THE UNIVERSITY of EDINBURGH

State space models and Kalman filtering (L2)

Víctor Elvira
School of Mathematics
University of Edinburgh
(victor.elvira@ed.ac.uk)

PhD course on Bayesian filtering and Monte Carlo methods
UC3M, Madrid, February 3-7, 2025

Outline

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother

Nonlinear Kalman filters

Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

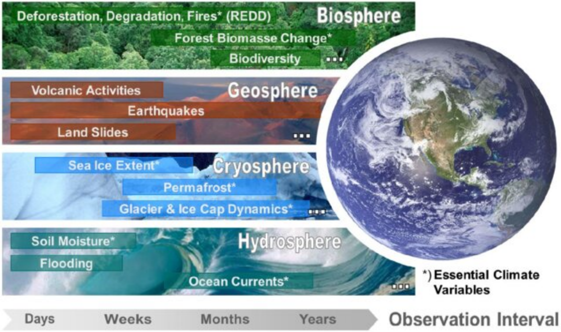
Dynamical systems

- **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

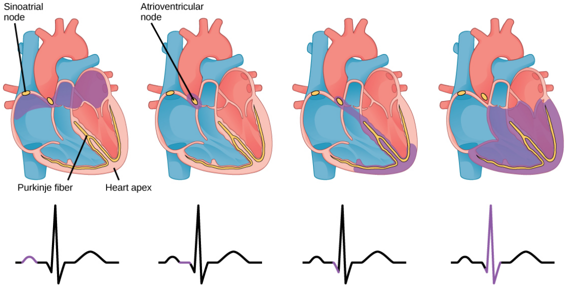
- **Dynamical systems** are composed of elementary **units** whose evolution depends on their **local features** and **interactions over time**.[1]
  - The Earth is formed by dynamical subsystems interacting at different scales in time and space (e.g., biosphere, atmosphere, etc.)



[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)



(a) An electrical impulse travels from the sinoatrial node to the walls of the atria, causing them to contract.

(b) The impulse reaches the atrioventricular node, which delays it by about 0.1 second.

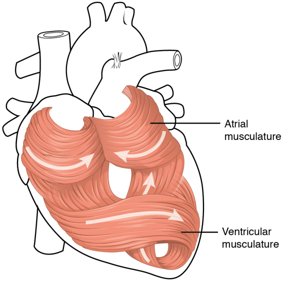
(c) Bundle branches carry signals from the atrioventricular node to the heart apex.

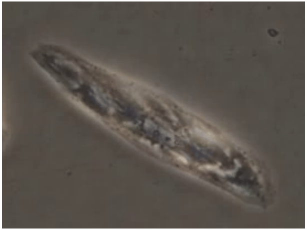
(d) The signal spreads through the ventricle walls, causing them to contract.

[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, machine learning, signal processing, ... AI?

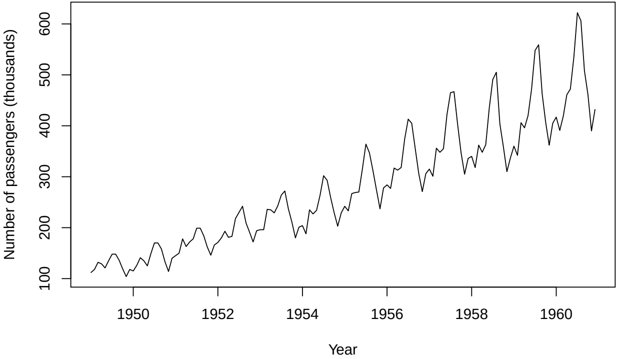
Time series: deterministic vs stochastic

- A time series is a collection of observations/measurements made sequentially through time.
- A time series is said to be **continuous** when observations are made continuously through time. The observations themselves may still be discrete. discrete or continuous).
- A time series is said to be **discrete** when observations are made at discrete time points (e.g. the air temperature measured each day). The observations y_t may be discrete or continuous.
  - **This lecture focus on discrete time series**, with equally spaced times (e.g. measurements are made at regular intervals).
    - Notation: $y_t \in \mathbb{R}^{d_y}$ made at times $t = 1, 2, 3, \ldots n$.
    - Remark: Time is measured in suitable units (e.g. minutes, days, years).
- Further reading:
  - Prado, R., & West, M. (2010). Time series: modeling, computation, and inference. CRC Press.
  - Kitagawa, G. (2010). Introduction to time series modeling. CRC press.

Time series: deterministic vs stochastic

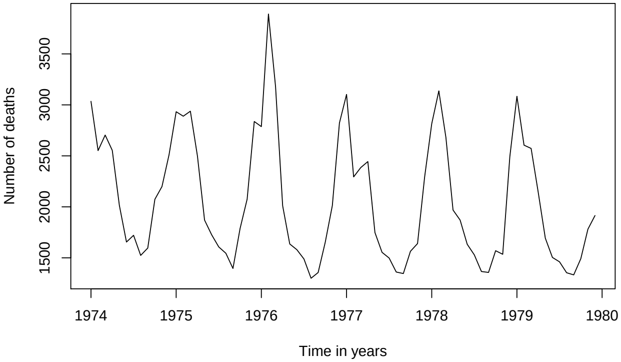
- Successive observations in a time series are often not independent.
- This means that past observations can be used to *predict* future observations.
- If, given the past observations $y_1, \ldots y_{t-1}$, the observation y_t can be predicted exactly, the times series is known as **deterministic**.
- If future observations cannot be predicted exactly, the time series is said to be **stochastic**.
- In a stochastic series, future observations will have a probability distribution.
    - If the observations are dependent, then this probability distribution is dependent on past observations in the series.
        - $p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$

Examples



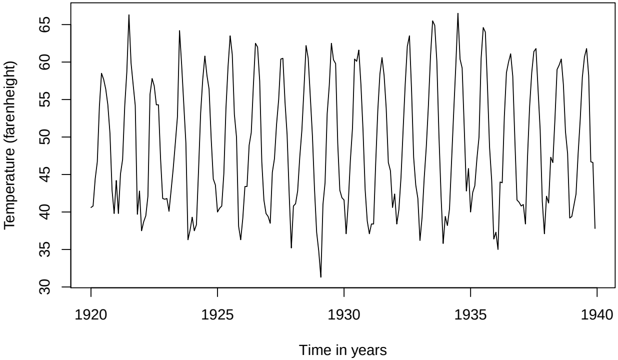
Monthly totals of international airline passengers in the USA, from January 1949 to December 1960.

Examples



Data on the monthly deaths from bronchitis, emphysema, and asthma in the UK, 1974-1979

Examples



Average air temperatures at Nottingham Castle in degrees Fahrenheit for 20 years, measured monthly.

Outline

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother

Nonlinear Kalman filters

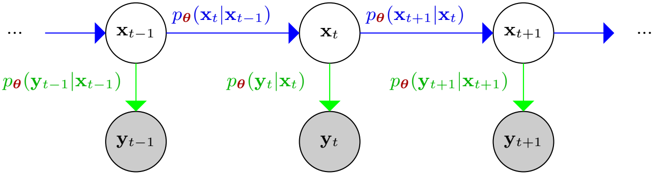
Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

1. Modeling: state-space models (SSM)

- A SSM models a sequence of hidden states $\mathbf{x}_t \in \mathbb{R}^{N_x}$, $t = 1, ..., T$.
  - it captures the state and dynamics of a system
- Time-series data are collected, $\mathbf{y}_t \in \mathbb{R}^{N_y}$, $t = 1, ..., T$:
  - noisy and partial version of the system state



- Probabilistic notation of a (simple) Markovian SSM:
  - state model $\rightarrow$ $p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\theta})$
  - observation model $\rightarrow$ $p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t) = p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta})$
  - prior on initial state $\rightarrow$ $p_{\boldsymbol{\theta}}(\mathbf{x}_0) = p(\mathbf{x}_0|\boldsymbol{\theta})$

2. Estimation/inference problems

- We sequentially observe data $\mathbf{y}_t$ related to the hidden state $\mathbf{x}_t$.
  - At time t, we have accumulated t observations, $\mathbf{y}_{1:t} \equiv \{\mathbf{y}_1, ..., \mathbf{y}_t\}$.
- Wish list:
  - prediction of future observations and estimation of states (with uncertainty quantification)
    - **Filtering**: $p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{y}_{1:t})$ and joint $p_{\boldsymbol{\theta}}(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$
    - State prediction: $p_{\boldsymbol{\theta}}(\mathbf{x}_{t+\tau}|\mathbf{y}_{1:t}), \qquad \tau \geq 1$
    - Observation prediction: $p_{\boldsymbol{\theta}}(\mathbf{y}_{t+\tau}|\mathbf{y}_{1:t}), \qquad \tau \geq 1$
    - **Smoothing**: $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-\tau}|\mathbf{y}_{1:t}), \qquad \tau \geq 1$
  - estimation of model parameters (with interpretability)

