THE UNIVERSITY *of* EDINBURGH

Particle filtering (L4)

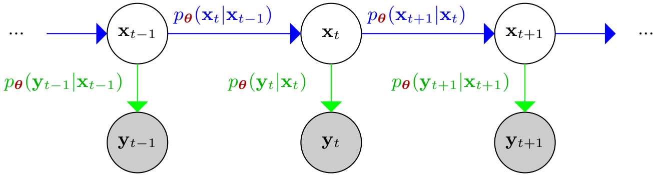
Víctor Elvira
School of Mathematics
University of Edinburgh
(victor.elvira@ed.ac.uk)

PhD course on Bayesian filtering and Monte Carlo methods
UPC, Barcelona, July 7-11, 2025

Outline

1. Modeling: state-space models (SSM)

- Time-series data are collected, $\mathbf{y}_t \in \mathbb{R}^{N_y}$, $t = 1, ..., T$:
- A SSM models a sequence of hidden states $\mathbf{x}_t \in \mathbb{R}^{N_x}$, $t = 1, ..., T$.



- 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 $\rightarrow$ $p_{\boldsymbol{\theta}}(\mathbf{x}_0) = p(\mathbf{x}_0|\boldsymbol{\theta})$

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

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

Outline

Monte Carlo methods: a bit of history

- A methodology that comes to the rescue for solving most difficult problems of inference is based on **drawing/simulation** of samples.
  - e.g., in Bayesian inference, for most of models of interest, it is usually **impossible** to find posteriors distributions nor simulate from them.
- The Monte Carlo methods were born in Los Alamos, New Mexico (USA) in the 1940s around the Manhattan project
  - associated to the Electronic Numerical Integrator and Computer **(ENIAC)**, one of the first electronic general-purpose computers.
- Foundational works of Stanislaw Ulam (1909-1984) and John von Neumann (1903-1957)
  - they invented the **inversion** and **accept-reject** techniques
  - independently developed by Enrico Fermi (1901-1954)
  - beautiful story at[1]
- Led to the **Metropolis algorithm** by Nicolas Metropolis in 1953
  - the first **Markov chain Monte Carlo (MCMC)** algorithm
  - listed among the "10 algorithms with the greatest influence on the development of science and engineering in the 20th century", by the American Institute of Physics and the IEEE Computer Society in 2000.

---

https://poole.ncsu.edu/thought-leadership/article/oppenheimer-ulam-and-risk-analytics-the-legacy-of-wwii-scientists-on-contemporary-computing/

[1] N. Metropolis et al. "The beginning of the Monte Carlo method". In: *Los Alamos Science* 15.584 (1987), pp. 125–130.

Monte Carlo basics

- Goal:[2] approximate the integral

$$I(f) \equiv \mathrm{E}_{\pi(\mathbf{x})}[f(\mathbf{x})] = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$$

- If we can sample $\mathbf{x}^{(n)} \sim \pi(\mathbf{x})$, $n = 1, ..., N$, then the target can be approximated as

$$\pi(\mathbf{x}) \approx \pi^N(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\mathbf{x}^{(n)}}(\mathbf{x})$$

and the moment $I(f)$ can be approximated as

$$\boxed{I(f) \approx \bar{I}_N(f) = \frac{1}{N}\sum_{n=1}^{N} f\left(\mathbf{x}^{(n)}\right)}$$

[2] C. P. Robert, G. Casella, and G. Casella. *Monte Carlo statistical methods*. Vol. 2. Springer, 1999.

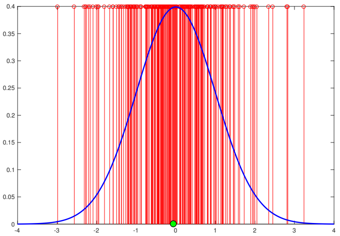
Monte Carlo basics

- We can approximate any integral involving $\pi(\mathbf{x})$: integral + dirac = sum!

$$\mathrm{E}_{\pi(\mathbf{x})}[f(\mathbf{x})] = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \approx \int f(\mathbf{x})\pi^{N}(\mathbf{x})d\mathbf{x} \approx \int f(\mathbf{x}) \left(\frac{1}{N} \sum_{n=1}^{N} \delta_{\mathbf{x}^{(n)}}(\mathbf{x}) \right) d\mathbf{x}$$
$$\approx \frac{1}{N} \sum_{n=1}^{N} \int f(\mathbf{x}) \delta_{\mathbf{x}^{(n)}}(\mathbf{x}) d\mathbf{x}$$
$$= \frac{1}{N} \sum_{n=1}^{N} f\left(\mathbf{x}^{(n)}\right)$$

- e.g. the mean of the distribution ($h(\mathbf{x}) = \mathbf{x}$) approximated by $N = 200$.

$$\hat{\mathbf{x}} = \mathrm{E}_{\pi(\mathbf{x})}[\mathbf{x}] = \int \mathbf{x}\pi(\mathbf{x})d\mathbf{x} \approx \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}^{(n)} = -0.0561$$



Importance sampling basics

- Unfortunately we only know how to draw samples from very few distributions.
  - In the general case, we cannot use the raw/direct/standard Monte Carlo.
  - **Importance sampling (IS)** is a Monte Carlo method that allows to approximate integrals over complicated distributions.[3]
- Same problem:

$$I(f) \equiv \mathrm{E}_{\pi(\mathbf{x})}[f(\mathbf{x})] = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{\pi(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x}$$

where $q(\mathbf{x})$ is the proposal density where the samples are now drawn

Standard **MC** estimator: $\bar{I}_N(f) = \frac{1}{N}\sum_{n=1}^{N} f\left(\mathbf{x}^{(n)}\right), \qquad \mathbf{x}^{(n)} \sim \pi(\mathbf{x})$

Basic **IS** estimator: $\hat{I}_N(f) = \frac{1}{N}\sum_{n=1}^{N} f\left(\mathbf{x}^{(n)}\right)\frac{\pi\left(\mathbf{x}^{(n)}\right)}{q\left(\mathbf{x}^{(n)}\right)}, \qquad \mathbf{x}^{(n)} \sim q(\mathbf{x})$

[3] V. Elvira and L. Martino. "Advances in importance sampling". In: *arXiv preprint arXiv:2102.05407* (2021).

Importance sampling basics

- Basic **IS** estimator:

$$\hat{I}_N(f) = \frac{1}{N}\sum_{n=1}^{N} f\left(\mathbf{x}^{(n)}\right)\frac{\pi\left(\mathbf{x}^{(n)}\right)}{q\left(\mathbf{x}^{(n)}\right)}, \qquad \mathbf{x}^{(n)} \sim q(\mathbf{x})$$

  - $W^{(n)} = \frac{\pi\left(\mathbf{x}^{(n)}\right)}{q\left(\mathbf{x}^{(n)}\right)}$, $n = 1, ..., N$, are the **importance weights**
  - only constraint: $\pi(\mathbf{x})$ must be evaluated
- Unfortunately, sometimes we have only access to $\gamma(\mathbf{x})$, where $\pi(\mathbf{x}) = \frac{\gamma(\mathbf{x})}{Z}$, Z is the normalizing constant:
  - basic IS estimator is not possible $\Rightarrow$ self-normalize estimator:

$$\widetilde{I}_N(f) = \sum_{n=1}^{N} f\left(\mathbf{x}^{(n)}\right) w^{(n)}, \qquad \mathbf{x}^{(n)} \sim q(\mathbf{x})$$

    where $w^{(n)} = \frac{W^{(n)}}{\sum_{j=1}^{N} W^{(n)}}$ are the normalized weights ($\sum_{j=1}^{N} w^{(j)} = 1$).
    - here the weights $W^{(n)}$ can be computed by evaluating $\gamma(\mathbf{x})$ instead.
  - Approximation of the targeted distribution

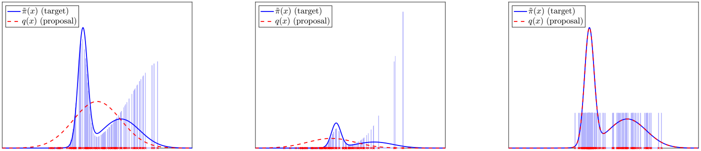
$$\pi(\mathbf{x}) \approx \pi_{\text{IS}}^{N}(\mathbf{x}) = \sum_{n=1}^{N} w^{(n)}\delta_{\mathbf{x}^{(n)}}(\mathbf{x})$$

The variance in IS and the need of better proposals

- A good proposal $q(\mathbf{x})$ is key for the efficiency of IS.
- Variance of the UIS estimator of $I = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$:

$$\mathrm{Var}_{\pi(\mathbf{x})}(\widehat{I}) = \frac{1}{N}\int \frac{f^2(\mathbf{x})\pi^2(\mathbf{x})}{q(\mathbf{x})}d\mathbf{x} - \frac{I^2}{N}$$

- optimal UIS proposal: $q(\mathbf{x}) \propto |f(\mathbf{x})|\pi(\mathbf{x})$
- for a generic $f(\mathbf{x})$, $q(\mathbf{x})$ should be as *close* as possible to $\pi(\mathbf{x})$



- Very difficult to find a good $q(\mathbf{x})$ a priori:
  - $\pi(\mathbf{x})$ can be only evaluated (up to a normalizing constant)
  - $\pi(\mathbf{x})$ may be multimodal, skewed, heavy tailed
- A posteriori metric: $\widehat{\mathrm{ESS}} = \frac{1}{\sum_{n=1}^{N}\bar{w}_n^2}$, although it presents serious problems[4]
- Use **multiple** proposals (**MIS**) and **explore** the space (**AIS**).

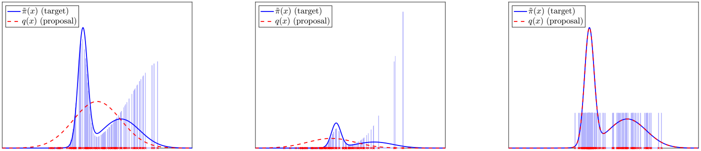
[4] V. Elvira, L. Martino, and C. P. Robert. "Rethinking the effective sample size". In: *International Statistical Review* 90.3 (2022), pp. 525–550.

The variance in IS and the need of better proposals

- A good proposal $q(\mathbf{x})$ is key for the efficiency of IS.
- Variance of the UIS estimator of $I = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$:

$$\mathrm{Var}_{\pi(\mathbf{x})}(\widehat{I}) = \frac{1}{N}\int \frac{f^2(\mathbf{x})\pi^2(\mathbf{x})}{q(\mathbf{x})}d\mathbf{x} - \frac{I^2}{N}$$

- optimal UIS proposal: $q(\mathbf{x}) \propto |f(\mathbf{x})|\pi(\mathbf{x})$
- for a generic $f(\mathbf{x})$, $q(\mathbf{x})$ should be as *close* as possible to $\pi(\mathbf{x})$



- Very difficult to find a good $q(\mathbf{x})$ a priori:
  - $\pi(\mathbf{x})$ can be only evaluated (up to a normalizing constant)
  - $\pi(\mathbf{x})$ may be multimodal, skewed, heavy tailed
- A posteriori metric: $\widehat{\mathrm{ESS}} = \frac{1}{\sum_{n=1}^{N}\bar{w}_n^2}$, although it presents serious problems[4]
- Use **multiple** proposals (**MIS**) and **explore** the space (**AIS**).

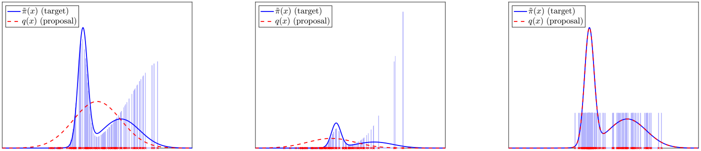
[4] V. Elvira, L. Martino, and C. P. Robert. "Rethinking the effective sample size". In: *International Statistical Review* 90.3 (2022), pp. 525–550.

The variance in IS and the need of better proposals

- A good proposal $q(\mathbf{x})$ is key for the efficiency of IS.
- Variance of the UIS estimator of $I = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$:

$$\mathrm{Var}_{\pi(\mathbf{x})}(\widehat{I}) = \frac{1}{N}\int \frac{f^2(\mathbf{x})\pi^2(\mathbf{x})}{q(\mathbf{x})}d\mathbf{x} - \frac{I^2}{N}$$

- optimal UIS proposal: $q(\mathbf{x}) \propto |f(\mathbf{x})|\pi(\mathbf{x})$
- for a generic $f(\mathbf{x})$, $q(\mathbf{x})$ should be as *close* as possible to $\pi(\mathbf{x})$



- Very difficult to find a good $q(\mathbf{x})$ a priori:
  - $\pi(\mathbf{x})$ can be only evaluated (up to a normalizing constant)
  - $\pi(\mathbf{x})$ may be multimodal, skewed, heavy tailed
- A posteriori metric: $\widehat{\mathrm{ESS}} = \frac{1}{\sum_{n=1}^{N}\bar{w}_n^2}$, although it presents serious problems[4]
- Use **multiple** proposals (**MIS**) and **explore** the space (**AIS**).

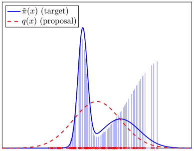
[4] V. Elvira, L. Martino, and C. P. Robert. "Rethinking the effective sample size". In: *International Statistical Review* 90.3 (2022), pp. 525–550.

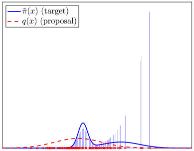
The variance in IS and the need of better proposals

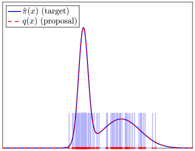
- A good proposal $q(\mathbf{x})$ is key for the efficiency of IS.
- Variance of the UIS estimator of $I = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$:

$$\mathrm{Var}_{\pi(\mathbf{x})}(\widehat{I}) = \frac{1}{N}\int \frac{f^2(\mathbf{x})\pi^2(\mathbf{x})}{q(\mathbf{x})}d\mathbf{x} - \frac{I^2}{N}$$

- optimal UIS proposal: $q(\mathbf{x}) \propto |f(\mathbf{x})|\pi(\mathbf{x})$
- for a generic $f(\mathbf{x})$, $q(\mathbf{x})$ should be as *close* as possible to $\pi(\mathbf{x})$





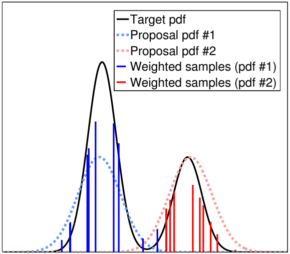


- Very difficult to find a good $q(\mathbf{x})$ a priori:
  - $\pi(\mathbf{x})$ can be only evaluated (up to a normalizing constant)
  - $\pi(\mathbf{x})$ may be multimodal, skewed, heavy tailed
- A posteriori metric: $\widehat{\mathrm{ESS}} = \frac{1}{\sum_{n=1}^N \bar{w}_n^2}$, although it presents serious problems[4]
- Use **multiple** proposals (**MIS**) and **explore** the space (**AIS**).

[4] V. Elvira, L. Martino, and C. P. Robert. "Rethinking the effective sample size". In: *International Statistical Review* 90.3 (2022), pp. 525–550.

Multiple Importance Sampling (MIS): Basics

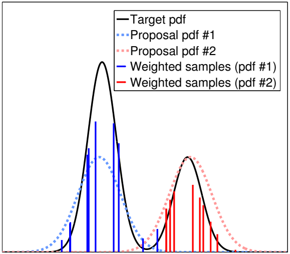
- Set of N available proposal pdfs $\{q_n(\mathbf{x})\}_{n=1}^{N}$.
  - Example $N = 2$:



- For simplicity, **we simulate** N **samples** in total from the set $\{q_n(\mathbf{x})\}_{n=1}^{N}$, but how?
  1. **Sampling:** $\mathbf{x}_n \sim ?$, $\quad n = 1, ..., N$.
  2. **Weighting:** $W_n = ?$, $\quad n = 1, ..., N$.
- Most known MIS algorithms focus just in the adaptation (AIS):
  - Implement **sampling** and **weighting** in a different way without much justification.

Multiple Importance Sampling (MIS): Basics

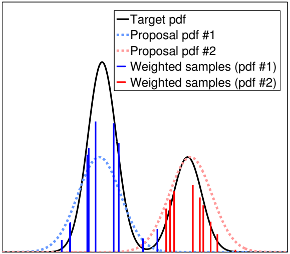
- Set of N available proposal pdfs $\{q_n(\mathbf{x})\}_{n=1}^N$.
  - Example $N = 2$:



- For simplicity, **we simulate** N **samples** in total from the set $\{q_n(\mathbf{x})\}_{n=1}^N$, but how?
  1. **Sampling:** $\mathbf{x}_n \sim ?, \qquad n = 1, ..., N.$
  2. **Weighting:** $W_n = ?, \qquad n = 1, ..., N.$
- Most known MIS algorithms focus just in the adaptation (AIS):
  - Implement **sampling** and **weighting** in a different way without much justification.

Multiple Importance Sampling (MIS): Basics

- Set of N available proposal pdfs $\{q_n(\mathbf{x})\}_{n=1}^{N}$.
  - Example $N = 2$:



- For simplicity, **we simulate** N **samples** in total from the set $\{q_n(\mathbf{x})\}_{n=1}^{N}$, but how?
  1. **Sampling:** $\mathbf{x}_n \sim ?, \qquad n = 1, ..., N.$
  2. **Weighting:** $W_n = ?, \qquad n = 1, ..., N.$
- Most known MIS algorithms focus just in the adaptation (AIS):
  - Implement **sampling** and **weighting** in a different way without much justification.

Multiple Importance Sampling: Interpretations

a) Population Monte Carlo [Cappe04]:
   1. **Sampling:** $\mathbf{x}_n \sim q_n(\mathbf{x}), \qquad n = 1, ..., N.$
   2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{q_n(\mathbf{x}_n)}, \qquad n = 1, ..., N.$
   - Estimators can be unstable.[5]

b) M-PMC [Douc07a,Cappe08]:
   1. **Sampling:** $\mathbf{x}_n \overset{i.i.d}{\sim} \psi(\mathbf{x}), \qquad n = 1, ..., N.$
   2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{\psi(\mathbf{x}_n)}, \qquad n = 1, ..., N.$
   - Some proposals can be used more than once, others can be not used.

$$\boxed{\psi(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} q_j(\mathbf{x})}$$

c) Deterministic mixture (DM)
   1. **Sampling:** $\mathbf{x}_n \sim q_n(\mathbf{x}), \qquad n = 1, ..., N.$
   2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{\psi(\mathbf{x}_n)}, \qquad n = 1, ..., N.$
   - $\mathbf{x}_n$ is not drawn from ψ, but estimators are consistent and efficient.

- Several questions:
  - Why all these **sampling/weighting** schemes are valid?
  - Are some schemes better than others?
  - Are there other valid schemes?
  - Novel theoretical framework for MIS $\Rightarrow$ Generalized MIS

[5] O. Cappé, A. Guillin, J.-M. Marin, and C. P. Robert. "Population monte carlo". In: *Journal of Computational and Graphical Statistics* 13.4 (2004), pp. 907–929.

Multiple Importance Sampling: Interpretations

a) Population Monte Carlo [Cappe04]:

1. **Sampling:** $\mathbf{x}_n \sim q_n(\mathbf{x}), \qquad n = 1, ..., N.$
2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{q_n(\mathbf{x}_n)}, \qquad n = 1, ..., N.$

- Estimators can be unstable.

b) M-PMC [Douc07a,Cappe08]:

1. **Sampling:** $\mathbf{x}_n \overset{i.i.d}{\sim} \psi(\mathbf{x}), \qquad n = 1, ..., N.$
2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{\psi(\mathbf{x}_n)}, \qquad n = 1, ..., N.$

$$\boxed{\psi(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} q_j(\mathbf{x})}$$

- Some proposals can be used more than once, others can be not used.[5][6]

c) Deterministic mixture (DM)

1. **Sampling:** $\mathbf{x}_n \sim q_n(\mathbf{x}), \qquad n = 1, ..., N.$
2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{\psi(\mathbf{x}_n)}, \qquad n = 1, ..., N.$

- $\mathbf{x}_n$ is not drawn from ψ, but estimators are consistent and efficient.

- Several questions:
  - Why all these **sampling/weighting** schemes are valid?
  - Are some schemes better than others?
  - Are there other valid schemes?
  - Novel theoretical framework for MIS ⇒ Generalized MIS

[5] R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. "Minimum variance importance sampling via population Monte Carlo". In: *ESAIM: Probability and Statistics* 11 (2007), pp. 427–447.

[6] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. "Adaptive importance sampling in general mixture classes". In: *Statistics and Computing* 18 (2008), pp. 447–459.

Multiple Importance Sampling: Interpretations

a) Population Monte Carlo [Cappe04]:

1. **Sampling:** $\mathbf{x}_n \sim q_n(\mathbf{x}), \qquad n = 1, ..., N.$
2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{q_n(\mathbf{x}_n)}, \qquad n = 1, ..., N.$
   - Estimators can be unstable.

b) M-PMC [Douc07a,Cappe08]:

1. **Sampling:** $\mathbf{x}_n \overset{i.i.d}{\sim} \psi(\mathbf{x}), \qquad n = 1, ..., N.$
2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{\psi(\mathbf{x}_n)}, \qquad n = 1, ..., N.$

$$\boxed{\psi(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} q_j(\mathbf{x})}$$

   - Some proposals can be used more than once, others can be not used.

c) Deterministic mixture (DM)

1. **Sampling:** $\mathbf{x}_n \sim q_n(\mathbf{x}), \qquad n = 1, ..., N.$
2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{\psi(\mathbf{x}_n)}, \qquad n = 1, ..., N.$
   - $\mathbf{x}_n$ is not drawn from ψ, but estimators are consistent and efficient.[5][6]

- Several questions:
  - Why all these **sampling**/**weighting** schemes are valid?
  - Are some schemes better than others?
  - Are there other valid schemes?
  - Novel theoretical framework for MIS $\Rightarrow$ Generalized MIS

[5] E. Veach and L. J. Guibas. "Optimally combining sampling techniques for Monte Carlo rendering". In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques.* 1995, pp. 419–428.

[6] A. Owen and Y. Zhou. "Safe and effective importance sampling". In: *Journal of the American Statistical Association* 95.449 (2000), pp. 135–143.

Multiple Importance Sampling: Interpretations

a) Population Monte Carlo [Cappe04]:

1. **Sampling:** $\mathbf{x}_n \sim q_n(\mathbf{x}), \qquad n = 1, ..., N.$
2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{q_n(\mathbf{x}_n)}, \qquad n = 1, ..., N.$
   - Estimators can be unstable.

b) M-PMC [Douc07a,Cappe08]:

1. **Sampling:** $\mathbf{x}_n \overset{i.i.d}{\sim} \psi(\mathbf{x}), \qquad n = 1, ..., N.$
2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{\psi(\mathbf{x}_n)}, \qquad n = 1, ..., N.$
   - Some proposals can be used more than once, others can be not used.

$$\boxed{\psi(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} q_j(\mathbf{x})}$$

c) Deterministic mixture (DM)

1. **Sampling:** $\mathbf{x}_n \sim q_n(\mathbf{x}), \qquad n = 1, ..., N.$
2. **Weighting:** $W_n = \frac{\pi(\mathbf{x}_n)}{\psi(\mathbf{x}_n)}, \qquad n = 1, ..., N.$
   - $\mathbf{x}_n$ is not drawn from ψ, but estimators are consistent and efficient.

- Several questions:
  - