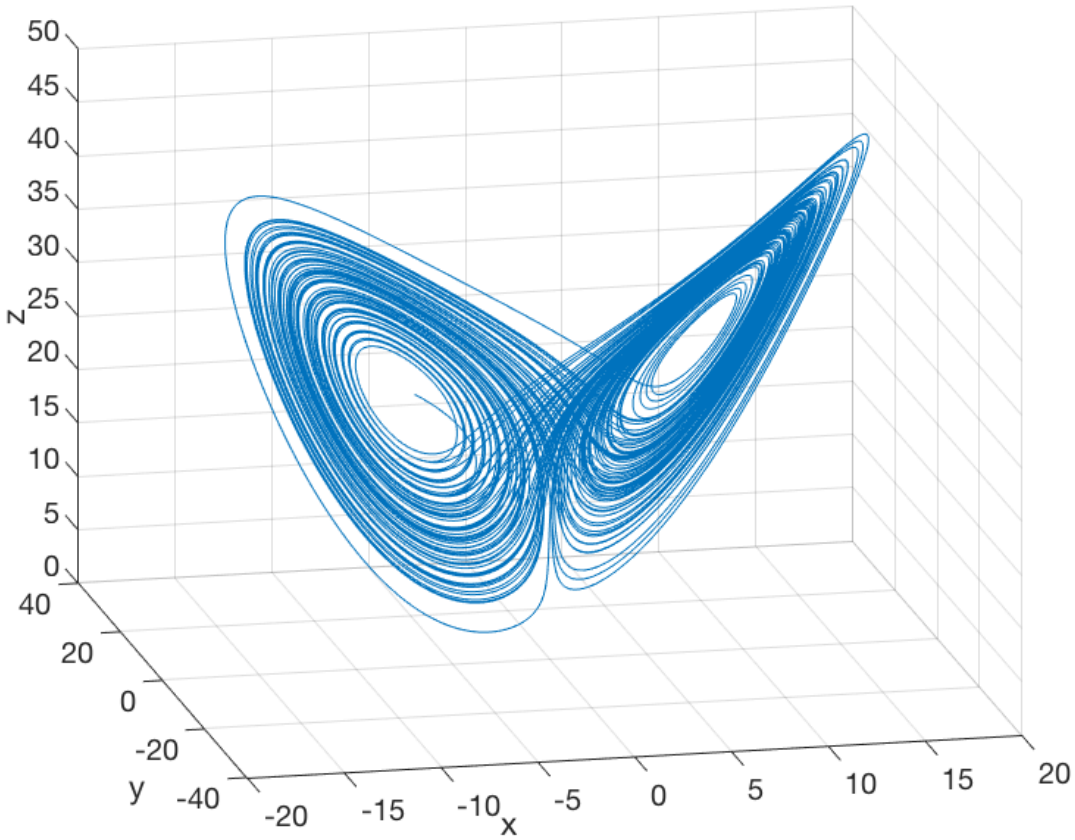
Experimental results of estimating $\mathbf{A}$ with GraphEM



True graph associated to $\mathbf{A}$ (left) and GraphEM estimate (right) for dataset C.

Outline

Motivating example: Lorenz 63



[6] E. N. Lorenz. "Deterministic nonperiodic flow". In: *Journal of atmospheric sciences* 20.2 (1963), pp. 130–141.

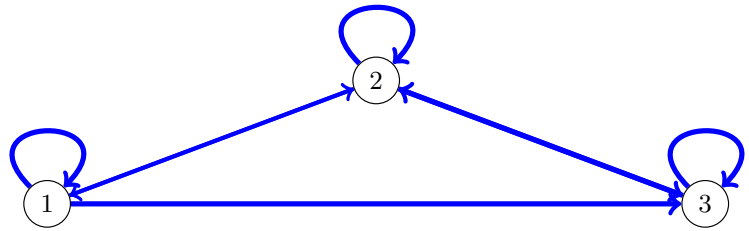
Motivating example: Lorenz 63

- Lorenz 63 equations:

$$\begin{aligned} dx_1 &= -\sigma(x_1 - x_2), \\ dx_2 &= \rho x_1 - x_2 - x_1 x_3, \\ dx_3 &= x_1 x_2 - \beta x_3, \end{aligned}$$

  - $(\sigma, \rho, \beta) = (10, 28, \frac{8}{3})$: static parameters leading to a **chaotic** behavior.
- adjacency matrix:

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$



- Discretized (Euler-Maruyama) with Δt:

$$\begin{aligned} x_{t,1} &= x_{t-1,1} + \Delta t(\sigma(x_{t-1,2} - x_{t-1,1})) + q_{t,1} \\ &= (1 - \sigma\Delta t) \cdot x_{t-1,1} + \sigma\Delta t \cdot x_{t-1,2} + q_{t,1}, \\ x_{t,2} &= x_{t-1,2} + \Delta t(x_{t-1,1}(\rho - x_{t-1,3}) - x_{t-1,2}) + q_{t,2} \\ &= \rho\Delta t \cdot x_{t-1,1} + (1 - \Delta t) \cdot x_{t-1,2} - \Delta t \cdot x_{t-1,1} x_{t-1,3} + q_{t,2}, \\ x_{t,3} &= x_{t-1,3} + \Delta t(x_{t-1,1} x_{t-1,2} - \beta x_{t-1,3}) + q_{t,3} \\ &= (1 - \beta\Delta t) \cdot x_{t-1,3} + \Delta t \cdot x_{t-1,1} x_{t-1,2} + q_{t,3}, \end{aligned}$$

where $q_{t,j} \sim \mathcal{N}(0, \Delta t)$.

A polynomial SSM

- Consider instead a d-degree polynomial model on $\mathbf{x}_t \in \mathbb{R}^{N_x}$:

$$\mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{C}) := \mathcal{N}(\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{C}; \mathbf{D}), \mathbf{Q}) \tag{1}$$

with

$$f_k(\mathbf{x}_{t-1}, \mathbf{C}; \mathbf{D}) = \sum_{i=1}^{M} \left(\mathbf{C}_{k,i} \cdot \prod_{j=1}^{N_x} x_{t-1,j}^{D_{ij}} \right), \quad k = 1, ..., K. \tag{2}$$

  - d is the maximum degree of the monomials
  - $M = \sum_{n=0}^{d} \binom{n+N_x-1}{N_x-1}$ the number of monomials up to degree d in N_x variables
  - $\mathbf{D} \in \mathbb{R}^{N_x \times M}$ a fixed integer matrix of monomial degrees associated with $\mathbf{C}$
  - $\mathbf{C} \in \mathbb{R}^{N_x \times M}$ is an **unknown** matrix of real numbers with the coefficients of the monomials,

A polynomial SSM: example in Lorenz 63

- Example: discretized (Euler-Maruyama) Lorenz 63 with Δt:

$$
\begin{aligned}
x_{t,1} &= (1 - \sigma\Delta t) \cdot x_{t-1,1} + \sigma\Delta t \cdot x_{t-1,2} + q_{t,1}, \\
x_{t,2} &= \rho\Delta t \cdot x_{t-1,1} + (1 - \Delta t) \cdot x_{t-1,2} - \Delta t \cdot x_{t-1,1} x_{t-1,3} + q_{t,2}, \\
x_{t,3} &= (1 - \beta\Delta t) \cdot x_{t-1,3} + \Delta t \cdot x_{t-1,1} x_{t-1,2} + q_{t,3},
\end{aligned}
$$

  where $q_{t,i} \sim \mathcal{N}(0, \Delta t)$.

- Set degree $d = 2$. For $N_x = 3 \Rightarrow M = 10$ monomials:

$$\mathcal{M} = \{1, x_1, x_2, x_3, x_1^2, x_1 x_2, x_1 x_3, x_2^2, x_2 x_3, x_3^2\}$$

  encoded in the degree matrix

$$\mathbf{D} = \begin{pmatrix} 0 & 1 & 0 & 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 \end{pmatrix},$$

- Lorenz 63 **exactly recovered** with coefficient matrix

$$\mathbf{C} = \begin{pmatrix} 0 & 1 - \sigma\Delta t & \sigma\Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho\Delta t & 1 - \Delta t & 0 & 0 & 0 & -\Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 - \beta\Delta t & 0 & \Delta t & 0 & 0 & 0 & 0 \end{pmatrix}$$

- GraphGrad algorithm:[7] MAP estimator of $\mathbf{C}$ with $||\mathbf{C}||_1$ penalty.
  - stochastic proximal-gradient method with differentiable PFs

[7] B. Cox, E. Chouznoux, and V. Elvira. "GraphGrad: Efficient Estimation of Sparse Polynomial Representations for General State-Space Models". In: *IEEE Transactions on Signal Processing* 72 (2025), pp. 1562–1576.

GraphGrad algorithm

- GraphGrad algorithm:[8]
  - MAP estimator $\mathbf{C}$, given $\mathbf{y}_{1:T}$ under a sparsity inducing penalty
    - first-order optimisation scheme (stochastic proximal-gradient method)

$$\widehat{\mathbf{C}} = \underset{\mathbf{C}\in\mathbb{R}^{N_x\times M}}{\operatorname{argmin}} \mathcal{L}(\mathbf{C}|\mathbf{y}_{1:T},\lambda) = \underset{\mathbf{C}\in\mathbb{R}^{N_x\times M}}{\operatorname{argmin}} \ell(\mathbf{C}) + \lambda\mathcal{L}_0(\mathbf{C}), \tag{3}$$

where $\ell(\mathbf{C}) = -\log(p(\mathbf{y}_{1:T}|\mathbf{C}))$, and $\mathcal{L}_0(\mathbf{C}) = ||\mathbf{C}||_1$