- Bayesian/probabilistic inference:
  - we compute or approximate pdfs of unknowns when possible (instead of point-wise estimates)

Bayesian filtering

- Bayesian rule for the joint:

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

- Filtering distribution as a marginal:

$$p(\mathbf{x}_T|\mathbf{y}_{1:T}) = \int p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})d\mathbf{x}_1 d\mathbf{x}_2 ... d\mathbf{x}_{T-1}$$

- Problems:
  - Dimension: $\mathbf{x}_{1:T} \in \mathbb{R}^{T \cdot d_x}$
  - When we receive $\mathbf{y}_t$, we don't want to reprocess $\mathbf{y}_{1:t-1}$

Goal: **efficient and sequential** Bayesian inference

Sequential optimal filtering

- Filtering Problem:
  - Distribution of $\mathbf{x}_t$ given all the obs. up to time t, $p(\mathbf{x}_t|\mathbf{y}_{1:t})$
  - Recursively from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ updating with the new $\mathbf{y}_t$
- Optimal filtering:
  1. **Prediction** step:

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

  2. **Update** step:

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

- Interest in integrals of the form: $I(f) = \int f(\mathbf{x}_t)p(\mathbf{x}_t|y_{1:t})d\mathbf{x}_t$
  - e.g., the mean, $I(f) = \int \mathbf{x}_t p(\mathbf{x}_t|y_{1:t})d\mathbf{x}_t$

Outline

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother

Nonlinear Kalman filters

Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

The linear-Gaussian model

- The linear-Gaussian model 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 $\rightarrow$ $p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t \mathbf{x}_t, \mathbf{R}_t)$
- **Kalman filter**: obtains the filtering pdfs $p(\mathbf{x}_t|\mathbf{y}_{1:t})$at each t (if known $\boldsymbol{\theta}$)
  - Gaussian pdfs (i.e., compute means and covariance matrices)
  - Efficient processing of $\mathbf{y}_t$ from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
    - only $\mathbf{y}_t$ is processed at time t
- **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
  - requires a backward reprocessing, refining the Kalman estimates

The linear-Gaussian model

- The linear-Gaussian model 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 $\rightarrow$ $p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t \mathbf{x}_t, \mathbf{R}_t)$
- **Kalman filter**: obtains the filtering pdfs $p(\mathbf{x}_t|\mathbf{y}_{1:t})$at each t (if known $\boldsymbol{\theta}$)
  - Gaussian pdfs (i.e., compute means and covariance matrices)
  - Efficient processing of $\mathbf{y}_t$ from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
    - only $\mathbf{y}_t$ is processed at time t
- **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t|\mathbf{y}_{1:T})$
  - requires a backward reprocessing, refining the Kalman estimates

The linear-Gaussian model

- The linear-Gaussian model 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 $\rightarrow$ $p(\mathbf{y}_t | \mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t \mathbf{x}_t, \mathbf{R}_t)$
- **Kalman filter**: obtains the filtering pdfs $p(\mathbf{x}_t | \mathbf{y}_{1:t})$at each t (if known $\boldsymbol{\theta}$)
  - Gaussian pdfs (i.e., compute means and covariance matrices)
  - Efficient processing of $\mathbf{y}_t$ from $p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$
    - only $\mathbf{y}_t$ is processed at time t
- **Rauch-Tung-Striebel (RTS) smoother**: obtains $p(\mathbf{x}_t | \mathbf{y}_{1:T})$
  - requires a backward reprocessing, refining the Kalman estimates

Outline

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother

Mini-project 2: KF

Nonlinear Kalman filters

Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

Kalman Filter: a bit of history

- Rudolf E. Kálmán (Hungary 1930 - USA 2016) developed the famous Kalman filter algorithm[2]
  - The second paper was rejected by an electrical engineering journal with a comment of a referee saying "it cannot possibly be true" (now it has +9k citations)[3]
- The on-board computer that guided the descent of the **Apollo 11** lunar module to the moon had a Kalman filter to track its trajectory!

[2] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82 (1960), pp. 35–45.

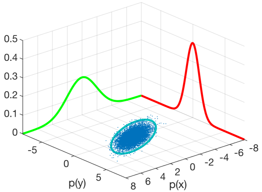
[3] R. E. Kalman and R. S. Bucy. "New results in linear filtering and prediction theory". In: (1961).

Kalman filter: Gaussian properties

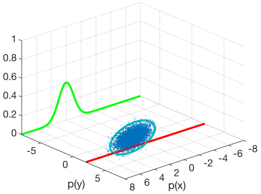
- **Gaussian distribution**:
  1. **Product** of two Gaussian distributions is still a Gaussian distribution:
  $$p(a|b)p(b) = p(a, b).$$
  $p(a, b)$ is Gaussian.
  2. **Marginalization** of a joint Gaussian distribution is still Gaussian:
  $$p(a) = \int p(b, a)db,$$
  marginalizing b, $p(a)$ is also Gaussian
  3. **Conditional** of a joint Gaussian distribution is still Gaussian (equivalent to first point):
  $$p(a|b) = \frac{p(a, b)}{p(b)}.$$

Kalman filter: Gaussian properties (graphical)

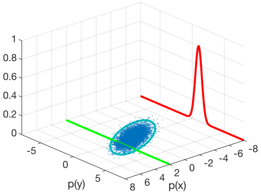
- Marginals of a bi-variate Gaussian distribution are Gaussian:



- Conditionals of a bi-variate Gaussian distribution are Gaussian:



$p(x|y=2)$



$p(y|x=2)$

Kalman Filter: prediction step

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 $p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \equiv \mathcal{N}(\mathbf{x}_t^-, \mathbf{P}_t^-)$
  - Mean: $\mathbf{x}_t^- = \mathbf{A}_t\mathbf{m}_{t-1}$
  - Variance: $\mathbf{P}_t^- = \mathbf{A}_t\mathbf{P}_{t-1}\mathbf{A}_t^T + \mathbf{Q}_t$

- Interpretation:
  - The mean is projected by the propagation matrix $\mathbf{A}_t$
  - The **uncertainty** is propagated 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: $\mathbf{m}_t = \mathbf{x}_t^- + \mathbf{K}_t \left(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^-\right)$
  - Variance: $\mathbf{P}_t = (\boldsymbol{I} - \mathbf{K}_t\mathbf{H}_t)\,\mathbf{P}_t^-$

  where $\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}_t^T\left(\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t\right)^{-1}$ 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 $\mathbf{H}_t$ and divided by the covariance of the residual/error.

Kalman summary and RTS smoother

- 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 → $p(\mathbf{y}_t|\mathbf{x}_t) \equiv \mathcal{N}(\mathbf{y}_t; \boldsymbol{H}_t\mathbf{x}_t, \mathbf{R}_t)$

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 $\boldsymbol{\theta} = [\mathbf{m}_0, \mathbf{P}_0, \{\mathbf{A}_t, \mathbf{Q}_t, \mathbf{H}_t, \mathbf{R}_t\}_{t=1}^T]$ are unknown ?

- even constant $\boldsymbol{\theta} = [\mathbf{m}_0, \mathbf{P}_0, \mathbf{A}, \mathbf{Q}, \mathbf{H}, \mathbf{R}]$ can be extremely challenging.

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 $\boldsymbol{\theta} = [\mathbf{m}_0, \mathbf{P}_0, \{\mathbf{A}_t, \mathbf{Q}_t, \mathbf{H}_t, \mathbf{R}_t\}_{t=1}^T]$ are unknown ?

- even constant $\boldsymbol{\theta} = [\mathbf{m}_0, \mathbf{P}_0, \mathbf{A}, \mathbf{Q}, \mathbf{H}, \mathbf{R}]$ can be extremely challenging.

Outline

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother
- Mini-project 2: KF

Nonlinear Kalman filters

Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

Mini-project 2: Kalman filter for 1D motion tracking (1/2)

Goal: Implement a Kalman filter (KF) for tracking the position and velocity of an object moving in one dimension.

State-space model:
Hidden state:

$$\mathbf{x}_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix}$$

where:

- p_t is the **position** at time t,
- v_t is the **velocity** at time t.

State evolution: (constant acceleration model)

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{q}_t, \quad \mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$
$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \frac{\Delta t^4}{4}\sigma_a^2 & \frac{\Delta t^3}{2}\sigma_a^2 \\ \frac{\Delta t^3}{2}\sigma_a^2 & \Delta t^2 \sigma_a^2 \end{bmatrix}$$

Observation model:

$$y_t = H\mathbf{x}_t + r_t, \quad r_t \sim \mathcal{N}(0, R)$$
$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Simulation parameters:

- $\Delta t = 1$, $\sigma_a = 0.5$, $R = 1$.
- Initial state: $\hat{\mathbf{x}}_0 = [0, 1]^\top$.
- Initial covariance: $P_0 = 10 I_2$.

Mini-project 2: Tasks (2/2)

Tasks:

1. **Simulate the system:**
   - Generate a **ground-truth trajectory** $\mathbf{x}_t$ over T time steps.
   - Simulate **noisy position measurements** y_t.
2. **Implement the Kalman filter:**
   - Use the standard **prediction** and **update** steps.
   - Estimate **position and velocity** over time.
3. **Evaluate the KF performance:**
   - Compare **estimated position** $\hat{p}_t$ with the true p_t.
     - Compute the **Mean Squared Error (MSE)** of position estimates.
     - You can also average over many data generation processes (recall KF is deterministic given the data)
   - Plot **ground truth, noisy measurements, and KF estimates**.
4. **Beyond (some ideas):**
   - play with the model parameters, for instance the initial velocity or the element $A(2,2)$
   - extension to a 2D motion model ($d_x = 4$)
     - you can draw trajectories in the plane
   - implement RTS smoother and compare MSE w.r.t. to true p_t
   - experiment model misspecified/mismatch scenarios (e.g., consider in inference values of σ^2 and R that are different than during data generation process)

Possible values (please experiment!):

- $T = 50$, $\Delta t = 1$, $\sigma_a = 0.5$, $R = 1$.
  - play with σ^2 and R (fix one and play with larger/smaller value of the other one, interpret the results)
- $\hat{\mathbf{x}}_0 = [0, 1]^\top$, $P_0 = 10 I_2$.

Outline

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother

Nonlinear Kalman filters

Mini-project

Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

The world is not linear-Gaussian: Lorenz model

- There was a time where the universe "was" all linear-Gaussian but...
  - solving real-world and interesting problems requires complicated models.
- Example: Lorenz system: non-linear and continuous time model (stochastic version)[4]

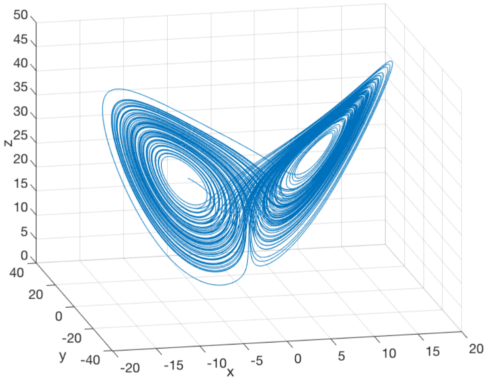
$$
\begin{aligned}
dX_1 &= -\mathsf{s}(X_1 - Y_1) + U_1, \\
dX_2 &= \mathsf{r}X_1 - X_2 - X_1 X_3 + U_2, \\
dX_3 &= X_1 X_2 - \mathsf{b}X_3 + U_3,
\end{aligned}
$$

  - U_1, U_2, U_3 are some noise process
  - $(\mathsf{s}, \mathsf{r}, \mathsf{b}) = \left(10, 28, \frac{8}{3}\right)$ are static model parameters broadly used in the literature since they lead to a **chaotic** behavior.
  - product of variables, continuous time, non-Markov behavior...

[4] **lorenz1963deterministic**.

The world is not linear-Gaussian: Lorenz model

Chaos: When the present determines the future, but the approximate present does not approximately determine the future.



The world is not linear-Gaussian: discretized Lorenz model

- Continuous-time Lorenz model $\Rightarrow$ discrete-time approximation
  - Euler-Maruyama integration with integration step $\Delta = 10^{-3}$

$$
\begin{aligned}
X_{1,t} &= X_{1,t-1} - \Delta \mathsf{s}(X_{1,t-1} - X_{2,t-1}) + \sqrt{\Delta} U_{1,t}, \\
X_{2,t} &= X_{2,t-1} + \Delta(\mathsf{r} X_{1,t-1} - X_{2,t-1} - X_{1,t-1} X_{3,t-1}) + \sqrt{\Delta} U_{2,t}, \\
X_{3,t} &= X_{3,t-1} + \Delta(X_{1,t-1} X_{2,t-1} - \mathsf{b} X_{3,t-1}) + \sqrt{\Delta} U_{3,t},
\end{aligned}
$$

  - $\{U_{i,t}\}_{t=0,1,\ldots}$, $i = 1, 2, 3$, are independent sequences of i.i.d. Gaussian random variables with zero mean and unit variance.
  - Markov model and also Gaussian, but still non-linear

The world is not linear-Gaussian: discretized Lorenz model

- Continuous-time Lorenz model $\Rightarrow$ discrete-time approximation
  - Euler-Maruyama integration with integration step $\Delta = 10^{-3}$

$$
\begin{aligned}
X_{1,t} &= X_{1,t-1} - \Delta \mathsf{s}(X_{1,t-1} - X_{2,t-1}) + \sqrt{\Delta} U_{1,t}, \\
X_{2,t} &= X_{2,t-1} + \Delta(\mathsf{r} X_{1,t-1} - X_{2,t-1} - X_{1,t-1} X_{3,t-1}) + \sqrt{\Delta} U_{2,t}, \\
X_{3,t} &= X_{3,t-1} + \Delta(X_{1,t-1} X_{2,t-1} - \mathsf{b} X_{3,t-1}) + \sqrt{\Delta} U_{3,t},
\end{aligned}
$$

  - $\{U_{i,t}\}_{t=0,1,\ldots}$, $i = 1, 2, 3$, are independent sequences of i.i.d. Gaussian random variables with zero mean and unit variance.
  - Markov model and also Gaussian, but still non-linear

The world is not linear-Gaussian: discretized Lorenz model

- Continuous-time Lorenz model $\Rightarrow$ discrete-time approximation
  - Euler-Maruyama integration with integration step $\Delta = 10^{-3}$

$$
\begin{aligned}
X_{1,t} &= X_{1,t-1} - \Delta \mathsf{s}(X_{1,t-1} - X_{2,t-1}) + \sqrt{\Delta} U_{1,t}, \\
X_{2,t} &= X_{2,t-1} + \Delta(\mathsf{r} X_{1,t-1} - X_{2,t-1} - X_{1,t-1} X_{3,t-1}) + \sqrt{\Delta} U_{2,t}, \\
X_{3,t} &= X_{3,t-1} + \Delta(X_{1,t-1} X_{2,t-1} - \mathsf{b} X_{3,t-1}) + \sqrt{\Delta} U_{3,t},
\end{aligned}
$$

  - $\{U_{i,t}\}_{t=0,1,\ldots}$, $i = 1, 2, 3$, are independent sequences of i.i.d. Gaussian random variables with zero mean and unit variance.
  - Markov model and also Gaussian, but still non-linear

The world is not linear-Gaussian: discretized Lorenz model

- Continuous-time Lorenz model $\Rightarrow$ discrete-time approximation
  - Euler-Maruyama integration with integration step $\Delta = 10^{-3}$

$$
\begin{aligned}
X_{1,t} &= X_{1,t-1} - \Delta \mathsf{s}(X_{1,t-1} - X_{2,t-1}) + \sqrt{\Delta} U_{1,t}, \\
X_{2,t} &= X_{2,t-1} + \Delta(\mathsf{r} X_{1,t-1} - X_{2,t-1} - X_{1,t-1} X_{3,t-1}) + \sqrt{\Delta} U_{2,t}, \\
X_{3,t} &= X_{3,t-1} + \Delta(X_{1,t-1} X_{2,t-1} - \mathsf{b} X_{3,t-1}) + \sqrt{\Delta} U_{3,t},
\end{aligned}
$$

  - $\{U_{i,t}\}_{t=0,1,\ldots}$, $i = 1, 2, 3$, are independent sequences of i.i.d. Gaussian random variables with zero mean and unit variance.
  - Markov model and also Gaussian, but still non-linear

Kalman filtering for nonlinear systems

- The Kalman filter is exact for linear and Gaussian models only.
- However, Kalman-like approximations are possible for nonlinear models.
- The most popular approaches include
  - Linearisation: the extended Kalman filter (EKF)[5]
  - Numerical integration: the unscented Kalman filter (UKF)[6], and quadrature/cubature Kalman filters (QKF)[7].
  - Monte Carlo & Kalman updates: ensemble Kalman filter[8].

[5] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Englewood Cliffs, 1979.

[6] S. J. Julier and J. Uhlmann. "Unscented filtering and nonlinear estimation". In: *Proceedings of the IEEE* 92.2 (Mar. 2004), pp. 401–422.

[7] I. Arasaratnam, S. Haykin, and R. J. Elliott. "Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature". In: *Proceedings of the IEEE* 95.5 (2007), pp. 953–977, I. Arasaratnam and S. Haykin. "Cubature kalman filters". In: *IEEE Transactions on Automatic Control* 54.6 (2009), pp. 1254–1269.

[8] G. Evensen. "The ensemble Kalman filter: Theoretical formulation and practical implementation". In: *Ocean dynamics* 53.4 (2003), pp. 343–367.

Linearisation

- Nonlinear dynamical system:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{q}_t, \quad \mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{r}_t$$

- The classical approach to nonlinear filtering is to linearise $f(\cdot)$ and $h(\cdot)$ using Taylor's theorem.
- Example: if $\mathbf{x}_t = \mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{q}_t$ but $\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{r}_t$, then

$$\mathbf{y}_t \approx h(\mathbf{x}_t^-) + \overbrace{\mathbf{J}_t(\mathbf{x}_t^-)}^{=\boldsymbol{H}_t}(\mathbf{x}_t - \mathbf{x}_t^-) + \mathbf{r}_t$$

  where $\mathbf{J}_t(\mathbf{m}_t^-)$ is the Jacobian matrix evaluated at $\mathbf{x}_t^-$,

$$\mathbf{J}_t = \begin{bmatrix} \frac{\partial h_1}{\partial x_{1,n}} & \frac{\partial h_1}{\partial x_{2,n}} & \cdots & \frac{\partial h_1}{\partial x_{d_x,n}} \\ \frac{\partial h_2}{\partial x_{1,n}} & \frac{\partial h_2}{\partial x_{2,n}} & \cdots & \frac{\partial h_2}{\partial x_{d_x,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_{d_y}}{\partial x_{1,n}} & \frac{\partial h_{d_y}}{\partial x_{2,n}} & \cdots & \frac{\partial h_{d_y}}{\partial x_{d_x,n}} \end{bmatrix}_{d_y \times d_x}$$

- If the state equation is nonlinear, then we linearise it around $\mathbf{m}_{t-1}$.

The extended Kalman filter

- Extended Kalman filter (EKF) for a nonlinear likelihood

$$\begin{aligned}\mathbf{x}_t &= \mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{q}_t, \quad \mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q}_t) \\ \mathbf{y}_t &\approx h(\mathbf{x}_t^-) + \mathbf{J}_t(\mathbf{x}_t^-)(\mathbf{x}_t - \mathbf{x}_t^-) + \mathbf{r}_t, \quad \mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t)\end{aligned}$$

$$\text{Prediction:} \quad \begin{cases} \mathbf{P}_t^- = \mathbf{A}_t P_{t-1}\mathbf{A}_t^\top + \mathbf{Q}_t \\ \mathbf{x}_t^- = \mathbf{A}_t\mathbf{m}_{t-1} \end{cases}$$

$$\text{Update:} \quad \begin{cases} \mathbf{S}_t = \mathbf{J}_t(\mathbf{x}_t^-)\mathbf{P}_t^-\mathbf{J}_t(\mathbf{x}_t^-)^\top + \mathbf{R}_t \\ \mathbf{m}_t = \mathbf{x}_t^- + \mathbf{P}_t^-\mathbf{J}_t(\mathbf{x}_t^-)^\top\mathbf{S}_t^{-1}\left(\mathbf{y}_t - h(\mathbf{x}_t^-)\right) \\ P_t = \mathbf{P}_t^- - \mathbf{P}_t^-\mathbf{J}_t(\mathbf{x}_t^-)^\top\mathbf{S}_t^{-1}\mathbf{J}_t(\mathbf{x}_t^-)\mathbf{P}_t^- \end{cases}$$

- Exercise: derive the EKF for nonlinear transition model
  - check the EKF in[9] (or same book of 2023 edition, Section 7.2)

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

Numerical integration with reference points

- Consider the problem of computing integrals w.r.t. a Gaussian pdf
$$\int f(x)\mathcal{N}(x;m,C)\mathrm{d}x \tag{1}$$
where $\mathcal{N}(x;m,C)$ is the Gaussian pdf with mean m and covariance C.
- There are several schemes that enable the approximation of (1) using a deterministic set of weighted points $\{x^j,\lambda^j\}_{j=1}^J$, namely,
$$\int f(x)\mathcal{N}(x;m,C)\mathrm{d}x \approx \sum_{j=1}^{J} \lambda^j f(x^j).$$
- Such approximations come in different "flavours": σ-points[10], quadrature methods[11], cubature schemes[12].

[10] S. J. Julier and J. Uhlmann. "Unscented filtering and nonlinear estimation". In: *Proceedings of the IEEE* 92.2 (Mar. 2004), pp. 401–422, H. M. Menegaz, J. Y. Ishihara, G. A. Borges, and A. N. Vargas. "A systematization of the unscented Kalman filter theory". In: *IEEE Transactions on automatic control* 60.10 (2015), pp. 2583–2598.
[11] I. Arasaratnam, S. Haykin, and R. J. Elliott. "Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature". In: *Proceedings of the IEEE* 95.5 (2007), pp. 953–977.
[12] I. Arasaratnam and S. Haykin. "Cubature kalman filters". In: *IEEE Transactions on Automatic Control* 54.6 (2009), pp. 1254–1269, B. Jia, M. Xin, and Y. Cheng. "High-degree cubature Kalman filter". In: *Automatica* 49.2 (2013), pp. 510–518.

Numerical integration with reference points

- The key concept is the following:
  - In MC and for a standard normal, we implicitly approximated the target distribution by a set of random points that are more likely to be around the mean.
  - In the case of reference/quadrature/cubature/deterministic points, the "samples" follow a similar principle but they are chosen deterministically:
    - it is not possible to do a variance analysis nor there is a consistency results (unless we have rules to take the number of points to infinity)
- Example: the spherical-radial cubature rule of degree 3[13]. If $x \sim \mathcal{N}(m, C)$ is d-dimensional, $C = SS^\top$ and S_j denotes j-th column of S, then

$$
\begin{aligned}
x^j &= m + \sqrt{d} S_j, \quad j = 1, \dots, d \\
x^j &= m - \sqrt{d} S_{j-d}, \quad j = d+1, \dots, 2d \\
\lambda^j &= \frac{1}{2d} \quad \forall j
\end{aligned}
$$

[13] B. Jia, M. Xin, and Y. Cheng. "High-degree cubature Kalman filter". In: *Automatica* 49.2 (2013), pp. 510–518.

Kalman filtering with reference points

- General description of unscented/quadrature/cubature Kalman filters.
- Let $X_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$ and assume the model

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{q}_t, \quad \mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{r}_t, \quad \mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q}_t), \quad \mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t).$$

- Kalman filter with reference points

- **Prediction**: assume $p(\mathbf{x}_{t-1}|y_{1:t-1}) \approx \mathcal{N}(\mathbf{x}_{t-1}; \mathbf{m}_{t-1}, \mathbf{P}_{t-1})$; then
  - compute $\{\mathbf{x}_{t-1}^j, \lambda_{t-1}^j\}_{j=1}^J$ from $\mathcal{N}(\mathbf{x}_{t-1}; \mathbf{m}_{t-1}, \mathbf{P}_{t-1})$ and
  - let $\chi_t^j = f(\mathbf{x}_{t-1}^j)$ for $j = 1, \ldots, J$;
  - predictive mean: $\mathbf{x}_t^- = \sum_{j=1}^J \lambda_{t-1}^j \chi_n^j$;
  - predictive covariance: $\mathbf{P}_t^- = \sum_{j=1}^J (\chi_t^j - \mathbf{x}_t^-)(\chi_n^j - \mathbf{x}_t^-)^\top \lambda_{t-1}^j + \mathbf{Q}_t$.

Kalman filtering with reference points

- Kalman filter w/ reference points (cont)

- **Update**:
  - compute $\{\mathbf{x}_t^{j-}, \lambda_t^{j-}\}_{j=1}^J$ from $\mathcal{N}(\mathbf{x}_t; \mathbf{x}_t^-, \mathbf{P}_t^-)$ and
  - let $\eta_t^j = h(\mathbf{x}_t^{j-})$ for $j = 1, \ldots, J$;
  - predicted observation: $\hat{\mathbf{y}}_t = \sum_{j=1}^J \lambda_t^{j-} \eta_t^{j-}$;
  - cross-covariance $P_t^{xy} = \sum_{j=1}^J (\mathbf{x}_t^{j-} - \mathbf{x}_t^-)(\eta_t^j - \hat{\mathbf{y}}_t)^\top \lambda_t^{j-}$
  - observation covariance: $\mathbf{S}_t = \sum_{j=1}^J (\eta_t^{j-} - \hat{\mathbf{y}}_t)(\eta_t^j - \hat{\mathbf{y}}_t)^\top \lambda_t^{j-} + \mathbf{R}_t$
  - Kalman gain: $\mathbf{K}_t = P_t^{xy} \mathbf{S}_t^{-1}$
  - mean: $\mathbf{m}_t = \mathbf{x}_t^- + \mathbf{K}_t (\mathbf{y}_t - \hat{\mathbf{y}}_t)$;
  - covariance: $P_t = \mathbf{P}_t^- - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^\top = \mathbf{P}_t^- - P_t^{xy} \mathbf{S}_t^{-1} (P_t^{xy})^\top$

Outline

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother

Nonlinear Kalman filters
- Mini-project

Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

Mini-project: CKF for the Lorenz 63 model

- Design and implement a cubature Kalman filter (CKF) and an unscented Kalman filter (UKF) for the stochastic Lorenz 63 model with nonlinear observations.
- State equation: stochastic Lorenz 63

$$\begin{aligned}\mathrm{d}X_1 &= -s(X_1 - Y_1) + \sigma \mathrm{d}W_1,\\ \mathrm{d}X_2 &= rX_1 - X_2 - X_1X_3 + \sigma \mathrm{d}W_2,\\ \mathrm{d}X_3 &= X_1X_2 - bX_3 + \sigma \mathrm{d}W_3,\end{aligned}$$

  where the $W_i(t)$'s are standard Wiener processes, σ is a constant, and the parameters $(s, r, b) = \left(10, 28, \frac{8}{3}\right)$ yield chaotic dynamics. Discretised via Euler-Maruyama with time-step h we have

$$\begin{aligned}X_{1,t} &= X_{1,t-1} - hs\left(X_{1,t-1} - X_{2,t-1}\right) + \sigma\sqrt{h}Z_{1,t},\\ X_{2,t} &= X_{2,t-1} + h\left(rX_{1,t-1} - X_{2,t-1} - X_{1,t-1}X_{3,t-1}\right) + \sigma\sqrt{h}Z_{2,t},\\ X_{3,t} &= X_{3,t-1} + h\left(X_{1,t-1}X_{2,t-1} - bX_{3,t-1}\right) + \sigma\sqrt{h}Z_{3,t},\end{aligned}$$

  where $Z_{i,t} \sim \mathcal{N}(0,1)$. The state is $\mathbf{x}_t = [X_{1,t}, X_{2,t}, X_{3,t}]^\top$.

Mini-project: CKF for the Lorenz 63 model

- Observations:

$$\begin{aligned} Y_{1,t} &= \frac{1}{10} X_{1,t} X_{2,t} + \sigma_u U_{1,t} \\ Y_{2,t} &= \frac{1}{10} X_{1,t} X_{3,t} + \sigma_u U_{2,t} \end{aligned}$$

  where σ_u is a constant and $U_{i,t} \sim \mathcal{N}(0,1)$. We denote $\mathbf{y}_t = [Y_{1,t}, Y_{2,t}]^\top$.
  Assume that observations are collected only every B discrete-time steps (i.e., when $t = kP$, $k = 1, 2, \ldots$). In the absence of observations, only the prediction step of the CKF has to be taken.
- The simulation code should generate the ground-truth signal $X_{0:T}$ and the observations $Y_{1:T}$ for some time horizon T. All model parameters should be user-selected, including T, the time step h, σ and σ_u, B, and $\{s, r, b\}$.
- Initial mean $\hat{x}_0 = [-5.9165; -5.5233; 24.5723]^\top$ (a point in the attractor of the deterministic Lorenz 63 with $(s, r, b) = (10, 28, \frac{8}{3})$.
- Reference values: $(s, r, b) = (10, 28, \frac{8}{3})$, initial covariance $P_0 = 20I$, $\sigma = \frac{1}{2}$, $\sigma_u = 2$, time step $h = 10^{-3}$, gap between observations $B = 20$, length of the simulation $T = 20/h = 20,000$ discrete time units.

Outline

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother

Nonlinear Kalman filters

Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

Learning model parameters in SSMs

- Options to learn model parameters $\boldsymbol{\theta}$ in general SSMs ($\boldsymbol{\theta} = [\mathbf{m}_0, \mathbf{P}_0, \mathbf{A}, \mathbf{Q}, \mathbf{H}, \mathbf{R}]$ in LG-SSM):
  1. Maximum-likelihood (point-wise estimate $\widehat{\boldsymbol{\theta}}$)
     - no prior knowledge is assumed
  2. Maximum a posteriori (point-wise estimate $\widehat{\boldsymbol{\theta}}$)
     - prior knowledge is incorporated and can help the inference
  3. Fully Bayesian approach: compute the posterior $p(\boldsymbol{\theta}|y_{1:T})$
     - even more complicated problem
     - Monte Carlo methods are generally used to obtain samples from $p(\boldsymbol{\theta}|y_{1:T})$

1. Maximum-likelihood estimation

- Goal:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \;\; p(\mathbf{y}_{1:T}|\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \;\; p(\mathbf{y}_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}) \tag{2}$$

  - partial normalizing constant $p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta})$:
    - computed by KF in LG-SSMs
    - approximated by PFs in other SSMs
  - equivalent to minimize the energy function

$$\begin{aligned}
\varphi(\boldsymbol{\theta}) &= -\log\left(p(\mathbf{y}_{1:T}|\boldsymbol{\theta})\right) &(3)\\
&= -\log\left(p(\mathbf{y}_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta})\right) &(4)\\
&= \underbrace{-\log\left(p(\mathbf{y}_1|\boldsymbol{\theta})\right)}_{\varphi_1(\boldsymbol{\theta})} + \sum_{t=2}^{T} \underbrace{-\log\left(p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta})\right)}_{\varphi_t(\boldsymbol{\theta})} &(5)\\
&= \sum_{t=1}^{T} \varphi_t(\boldsymbol{\theta}) &(6)
\end{aligned}$$

1. Maximum-likelihood estimation

- Numerical approaches for ML estimation:
  1. Gradient-based methods:
     - Option A:[14] obtain gradient of the energy function (sensitivity equations) $\nabla_{\boldsymbol{\theta}} \varphi(\boldsymbol{\theta})$
     - Option B:[15] through the Fisher identity (which uses the smoothing distribution)

$$\nabla_{\boldsymbol{\theta}} \varphi(\boldsymbol{\theta}) = \int \nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{x}_{1:T}, \mathbf{y}_{1:T} | \boldsymbol{\theta}\right) p\left(\mathbf{x}_{1:T} | \mathbf{y}_{1:T}, \boldsymbol{\theta}\right) d\mathbf{x}_{1:T} \tag{7}$$

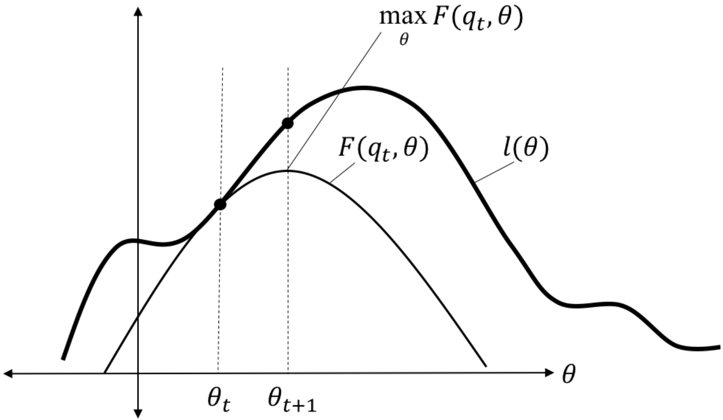
  2. Expectation-maximization (EM) algorithm:[16]
     - turns a complicated optimization problem into a sequence of easier problems
     - can be more stable numerically, ensures convergence, and may run faster

[14] D. Nagakura. "Computing exact score vectors for linear Gaussian state space models". In: *Communications in Statistics-Simulation and Computation* 50.8 (2021), pp. 2313–2326.

[15] https://www.almoststochastic.com/2014/06/fishers-identity.html

[16] 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.

Expectation-maximization approach for ML



(credit to M. N. Bernstein)

Expectation-maximization approach for ML

- Expectation-maximization (EM): iterative ML estimate
  - Algorithm introduced in [17]
  - Application to LG-SSMs in [18]
  - Based on the majorizing function property

$$\log\left(p(\mathbf{y}_{1:T}|\boldsymbol{\theta})\right) \geq F[q(\mathbf{x}_{0:T}),\boldsymbol{\theta}], \tag{8}$$

    where

$$F[q(\mathbf{x}_{0:T}),\boldsymbol{\theta}] = \int q(\mathbf{x}_{0:T}) \log \frac{p(\mathbf{x}_{0:T},\mathbf{y}_{1:T}|\boldsymbol{\theta})}{q(\mathbf{x}_{0:T})} d\mathbf{x}_{0:T} \tag{9}$$

    for any arbitrary pdf $q(\mathbf{x}_{0:T})$.
  - It is possible to maximize $\log\left(p(\mathbf{y}_{1:T}|\boldsymbol{\theta})\right)$ by iteratively maximizing the minorizing function $F[q(\mathbf{x}_{0:T}),\boldsymbol{\theta}]$
    - equivalent to minimize $\varphi(\boldsymbol{\theta})$ by minimizing $-F[q(\mathbf{x}_{0:T}),\boldsymbol{\theta}]$

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), pp. 1–38. ISSN: 00359246. URL: http://www.jstor.org/stable/2984875.

[18] 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.

Expectation-maximization approach for ML

- Maximize the minorizing function $F[q(\mathbf{x}_{0:T}), \boldsymbol{\theta}]$ w.r.t. functional q and parameter $\boldsymbol{\theta}$ via coordinate ascent:

Generic EM

- Initialization of $\boldsymbol{\theta}^{(0)}$ and function $q^{(0)}$.
- For $i = 1, 2, \ldots$

E-step $q^{(i)} = \underset{q}{\operatorname{argmax}} \; F[q(\mathbf{x}_{0:T}), \boldsymbol{\theta}^{(i-1)}]$.

M-step $\boldsymbol{\theta}^{(i)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \; F[q^{(i-1)}(\mathbf{x}_{0:T}), \boldsymbol{\theta}]$.

- Possible to show that the E-step solution is the smoothing distribution[19]

$$q^{(i)}(\mathbf{x}_{0:T}) = p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)}) \tag{10}$$

[19] R. M. Neal and G. E. Hinton. "A view of the EM algorithm that justifies incremental, sparse, and other variants". In: *Learning in graphical models*. Springer, 1998, pp. 355–368.

Expectation-maximization approach for ML

- Then, plugging $q^{(i)}(\mathbf{x}_{0:T}) = p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)})$ in (9), the M-step consists in maximizing:

$$F[q^{(i)}(\mathbf{x}_{0:T}), \boldsymbol{\theta}] = \int p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)}) \log \frac{p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T}|\boldsymbol{\theta})}{p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)})} d\mathbf{x}_{0:T}$$

$$= \underbrace{\int p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)}) \log \left(p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T}|\boldsymbol{\theta})\right) d\mathbf{x}_{0:T}}_{\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})}$$

$$\underbrace{- \int p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)}) \log \left(p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)})\right) d\mathbf{x}_{0:T}}_{\text{constant w.r.t. } \boldsymbol{\theta}}$$

EM algorithm for ML in generic SSMs

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

E-step compute $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$

M-step compute $\boldsymbol{\theta}^{(i)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \; \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$.

Expectation-maximization approach

- M-step: maximize

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) = \int p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)}) \log \left(p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T}|\boldsymbol{\theta}) \right) d\mathbf{x}_{0:T}$$

  - $p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)})$: smoothing distribution given $\boldsymbol{\theta}^{(i-1)}$
  - $p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T}|\boldsymbol{\theta}) = p(\mathbf{x}_0|\boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}) \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t)$: joint distribution of states and observations (as a function of $\boldsymbol{\theta}$)
- We need:
  - (E-step) $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$ to be closed-form
  - (M-step) Solution to $\frac{\partial \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})}{\partial \boldsymbol{\theta}} = 0$ (or iterative optimization method in M-step)

Expectation-maximization algorithm for LG-SSMs

- In LG-SSM:
  - joint smoothing $p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i-1)})$ is Gaussian
  - tractable integral to obtain:

$$
\begin{aligned}
\mathcal{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}\right) = & -\frac{1}{2}\log|2\pi\mathbf{P}_0(\boldsymbol{\theta})| - \frac{T}{2}\log|2\pi\mathbf{Q}(\boldsymbol{\theta})| - \frac{T}{2}\log|2\pi\mathbf{R}(\boldsymbol{\theta})| \\
& - \frac{1}{2}\operatorname{tr}\left\{\mathbf{P}_0^{-1}(\boldsymbol{\theta})\left[\mathbf{P}_0^{\mathrm{s}} + (\mathbf{m}_0^{\mathrm{s}} - \mathbf{m}_0(\boldsymbol{\theta}))(\mathbf{m}_0^{\mathrm{s}} - \mathbf{m}_0(\boldsymbol{\theta}))^\top\right]\right\} \\
& - \frac{T}{2}\operatorname{tr}\left\{\mathbf{Q}^{-1}(\boldsymbol{\theta})\left[\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^\top(\boldsymbol{\theta}) - \mathbf{A}(\boldsymbol{\theta})\mathbf{C}^\top + \mathbf{A}(\boldsymbol{\theta})\boldsymbol{\Phi}\mathbf{A}^\top(\boldsymbol{\theta})\right]\right\} \\
& - \frac{T}{2}\operatorname{tr}\left\{\mathbf{R}^{-1}(\boldsymbol{\theta})\left[\mathbf{D} - \mathbf{B}\mathbf{H}^\top(\boldsymbol{\theta}) - \mathbf{H}(\boldsymbol{\theta})\mathbf{B}^\top + \mathbf{H}(\boldsymbol{\theta})\boldsymbol{\Sigma}\mathbf{H}^\top(\boldsymbol{\theta})\right]\right\},
\end{aligned}
$$

where the following quantities are computed from the results of RTS smoother run under parameter values $\boldsymbol{\theta}^{(i-1)}$:

$$
\boldsymbol{\Sigma} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{P}_t^{\mathrm{s}} + \mathbf{m}_t^{\mathrm{s}}\left[\mathbf{m}_t^{\mathrm{s}}\right]^\top, \boldsymbol{\Phi} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{P}_{t-1}^{\mathrm{s}} + \mathbf{m}_{t-1}^{\mathrm{s}}\left[\mathbf{m}_{t-1}^{\mathrm{s}}\right]^\top,
$$

$$
\mathbf{B} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t\left[\mathbf{m}_t^{\mathrm{s}}\right]^\top, \mathbf{C} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{P}_t^{\mathrm{s}}\mathbf{G}_{t-1}^\top + \mathbf{m}_t^{\mathrm{s}}\left[\mathbf{m}_{t-1}^{\mathrm{s}}\right]^\top,\ \mathbf{D} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_k\mathbf{y}_k^\top.
$$

Expectation-maximization algorithm for LG-SSMs

EM algorithm for generic LG-SSMs

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

E-step run the RTS smoother and obtain closed-form $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$

M-step compute $\boldsymbol{\theta}^{(i)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$.

- If all parameters in $\boldsymbol{\theta}$ are known except one, M-step has closed form solution
  - otherwise more advanced optimisation methods are needed (block-alternating, gradient descent, proximal methods,...)
- For instance, if only $\mathbf{A}$ is unknown, the M-step optimizes

$$\mathcal{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}\right) = -\frac{T}{2} \operatorname{tr}\left\{\mathbf{Q}^{-1}(\boldsymbol{\theta})\left[\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^\top(\boldsymbol{\theta}) - \mathbf{A}(\boldsymbol{\theta})\mathbf{C}^\top + \mathbf{A}(\boldsymbol{\theta})\boldsymbol{\Phi}\mathbf{A}^\top(\boldsymbol{\theta})\right]\right\} + \mathrm{ct}_{/\mathbf{A}}$$

with

$$\boldsymbol{\Sigma} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{P}_t^{\mathrm{s}} + \mathbf{m}_t^{\mathrm{s}}\left[\mathbf{m}_t^{\mathrm{s}}\right]^\top, \boldsymbol{\Phi} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{P}_{t-1}^{\mathrm{s}} + \mathbf{m}_{t-1}^{\mathrm{s}}\left[\mathbf{m}_{t-1}^{\mathrm{s}}\right]^\top,$$

$$\mathbf{C} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{P}_t^{\mathrm{s}}\mathbf{G}_{t-1}^\top + \mathbf{m}_t^{\mathrm{s}}\left[\mathbf{m}_{t-1}^{\mathrm{s}}\right]^\top.$$

  - the closed-form solution is $\mathbf{A}^{(i)} = \mathbf{C}\boldsymbol{\Phi}^{-1}$

2. Maximum a posteriori (MAP) estimation

- MAP goal:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \;\; p(\boldsymbol{\theta}|\mathbf{y}_{1:T}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \;\; p(\mathbf{y}_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \tag{11}$$

- equivalent to

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \;\; \varphi(\boldsymbol{\theta}), \tag{12}$$

with

$$\begin{aligned}
\varphi(\boldsymbol{\theta}) &= -\log\left(p(\mathbf{y}_{1:T}|\boldsymbol{\theta})\right) - \log\left(p(\boldsymbol{\theta})\right) && (13)\\
&= -\log\left(p(\mathbf{y}_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta})\right) - \log\left(p(\boldsymbol{\theta})\right) && (14)\\
&= \underbrace{-\log\left(p(\mathbf{y}_1|\boldsymbol{\theta})\right)}_{\varphi_1(\boldsymbol{\theta})} + \sum_{t=2}^{T} \underbrace{-\log\left(p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta})\right)}_{\varphi_t(\boldsymbol{\theta})} - \log\left(p(\boldsymbol{\theta})\right) && (15)\\
&= \sum_{t=1}^{T} \varphi_t(\boldsymbol{\theta}) - \log\left(p(\boldsymbol{\theta})\right) && (16)
\end{aligned}$$

- MAP requires similar numerical (gradient-based and EM-based) methods can be used, with extra complications depending on $p(\boldsymbol{\theta})$

3. Fully Bayesian approach

- It is possible to do augmented inference on all unknowns, $p(\boldsymbol{\theta}, \mathbf{x}_{0:T}|\mathbf{y}_{1:T})$ and the marginalize to obtain

$$p(\boldsymbol{\theta}|\mathbf{y}_{1:T}) = \int p(\boldsymbol{\theta}, \mathbf{x}_{0:T}|\mathbf{y}_{1:T}) d\mathbf{x}_{0:T} \tag{17}$$

  - the full posterior and the marginalization are in general intractable
- Many methods based on approximating $p(\boldsymbol{\theta}|\mathbf{y}_{1:T})$ by a particle approximation $p(\boldsymbol{\theta}|\mathbf{y}_{1:T}) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\boldsymbol{\theta}_n}(\boldsymbol{\theta})$, e.g., particle MCMC[20]

Particle Metropolis-Hastings algorithm

- Initialization of $\boldsymbol{\theta}^{(0)}$.
- For $n = 1, 2, \ldots, N$
  1. Simulate a candidate sample $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}_{n-1})$
  2. Compute the acceptance probability
     $\alpha = \min\left\{1, \frac{p(\mathbf{y}_{1:T}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}^*)}{p(\mathbf{y}_{1:T}|\boldsymbol{\theta}_{n-1})p(\boldsymbol{\theta}_{n-1})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_{n-1})}\right\}$
  3. Simulate a uniform r.v. $u \sim \mathcal{U}(0,1)$ and set

$$\boldsymbol{\theta}_n = \begin{cases} \boldsymbol{\theta}^*, & \text{if } u \leq \alpha \\ \boldsymbol{\theta}_{n-1}, & \text{otherwise.} \end{cases}$$

[20] 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.

Outline

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother

Nonlinear Kalman filters

Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

Goal

- LG-SSMs: goal is to learn the state model
  - we consider $\mathbf{H}_t$ and $\mathbf{R}_t$ known and constant $\mathbf{A}_t = \mathbf{A}$ and $\mathbf{Q}_t = \mathbf{Q}$
  - goal: estimate $\boldsymbol{\theta} = [\mathbf{A}; \mathbf{Q}]$ through MAP

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

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.
- **Majorization-minimization** methodology to estimate $\mathbf{A}$ and $\mathbf{Q}$

A graphical perspective on $\mathbf{A}$

- **Goal.** Estimation of matrix $\mathbf{A}$ (a) introducing **prior knowledge**, and (b) under a novel **interpretation** of $\mathbf{A}$:

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

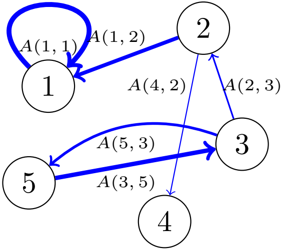
- $\mathbf{A}$ interpreted as a **sparse directed graph**

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

- $\mathbf{x}_t \in \mathbb{R}^{N_x}$ 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:

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

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



A graphical perspective on $\mathbf{A}$

- **Goal.** Estimation of matrix $\mathbf{A}$ (a) introducing **prior knowledge**, and (b) under a novel **interpretation** of $\mathbf{A}$:

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

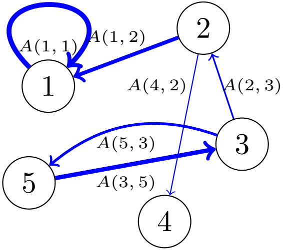
- $\mathbf{A}$ interpreted as a **sparse directed graph**

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

- $\mathbf{x}_t \in \mathbb{R}^{N_x}$ 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:

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

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



A graphical perspective on $\mathbf{A}$

- **Goal.** Estimation of matrix $\mathbf{A}$ (a) introducing **prior knowledge**, and (b) under a novel **interpretation** of $\mathbf{A}$:

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

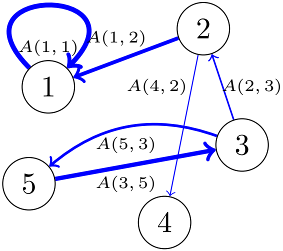
- $\mathbf{A}$ interpreted as a **sparse directed graph**

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

- $\mathbf{x}_t \in \mathbb{R}^{N_x}$ 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:

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

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



A graphical perspective on $\mathbf{A}$

- **Goal.** Estimation of matrix $\mathbf{A}$ (a) introducing **prior knowledge**, and (b) under a novel **interpretation** of $\mathbf{A}$:

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

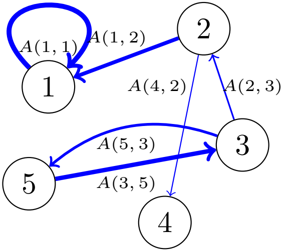
- $\mathbf{A}$ interpreted as a **sparse directed graph**

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

- $\mathbf{x}_t \in \mathbb{R}^{N_x}$ 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:

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

- $A(i,j) \neq 0 \Rightarrow x_{t-1,j}$ 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!)

- Let us consider two time-series $\mathbf{y}_i = [\mathbf{y}_{1,i}, \mathbf{y}_{2,i}, ..., \mathbf{y}_{T,i}]$ and $\mathbf{y}_j = [\mathbf{y}_{1,j}, \mathbf{y}_{2,j}, ..., \mathbf{y}_{T,j}]$
- We say that $\mathbf{y}_j$ Granger-causes $\mathbf{y}_i$ (order $p = 1$) if
  - when fitting the two auto-regressive (AR) models
    - (A) $\mathbf{y}_{t,i} = a_1 \mathbf{y}_{t-1,i} + \varepsilon_t$
    - (B) $\mathbf{y}_{t,i} = a_1 \mathbf{y}_{t-1,i} + b_1 \mathbf{y}_{t-1,j} + \gamma_t$
  - $\mathrm{Var}(\gamma_t) << \mathrm{Var}(\varepsilon_t)$

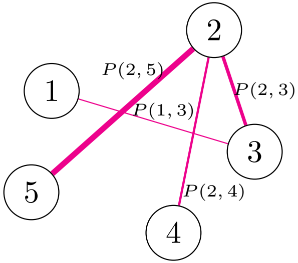
A graphical modeling $\mathbf{P} = \mathbf{Q}^{-1}$

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

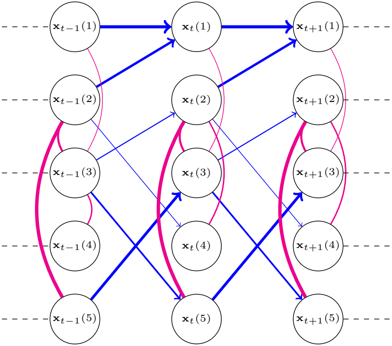
- $\mathbf{P} = \mathbf{Q}^{-1}$ interpreted as **sparse undirected graph** (Gaussian graphical models).

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

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



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

Dynamical systems

State-space models (SSMs)

Linear-Gaussian model and Kalman filter

Kalman filter and RTS smoother

Nonlinear Kalman filters

Learning model parameters in SSMs

A doubly graphical perspective on LG-SSM

Estimation of $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSM

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

Proposed penalized formulation

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

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

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

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-like approach

- EM-like approach: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 ($\mathcal{Q}(\mathbf{A}, \mathbf{P}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \geq \mathcal{L}(\mathbf{A}, \mathbf{P})$, $\forall(\mathbf{A}, \mathbf{P})$).
  - 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 $\mathbf{P}^{(i+1)}$.

DGLASSO algorithm

- **Block alternating majorization-minimization technique**:
  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 $(\theta_A, \theta_P) > 0$, to **stabilize** the minimization process and guarantee convergence of iterates.
- 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[21]
  - both optimization steps (b) and (d) require modern optmisation algorithms

[21] J. Friedman, T. Hastie, and R. Tibshirani. "Sparse inverse covariance estimation with the graphical lasso". In: *Biostatistics* 9.3 (2008), pp. 432–441.

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 $\mathcal{L}$.

• Proof based on the work[22]

• In practice, inner mininimization steps (b) and (d) using a Dykstra proximal splitting solver.[23]

[22] 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.

[23] 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.

Summary of the GraphEM algorithm

- DGLASSO generalises our previous GraphEM,[24] 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 $\mathbf{A}^{(i)} = \mathrm{argmin}_{\mathbf{A}} \left(\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i-1)})\right)$ 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 $\mathcal{L}_0 = \sum_{m=1}^{M} f_m$

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

SpaRJ algorithm

- SpaRJ[25] (*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 $p(\mathbf{A}|\mathbf{y}_{1:T})$.[26]
  - 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.

[25] 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.

[26] 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[25] (*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 $p(\mathbf{A}|\mathbf{y}_{1:T})$.[26]
  - 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.

[25] 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.

[26] 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[25] (*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 $p(\mathbf{A}|\mathbf{y}_{1:T})$.[26]
  - 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.

[25] 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.
[26] 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[25] (*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 $p(\mathbf{A}|\mathbf{y}_{1:T})$.[26]
  - 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.

---

[25] 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.

[26] 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 $\mathbf{A}'$
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

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

- Four synthetic datasets with $\mathbf{H} = \mathbf{Id}$ and block-diagonal matrix $\mathbf{A}$, composed with b blocks of size $(b_j)_{1 \le j \le b}$, 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	$(b_j)_{1 \le j \le b}$	$(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$
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)[27]
  - Granger-causality approaches: pairwise Granger Causality (PGC) and conditional Granger Causality (CGC)[28]

[27] S. Sarkka. *Bayesian Filtering and Smoothing*. Ed. by C. U. Press. 2013.
[28] 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

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

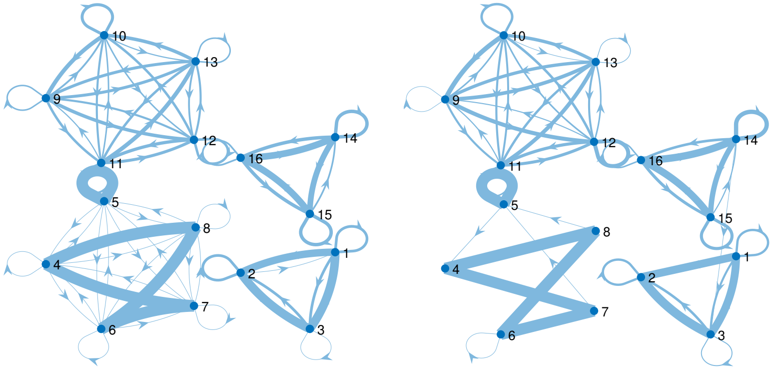
Dataset	N_x	$(b_j)_{1 \leq j \leq b}$	$(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$
A	9	$(3,3,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
B	9	$(3,3,3)$	$(1, 1, 10^{-4})$
C	16	$(3,5,5,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
D	16	$(3,5,5,3)$	$(1, 1, 10^{-4})$

- GraphEM (DGLASSO with known $\mathbf{Q}$) is compared with:
  - Maximum likelihood EM (MLEM)[27]
  - Granger-causality approaches: pairwise Granger Causality (PGC) and conditional Granger Causality (CGC)[28]

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

[28] 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

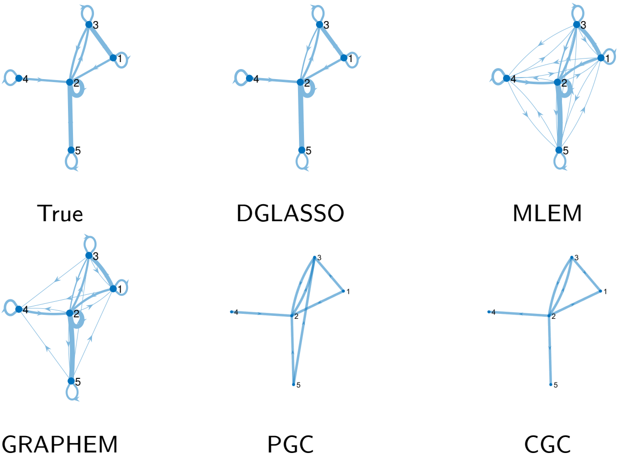


True graph associated to $\mathbf{A}$ (left) and GraphEM estimate (right) for dataset C.

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



Graph inference results on an example from WeathN5a dataset.[29]

[29] J. Runge, X.-A. Tibau, M. Bruhns, J. Muñoz-Marí, and G. Camps-Valls. "The causality for climate competition". In: *NeurIPS 2019 Competition and Demonstration Track*. Pmlr. 2020, pp. 110–120.

Computational complexity of DGLASSO

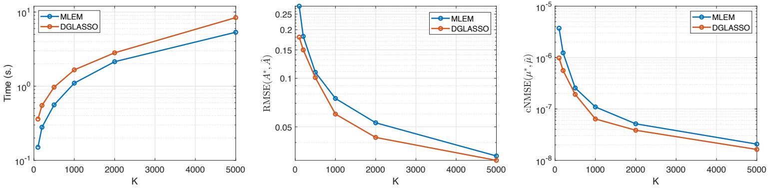


Figure 6: Evolution of the complexity time (left), RMSE$(\mathbf{A}^*, \widehat{\mathbf{A}})$ (middle) and cNMSE$(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$ (right) metrics, as a function of the time series length K, for experiments on dataset A averaged over 50 runs.

Convergence of SpaRJ and GarphEM with data

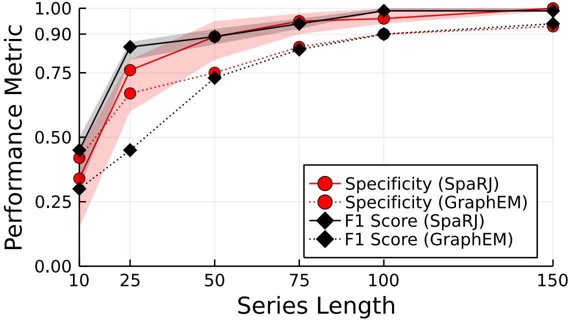


Figure: 3×3 system with known isotropic state covariance.

Convergence of SpaRJ with iterations

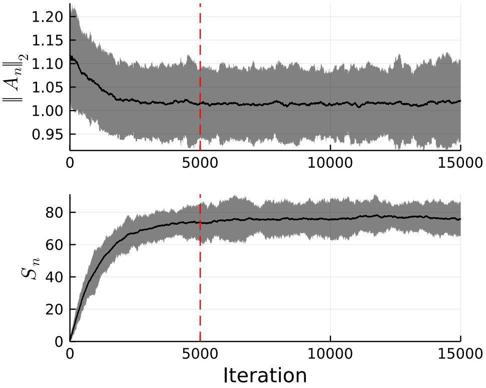


Figure: Progression of sample metrics in a 12×12.

SpaRJ with real world data

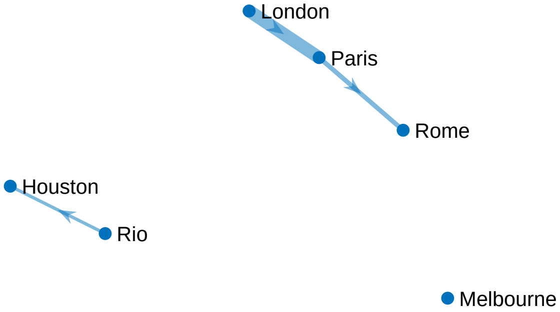


Figure: Average daily temperature of 324 cities from 1995 to 2021, curated by the United States Environmental Protection Agency.

Conclusion

- SSMs are very powerful tools but still underdeveloped due to conceptual and computational limitations.
- Even LG-SSMs require significant research for modeling and parameter estimation.
- Novel graphical interpretation on matrices $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSMs.
  - Algorithms to estimate sparse model parameters: GraphEM, DGLASSO (point-wise) and SpaRJ (fully Bayesian).
  - strong model interpretation
  - theoretical guarantees
  - good performance
- 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 limitations.
- Even LG-SSMs require significant research for modeling and parameter estimation.
- Novel graphical interpretation on matrices $\mathbf{A}$ and $\mathbf{Q}$ in LG-SSMs.
  - Algorithms to estimate sparse model parameters: GraphEM, DGLASSO (point-wise) and SpaRJ (fully Bayesian).
  - strong model interpretation
  - theoretical guarantees
  - good performance
- This is a challenging problem with many exciting ongoing methodological and applied avenues ahead!

Useful book: S. Sarkka and L. Svensson. Bayesian filtering and smoothing. Vol. 17. Cambridge university press, 2023.

GraphEM paper: V. Elvira, É. Chouzenoux, "Graphical Inference in Linear-Gaussian State-Space Models", *IEEE Transactions on Signal Processing*, Vol. 70, pp. 4757-4771, 2022.

SpaRJ: 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.

DGLASSO: E. Chouzenoux and V. Elvira, "Sparse Graphical Linear Dynamical Systems, submitted, 2023. https://arxiv.org/abs/2307.03210

GraphIT paper: 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-Markovian models: 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.