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) and B. Cox (University of Edinburgh, UK)

Institute of Mathematics, University of Potsdam
December 6, 2024

Outline

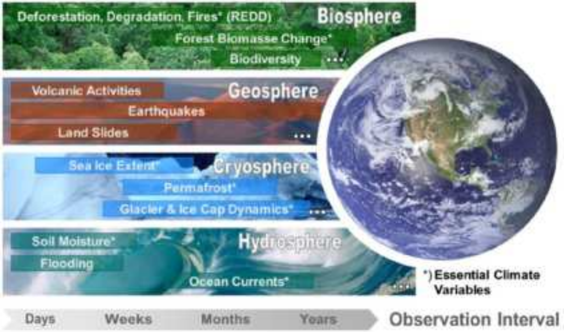
Dynamical systems

- **Dynamical systems** are composed of elementary **units** whose evolution depends on their **local features** and **interactions over time**.[1]

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

Dynamical systems

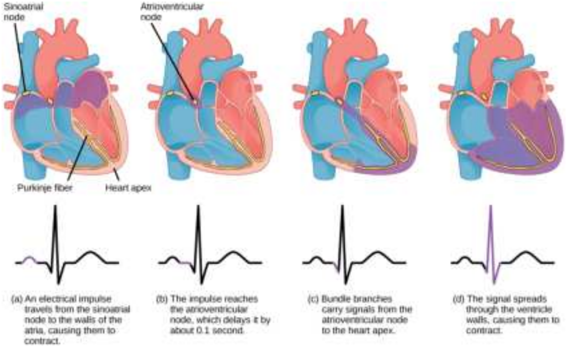
- **Dynamical systems** are composed of elementary **units** whose evolution depends on their **local features** and **interactions over time**.[1]
  - The Earth is formed by dynamical subsystems interacting at different scales in time and space (e.g., biosphere, atmosphere, etc.)



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

Dynamical systems

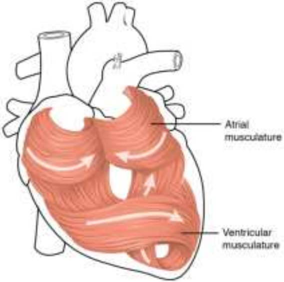
- **Dynamical systems** are composed of elementary **units** whose evolution depends on their **local features** and **interactions over time**.[1]
  - The heart is a dynamical system at different scales (electrical and physical)

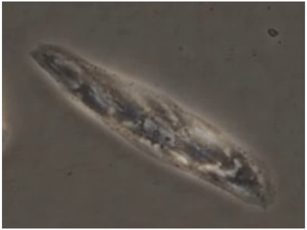


[1] D. J. Watts and S. H. Strogatz. "Collective dynamics of small-world networks". In: *Nature* 393.6684 (1998), pp. 440–442.

Dynamical systems

- **Dynamical systems** are composed of elementary **units** whose evolution depends on their **local features** and **interactions over time**.[1]
  - The heart is a dynamical system at different scales (electrical and physical)





[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 systems** are composed 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)
    - heart electro-dynamics
    - popluation ecology (pray-predator interactions)
    - climate
    - brain
    - robotics with target tracking, positioning, navigation
    - wireless communications in automobiles
    - financial markets

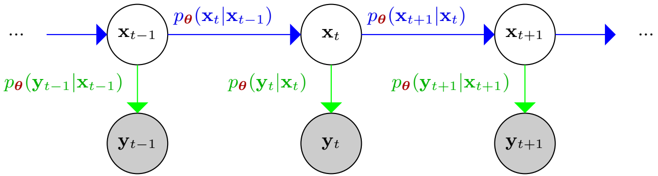
[1] D. J. Watts and S. H. Strogatz. "Collective dynamics of small-world networks". In: *Nature* 393.6684 (1998), pp. 440–442.

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 $\mathbf{x}_t$.
  - $\mathbf{y}_{1:t} \equiv \{\mathbf{y}_1, ..., \mathbf{y}_t\}$.
- 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 (with interpretability)

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

The linear-Gaussian model

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

    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 → $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \equiv \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t \mathbf{x}_{t-1}, \mathbf{Q}_t)$
    - Observations → $p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t \mathbf{x}_t, \mathbf{R}_t)$
- Methods (known $\boldsymbol{\theta}$):
  - **Kalman filter**: obtains the filtering pdfs $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ at each t
    - Gaussian pdfs (i.e., compute means and covariance matrices)
    - Efficient processing of $\mathbf{y}_t$ from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
    - requires a backward reprocessing, refining the Kalman estimates
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - expectation-maximization (EM)
    - maximum likelihood (ML)
  - fully Bayesian Monte Carlo methods
    - particle Metropolis
    - particle Gibbs
    - ...

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
    - Gaussian pdfs (i.e., compute means and covariance matrices)
    - Efficient processing of $\mathbf{y}_t$ from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
    - requires a backward reprocessing, refining the Kalman estimates
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - expectation-maximization (EM)
    - maximum likelihood (ML)
  - fully Bayesian Monte Carlo methods
    - particle Metropolis
    - particle Gibbs
    - ...

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
    - Gaussian pdfs (i.e., compute means and covariance matrices)
    - Efficient processing of $\mathbf{y}_t$ from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
    - requires a backward reprocessing, refining the Kalman estimates
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - expectation-maximization (EM)
    - maximum likelihood (ML)
  - fully Bayesian Monte Carlo methods
    - particle Metropolis
    - particle Gibbs
    - ...

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
    - Gaussian pdfs (i.e., compute means and covariance matrices)
    - Efficient processing of $\mathbf{y}_t$ from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
    - requires a backward reprocessing, refining the Kalman estimates
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - expectation-maximization (EM)
    - maximum likelihood (ML)
  - fully Bayesian Monte Carlo methods
    - particle Metropolis
    - particle Gibbs
    - ...

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
    - Gaussian pdfs (i.e., compute means and covariance matrices)
    - Efficient processing of $\mathbf{y}_t$ from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
    - requires a backward reprocessing, refining the Kalman estimates
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - expectation-maximization (EM)
    - maximum likelihood (ML)
  - fully Bayesian Monte Carlo methods
    - particle Metropolis
    - particle Gibbs
    - ...

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
    - Gaussian pdfs (i.e., compute means and covariance matrices)
    - Efficient processing of $\mathbf{y}_t$ from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
  - **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
    - requires a backward reprocessing, refining the Kalman estimates
- Methods (unknown $\boldsymbol{\theta}$; build upon KF/RTS):
  - Point-wise:
    - expectation-maximization (EM)
    - maximum likelihood (ML)
  - fully Bayesian Monte Carlo methods
    - particle Metropolis
    - particle Gibbs
    - ...

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 $\boldsymbol{\theta} = [\mathbf{A}; \mathbf{Q}]$

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 $\mathbf{A}$

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

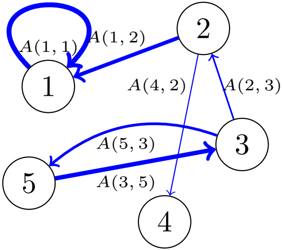
- $\mathbf{A}$ interpreted as a **sparse directed graph**

- $\mathbf{x}_t \in \mathbb{R}^{N_x}$ contains N_x time-series
  - each of them represents the latent process in a node in the graph
- $A(i,j)$ 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 $x_{t,i}$.

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



A graphical perspective on $\mathbf{A}$

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

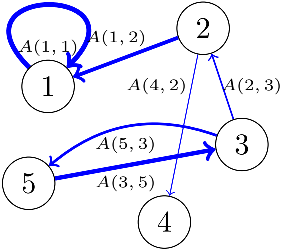
- $\mathbf{A}$ interpreted as a **sparse directed graph**

  - $\mathbf{x}_t \in \mathbb{R}^{N_x}$ contains N_x time-series
    - each of them represents the latent process in a node in the graph
  - $A(i,j)$ 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 $x_{t,i}$.

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



A graphical perspective on $\mathbf{A}$

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

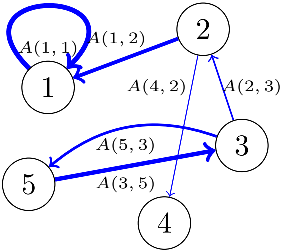
- $\mathbf{A}$ interpreted as a **sparse directed graph**

  - $\mathbf{x}_t \in \mathbb{R}^{N_x}$ contains N_x time-series
    - each of them represents the latent process in a node in the graph
  - $A(i,j)$ 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 $x_{t,i}$.

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

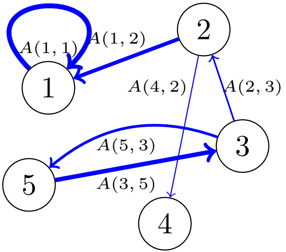


A graphical perspective on $\mathbf{A}$

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

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

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



- $\mathbf{x}_t \in \mathbb{R}^{N_x}$ contains N_x time-series
  - each of them represents the latent process in a node in the graph
- $A(i,j)$ 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 $\mathbf{A}$

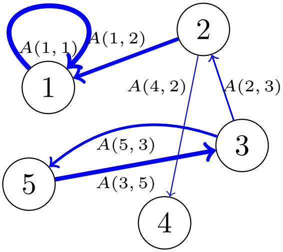
$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{q}_t, \qquad \mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q})$$

- $\mathbf{A}$ interpreted as a **sparse directed graph**
  - $\mathbf{x}_t \in \mathbb{R}^{N_x}$ contains N_x time-series
    - each of them represents the latent process in a node in the graph
  - $A(i,j)$ 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 $x_{t,i}$.

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



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!)

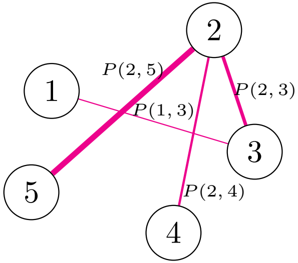
A graphical modeling $\mathbf{P} = \mathbf{Q}^{-1}$

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

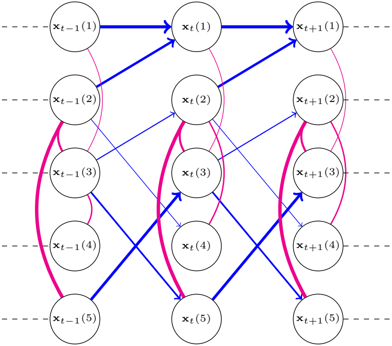
- $\mathbf{P} = \mathbf{Q}^{-1}$ interpreted as **sparse undirected graph** (Gaussian graphical models).

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

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



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:

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

- 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

Proposed penalized formulation

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

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

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:

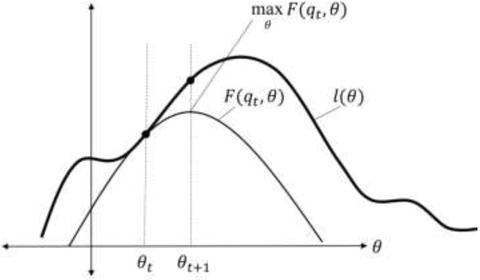
$$\mathcal{L}_{1:T}(\mathbf{A}, \mathbf{P}) = \sum_{t=1}^{T} \tfrac{1}{2} \log |2\pi \mathbf{S}_t(\mathbf{A}, \mathbf{P})| + \frac{1}{2} \mathbf{z}_t(\mathbf{A}, \mathbf{P})^\top \mathbf{S}_t(\mathbf{A}, \mathbf{P})^{-1} \mathbf{z}_t(\mathbf{A}, \mathbf{P}).$$

- 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

EM approach for ML estimation



(credit to M. N. Bernstein)

- EM approach for ML:[2] 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 $\mathbf{P}^{(i+1)}$.

[2] 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.

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

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

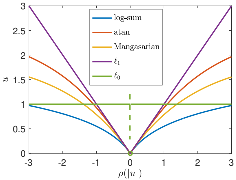
- converges to a critical point of $\mathcal{L}(\mathbf{A}, \mathbf{P})$.

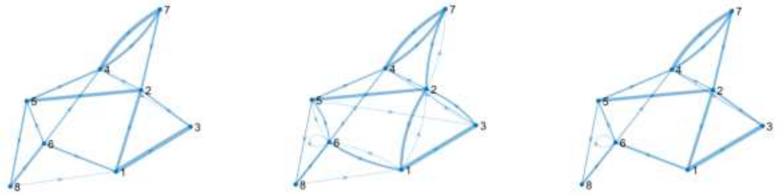
- Proof based on the recent work.[5]

[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.

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[6] 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[7] (c) GraphIT

[6] 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.

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

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 $T = 10^3$, $\mathbf{Q} = \sigma_{\mathbf{Q}}^2 \mathbf{Id}$, $\mathbf{R} = \sigma_{\mathbf{R}}^2 \mathbf{Id}$, $\mathbf{P}_0 = \sigma_{\mathbf{P}}^2 \mathbf{Id}$.

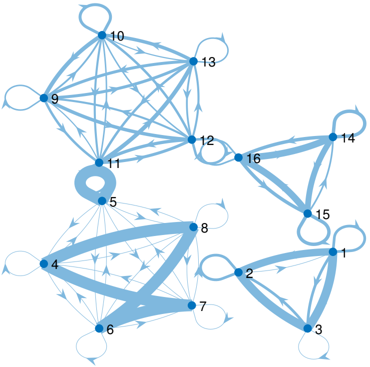
Dataset	N_x	$(b_j)_{1 \leq j \leq b}$	$(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$
A	9	$(3,3,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
B	9	$(3,3,3)$	$(1, 1, 10^{-4})$
C	16	$(3,5,5,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
D	16	$(3,5,5,3)$	$(1, 1, 10^{-4})$

- GraphEM (DGLASSO with known $\mathbf{Q}$) is compared with:
  - Maximum likelihood EM (MLEM)[8]
  - Granger-causality approaches: pairwise Granger Causality (PGC) and conditional Granger Causality (CGC)[9]

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

[9] 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

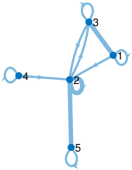


True graph associated to $\mathbf{A}$ (left) and GraphEM estimate (right) for dataset C.

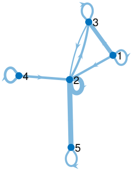
Experimental results of estimating $\mathbf{A}$ with GraphEM

	method	RMSE	accur.	prec.	recall	spec.	F1
A	GraphEM	0.081	0.9104	0.9880	0.7407	0.9952	**0.8463**
	MLEM	0.149	0.3333	0.3333	1	0	0.5
	PGC	-	0.8765	0.9474	0.6667	0.9815	0.7826
	CGC	-	0.8765	1	0.6293	1	0.7727
B	GraphEM	0.082	0.9113	0.9914	0.7407	0.9967	**0.8477**
	MLEM	0.148	0.3333	0.3333	1	0	0.5
	PGC	-	0.8889	1	0.6667	1	0.8
	CGC	-	0.8889	1	0.6667	1	0.8
C	GraphEM	0.120	0.9231	0.9401	0.77	0.9785	**0.8427**
	MLEM	0.238	0.2656	0.2656	1	0	0.4198
	PGC	-	0.9023	0.9778	0.6471	0.9949	0.7788
	CGC	-	0.8555	0.9697	0.4706	0.9949	0.6337
D	GraphEM	0.121	0.9247	0.9601	0.7547	0.9862	**0.8421**
	MLEM	0.239	0.2656	0.2656	1	0	0.4198
	PGC	-	0.8906	0.9	0.6618	0.9734	0.7627
	CGC	-	0.8477	0.9394	0.4559	0.9894	0.6139

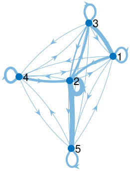
Experimental results: Realistic weather datasets



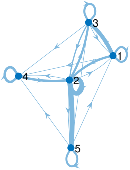
True



DGLASSO



MLEM



GRAPHEM

PGC

CGC

Graph inference results on an example from WeathN5a dataset.[10]

[10] J. Runge, X.-A. Tibau, M. Bruhns, J. Muoz-Mar, and G. Camps-Valls. The causality for climate competition. In Proceedings of the NeurIPS 2019 Competition and Demonstration Track, volume 123, pages 110–120, 2020.

Computational complexity of DGLASSO

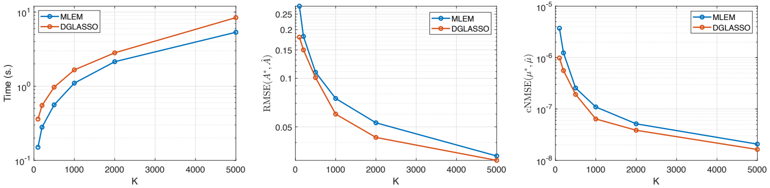
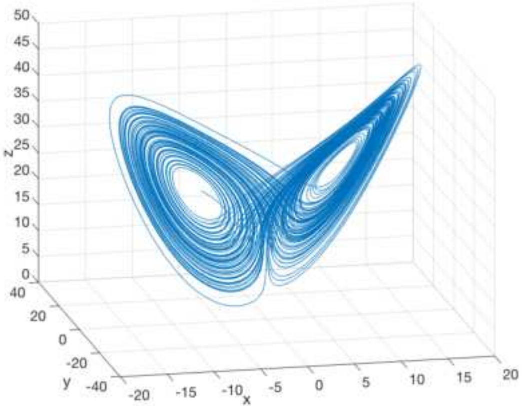


Figure 6: Evolution of the complexity time (left), RMSE$(\mathbf{A}^*, \widehat{\mathbf{A}})$ (middle) and cNMSE$(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$ (right) metrics, as a function of the time series length K, for experiments on dataset A averaged over 50 runs.

Outline

Motivating example: Lorenz 63

- Lorenz system: non-linear and continuous time model[11]
  - it can have chaotic behavior
    - when the present determines the future, but the approximate present does not approximately determine the future.
  - it captures the essence of atmospheric convection.



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

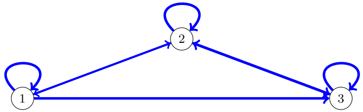
Motivating example: Lorenz 63

- Lorenz 63 equations:

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

  - $(\sigma, \rho, \beta) = \left(10, 28, \frac{8}{3}\right)$: 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

We consider d-degree polynomial model on $\mathbf{x}_t \in \mathbb{R}^{N_x}$:

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

with

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

- 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

- $N_x = 3$ dimensions in the Lorenz 63 model
  - exactly represented with max degree of polynomial: $d = 2$
- Set of $M = 10$ monomials up to degree $d = 2$:

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

with associated degree matrix

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

- Discretized (Euler-Maruyama) 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 $q_{t,i} \sim \mathcal{N}(0, \Delta t)$.

- The M monomials are used at each dimension through coefficients in

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

GraphGrad algorithm

- GraphGrad algorithm:[12]
  - observe $\mathbf{y}_{1:T}$ associated to $\mathbf{x}_{1:T}$ (e.g., noisy version of one unique dimension)
  - learn the coefficient matrix $\mathbf{C}$ using a MAP estimator under a sparsity inducing penalty
    - first-order optimisation scheme (proximal-gradient method)

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

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

  - gradients of the log-likelihood, approximated via diff. particle filtering

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

  - penalty term R using its proximity operator, which is both faster and avoids the requirement for R to be differentiable.

[12] B. Cox, E. Chouznoux, and V. Elvira. "GraphGrad: Efficient Estimation of Sparse Polynomial Representations for General State-Space Models". In: *arXiv preprint arXiv:2411.15637* (2024).

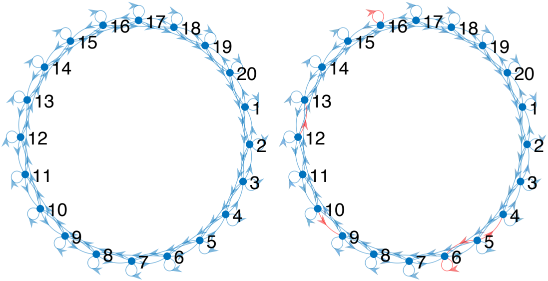
GraphGrad in Lorenz 96

Lorenz 96 system with variable dimension:[13]

$$x_{t+1,i} = (1-\Delta t)x_{t,i} + \Delta t x_{t,i-1}x_{t,i+1} - \Delta t x_{t,i-2} + F\Delta t + \sqrt{\Delta t}\cdot q_{t+1,i},$$
$$y_{t+1,i} = x_{t+1,i} + \sqrt{\Delta t}\cdot r_{i,t+1}, \qquad (5)$$

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 system)
- $N_x = N_y = 20$
  - if $d = 2$, $\mathbf{C}$ has 4620 parameters
  - if $d = 3$, $\mathbf{C}$ has 35420 parameters
- True graph vs GraphGrad estimation:



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

Outline

Ongoing extensions: beyond Markovianity

- Non-Markovian LG-SSM:[14]
  - Unobserved state $\rightarrow$ $\mathbf{x}_t = \sum_{i=1}^{P} \mathbf{A}_i \mathbf{x}_{t-i} + \mathbf{q}_t$
  - Observations $\rightarrow$ $\mathbf{y}_t = \boldsymbol{H}_t \mathbf{x}_t + \mathbf{r}_t$
- 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

$$\begin{cases} \mathbf{z}_t = \check{\mathbf{A}} \mathbf{z}_{t-1} + \check{\mathbf{q}}_t, \\ \mathbf{y}_t = \check{\mathbf{H}} \mathbf{z}_t + \mathbf{r}_t, \end{cases} \tag{6}$$

where we define

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

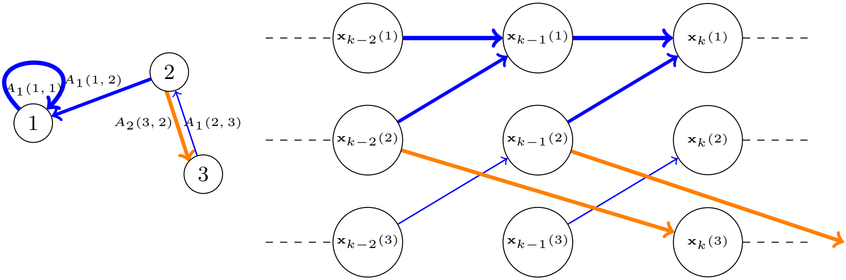
$$\check{\mathbf{H}} = [\mathbf{H} \ (0)] \in \mathbb{R}^{N_y \times pN_x}, \ \check{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & (0) \\ (0) & (0) \end{bmatrix} \in \mathbb{R}^{pN_x \times pN_x},$$

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

[14] 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.

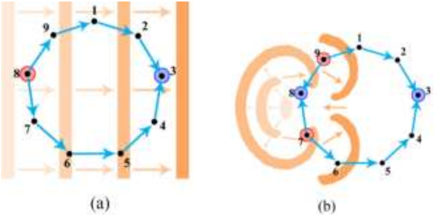
Beyond Markovianity

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



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[15]



- 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-peroming methods and solid theory

[15] 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.

Outline

SpaRJ algorithm

- SpaRJ[16] (*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 $p(\mathbf{A}|\mathbf{y}_{1:T})$.[17]
  - 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.

[16] 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.

[17] 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[16] (*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 $p(\mathbf{A}|\mathbf{y}_{1:T})$.[17]
  - 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.

[16] 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.

[17] 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[16] (*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 $p(\mathbf{A}|\mathbf{y}_{1:T})$.[17]
  - 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.

[16] 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.
[17] 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[16] (*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 $p(\mathbf{A}|\mathbf{y}_{1:T})$.[17]
  - 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.

[16] 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.
[17] 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 A_0
Run Kf obtaining l_0 := log(p(y_{1:T}|A_0))p(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 A' using an MCMC sampler conditional on M'
    Step 3: MH accept-reject
    Evaluate Kalman filter with A := A'
    Set l' := log(p(y_{1:T}|A'))p(A')
    Compute log(a_r) := l' - l_{n-1} + c and Accept w.p. a_r:
    if Accept then
        Set M_n := M', A_n := A', l_n := log(p(y_{1:T}|A'))p(A')
    else
        Set M_n := M_{n-1}, A_n := 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

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

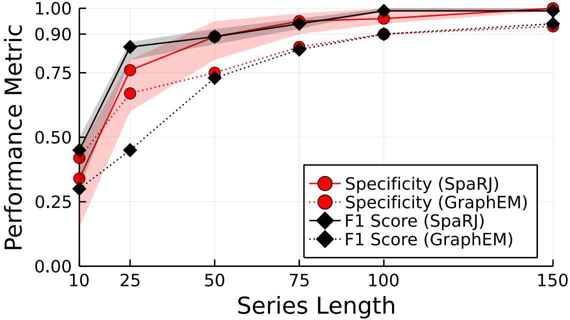


Figure: 3×3 system with known isotropic state covariance.

Convergence of SpaRJ with iterations

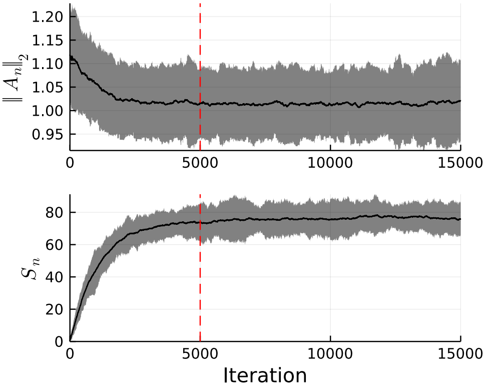


Figure: Progression of sample metrics in a 12×12.

SpaRJ with real world data

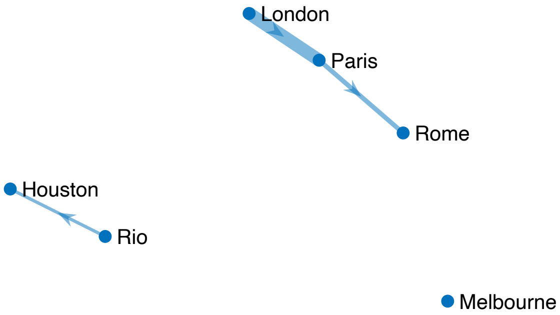


Figure: Average daily temperature of 324 cities from 1995 to 2021, curated by the United States Environmental Protection Agency.

Outline

Conclusion

- SSMs are very powerful tools but still underdeveloped due to conceptual and computational limiations.
- Novel graphical interpretation on matrices $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSMs.
- Algorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM is faster and allows explicit penalty functions (prior knowledge) beyond sparsity.
  - SpaRJ provides samples of the posterior allowing for uncertainty quantification.
- Algorithm to estimate both sparse $\mathbf{A}$ and $\mathbf{Q}$: DGLASSO (point-wise)
- All have solid theoretical guarantees and show good performance.
- Current efforts to go beyond Markovianity, linearity, Gaussianity and more uncertainty quantification.
  - This is a challenging problem with many exciting ongoing methodological and applied avenues ahead!