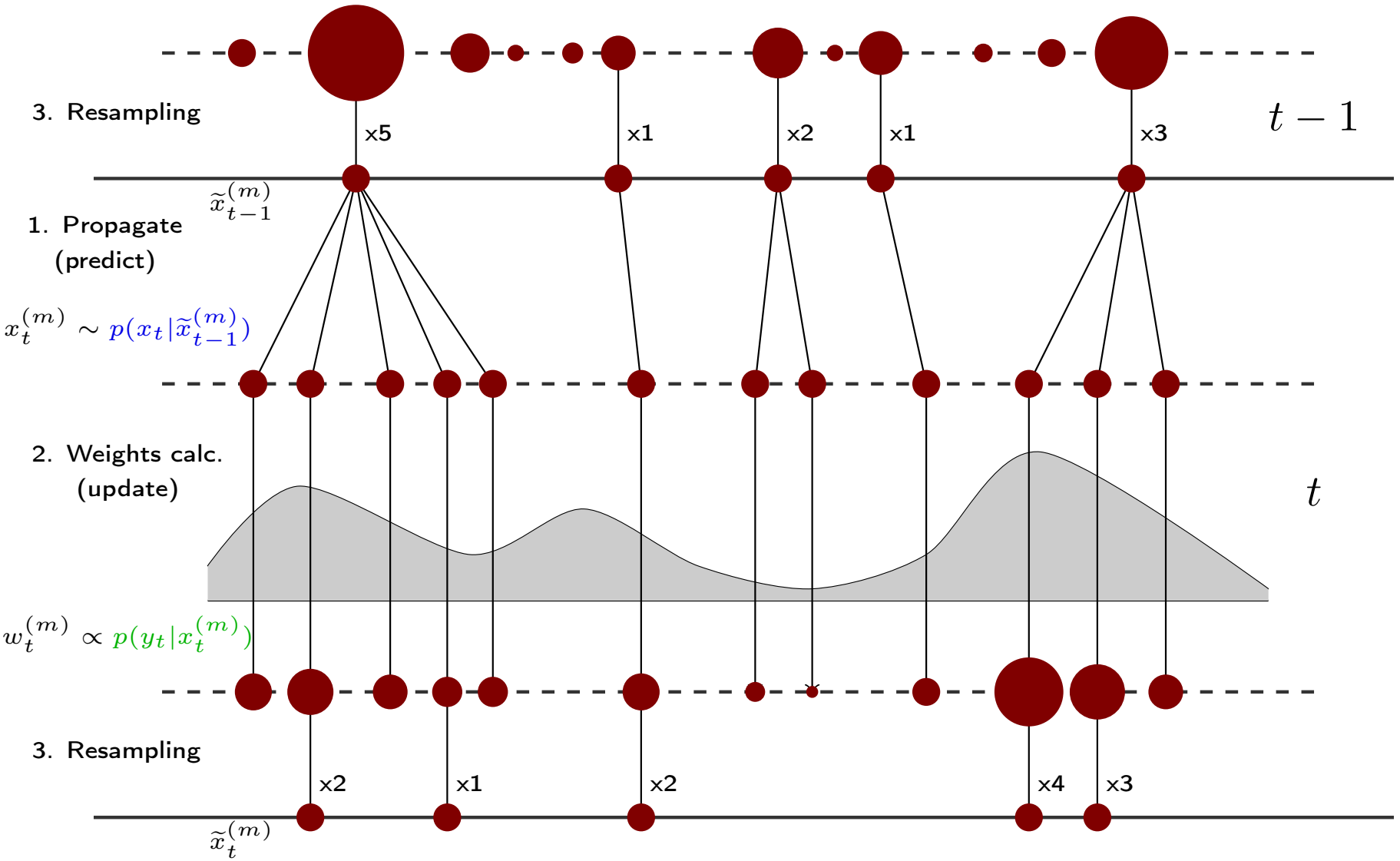
  - penalty term $\mathcal{L}_0$ using its proximity operator, which is both faster and avoids the requirement for $\mathcal{L}_0$ to be differentiable.
  - filtering distribution and likelihood function are approximated via **particle filters**

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{k=1}^{K} w_t^{(k)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(k)}).$$

$$p(\mathbf{y}_{1:T}|\mathbf{C}) \approx \prod_{t=1}^{T} \left(\frac{1}{K} \sum_{k=1}^{K} w_t^{(k)} \right), \tag{4}$$

[8] B. Cox, E. Chouznoux, and V. Elvira. "GraphGrad: Efficient Estimation of Sparse Polynomial Representations for General State-Space Models". In: *IEEE Transactions on Signal Processing* 72 (2025), pp. 1562–1576.

Bootstrap particle filter



Computational challenges in particle filtering

- Two main problems with $p(\mathbf{y}_{1:T}|\mathbf{C})$:
  - it is not differentiable w.r.t. $\mathbf{C}$ using standard particle filters
    - issue with sampling step is possible to circumvent via reparametrization trick
    - issue with resampling step is trickier[9]
  - it is too "picky"for large T, which translates in exploding gradients
    - batched version, processing only B consecutive observations at a time

[9] X. Chen and Y. Li. "An overview of differentiable particle filters for data-adaptive sequential Bayesian inference". In: *arXiv preprint arXiv:2302.09639* (2023).

S-GraphGrad

Algorithm Series GraphGrad algorithm (S-GraphGrad)

1: **Input:** Series of observations $\mathbf{y}_{1:T}$, number of steps S, penalty parameter λ, $\mathbb{N}^{N_x \times M}$ degree matrix $\mathbf{D}$, initial coefficient value $\mathbf{C}_0$, learning rate η.
2: **Output:** Sparse $\mathbb{R}^{N_x \times M}$ matrix $\mathbf{C}$ of polynomial coefficients.
3: **for** $s = 1, \ldots, S$ **do**
4: Estimate $\ell(\mathbf{C}_{s-1}|\mathbf{y}_{1:T})$ and $\nabla\ell(\mathbf{C}_{s-1}|\mathbf{y}_{1:T})$ by running differential PF with $p(\mathbf{x}_t|\mathbf{x}_{t-1}; \mathbf{C}_{s-1}, \mathbf{D})$ and observations $\mathbf{y}_{1:T}$.
5: Set $\widetilde{\mathbf{C}}_s = \text{update}_\eta(\mathbf{C}_{s-1}, \nabla\ell(\mathbf{C}_{s-1}|\mathbf{y}_{1:T}))$.
6: Set $\mathbf{C}_s = \mathcal{T}_{\eta\cdot\lambda}(\widetilde{\mathbf{C}}_s)$.
7: **end for**
8: Output $\mathbf{C} = \mathbf{C}_S$.

GraphGrad in Lorenz 63

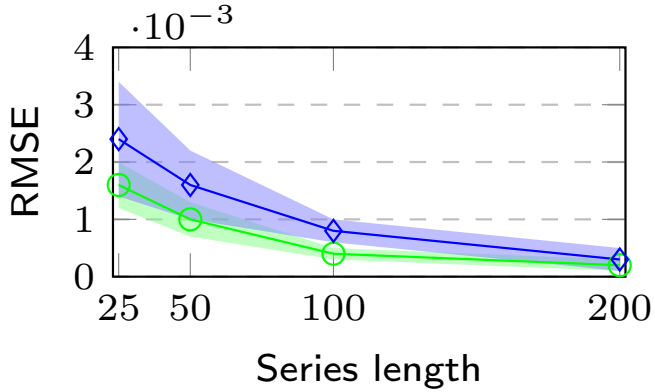


Figure: Comparison of B-GraphGrad (green) with pMLE (blue) over variable series length on the Lorenz 63 oscillator, with maximum polynomial degree $d = 2$.

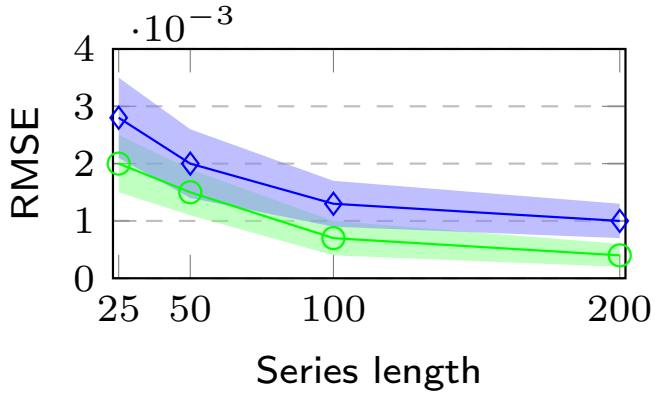


Figure: Comparison of B-GraphGrad (green) with pMLE (blue) over variable series length on the Lorenz 63 oscillator, with maximum polynomial degree $d = 3$.

GraphGrad in Lorenz 96

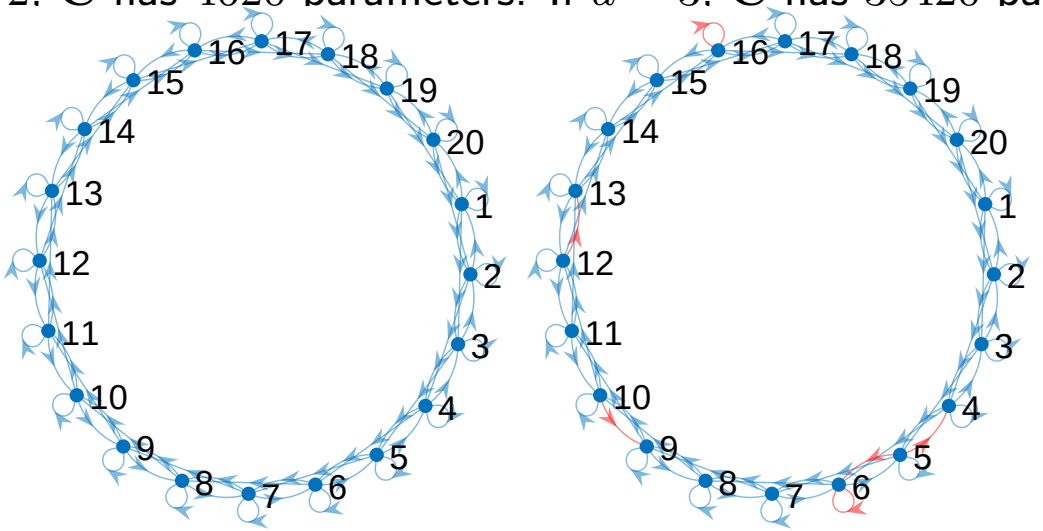
GraphGrad algorithm[10] on Lorenz 96 system:[11]

$$x_{t+1,i} = (1 - \Delta t)x_{t,i} + \Delta t x_{t,i-1}x_{t,i+1} - \Delta t x_{t,i-2} + F\Delta t + \sqrt{\Delta t} \cdot q_{t+1,i},$$

$$y_{t+1,i} = x_{t+1,i} + \sqrt{\Delta t} \cdot r_{i,t+1},$$

for $i = \{1, \ldots, N_x\}$, with $\mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R})$, $F = 8$ (chaotic) (5)

- $N_x = N_y = 20$
  - If $d = 2$, $\mathbf{C}$ has 4620 parameters. If $d = 3$, $\mathbf{C}$ has 35420 parameters



True graph GraphGrad estimate

[10] B. Cox, E. Chouznoux, and V. Elvira. "GraphGrad: Efficient Estimation of Sparse Polynomial Representations for General State-Space Models". In: *IEEE Transactions on Signal Processing* 72 (2025), pp. 1562–1576.

[11] E. N. Lorenz. "Predictability: A problem partly solved". In: *Proc. Seminar on predictability*. Vol. 1. 1. Reading. 1996.

