Modeling, filtering, and graph discovery in state-space models

Víctor Elvira
School of Mathematics
University of Edinburgh

Joint work with: M. F. Bugallo, N. Branchini,
E. Chouzenoux, P. M. Djuric, L. Martino

Bellairs Research Institute of McGill University, Barbados
December 14, 2021

Outline

Intro: State-space models (SSMs) and Bayesian filtering

Part I: Linear-Gaussian model and Kalman filter

Part I: GraphEM: Graph discovery in linear-Gaussian SSMs

Part II: Beyond linear-Gaussian SSMs and particle filters (PFs)

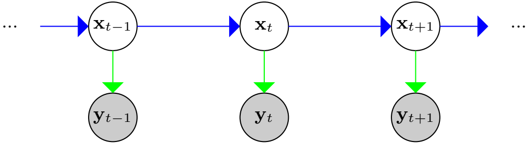
Part II: PFs from the MIS perspective

Motivation

- A large class of problems in statistics, machine learning, and signal processing requires **sequential processing of observed data**.
- Examples of **applications**:
  - Geophysical systems (atmosphere, oceans)
  - Robotics
  - Target tracking, positioning, navigation
  - Communications
  - Biomedical signal processing
  - Financial engineering
  - Ecology

Inference in State-Space Models (SSM)

- Let us consider:
  - a set of hidden states $\mathbf{x}_t \in \mathbb{R}^{d_x}$, $t = 1, ..., T$.
  - a set of observations $\mathbf{y}_t \in \mathbb{R}^{d_y}$, $t = 1, ..., T$.
- A SSM is an underlying hidden process of $\mathbf{x}_t$ that evolves and that, partially and noisily, expresses itself through $\mathbf{y}_t$.



- Two ways or describing the system:
  1. *Deterministic* notation:
     - Hidden state $\rightarrow$ $\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{q}_t)$
     - Observations $\rightarrow$ $\mathbf{y}_t = h(\mathbf{x}_t, \mathbf{r}_t)$

     where $\mathbf{q}_t$ and $\mathbf{r}_t$ are **random** noise vector (with known distributions of $\mathbf{q}_t$ and $\mathbf{r}_t$) and $g(\cdot)$ and $h(\cdot)$ are also known.
  2. *Probabilistic* notation:
     - Hidden state $\rightarrow$ $p(\mathbf{x}_t|\mathbf{x}_{t-1})$
     - Observations $\rightarrow$ $p(\mathbf{y}_t|\mathbf{x}_t)$

The estimation problem

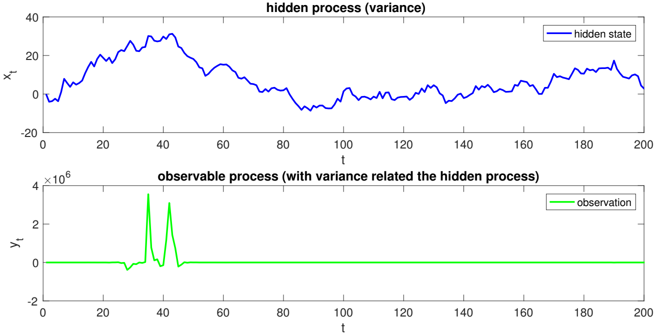
- We sequentially observe observations $\mathbf{y}_t$ related to the hidden state $\mathbf{x}_t$.
- At time t, we have accumulated t observations, $\mathbf{y}_{1:t} \equiv \{\mathbf{y}_1, ..., \mathbf{y}_t\}$.
- Interesting problems:
  - **Filtering**: estimate current state $\widehat{\mathbf{x}}_t$ given all observations $\mathbf{y}_{1:t}$
  - State prediction: predict the future state $\widehat{\mathbf{x}}_{t+\tau}$ given $\mathbf{y}_{1:t}$ $(\tau > 0)$
  - Obs. prediction: predict the future observation $\hat{\mathbf{y}}_{t+\tau}$ given $\mathbf{y}_{1:t}$
  - Better estimate a past state (aka smoothing): Estimate $\widehat{\mathbf{x}}_{t-\tau}$ given $\mathbf{y}_{1:t}$ $(\tau > 0)$
- We want to do it sequentially and efficiently.
  - At time t, we want to *process* only $\mathbf{y}_t$, but not reprocess all $\mathbf{y}_{1:t-1}$ (that were already processed!)

Example

- There are two interrelated random processes, one is observed and one is hidden.
  - e.g.,stochastic volatility model, very common in financial engineering

$$
\begin{aligned}
x_t &= 0.999x_{t-1} + q_t \\
y_t &= e^{\frac{x_t}{2}} r_t,
\end{aligned}
$$

  - with $q_t \sim \mathcal{N}(0,1)$ and $r_t \sim \mathcal{N}(0,1)$
  - **Goal**: estimate the hidden x_t given the observed $y_{1:t}$