Conclusion

- SSMs are very powerful tools but still underdeveloped due to conceptual and computational limiations.
- Novel graphical interpretation on matrices $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSMs.
- Algorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM is faster and allows explicit penalty functions (prior knowledge) beyond sparsity.
  - SpaRJ provides samples of the posterior allowing for uncertainty quantification.
- Algorithm to estimate both sparse $\mathbf{A}$ and $\mathbf{Q}$: DGLASSO (point-wise)
- All have solid theoretical guarantees and show good performance.
- Current efforts to go beyond Markovianity, linearity, Gaussianity and more uncertainty quantification.
  - This is a challenging problem with many exciting ongoing methodological and applied avenues ahead!

Conclusion

- SSMs are very powerful tools but still underdeveloped due to conceptual and computational limiations.
- Novel graphical interpretation on matrices $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSMs.
- Algorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM is faster and allows explicit penalty functions (prior knowledge) beyond sparsity.
  - SpaRJ provides samples of the posterior allowing for uncertainty quantification.
- Algorithm to estimate both sparse $\mathbf{A}$ and $\mathbf{Q}$: DGLASSO (point-wise)
- All have solid theoretical guarantees and show good performance.
- Current efforts to go beyond Markovianity, linearity, Gaussianity and more uncertainty quantification.
  - This is a challenging problem with many exciting ongoing methodological and applied avenues ahead!

Conclusion

- SSMs are very powerful tools but still underdeveloped due to conceptual and computational limiations.
- Novel graphical interpretation on matrices $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSMs.
- Algorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM is faster and allows explicit penalty functions (prior knowledge) beyond sparsity.
  - SpaRJ provides samples of the posterior allowing for uncertainty quantification.
- Algorithm to estimate both sparse $\mathbf{A}$ and $\mathbf{Q}$: DGLASSO (point-wise)
- All have solid theoretical guarantees and show good performance.
- Current efforts to go beyond Markovianity, linearity, Gaussianity and more uncertainty quantification.
  - This is a challenging problem with many exciting ongoing methodological and applied avenues ahead!

Conclusion

- SSMs are very powerful tools but still underdeveloped due to conceptual and computational limiations.
- Novel graphical interpretation on matrices $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSMs.
- Algorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM is faster and allows explicit penalty functions (prior knowledge) beyond sparsity.
  - SpaRJ provides samples of the posterior allowing for uncertainty quantification.
- Algorithm to estimate both sparse $\mathbf{A}$ and $\mathbf{Q}$: DGLASSO (point-wise)
- All have solid theoretical guarantees and show good performance.
- Current efforts to go beyond Markovianity, linearity, Gaussianity and more uncertainty quantification.
  - This is a challenging problem with many exciting ongoing methodological and applied avenues ahead!

Conclusion

- SSMs are very powerful tools but still underdeveloped due to conceptual and computational limiations.
- Novel graphical interpretation on matrices $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSMs.
- Algorithms to estimate only a sparse $\mathbf{A}$: GraphEM (point-wise) and SpaRJ (fully Bayesian).
  - GraphEM is faster and allows explicit penalty functions (prior knowledge) beyond sparsity.
  - SpaRJ provides samples of the posterior allowing for uncertainty quantification.
- Algorithm to estimate both sparse $\mathbf{A}$ and $\mathbf{Q}$: DGLASSO (point-wise)
- All have solid theoretical guarantees and show good performance.
- Current efforts to go beyond Markovianity, linearity, Gaussianity and more uncertainty quantification.
  - This is a 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): E. 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", arxiv:2411.15637, 2024.

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), Rhodes, Greece, June, 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), Seoul, Korea, April, 2024.