Outline

SpaRJ algorithm

- SpaRJ[12] (*sparse reversible jump*) is a fully probabilistic algorithm for the estimation of $\mathbf{A}$, i.e., obtains samples from $p(\mathbf{A}|\mathbf{y}_{1:T})$.
- The sparsity is imposed by transitioning among models of different complexity, defined hierarchically:
  - $M_n \in \{0,1\}^{N_x \times N_x}$: sparsity pattern sample
  - A_n: matrix $\mathbf{A}$ sample, with non-zero elements, $A(i,j)$ for $\{(i,j) : M_n(i,j) = 1\}$
- We use reversible jump MCMC (RJ-MCMC) to explore $p(\mathbf{A}|\mathbf{y}_{1:T})$.[13]
  - MCMC algorithm to simulate in spaces of varying dimension, e.g., the number of ones in the sparsity pattern, $|M_n|$.
- It requires to define:
  - transition kernels for the model jumps
  - mechanism to set values when jumping to a more complex model.

[12] B. Cox and V. Elvira. "Sparse Bayesian Estimation of Parameters in Linear-Gaussian State-Space Models". In: *IEEE Transactions on Signal Processing* 71 (2023), pp. 1922–1937.

[13] P. J. Green. "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination". In: *Biometrika* 82.4 (1995), pp. 711–732.

SpaRJ algorithm

- SpaRJ[12] (*sparse reversible jump*) is a fully probabilistic algorithm for the estimation of $\mathbf{A}$, i.e., obtains samples from $p(\mathbf{A}|\mathbf{y}_{1:T})$.
- The sparsity is imposed by transitioning among models of different complexity, defined hierarchically:
  - $M_n \in \{0,1\}^{N_x \times N_x}$: sparsity pattern sample
  - A_n: matrix $\mathbf{A}$ sample, with non-zero elements, $A(i,j)$ for $\{(i,j) : M_n(i,j) = 1\}$
- We use reversible jump MCMC (RJ-MCMC) to explore $p(\mathbf{A}|\mathbf{y}_{1:T})$.[13]
  - MCMC algorithm to simulate in spaces of varying dimension, e.g., the number of ones in the sparsity pattern, $|M_n|$.
- It requires to define:
  - transition kernels for the model jumps
  - mechanism to set values when jumping to a more complex model.

[12] B. Cox and V. Elvira. "Sparse Bayesian Estimation of Parameters in Linear-Gaussian State-Space Models". In: *IEEE Transactions on Signal Processing* 71 (2023), pp. 1922–1937.

[13] P. J. Green. "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination". In: *Biometrika* 82.4 (1995), pp. 711–732.

SpaRJ algorithm

- SpaRJ[12] (*sparse reversible jump*) is a fully probabilistic algorithm for the estimation of $\mathbf{A}$, i.e., obtains samples from $p(\mathbf{A}|\mathbf{y}_{1:T})$.
- The sparsity is imposed by transitioning among models of different complexity, defined hierarchically:
  - $M_n \in \{0,1\}^{N_x \times N_x}$: sparsity pattern sample
  - A_n: matrix $\mathbf{A}$ sample, with non-zero elements, $A(i,j)$ for $\{(i,j) : M_n(i,j) = 1\}$
- We use reversible jump MCMC (RJ-MCMC) to explore $p(\mathbf{A}|\mathbf{y}_{1:T})$.[13]
  - MCMC algorithm to simulate in spaces of varying dimension, e.g., the number of ones in the sparsity pattern, $|M_n|$.
- It requires to define:
  - transition kernels for the model jumps
  - mechanism to set values when jumping to a more complex model.

[12] B. Cox and V. Elvira. "Sparse Bayesian Estimation of Parameters in Linear-Gaussian State-Space Models". In: *IEEE Transactions on Signal Processing* 71 (2023), pp. 1922–1937.

[13] P. J. Green. "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination". In: *Biometrika* 82.4 (1995), pp. 711–732.

SpaRJ algorithm

- SpaRJ[12] (*sparse reversible jump*) is a fully probabilistic algorithm for the estimation of $\mathbf{A}$, i.e., obtains samples from $p(\mathbf{A}|\mathbf{y}_{1:T})$.
- The sparsity is imposed by transitioning among models of different complexity, defined hierarchically:
  - $M_n \in \{0,1\}^{N_x \times N_x}$: sparsity pattern sample
  - A_n: matrix $\mathbf{A}$ sample, with non-zero elements, $A(i,j)$ for $\{(i,j) : M_n(i,j) = 1\}$
- We use reversible jump MCMC (RJ-MCMC) to explore $p(\mathbf{A}|\mathbf{y}_{1:T})$.[13]
  - MCMC algorithm to simulate in spaces of varying dimension, e.g., the number of ones in the sparsity pattern, $|M_n|$.
- It requires to define:
  - transition kernels for the model jumps
  - mechanism to set values when jumping to a more complex model.

[12] B. Cox and V. Elvira. "Sparse Bayesian Estimation of Parameters in Linear-Gaussian State-Space Models". In: *IEEE Transactions on Signal Processing* 71 (2023), pp. 1922–1937.

[13] P. J. Green. "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination". In: *Biometrika* 82.4 (1995), pp. 711–732.

Pseudocode of SpaRJ

Input: Known SSM parameters $\{\bar{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}, \mathbf{H}\}$, observations $\{y_t\}_{t=1}^T$, hyper-parameters, number of iterations N, initial value $\mathbf{A}_0$
Output: Set of sparse samples $\{\mathbf{A}_n\}_{n=1}^N$

Initialization
Initialize M_0 as fully dense (all ones) and $\mathbf{A}_0$
Run Kf obtaining $l_0 := \log(\mathrm{p}(\mathbf{y}_{1:T}|\mathbf{A}_0))p(\mathbf{A}_0)$
for $n = 1, ..., N$ **do**
  Step 1: Propose model
  Propose a new sparsity pattern M', obtaining a symmetry correction of c.
  Step 2: Propose $\mathbf{A}'$
  Propose $\mathbf{A}'$ using an MCMC sampler conditional on M'
  Step 3: MH accept-reject
  Evaluate Kalman filter with $\mathbf{A} := \mathbf{A}'$
  Set $l' := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
  Compute $\log(a_r) := l' - l_{n-1} + c$ and *Accept* w.p. a_r:
  if *Accept* **then**
    Set $M_n := M'$, $\mathbf{A}_n := \mathbf{A}'$, $l_n := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
  else
    Set $M_n := M_{n-1}, \mathbf{A}_n := \mathbf{A}_{n-1}, l_n := l_{n-1}$
  end if
end for

Pseudocode of SpaRJ

Input: Known SSM parameters $\{\bar{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}, \mathbf{H}\}$, observations $\{y_t\}_{t=1}^T$, hyper-parameters, number of iterations N, initial value $\mathbf{A}_0$
Output: Set of sparse samples $\{\mathbf{A}_n\}_{n=1}^N$

Initialization
Initialize M_0 as fully dense (all ones) and $\mathbf{A}_0$
Run Kf obtaining $l_0 := \log(\mathrm{p}(\mathbf{y}_{1:T}|\mathbf{A}_0))p(\mathbf{A}_0)$
for $n = 1, ..., N$ **do**
Step 1: Propose model
Propose a new sparsity pattern M', obtaining a symmetry correction of c.
Step 2: Propose A′
Propose $\mathbf{A}'$ using an MCMC sampler conditional on M'
Step 3: MH accept-reject
Evaluate Kalman filter with $\mathbf{A} := \mathbf{A}'$
Set $l' := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
Compute $\log(a_r) := l' - l_{n-1} + c$ and *Accept* w.p. a_r:
if *Accept* **then**
Set $M_n := M'$, $\mathbf{A}_n := \mathbf{A}'$, $l_n := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
else
Set $M_n := M_{n-1}, \mathbf{A}_n := \mathbf{A}_{n-1}, l_n := l_{n-1}$
end if
end for

Pseudocode of SpaRJ

Input: Known SSM parameters $\{\bar{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}, \mathbf{H}\}$, observations $\{y_t\}_{t=1}^T$, hyper-parameters, number of iterations N, initial value $\mathbf{A}_0$
Output: Set of sparse samples $\{\mathbf{A}_n\}_{n=1}^N$

Initialization
Initialize M_0 as fully dense (all ones) and $\mathbf{A}_0$
Run Kf obtaining $l_0 := \log(\mathrm{p}(\mathbf{y}_{1:T}|\mathbf{A}_0))p(\mathbf{A}_0)$
for $n = 1, ..., N$ **do**
Step 1: Propose model
Propose a new sparsity pattern M', obtaining a symmetry correction of c.
Step 2: Propose $\mathbf{A}'$
Propose $\mathbf{A}'$ using an MCMC sampler conditional on M'
Step 3: MH accept-reject
Evaluate Kalman filter with $\mathbf{A} := \mathbf{A}'$
Set $l' := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
Compute $\log(a_r) := l' - l_{n-1} + c$ and *Accept* w.p. a_r:
if *Accept* **then**
Set $M_n := M'$, $\mathbf{A}_n := \mathbf{A}'$, $l_n := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
else
Set $M_n := M_{n-1}, \mathbf{A}_n := \mathbf{A}_{n-1}, l_n := l_{n-1}$
end if
end for

Pseudocode of SpaRJ

Input: Known SSM parameters $\{\bar{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}, \mathbf{H}\}$, observations $\{y_t\}_{t=1}^T$, hyper-parameters, number of iterations N, initial value $\mathbf{A}_0$
Output: Set of sparse samples $\{\mathbf{A}_n\}_{n=1}^N$

Initialization

Initialize M_0 as fully dense (all ones) and $\mathbf{A}_0$
Run Kf obtaining $l_0 := \log(\mathrm{p}(\mathbf{y}_{1:T}|\mathbf{A}_0))p(\mathbf{A}_0)$
for $n = 1, ..., N$ **do**

Step 1: Propose model

Propose a new sparsity pattern M', obtaining a symmetry correction of c.

Step 2: Propose $\mathbf{A}'$

Propose $\mathbf{A}'$ using an MCMC sampler conditional on M'

Step 3: MH accept-reject

Evaluate Kalman filter with $\mathbf{A} := \mathbf{A}'$
Set $l' := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
Compute $\log(a_r) := l' - l_{n-1} + c$ and *Accept* w.p. a_r:
if *Accept* **then**
Set $M_n := M'$, $\mathbf{A}_n := \mathbf{A}'$, $l_n := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
else
Set $M_n := M_{n-1}, \mathbf{A}_n := \mathbf{A}_{n-1}, l_n := l_{n-1}$
end if
end for

Pseudocode of SpaRJ

Input: Known SSM parameters $\{\bar{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}, \mathbf{H}\}$, observations $\{y_t\}_{t=1}^T$, hyper-parameters, number of iterations N, initial value $\mathbf{A}_0$
Output: Set of sparse samples $\{\mathbf{A}_n\}_{n=1}^N$

Initialization
Initialize M_0 as fully dense (all ones) and $\mathbf{A}_0$
Run Kf obtaining $l_0 := \log(\mathrm{p}(\mathbf{y}_{1:T}|\mathbf{A}_0))p(\mathbf{A}_0)$
for $n = 1, ..., N$ **do**
  Step 1: Propose model
  Propose a new sparsity pattern M', obtaining a symmetry correction of c.
  Step 2: Propose $\mathbf{A}'$
  Propose $\mathbf{A}'$ using an MCMC sampler conditional on M'
  Step 3: MH accept-reject
  Evaluate Kalman filter with $\mathbf{A} := \mathbf{A}'$
  Set $l' := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
  Compute $\log(a_r) := l' - l_{n-1} + c$ and *Accept* w.p. a_r:
  if *Accept* **then**
    Set $M_n := M'$, $\mathbf{A}_n := \mathbf{A}'$, $l_n := \log(p(\mathbf{y}_{1:T}|\mathbf{A}'))p(\mathbf{A}')$
  else
    Set $M_n := M_{n-1}, \mathbf{A}_n := \mathbf{A}_{n-1}, l_n := l_{n-1}$
  end if
end for

Convergence of SpaRJ and GraphEM with data

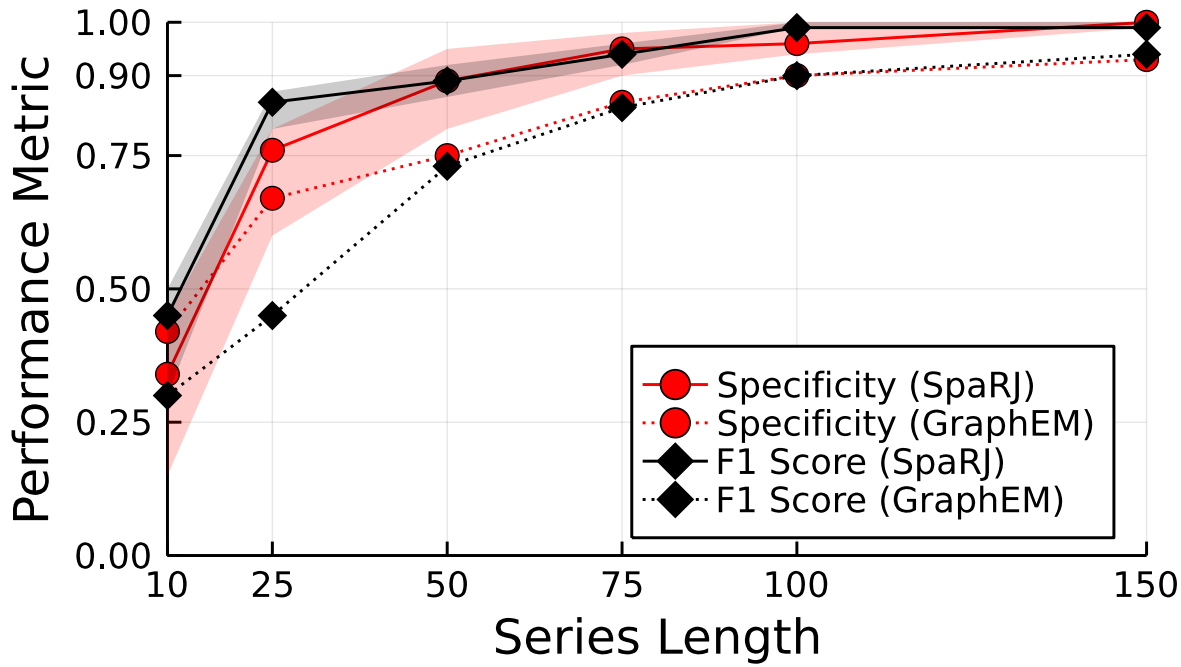


Figure: 3×3 system with known isotropic state covariance.

Convergence of SpaRJ with iterations

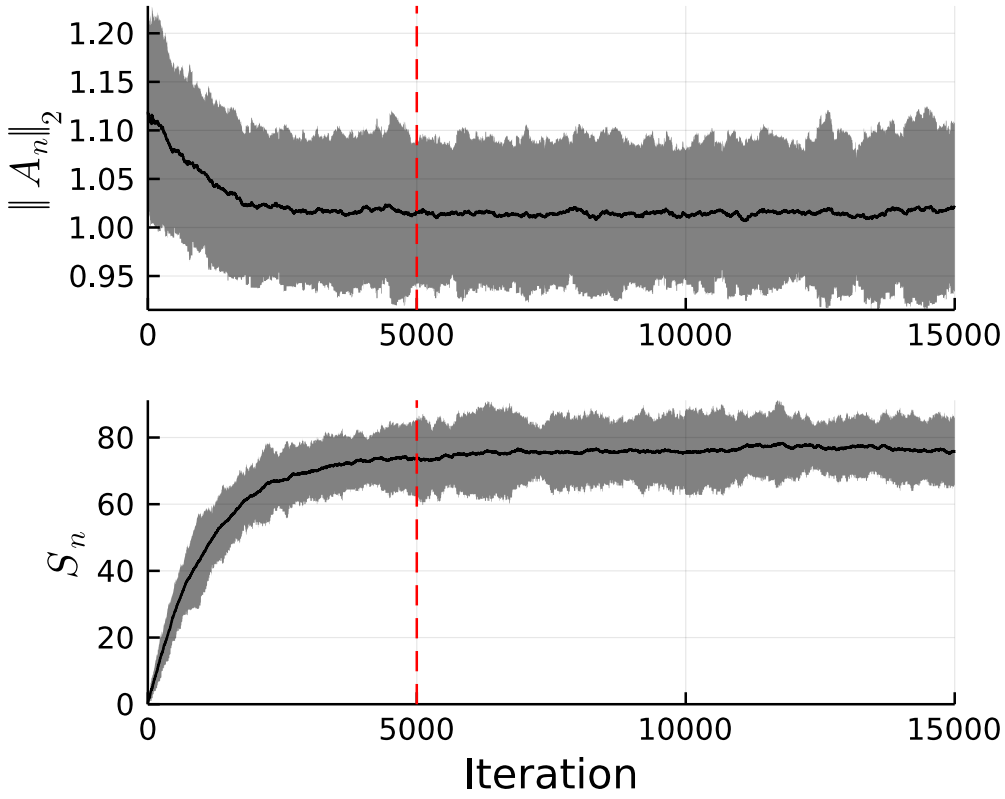


Figure: Progression of sample metrics in a 12×12.

SpaRJ with real world data

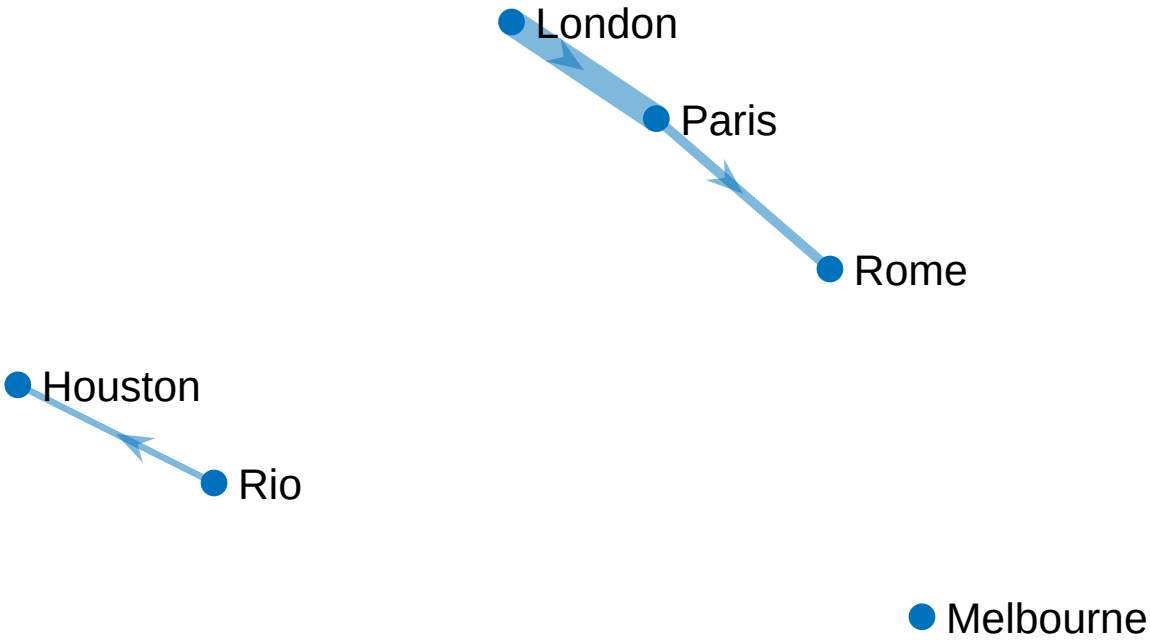


Figure: Average daily temperature of 324 cities from 1995 to 2021, curated by the United States Environmental Protection Agency.

Outline

Conclusion

- Novel graphical interpretation on SSMs
- LG-SSMs: lgorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM: faster and allows explicit penalty functions
  - SpaRJ: samples of the posterior of $\mathbf{A}$.
- DGLASSO: estimate sparse $\mathbf{A}$ and $\mathbf{Q}$ (point-wise)
- GraphGrad: approximate nonlinear SSMs with (highly sparse) polynomials
- Challenging problem with many exciting ongoing methodological and applied avenues ahead!

Conclusion

- Novel graphical interpretation on SSMs
- LG-SSMs: lgorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM: faster and allows explicit penalty functions
  - SpaRJ: samples of the posterior of $\mathbf{A}$.
- DGLASSO: estimate sparse $\mathbf{A}$ and $\mathbf{Q}$ (point-wise)
- GraphGrad: approximate nonlinear SSMs with (highly sparse) polynomials
- Challenging problem with many exciting ongoing methodological and applied avenues ahead!

Conclusion

- Novel graphical interpretation on SSMs
- LG-SSMs: lgorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM: faster and allows explicit penalty functions
  - SpaRJ: samples of the posterior of $\mathbf{A}$.
- DGLASSO: estimate sparse $\mathbf{A}$ and $\mathbf{Q}$ (point-wise)
- GraphGrad: approximate nonlinear SSMs with (highly sparse) polynomials
- Challenging problem with many exciting ongoing methodological and applied avenues ahead!

Conclusion

- Novel graphical interpretation on SSMs
- LG-SSMs: lgorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM: faster and allows explicit penalty functions
  - SpaRJ: samples of the posterior of $\mathbf{A}$.
- DGLASSO: estimate sparse $\mathbf{A}$ and $\mathbf{Q}$ (point-wise)
- GraphGrad: approximate nonlinear SSMs with (highly sparse) polynomials
- Challenging problem with many exciting ongoing methodological and applied avenues ahead!

Conclusion

- Novel graphical interpretation on SSMs
- LG-SSMs: lgorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM: faster and allows explicit penalty functions
  - SpaRJ: samples of the posterior of $\mathbf{A}$.
- DGLASSO: estimate sparse $\mathbf{A}$ and $\mathbf{Q}$ (point-wise)
- GraphGrad: approximate nonlinear SSMs with (highly sparse) polynomials
- Challenging problem with many exciting ongoing methodological and applied avenues ahead!

Thank you for your attention!

GraphEM (learn $\mathbf{A}$ in LG-SSMs): V. Elvira, É. Chouzenoux, "Graphical Inference in Linear-Gaussian State-Space Models", *IEEE Transactions on Signal Processing*, Vol. 70, pp. 4757-4771, 2022.

DGLASSO (learn $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSMs): É. Chouzenoux and V. Elvira, "Sparse Graphical Linear Dynamical Systems, Journal of Machine Learning Research, Vol. 25, No. 223, pp. 1-53, 2024

GraphGrad (learn $\mathbf{C}$ in non-linear SSMs): B. Cox, É. Chouzenoux, V. Elvira, "GraphGrad: Efficient Estimation of Sparse Polynomial Representations for General State-Space Models", *IEEE Transactions on Signal Processing*, Vol. 73, pp. 1562-1576, 2025.

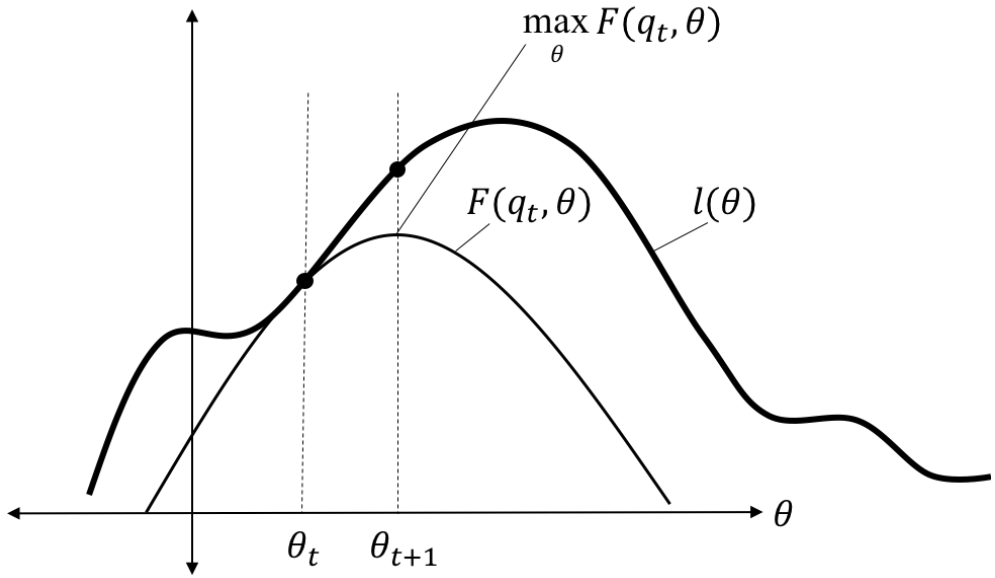
SpaRJ (probabilistic learning of $\mathbf{A}$ in LG-SSMs): B. Cox and V. Elvira, "Sparse Bayesian Estimation of Parameters in Linear-Gaussian State-Space Models", IEEE Transactions on Signal Processing, vol. 71, pp. 1922-1937, 2023.

GraphIT (better sparsity in $\mathbf{A}$ in LG-SSMs): E. Chouzenoux and V. Elvira, "Iterative reweighted ℓ_1 algorithm for sparse graph inference in state-space models", IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2023).

Non-LaGrangEM (learn $\mathbf{A}$ in non-Markovian LG-SSMs): E. Chouzenoux and V. Elvira, "Graphical Inference in Non-Markovian Linear-Gaussian State-space Models", IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2024).

EM approach for ML estimation

- EM approach for ML:[14] Initialize $(\mathbf{A}^{(0)}, \mathbf{P}^{(0)})$ and, at each iteration $i \geq 0$,
  - Majorizing function (E-step):
    - run KF/RTS smoother by setting $(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \in \mathbb{R}^{N_x \times N_x} \times \mathcal{S}_{N_x}$
    - build majorizing function ($\mathcal{Q}(\mathbf{A}, \mathbf{P}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \geq \mathcal{L}_{1:T}(\mathbf{A}, \mathbf{P})$, $\forall(\mathbf{A}, \mathbf{P})$).
  - Minimization step (M-step): Minimize $\mathcal{Q}(\mathbf{A}, \mathbf{P}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)})$ w.r.t. $\mathbf{A}$ and $\mathbf{P}$ to obtain $\mathbf{A}^{(i+1)}$ and $\mathbf{P}^{(i+1)}$.



(credit to M. N. Bernstein)

[14] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. Journal of Time Series Analysis, 3(4):253–264, 1982.

Summary of the GraphEM algorithm

- DGLASSO generalises our previous GraphEM,[15] where only $\mathbf{A}$ is unknown.

GraphEM algorithm

- Initialization of $\mathbf{A}^{(0)}$.
- For $i = 1, 2, \ldots$

E-step Run the Kalman filter and RTS smoother by setting $\mathbf{A}' := \mathbf{A}^{(i-1)}$ and construct $\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i-1)})$.

M-step Update $\mathbf{A}^{(i)} = \text{argmin}_{\mathbf{A}} \left(\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i-1)}) \right)$ using Douglas-Rachford algorithm (simpler version) or monotone+skew (MS) algorithm (generalized version).

- Flexible approach, valid as long as the proximity operators of $(f_m)_{2 \leq m \leq M}$ are available, with $\mathcal{L}_0 = \sum_{m=1}^{M} f_m$

[15] V. Elvira and É. Chouzenoux. "Graphical Inference in Linear-Gaussian State-Space Models". In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 4757–4771.

GraphEM in a nutshell

- **Goal.** MAP estimate of $\mathbf{A}$:

$$\mathbf{A}^* = \text{argmax}_{\mathbf{A}} p(\mathbf{A}|\mathbf{y}_{1:T}) = \text{argmax}_{\mathbf{A}} p(\mathbf{A}) p(\mathbf{y}_{1:T}|\mathbf{A})$$

  - Equivalent to minimizing $\mathcal{L}(\mathbf{A}) = -\log p(\mathbf{A}) - \log p(\mathbf{y}_{1:T}|\mathbf{A})$.
  - **Challenges**: evaluating $\mathcal{L}_{1:T}(\mathbf{A}) \equiv -\log p(\mathbf{y}_{1:T}|\mathbf{A})$ requires to run the KF:

$$\mathcal{L}_{1:T}(\mathbf{A}) = \sum_{t=1}^{T} \frac{1}{2} \log |2\pi \mathbf{S}_t(\mathbf{A})| + \frac{1}{2} \mathbf{z}_t(\mathbf{A})^\top \mathbf{S}_t(\mathbf{A})^{-1} \mathbf{z}_t(\mathbf{A}).$$

    - Function $\mathcal{L}_0(\mathbf{A}) \equiv -\log p(\mathbf{A})$ might be complicated (e.g., non smooth).
    - Non tractable minimization.
  - Simplest version of GraphEM:[16] an EM strategy to minimize a sequence of (tractable) majorizing approximations of $\mathcal{L}$.
    - Lasso regularization (Laplace prior) to promote a **sparse matrix** $\mathbf{A}$:

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad \mathcal{L}_0(\mathbf{A}) = \gamma \|\mathbf{A}\|_1, \qquad \gamma > 0.$$

[16] E. Chouzenoux and V. Elvira. "GraphEM: EM algorithm for blind Kalman filtering under graphical sparsity constraints". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 5840–5844.

Expression of EM steps

• Majorizing approximation (E-step): Run the Kalman filter/RTS smoother by setting the state matrix to $\mathbf{A}'$ and define[17]

$$\boldsymbol{\Sigma} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{P}_t^s + \mathbf{m}_t^s (\mathbf{m}_t^s)^\top,$$

$$\boldsymbol{\Phi} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{P}_{t-1}^s + \mathbf{m}_{t-1}^s (\mathbf{m}_{t-1}^s)^\top$$

$$\mathbf{C} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{P}_t^s \mathbf{G}_{t-1}^\top + \mathbf{m}_t^s (\mathbf{m}_{t-1}^s)^\top.$$

and build

$$\mathcal{Q}(\mathbf{A}; \mathbf{A}') = \frac{T}{2}\mathrm{tr}\left(\mathbf{Q}^{-1}(\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^\top - \mathbf{A}\mathbf{C}^\top + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^\top)\right) + \mathcal{L}_0(\mathbf{A}) + \mathrm{ct}_{/\mathbf{A}},$$

such that, for every $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$:

$$\mathcal{Q}(\mathbf{A}; \mathbf{A}') \geq \mathcal{L}(\mathbf{A}), \qquad \text{and} \qquad \mathcal{Q}(\mathbf{A}'; \mathbf{A}') = \mathcal{L}(\mathbf{A}').$$

• Upper bound optimization (M-step): The M-step consists in searching for a minimizer of $\mathcal{Q}(\mathbf{A}; \mathbf{A}')$ with respect to $\mathbf{A}$ ($\mathbf{A}'$ being fixed).

[17] R. H. Shumway and D. S. Stoffer. "An approach to time series smoothing and forecasting using the EM algorithm". In: *Journal of Time Series Analysis* 3.4 (1982), pp. 253–264.