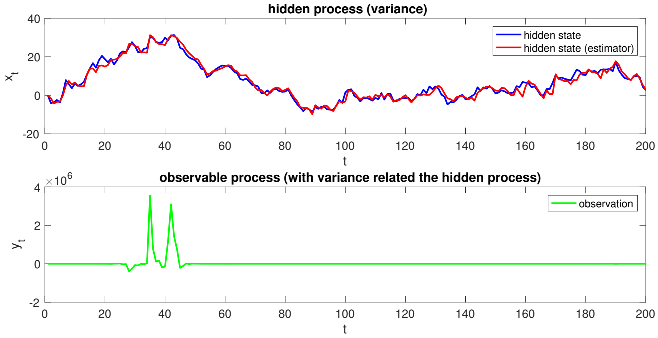


Example

- Consider the following stochastic volatility model, very common in financial engineering

$$
\begin{aligned}
x_t &= 0.999 x_{t-1} + q_t \\
y_t &= e^{\frac{x_t}{2}} r_t,
\end{aligned}
$$

  - with $q_t \sim \mathcal{N}(0,1)$ and $r_t \sim \mathcal{N}(0,1)$
  - **Goal**: estimate the hidden x_t given the observed $y_{1:t}$



The Probabilistic/Bayesian Approach

- **Estimations are good, distributions are better**
- Instead of a single value $\widehat{\mathbf{x}}_t$, we give a probability for any single possible value of $\mathbf{x}_t$.

$$\widehat{\mathbf{x}}_t \Rightarrow p(\mathbf{x}_t | y_{1:t})$$

  - Measure of certainty.
- The basic problems again (probabilistic version!)
  - **Filtering**: $p(\mathbf{x}_t | \mathbf{y}_{1:t})$
  - State prediction: $p(\mathbf{x}_{t+\tau} | \mathbf{y}_{1:t}), \qquad \tau \geq 1$
  - Observation prediction: $p(\mathbf{y}_{t+\tau} | \mathbf{y}_{1:t}), \qquad \tau \geq 1$
  - Smoothing: $p(\mathbf{x}_{t-\tau} | \mathbf{y}_{1:t}), \qquad \tau \geq 1$
- We will focus on the **filtering** problem

Bayesian Filtering

- Bayesian rule: if we want to infer all states $\mathbf{x}_{1:T} \equiv \{\mathbf{x}_1, ..., \mathbf{x}_T\}$,

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

- If we want to infer just a particular state $\mathbf{x}_t$ (smoothing): marginalization

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

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

efficient Bayesian **sequential** 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 (at time t):
  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, $I(f) = \int \mathbf{x}_t p(\mathbf{x}_t|y_{1:t})d\mathbf{x}_t$
  - Usually the posterior cannot be analytically computed!

Outline

The linear-Gaussian Model

- The linear-Gaussian model is arguably the most relevant SSM:
- *Deterministic* 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
  - Gaussian pdfs, with means and covariances matrices are calculated at each t
  - Efficient processing of $\mathbf{y}_t$, obtaining $p(\mathbf{x}_t | \mathbf{y}_{1:t})$ from $p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$ (intermediate $p(\mathbf{x}_t | \mathbf{y}_{1:t-1})$ result)
- **Rauch-Tung-Striebel (RTS) smoother**: obtains the smoothing distribution $p(\mathbf{x}_{1:T} | \mathbf{y}_{1:T})$, i.e., posterior of the whole trajectory
  - requires a backwards reprocessing, refining the Kalman estimates

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

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

Kalman Filter: update step

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

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

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

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

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

Kalman summary and RTS smoother

Kalman filter

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

  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 $t = T, \dots, 1$

  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 some model parameters are unknown ?

Outline

Intro: State-space models (SSMs) and Bayesian filtering

Part I: Linear-Gaussian model and Kalman filter

Part I: GraphEM: Graph discovery in linear-Gaussian SSMs

Part II: Beyond linear-Gaussian SSMs and particle filters (PFs)

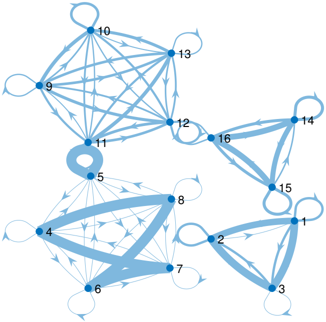
Part II: PFs from the MIS perspective

- Recall the linear-Gaussian system:
  - 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 $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t)$.
- In practice, most of these parameters are **unknown**: $\mathbf{A}_t$, $\mathbf{H}_t$, $\mathbf{Q}_t$, $\mathbf{R}_t$.
  - A common assumption is that they are static, i.e., $\mathbf{A}$, $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{R}$.

- The most challenging parameter to estimate (but also interesting) is $\mathbf{A}$:
  - **Graph discovery perspective**: $\mathbf{x}_t \in \mathbb{R}^{N_x}$ contains N_x unidimensional time-series, each of them acquired in a node of a graph (with N_x total nodes)
  - The elements $a_{i,j}$ of $\mathbf{A}$ represents, the linear effect of node j at time $t-1$ in the update of the signal of node i at time t:

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



- GraphEM: An expectation-maximization (EM) method within Kalman filters for the estimation of $\mathbf{A}$ (along with the hidden states).[1]

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

GraphEM in a nutshell

- **Goal:** Find $\mathbf{A}^*$ that maximizes $p(\mathbf{A}|\mathbf{y}_{1:T}) \propto p(\mathbf{A})p(\mathbf{y}_{1:T}|\mathbf{A})$, i.e., the MAP estimate of $\mathbf{A}$
  - Equivalent to minimizing $\varphi_T(\mathbf{A}) = -\log p(\mathbf{A}) - \log p(\mathbf{y}_{1:T}|\mathbf{A})$.
  - **Challenge**: evaluating $p(\mathbf{y}_{1:T}|\mathbf{A})$ (or $\varphi_T(\mathbf{A})$) requires to run Kalman filter:

$$\varphi_T(\mathbf{A}) = -\log p(\mathbf{A}) + \sum_{t=1}^{T} \frac{1}{2} \log |2\pi \mathbf{S}_t(\mathbf{A})| + \frac{1}{2} \mathbf{z}_t(\mathbf{A})^\top \mathbf{S}_t(\mathbf{A})^{-1} \mathbf{z}_t(\mathbf{A}) \quad (1)$$

    - Non tractable minimization.