Computation of the M-step

- Convex non-smooth minimization problem

$$\text{argmin}_{\mathbf{A}} \underbrace{\mathcal{Q}(\mathbf{A};\mathbf{A}')}_{f(\mathbf{A})} = \text{argmin}_{\mathbf{A}} \underbrace{\frac{T}{2}\text{tr}\left(\mathbf{Q}^{-1}(\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^\top - \mathbf{A}\mathbf{C}^\top + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^\top)\right)}_{f_1(\mathbf{A})=\text{upper bound of } -\log\left(p(\mathbf{y}_{1:T}|\mathbf{A})\right)} + \underbrace{\gamma\|\mathbf{A}\|_1}_{\substack{f_2(\mathbf{A})=-\log p(\mathbf{A}) \\ \textbf{(prior)}}}$$

Proximal splitting approach: The proximity operator of $f : \mathbb{R}^{N_x \times N_x} \to \mathbb{R}$ is defined

$$\text{prox}_f(\widetilde{\mathbf{A}}) = \text{argmin}_{\mathbf{A}} \left(f(\mathbf{A}) + \frac{1}{2}\|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2 \right).$$

Douglas-Rachford algorithm in GraphEM

- Set $\mathbf{Z}_0 \in \mathbb{R}^{N_x \times N_x}$ and $\theta \in (0, 2)$.
- For $n = 1, 2, \ldots$

$$\begin{aligned} \mathbf{A}_n &= \text{prox}_{\theta f_2}(\mathbf{Z}_n) \\ \mathbf{V}_n &= \text{prox}_{\theta f_1}(2\mathbf{A}_n - \mathbf{Z}_n) \\ \mathbf{Z}_{n+1} &= \mathbf{Z}_n + \theta(\mathbf{V}_n - \mathbf{A}_n) \end{aligned}$$

✓ $\{\mathbf{A}_n\}_{n\in\mathbb{N}}$ guaranteed to converge to a minimizer of $\mathcal{Q}(\mathbf{A};\mathbf{A}') = f_1 + f_2$

✓ Both involved proximity operators have closed form solution.

Generic GraphEM algorithm

- generic GraphEM allows for a larger family of priors (and several):[18]

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad \mathcal{Q}(\mathbf{A}; \mathbf{A}') = \sum_{m=1}^{M} f_m(\mathbf{A}), \tag{6}$$

  - $f_1(\mathbf{A})$ is still an upper bound of $-\log\left(p(\mathbf{y}_{1:T}|\mathbf{A})\right)$
  - $f_M(\mathbf{A}) = \gamma\|\mathbf{A}\|_1$ (sparsity promoter)
  - other losses $\{f_m(\mathbf{A})\}_{m=2}^{M-1}$ promote properties in $\mathbf{A}$ (e.g., stability)
- The inference now requires a more sophisticated optimization algorithm in the M-step, the monotone+skew algorithm.

MS algorithm for a generic GraphEMs (M-step)

- Set $\mathbf{V}_0^m = \mathbf{A}'$ $\forall m \in \{1, \ldots, M\}$, and stepsizes $\lambda \in (0, \frac{1}{M})$, $\gamma \in [\lambda, \frac{1-\lambda}{M-1}]$.
- For $n = 1, 2, \ldots$

$$\begin{aligned}
&\mathbf{W}_n^m = \mathbf{V}_n^m + \gamma \mathbf{V}_n^M \ (\forall m \in \{1, \ldots, M-1\}) \\
&\mathbf{W}_n^M = \mathbf{V}_n^M - \gamma \textstyle\sum_{m=1}^{M-1} \mathbf{V}_n^m \\
&\mathbf{A}_n^m = \mathbf{W}_n^m - \gamma \operatorname{prox}_{f_m/\gamma}(\mathbf{W}_n^m) \ (\forall m \in \{1, \ldots, M-1\}) \\
&\mathbf{A}_n^M = \operatorname{prox}_{\gamma f_M}(\mathbf{W}_n^M) \\
&\mathbf{Z}_n^m = \mathbf{A}_n^m + \gamma \mathbf{A}_n^M \ (\forall m \in \{1, \ldots, M-1\}) \\
&\mathbf{Z}_n^M = \mathbf{A}_n^M - \gamma \textstyle\sum_{m=1}^{M-1} \mathbf{A}_n^m \\
&\mathbf{V}_{n+1}^m = \mathbf{V}_n^m - \mathbf{W}_n^m + \mathbf{Z}_n^m \ (\forall m \in \{1, \ldots, M\})
\end{aligned}$$

[18] V. Elvira and É. Chouzenoux. "Graphical Inference in Linear-Gaussian State-Space Models". In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 4757–4771.

Theoretical guarantees of GraphEM

Theorem

Assume that the prior term $\mathcal{L}_0$ is proper, convex, lower semicontinuous. Under mild technical assumptions (qualification conditions),

- $\{\mathcal{L}(\mathbf{A}^{(i)})\}_{i\in\mathbb{N}}$ is a decreasing sequence converging to a finite limit $\mathcal{L}^*$.
- The sequence of iterates $\{\mathbf{A}^{(i)}\}_{i\in\mathbb{N}}$ has a cluster point (i.e., one can extract a converging subsequence)
- Let $\mathbf{A}^*$ a cluster point (i.e., the limit of a converging subsequence) of $\{\mathbf{A}^{(i)}\}_{i\in\mathbb{N}}$. Then, $\mathcal{L}(\mathbf{A}^*) = \mathcal{L}^*$ and $\mathbf{A}^*$ is a critical point of $\mathcal{L}$, i.e., $\nabla\mathcal{L}_{1:T}(\mathbf{A}^*) \in \partial\mathcal{L}_0(\mathbf{A}^*)$.

Data description and numerical settings

- Four synthetic datasets with $\mathbf{H} = \mathbf{Id}$, size $N_x = N_y = 9$, and randomly generated ground truth sparse matrices $\mathbf{A}^*$ and $\mathbf{P}^*$ (block diagonal 3×3) with varying conditioning for $\mathbf{Q}^* = (\mathbf{P}^*)^{-1}$.
  We set $K = 10^3$ and $\mathbf{R} = \sigma_{\mathbf{R}}^2 \mathbf{Id}$, $\mathbf{P}_0 = \sigma_0^2 \mathbf{Id}$ with $(\sigma_{\mathbf{R}}, \sigma_0) = (10^{-1}, 10^{-4})$.

- **Goal:** (i) Given $\{\mathbf{y}_k\}_{k=1}^K$, and $(\mathbf{H}, \mathbf{R}, \mathbf{P}_0)$, provide estimates $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$ of $(\mathbf{A}^*, \mathbf{P}^*)$, evaluated by **RMSE and $\mathbf{F}_1$ metrics**, (ii) Given a new test data, compute the **the predictive distribution means** by KF/RTS using the estimated model parameters, evaluated by **cNMSE** and loss metrics.

- DGLASSO, is compared with:
  - Maximum likelihood EM (MLEM): DGLASSO model with $\lambda_A = \lambda_P = 0$.
  - GRAPHEM approach [Elvira et al., 2022]: MAP estimate of $\mathbf{A}$, while fixing $\widehat{\mathbf{Q}} = \sigma_Q^2 \mathrm{Id}$ with finetuned σ_Q.
  - GLASSO approach [Friedman et al., 2008]: MAP estimate of $\mathbf{P}$, fixing $\widehat{\mathbf{A}} = \mathbf{0}$ and neglecting $\mathbf{R}$.
  - rGLASSO approach [Benfenati et al., 2020]: MAP estimate of $\mathbf{P}$, fixing $\widehat{\mathbf{A}} = \mathbf{0}$.
  - Pairwise Granger Causality (PGC) / conditional Granger Causality (CGC) based on sparse vector autoregressive (VAR) models [Luengo et al., 2019].

- Manual finetuning of hyperparameters (e.g., ℓ_1 penalty weight) on a single realization (see more details in paper). Results are averaged on 50

Convergence theorem

Assuming exact resolution of both inner steps (b) and (d), the sequence $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i\in\mathbb{N}}$ produced by DGLASSO algorithm:

- satisfies

$$(\forall i \in \mathbb{N}) \quad \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) \leq \mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}), \text{ and}$$

- converges to a critical point of $\mathcal{L}(\mathbf{A}, \mathbf{P})$.

- Proof based on the recent work.[19]

[19] D. N. Phan, N. Gillis, et al. "An inertial block majorization minimization framework for nonsmooth nonconvex optimization". In: *Journal of Machine Learning Research* 24.18 (2023), pp. 1–41.

Experimental results of estimating $\mathbf{A}$ with GraphEM (simplified DGLASSO)

- Four synthetic datasets with $\mathbf{H} = \mathbf{Id}$ and block-diagonal matrix $\mathbf{A}$, composed with b blocks of size $(b_j)_{1 \leq j \leq b}$, so that $N_y = N_x = \sum_{j=1}^{b} b_j$. We set $T = 10^3$, $\mathbf{Q} = \sigma_{\mathbf{Q}}^2 \mathbf{Id}$, $\mathbf{R} = \sigma_{\mathbf{R}}^2 \mathbf{Id}$, $\mathbf{P}_0 = \sigma_{\mathbf{P}}^2 \mathbf{Id}$.

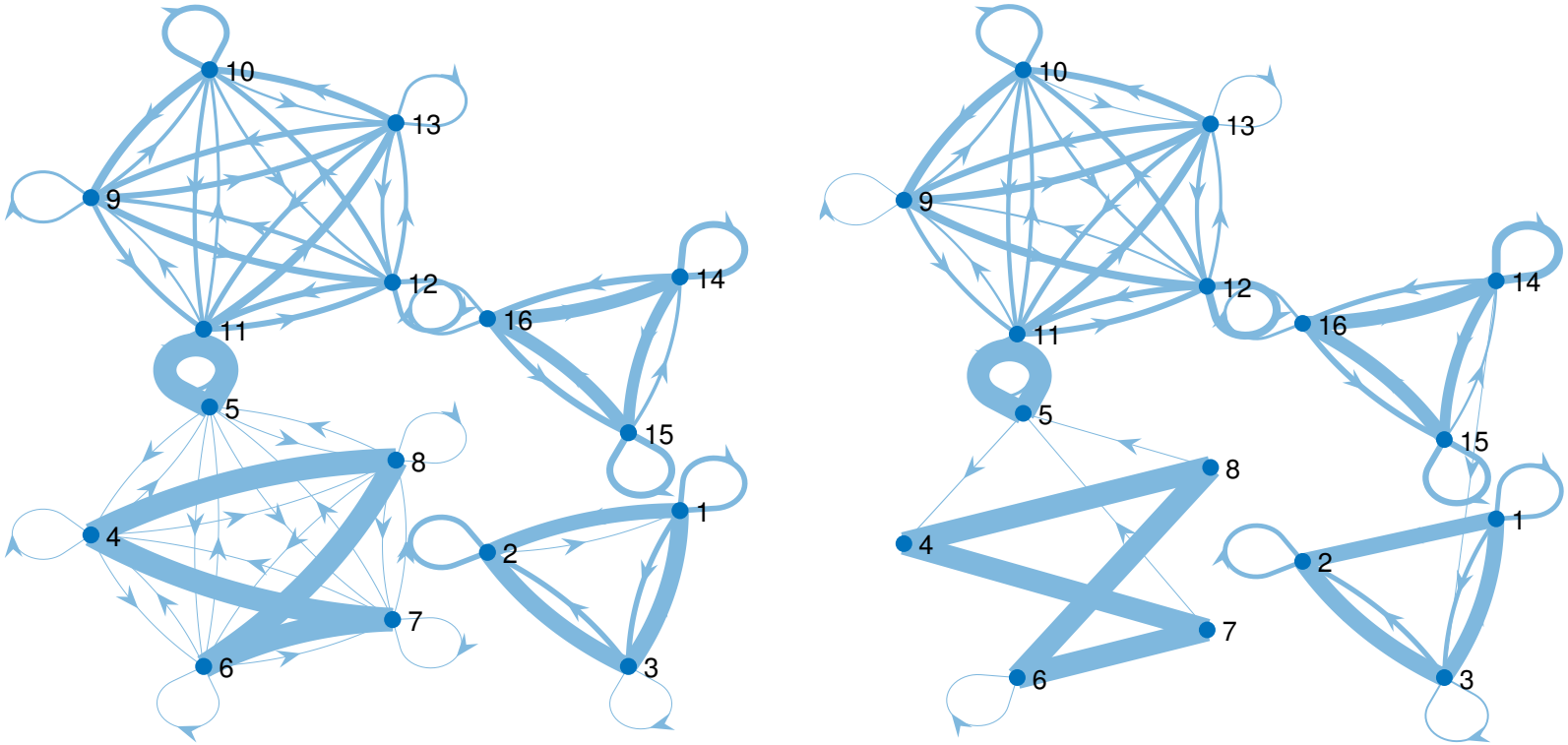
Dataset	N_x	$(b_j)_{1 \leq j \leq b}$	$(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$
A	9	$(3,3,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
B	9	$(3,3,3)$	$(1, 1, 10^{-4})$
C	16	$(3,5,5,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
D	16	$(3,5,5,3)$	$(1, 1, 10^{-4})$

- GraphEM (DGLASSO with known $\mathbf{Q}$) is compared with:
  - Maximum likelihood EM (MLEM)[20]
  - Granger-causality approaches: pairwise Granger Causality (PGC) and conditional Granger Causality (CGC)[21]

[20] S. Sarkka. *Bayesian Filtering and Smoothing*. Ed. by C. U. Press. 2013.

[21] D. Luengo, G. Rios-Munoz, V. Elvira, C. Sanchez, and A. Artes-Rodriguez. "Hierarchical algorithms for causality retrieval in atrial fibrillation intracavitary electrograms". In: *IEEE journal of biomedical and health informatics* 23.1 (2018), pp. 143–155.

Experimental results of estimating $\mathbf{A}$ with GraphEM



True graph associated to $\mathbf{A}$ (left) and GraphEM estimate (right) for dataset C.

Computational complexity of DGLASSO

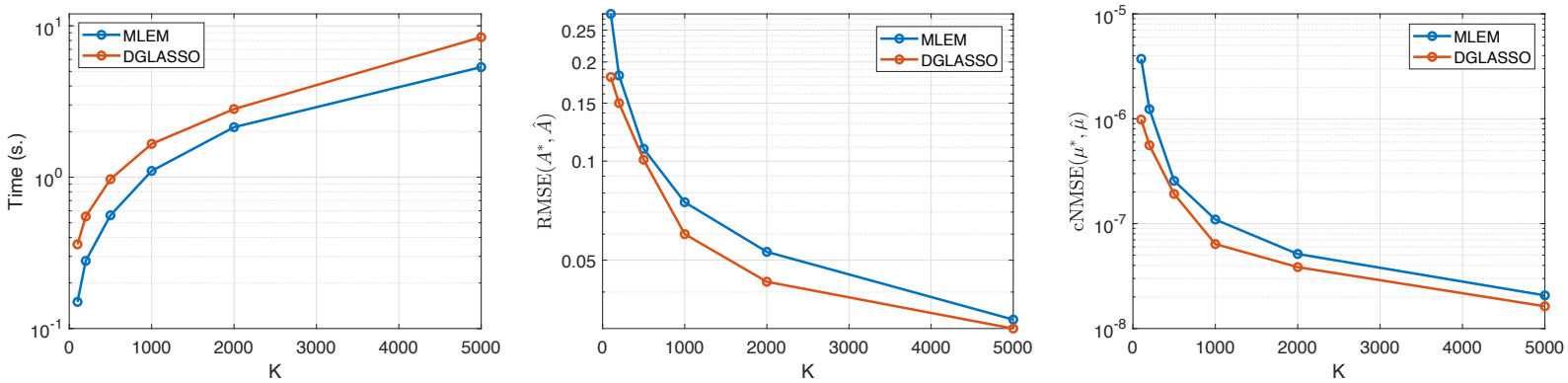


Figure 6: Evolution of the complexity time (left), RMSE($\mathbf{A}^*, \widehat{\mathbf{A}}$) (middle) and cNMSE($\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}}$) (right) metrics, as a function of the time series length K, for experiments on dataset A averaged over 50 runs.

Performance of DGLASSO (toy example)

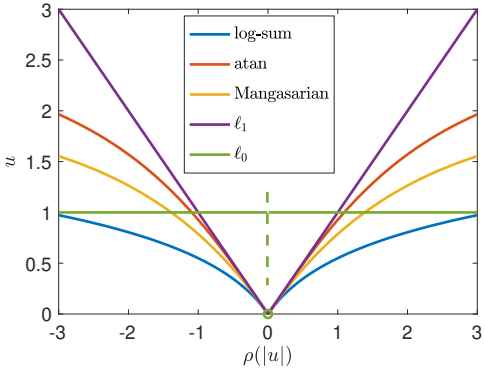
		Estimation of $\mathbf{A}$			Estimation of $\mathbf{P}$			Estim. $\mathbf{Q}$	State distrib.		Predictive distrib.	
	Method	RMSE	AUC	F1	RMSE	AUC	F1	RMSE	cNMSE($\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}}$)	cNMSE($\boldsymbol{\mu}^{s*}, \widehat{\boldsymbol{\mu}}^s$)	cNMSE($\boldsymbol{\nu}^*, \widehat{\boldsymbol{\nu}}$)	$\mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P})$
Dataset A	DGLASSO	0.061	0.843	**0.641**	**0.082**	0.778	0.698	**0.083**	$\mathbf{6.394 \times 10^{-8}}$	$\mathbf{1.050 \times 10^{-7}}$	$\mathbf{2.984 \times 10^{-4}}$	**12 307.169**
	MLEM	0.076	0.817	0.500	0.105	0.857	0.500	0.102	1.095×10^{-7}	1.803×10^{-7}	4.843×10^{-4}	12 341.205
	GLASSO	NA	NA	NA	0.818	0.804	0.496	1 073.510	4.485×10^{-6}	7.180×10^{-6}	1.000	28 459.294
	rGLASSO	NA	NA	NA	0.764	**0.924**	0.598	31.689	2.826×10^{-6}	5.492×10^{-6}	1.000	22 957.693
	GRAPHEM	**0.045**	**0.895**	**0.847**	NA	NA	NA	NA	4.364×10^{-6}	6.944×10^{-6}	2.980×10^{-4}	29 035.030
Dataset B	DGLASSO	0.068	0.833	0.603	**0.070**	0.893	**0.835**	**0.071**	$\mathbf{7.490 \times 10^{-8}}$	$\mathbf{1.236 \times 10^{-7}}$	$\mathbf{3.281 \times 10^{-4}}$	**11 806.744**
	MLEM	0.080	0.815	0.500	0.106	0.898	0.500	0.100	1.299×10^{-7}	2.133×10^{-7}	4.619×10^{-4}	11 833.448
	GLASSO	NA	NA	NA	0.827	0.826	0.505	341.873	5.069×10^{-6}	8.072×10^{-6}	1.000	27 744.964
	rGLASSO	NA	NA	NA	0.734	**0.930**	0.608	33.896	3.215×10^{-6}	6.187×10^{-6}	1.000	22 530.036
	GRAPHEM	**0.047**	**0.893**	**0.848**	NA	NA	NA	NA	5.158×10^{-6}	8.036×10^{-6}	2.912×10^{-4}	29 031.412
Dataset C	DGLASSO	0.070	0.829	0.581	**0.090**	0.954	**0.830**	**0.078**	$\mathbf{1.896 \times 10^{-7}}$	$\mathbf{2.994 \times 10^{-7}}$	$\mathbf{3.956 \times 10^{-4}}$	**10 311.104**
	MLEM	0.081	0.810	0.500	0.097	**0.974**	0.500	0.094	2.583×10^{-7}	4.180×10^{-7}	5.053×10^{-4}	10 326.410
	GLASSO	NA	NA	NA	0.901	0.805	0.489	3.926×10^{17}	0.012	0.012	1.000	26 634.892
	rGLASSO	NA	NA	NA	0.805	0.928	0.614	29.530	7.195×10^{-6}	1.320×10^{-5}	1.000	21 322.247
	GRAPHEM	**0.049**	**0.892**	**0.857**	NA	NA	NA	NA	1.055×10^{-5}	1.641×10^{-5}	3.912×10^{-4}	29 023.369
Dataset D	DGLASSO	0.073	0.835	0.575	**0.083**	**1.000**	0.598	**0.080**	$\mathbf{5.127 \times 10^{-7}}$	$\mathbf{8.243 \times 10^{-7}}$	$\mathbf{3.373 \times 10^{-4}}$	**7 911.943**
	MLEM	0.098	0.808	0.500	0.095	**1.000**	0.500	0.084	6.296×10^{-7}	1.027×10^{-6}	4.219×10^{-4}	7 923.850
	GLASSO	NA	NA	NA	0.964	0.941	0.550	187.823	2.348×10^{-5}	3.701×10^{-5}	1.000	23 684.178
	rGLASSO	NA	NA	NA	0.882	0.956	**0.645**	28.703	1.886×10^{-5}	3.239×10^{-5}	1.000	20 100.491
	GRAPHEM	**0.061**	**0.892**	**0.864**	NA	NA	NA	NA	2.503×10^{-5}	3.839×10^{-5}	3.743×10^{-4}	29 016.321

Performance of DGLASSO (climate model)

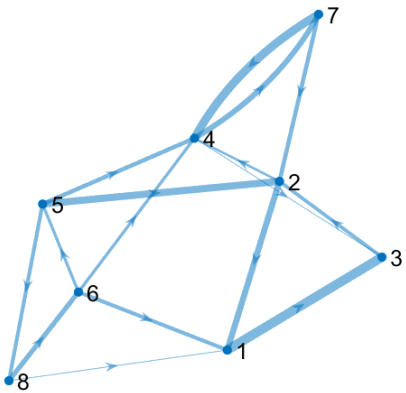
	method	RMSE	accur.	prec.	recall	spec.	F1	Time (s.)
WeathN5a	DGLASSO	**0.108**	**0.937**	0.894	0.998	0.894	**0.937**	0.608
	MLEM	0.140	0.413	0.413	**1.000**	0.000	0.584	0.596
	GRAPHEM	0.127	0.703	0.595	**1.000**	0.496	0.742	0.606
	PGC	-	0.772	**0.902**	0.515	**0.953**	0.652	0.019
	CGC	-	0.672	0.828	0.285	0.945	0.415	0.026
WeathN5b	DGLASSO	**0.166**	**0.773**	0.668	0.992	0.619	**0.788**	0.630
	MLEM	0.197	0.413	0.413	**1.000**	0.000	0.584	0.376
	GRAPHEM	0.186	0.629	0.536	**1.000**	0.368	0.694	0.470
	PGC	-	0.675	**0.677**	0.469	0.819	0.544	0.017
	CGC	-	0.634	0.659	0.263	**0.895**	0.369	0.023
WeathN10a	DGLASSO	**0.202**	**0.948**	**0.898**	0.925	0.954	**0.890**	1.363
	MLEM	0.264	0.219	0.219	**1.000**	0.000	0.359	0.834
	GRAPHEM	0.224	0.511	0.311	1.000	0.374	0.473	1.445
	PGC	-	0.879	0.904	0.504	**0.983**	0.644	0.232
	CGC	-	0.773	0.539	0.211	0.932	0.278	0.358
WeathN10b	DGLASSO	**0.192**	**0.866**	**0.633**	0.994	0.829	**0.769**	0.557
	MLEM	0.342	0.219	0.219	**1.000**	0.000	0.359	0.989
	GRAPHEM	0.219	0.855	0.620	0.994	0.816	0.757	0.655
	PGC	-	0.799	0.558	0.473	0.890	0.506	0.154
	CGC	-	0.750	0.407	0.218	**0.900**	0.265	0.178

Outline

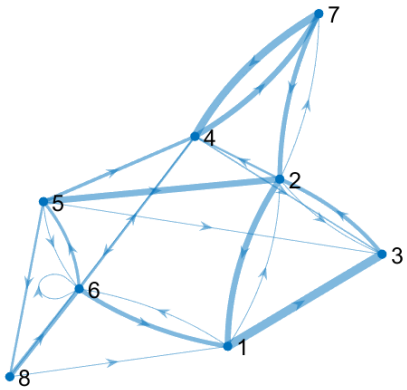
Beyond ℓ_1 norm



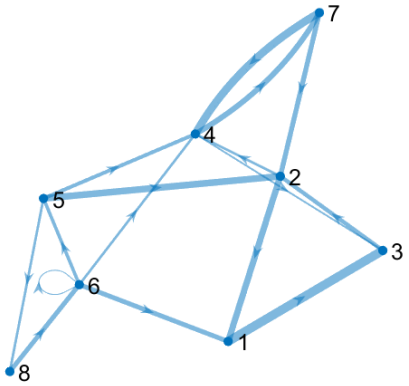
- DGLASSO requires the penalty term $\mathcal{L}_0(\mathbf{A})$ to be convex (e.g., ℓ_1 norm but not only).
- Non-convex penalties closer to pseudo-norm ℓ_0 would be better (SCAD, MCP, CEL0)
  - GraphIT algorithm[22] implements an iterative reweighted (IR) scheme
    - MM framework: $\mathcal{L}_0(\mathbf{A})$ is approximated by a surrogate convex function



(a) True graph



(b) GraphEM/DGLASSO[23]



(c) GraphIT

[22] E. Chouzenoux and V. Elvira. "GraphIT: Iterative reweighted ℓ_1 algorithm for sparse graph inference in state-space models". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5.

[23] V. Elvira and É. Chouzenoux. "Graphical Inference in Linear-Gaussian State-Space Models". In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 4757–4771.

Outline

Ongoing extensions: beyond Markovianity

- Non-Markovian LG-SSM:[24]
  - Unobserved state $\rightarrow \quad \mathbf{x}_t = \sum_{i=1}^{P} \mathbf{A}_i \mathbf{x}_{t-i} + \mathbf{q}_t$
  - Observations $\rightarrow \quad \mathbf{y}_t = \boldsymbol{H}_t \mathbf{x}_t + \mathbf{r}_t$
- Standard filtering and smoothing approach with known $\{A_i\}_{i=1}^{P}$
  - stacking (columnwise) the p consecutive states into $\mathbf{z}_t = [\mathbf{x}_t; \mathbf{x}_{t-1}; \ldots; \mathbf{x}_{t-p+1}] \in \mathbb{R}^{pN_x}$
  - run KF and RTS in the extended model

$$\begin{cases} \mathbf{z}_t = \check{\mathbf{A}} \mathbf{z}_{t-1} + \check{\mathbf{q}}_t, \\ \mathbf{y}_t = \check{\mathbf{H}} \mathbf{z}_t + \mathbf{r}_t, \end{cases} \tag{7}$$

where we define

$$\check{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_p \\ \boldsymbol{I} & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ (0) & & \boldsymbol{I} & 0 \end{bmatrix} \in \mathbb{R}^{pN_x \times pN_x},$$

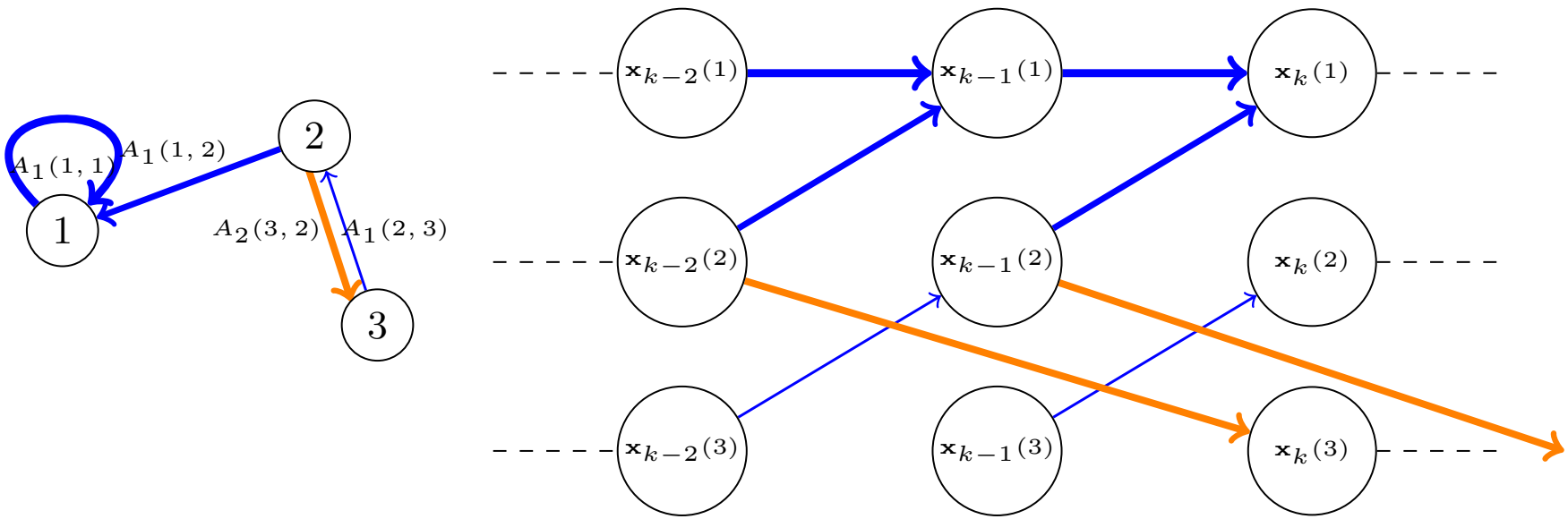
$$\check{\mathbf{H}} = [\mathbf{H}\ (0)] \in \mathbb{R}^{N_y \times pN_x}, \ \check{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & (0) \\ (0) & (0) \end{bmatrix} \in \mathbb{R}^{pN_x \times pN_x},$$

$\check{\mathbf{q}}_t \sim \mathcal{N}(0, \check{\mathbf{Q}})$, and $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R})$

[24] E. Chouzenoux and V. Elvira. "Graphical Inference in Non-Markovian Linear-Gaussian State-space Models"". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2024, pp. 1–5.

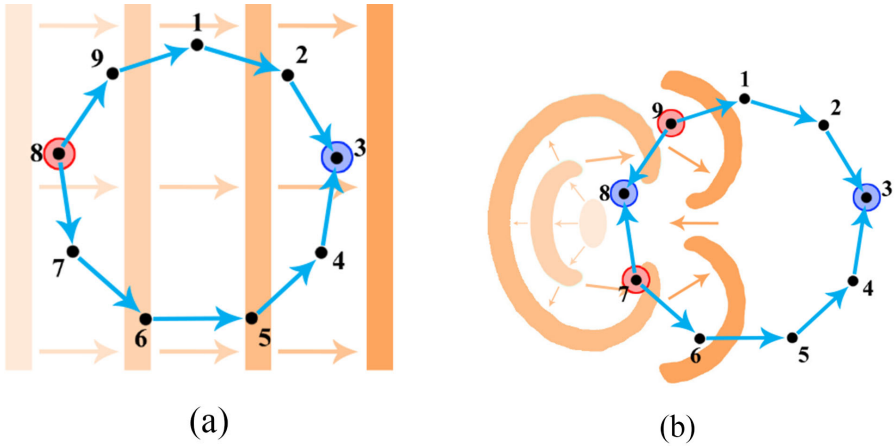
Beyond Markovianity

$$\mathbf{A}_1 = \begin{pmatrix} 0.9 & 0.7 & 0 \\ 0 & 0 & -0.3 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.8 & 0 \end{pmatrix}.$$



Ongoing extensions: beyond Markovianity

- LaGrangEM (ICASSP 2024): learn $\check{\mathbf{A}}$ non-Markovian models including desirable properties and interpretability, e.g.,
  - acyclic graph
  - sparsity
  - only one-lag interaction at maximum betwen nodes (more sparsity!)
    - reasonable in some physical models
  - one input arrow at maximum at each node (even more sparsity!)
    - strong connection with modern Granger causality models[25]



- So far, great results but with intermediate/post-processing mapping steps which may compromise the theoretical guarantees (?)
  - ongoing work in bridging the gap between well-performing methods and solid theory

[25] D. Luengo, G. Rios-Munoz, V. Elvira, C. Sanchez, and A. Artes-Rodriguez. "Hierarchical algorithms for causality retrieval in atrial fibrillation intracavitary electrograms". In: *IEEE journal of biomedical and health informatics* 23.1 (2018), pp. 143–155.