- EM strategy: Minimize a sequence of tractable approximations of φ_T satisfying a majorizing property.

- Lasso regularization (prior): In order to limit the degrees of freedom in the parametric model, we choose the prior to promote a sparse matrix $\mathbf{A}$.

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad -\log p(\mathbf{A}) \equiv \varphi_0(\mathbf{A}) = \gamma \|\mathbf{A}\|_1, \qquad \gamma > 0.$$

Expression of EM steps

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

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

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

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

Then, as a consequence of, we can build

$$\mathcal{Q}(\mathbf{A};\mathbf{A}') = \frac{T}{2}\mathrm{tr}\left(\mathbf{Q}^{-1}(\mathbf{\Sigma} - \mathbf{C}\mathbf{A}^\top - \mathbf{A}\mathbf{C}^\top + \mathbf{A}\mathbf{\Phi}\mathbf{A}^\top)\right) + \varphi_0(\mathbf{A}) + \mathcal{C},$$

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

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

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

Computation of the M-step

- Minimization problem:

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

  - Convex non-smooth minimization problem

- Proximal splitting approach: The proximity operator of $f : \mathbb{R}^{N_x \times N_x} \to \mathbb{R}$ is defined[2]

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

Douglas-Rachford algorithm

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

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

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

✓ Both involved proximity operators have closed form solution.

[2] P.L. Combettes and JC. Pesquet. "Proximal Splitting Methods in Signal Processing.". In: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* 49 (2011), pp. 185–212.

GraphEM algorithm

GraphEM algorithm

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

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

M-step Update $\mathbf{A}^{(i)} = \text{argmin}_{\mathbf{A}} \left(\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i-1)})\right)$ using Douglas-Rachford algorithm.

✓ Flexible approach, valid as long as the proximity operator of f_2 is available.

✓ sound convergence properties of the EM algorithm

- monotonical decrease and convergence of $\{\varphi_T(\mathbf{A}^{(i)})\}_{i \in \mathbb{N}}$ can be shown.

Data description and numerical settings

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

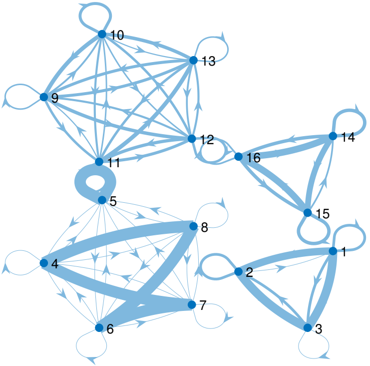
Dataset	N_x	$(b_j)_{1 \leq j \leq b}$	$(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$
A	9	$(3,3,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
B	9	$(3,3,3)$	$(1, 1, 10^{-4})$
C	16	$(3,5,5,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
D	16	$(3,5,5,3)$	$(1, 1, 10^{-4})$

- GraphEM is compared with:
  - Maximum likelihood EM (MLEM)[3]
  - Granger-causality approaches: pairwise Granger Causality (PGC) and conditional Granger Causality (CGC)[4]

[3] S. Sarkka. *Bayesian Filtering and Smoothing*. Ed. by Cambridge University Press. 3rd ed. 2013.

[4] D. Luengo et al. "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



True graph (left) and GraphEM estimate (right) for dataset C.

Experimental results

	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

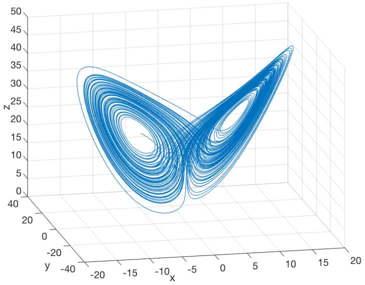
Conclusions and ongoing work

- GraphEM algorithm:
  - ✓ Interpretation of hidden states as a (causal) directed graph
  - ✓ Lasso penalization to promote sparsity
    - common in complex systems
    - reduces the implicit dimension
  - ✓ EM-based method with proximal splitting M-step
    - sound convergence guarantees
  - ✓ Good numerical performance compared to several techniques
- Ongoing work:
  - Extension to enforce multiple properties on $\mathbf{A}$
    - stability, block sparsity, positivity/negativity/etc (physically driving),...
    - requires a novel proximal-based method
  - application to Earth observation
  - Totally different approach for the same perspective on $\mathbf{A}$:
    - hierarchical algorithm with reversible jump MCMC on the sparsitiy levels of $\mathbf{A}$

Outline

Beyond linear-Gaussian SSMs

- The world is not linear-Gaussian: Lorenz model (chaotic model)



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

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

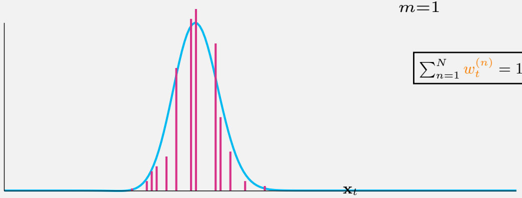
  - $\{U_{i,t}\}_{t=0,1,\ldots}$, $i = 1, 2, 3$, are independent sequences of i.i.d. Gaussian random variables with zero mean and unit variance.
  - Markov model and also Gaussian, but still non-linear

Particle Filtering

- Recall the generic SSM:
  - Hidden state model: $\rightarrow$ $p(\mathbf{x}_t|\mathbf{x}_{t-1})$
  - Observations model: $\rightarrow$ $p(\mathbf{y}_t|\mathbf{x}_t)$
- **Same goal**: Obtain the (now intractable) filtering distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ through particle filtering (PF) which is based on importance sampling (IS):

- IS in a nutshell:
  1. **Sampling.** $\mathbf{x}_t^{(m)} \sim \psi_t(\mathbf{x}_t)$, $m = 1, ..., M$
     - $\psi_t(\mathbf{x}_t)$ is the proposal and is key for the performance
  2. **Weighting.** $\widetilde{w}^{(m)} = \frac{p(\mathbf{x}^{(m)}|\mathbf{y}_{1:t})}{\psi_t(\mathbf{x}^{(m)})}$, $m = 1, ..., M$,
  3. **Normalize weights.** $w_t^{(n)} = \frac{\widetilde{w}^{(m)}}{\sum_{j=1}^{M} \widetilde{w}^{(j)}}$
- The distribution of interest (filtering) is approximated as:

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



$\boxed{\sum_{n=1}^{N} w_t^{(n)} = 1}$

The bootstrap PF (BPF)

- Bootstrap PF $\equiv$ Sequential Importance Resampling (SIR) based on importance sampling [Gordon, 1993]

(i) Initialization. At time $t = 0$, $\tilde{\mathbf{x}}_0^{(m)} \sim p(\mathbf{x}_0)$, $m = 1, \ldots, M$.

(ii) Recursive step. At time t,

1. **Prediction** (particles propagation): $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t | \tilde{\mathbf{x}}_{t-1}^{(m)})$
2. **Update** (weights calculation): compute the normalized weights as $w_t^{(m)} \propto p(\mathbf{y}_t | \mathbf{x}_t^{(m)})$
3. Multinomial **resampling** at every time step:
   - Set $\{\tilde{\mathbf{x}}_t^{(m)}\}_{m=1}^M$ is formed by sampling M times with replacement from the set $\{\mathbf{x}_t^{(m)}\}_{m=1}^M$ with associated probabilities $\{w_t^{(m)}\}_{m=1}^M$
   - **equivalent to**: simulate M i.i.d. samples from the approx. filtering distribution

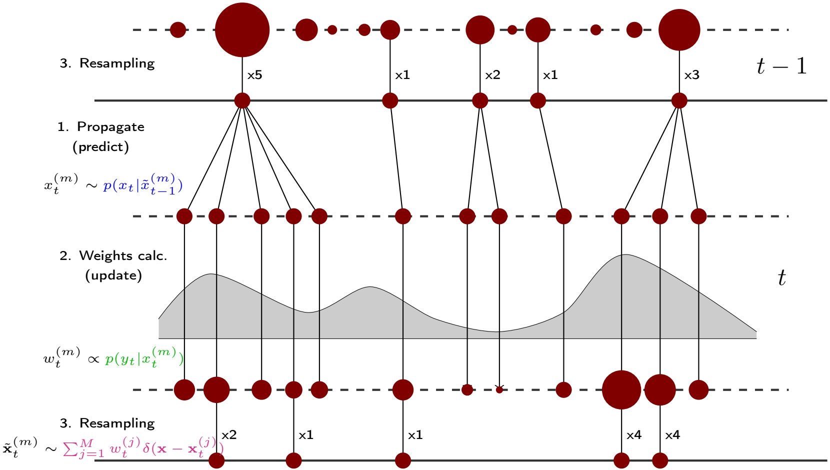
$$\tilde{\mathbf{x}}_t^{(m)} \sim p^M(\mathbf{x}_t | \mathbf{y}_{1:t}) \equiv \sum_{j=1}^{M} w_t^{(j)} \delta(\mathbf{x} - \mathbf{x}_t^{(j)})$$

- **Output.** The filtering distribution is now approximated as

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) \approx p^M(\mathbf{x}_t | \mathbf{y}_{1:t}) \equiv \sum_{j=1}^{M} w_t^{(j)} \delta(\mathbf{x} - \mathbf{x}_t^{(j)})$$

(instead of having a Gaussian filtering distribution as in Kalman)

Bootstrap PF (BPF)



Outline

Intro: State-space models (SSMs) and Bayesian filtering

Part I: Linear-Gaussian model and Kalman filter

Part I: GraphEM: Graph discovery in linear-Gaussian SSMs

Part II: Beyond linear-Gaussian SSMs and particle filters (PFs)

Part II: PFs from the MIS perspective

Multiple importance sampling (MIS)

- Multiple importance sampling (MIS) is an extension of IS when several proposals are available
  - Very active topic, and recent works show that there exist many sampling and weighting possibilities.[5]
- PFs are usually derived under the perspective of sampling trajectories, but rarely under the perspective of one time-step ahead, analyzing what the true proposal $\psi(\mathbf{x}_t)$ and the consequences.
- We propose an alternative way of deriving existing PFs[6] that
  - offers new insights about the implicit assumptions
  - helps to understand when you should use one or other PF
  - allows to propose new high-performance PFs

[5] V. Elvira et al. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.

[6] V. Elvira et al. "Elucidating the Auxiliary Particle Filter via Multiple Importance Sampling". In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 145–152.

A generic particle filtering from the MIS perspective

(i) Initialization. At time $t = 0$, $\mathbf{x}_0^{(m)} \sim p(\mathbf{x}_0)$, $w_0^{(m)} = 1/M$, $m = 1, \dots, M$.

(ii) Recursive step. At time $t > 0$,

1 **Proposal adaptation/selection.** Select the MIS proposal of the form

$$\psi_t(\mathbf{x}_t) = \sum_{j=1}^{M} \lambda_t^{(j)} p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(j)}),$$

2 **Sampling.** Draw samples according to

$$\mathbf{x}_t^{(m)} \sim \psi_t(\mathbf{x}_t), \qquad m = 1, ..., M.$$

3 **Weighting.** Compute the normalized IS weights by

$$w_t^{(m)} \propto \frac{p(\mathbf{x}_t^{(m)} | \mathbf{y}_{1:t})}{\psi_t(\mathbf{x}_t^{(m)})} \propto \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) p(\mathbf{x}_t^{(m)} | \mathbf{y}_{1:t-1})}{\psi_t(\mathbf{x}_t^{(m)})}$$

$$\approx \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(j)})}{\psi_t(\mathbf{x}_t^{(m)})} = \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M} \lambda_t^{(j)} p(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(j)})} \tag{2}$$

- Two questions:[6]
  1. Selection/adaptation of $\{\lambda_t^{(j)}\}_{j=1}^{M}$ to build $\psi_t(\mathbf{x}_t)$?
     - Recall: IS is efficient when $\psi_t(\mathbf{x}_t)$ is *close* to $p(\mathbf{x}_t | \mathbf{y}_{1:t})$
  2. Approximate $w_t^{(m)}$ in (2) to derive BPF and APF?

[6] Victor Elvira et al. "Elucidating the Auxiliary Particle Filter via Multiple Importance Sampling". In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 145–152.

BPF from the MIS perspective

(i) Initialization. At time $t = 0$, $\mathbf{x}_0^{(m)} \sim p(\mathbf{x}_0)$, and $w_0^{(m)} = 1/M$, $m = 1, \ldots, M$.

(ii) Recursive step. At time $t > 0$,

1. **Proposal adaptation/selection.** Select the MIS proposal of the form

$$\psi_t(\mathbf{x}_t) = \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)}), \qquad (\lambda_t^{(j)} = w_{t-1}^{(j)})$$

2. **Sampling.** Draw samples according to

$$\mathbf{x}_t^{(m)} \sim \psi_t(\mathbf{x}_t), \qquad m = 1, ..., M. \qquad \text{(equiv. resampling+propagation)}$$

3. **Weighting.** Compute the normalized IS weights by

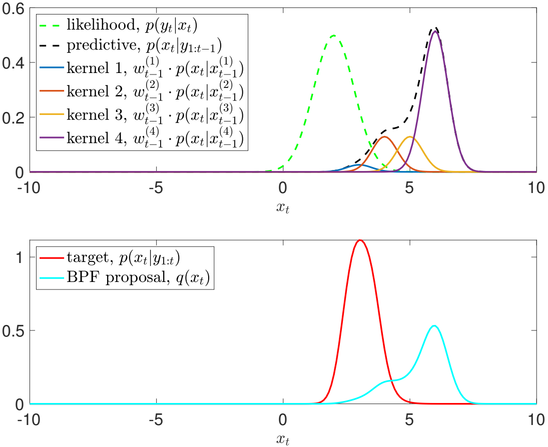
$$w_t^{(m)} \propto \frac{p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t})}{\psi_t(\mathbf{x}_t^{(m)})} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t-1})}{\psi_t(\mathbf{x}_t^{(m)})}$$

$$\approx \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\psi_t(\mathbf{x}_t^{(m)})} = \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}$$

$$= p(\mathbf{y}_t|\mathbf{x}_t^{(m)})$$

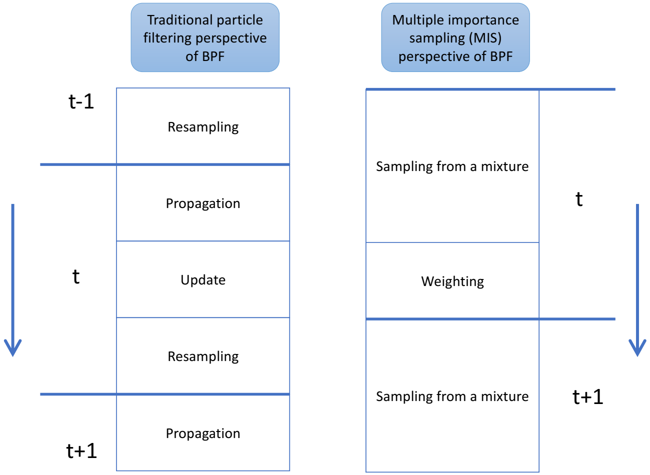
- Remark: the BPF matches just the prior of the numerator.

Toy example: BPF with $M = 4$ particles



- predictive, $p(x_t|y_{1:t-1}) = \sum_{j=1}^{M} w_{t-1}^{(j)} p(x_t|x_{t-1}^{(j)})$ with $w_{t-1} = [0.03,\ 0.16,\ 0.16,\ 0.65]$
- BPF proposal, $\psi_t^{\mathsf{BPF}}(x_t) = \sum_{j=1}^{M} \lambda_t^{(j)} p(x_t|x_{t-1}^{(j)})$, with $\lambda_t^{\mathsf{BPF}} = w_{t-1}^{(m)} = [0.03,\ 0.16,\ 0.16,\ 0.65]$

BPF from the MIS perspective



Auxiliary PF (APF)

- Proposed in [Pitt and Shephard, 1999] as an alternative to BPF of [Gordon, 1993]
  - APF improves sometimes the performance of BPF, but **not always.**

(i) Initialization. At time $t = 0$, $\mathbf{x}_0^{(m)} \sim p(\mathbf{x}_0)$, and $w_0^{(m)} = 1/M$, $m = 1, \ldots, M$.

(ii) Recursive step. At time $t > 0$,

1. **Modify weights before resampling.** Compute

$$\bar{\mathbf{x}}_t^{(m)} = \mathbb{E}_{p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})}[\mathbf{x}_t], \qquad m = 1, ..., M.$$

   and the normalized weights ($\sum_{m=1}^{M} \lambda_t^{(m)} = 1$)

$$\lambda_t^{(m)} \propto p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(m)}) w_{t-1}^{(m)}, \qquad m = 1, ..., M,$$

2. **Delayed resampling.** Select the indexes $i^{(m)} = j$, with probability proportional to $\lambda_t^{(j)}$, $m = 1, ...M$
3. **Prediction.** $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i^{(m)})})$, $\qquad m = 1, ..., M.$
4. **Update.** Compute the normalized weights as

$$w_t^{(m)} \quad \propto \quad \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})}{p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(i^{(m)})})}, \qquad m = 1, ..., M.$$

APF from the MIS perspective

(i) Initialization. At time $t=0$, $\mathbf{x}_0^{(m)} \sim p(\mathbf{x}_0)$, and $w_0^{(m)} = 1/M$, $m = 1, \ldots, M$.

(ii) Recursive step. At time $t > 0$,

1 **Proposal adaptation/selection.** The weight of each kernel in the mixture is amplified by the value of the likelihood at its center $\bar{\mathbf{x}}_t^{(m)} = \mathbb{E}_{p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})}[\mathbf{x}_t]$, i.e.,

$$\psi_t(\mathbf{x}_t) = \sum_{j=1}^{M} \lambda_t^{(j)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)}), \qquad \text{with} \quad \lambda_t^{(j)} \propto p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(j)}) w_{t-1}^{(j)}, \qquad j = 1, ..., M,$$

2 **Sampling.** Draw M i.i.d. samples from the mixture $\psi_t(\mathbf{x}_t)$, i.e.,

a) Select the indexes $i^{(m)} = j$, with probability $\propto \lambda_t^{(j)}$, $m = 1, \ldots M$

b) simulate $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i^{(m)})})$, $\qquad m = 1, \ldots M$.

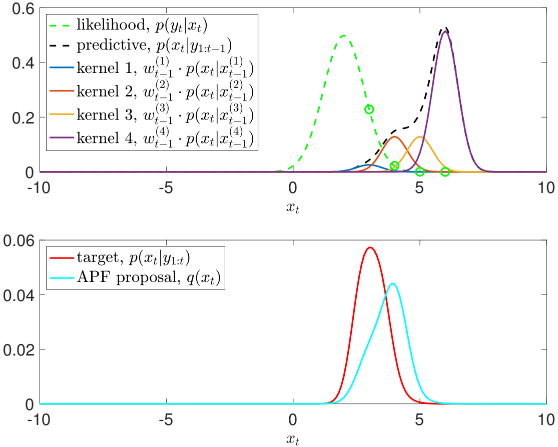
3 **Weighting.** Compute the normalized IS weights by

$$w_t^{(m)} \propto \frac{p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t})}{\psi_t(\mathbf{x}_t^{(m)})} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t-1})}{\sum_{j=1}^M \lambda_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})} \approx \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{j=1}^M w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^M \lambda_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}$$

$$\approx \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) w_{t-1}^{(i^{(m)})} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})}{\lambda_t^{(i^{(m)})} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) w_{t-1}^{(i^{(m)})} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})}{p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(i^{(m)})}) w_{t-1}^{(i^{(m)})} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})} = \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})}{p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(i^{(m)})})}$$

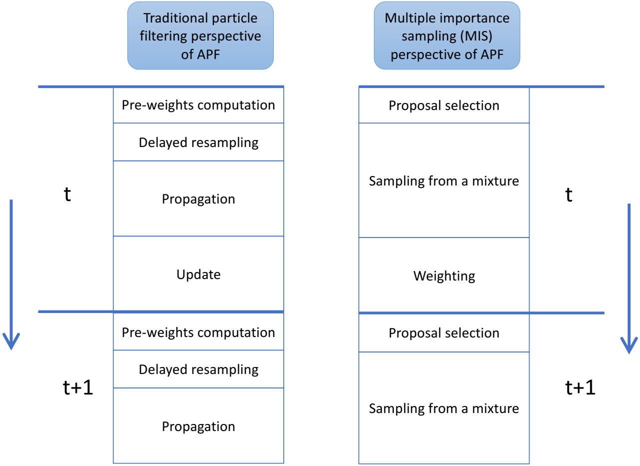
- Remark:
  - implicit assumption: kernels are far apart
  - the APF re-weights the kernels of the prior amplifying them with the likelihood (each of them, independently from the rest).

Toy example: APF with $M = 4$ particles



- predictive, $p(x_t|y_{1:t-1}) = \sum_{j=1}^{M} w_{t-1}^{(j)} p(x_t|x_{t-1}^{(j)})$ with $w_{t-1} = [0.03,\ 0.16,\ 0.16,\ 0.65]$
- APF proposal, $\psi_t^{\text{APF}}(x_t) = \sum_{j=1}^{M} \lambda_t^{(j)} p(x_t|x_{t-1}^{(j)})$, with $\lambda_t^{\text{APF}} = p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(m)}) w_{t-1}^{(m)} = [0.6713,\ 0.3221,\ 0.0065,\ 0.0001]$

Auxiliary PF (APF) from the MIS perspective



Improved APF (IAPF)

- IAPF: Based on this MIS interpretation, we improve the APF[7]
- It is in the proposed generic MIS framework:
  - The proposal is a **mixture of the same predictive kernels** as in BPF and APF

$$\psi_t(\mathbf{x}_t) = \sum_{j=1}^{M} \lambda_t^{(j)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)})$$

with

$$\lambda_t^{(j)} \propto \frac{p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(j)}) \sum_{k=1}^{M} w_{t-1}^{(k)} p(\bar{\mathbf{x}}_t^{(j)}|\mathbf{x}_{t-1}^{(k)})}{\sum_{k=1}^{M} p(\bar{\mathbf{x}}_t^{(j)}|\mathbf{x}_{t-1}^{(k)})}, \qquad j = 1, ..., M.$$

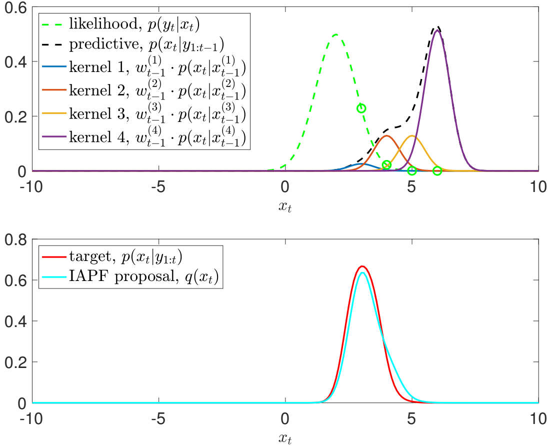
- **Interpration**:
  - reconstruction of the target distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ (numerator) with a mixture of kernels (denominator)[8]
  - the "amplification" $\lambda_t^{(j)}$ of j-th kernel, takes into account where all other kernels are placed (unlike APF)
  - APF fails when the kernels have important overlap
    - if kernels have few overlap, $\lambda_t^{(j)} \approx p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(j)}) w_{t-1}^{(j)}$ (IAPF reduces to APF)
- IS with no extra approximation:

$$w_t^{(m)} = \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M} \lambda_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})} \qquad m = 1, ..., M.$$

[7] V. Elvira et al. "In Search for Improved Auxiliary Particle Filters". In: *Signal Processing Conference (EUSIPCO), 2018 Proceedings of the 26th European*. IEEE. 2018, pp. 1–5.

[8] Elvira et al., "Generalized Multiple Importance Sampling".

Toy example: IAPF with $M = 4$ particles



- predictive, $p(x_t|y_{1:t-1}) = \sum_{j=1}^{M} w_{t-1}^{(j)} p(x_t|x_{t-1}^{(j)})$ with $w_{t-1} = [0.03,\ 0.16,\ 0.16,\ 0.65]$
- IAPF proposal, $\psi_t^{\text{IAPF}}(x_t) = \sum_{j=1}^{M} \lambda_t^{(j)} p(x_t|x_{t-1}^{(j)})$, with $\lambda_t^{\text{IAPF}} = [0.7657, 0.2276, 0.0066, 0.0001]$

Summary: PF framework from MIS perspective

(i) Initialization. At time $t = 0$, $\mathbf{x}_0^{(m)} \sim p(\mathbf{x}_0)$, and $w_0^{(m)} = 1/M$, $m = 1, \ldots, M$.

(ii) Recursive step. At time $t > 0$,

1. **Proposal adaptation/selection.** Select the MIS proposal of the form

$$\psi_t(\mathbf{x}_t) = \sum_{j=1}^{M} \lambda_t^{(j)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)}), \qquad \text{with} \qquad \lambda_t^{(j)} = ? \tag{3}$$

2. **Sampling.** Draw samples according to

$$\mathbf{x}_t^{(m)} \sim \psi_t(\mathbf{x}_t), \qquad m = 1, ..., M. \tag{4}$$

3. **Weighting.** Compute the normalized IS weights by

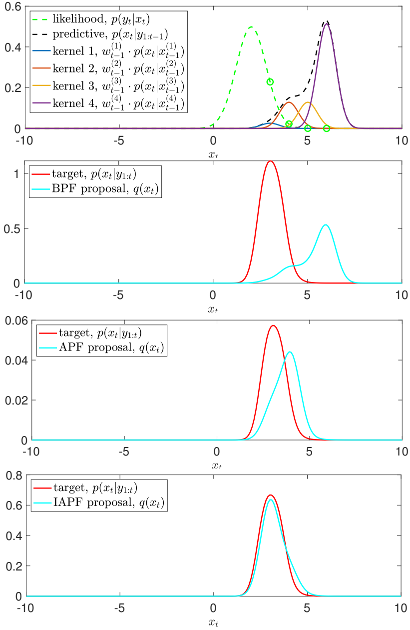
$$w_t^{(m)} = ? \tag{5}$$

	BPF	APF	IAPF
$\lambda_t^{(m)}$	$w_{t-1}^{(m)}$	$\propto p(\mathbf{y}_t\|\bar{\mathbf{x}}_t^{(m)})w_{t-1}^{(m)}$	$\propto \frac{p(\mathbf{y}_t\|\bar{\mathbf{x}}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\bar{\mathbf{x}}_t^{(m)}\|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M} p(\bar{\mathbf{x}}_t^{(m)}\|\mathbf{x}_{t-1}^{(j)})}$
$w_t^{(m)}$	$\propto p(\mathbf{y}_t\|\mathbf{x}_t^{(m)})$	$\propto \frac{p(\mathbf{y}_t\|\mathbf{x}_t^{(m)})}{p(\mathbf{y}_t\|\bar{\mathbf{x}}_t^{(i^m)})}$	$\propto \frac{p(\mathbf{y}_t\|\mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}\|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M} \lambda_t^{(j)} p(\mathbf{x}_t^{(m)}\|\mathbf{x}_{t-1}^{(j)})}$

- In all PFs:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{m=1}^{M} w_t^{(n)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(n)}\right)$$

Toy example: summary



Numerical result 1: channel estimation in wireless system

- We suppose a linear-Gaussian system described by

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{r}_t,$$
$$y_t = \mathbf{h}_t^\top \mathbf{x}_t + \mathbf{r}_t,$$

  - $\mathbf{h}_t = [h_t, h_{t-1}, ..., h_{t-d_x+1}]^\top$, last d_x transmitted pilots, $d_t \in \{-1, +1\}$,
  - $\mathbf{A} = 0.7\mathbf{I}$
  - $\mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q}), \mathbf{Q} = 5\mathbf{I}$
  - $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}), \mathbf{R} = 0.5$
- we set $T = 200$ time steps and $M = 100$ particles

d_x (dimension)	1	2	3	5	10
MSE (BPF)	0.0272	0.3762	0.9657	1.4705	2.9592
MSE (APF)	0.0709	0.8041	1.6041	2.2132	3.7187
MSE (IAPF)	**0.0062**	**0.1764**	**0.5176**	**0.8041**	**2.6931**

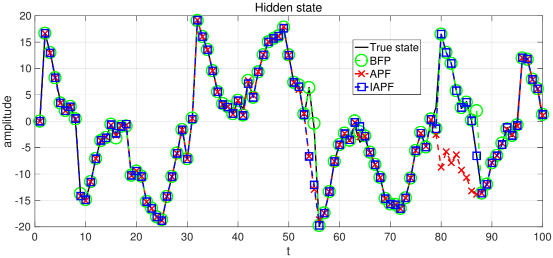
Numerical result 2: stochastic growth model

- We suppose a stochastic growth model

$$x_t = \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(\phi t) + u_t, \tag{6}$$

$$y_t = \frac{x_t^2}{20} + v_t, \tag{7}$$

where $\phi = 0.4$ is a frequency parameter (in rad/s), and u_t and v_t denote independent zero-mean univariate Gaussian r.v.'s with variance $\sigma_u^2 = 1$ and $\sigma_v^2 = 0.1$. $M = 100$ particles.



Conclusions and ongoing work

- APF has been used for a long time as an alternative to BPF
  - in many scenarios it works better but unclear when it fails
- Novel advances in MIS allow for reinterpreting PFs
  - **adapting-sampling-weighting** steps, instead of traditional **prediction-update-resampling**
  - APF is derived and the approximations/assumptions are explicit
- We also propose an IAPF that yields for a better proposal than APF, and hence, better performance
  - computationally expensive, but AIS techniques can be used to alleviate it
- Ongoing work for optimized (high-performance) yet efficient variants of APF: OAPF[9]
- This new interpretation paves the way for novel PFs but also for better understanding of the existing ones:
  - it is now easier to interpret which filter is more appropriate in each scenario

[9] N. Branchini and V. Elvira. "Optimized auxiliary particle filters: adapting mixture proposals via convex optimization". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 1289–1299.

Thank you for your attention!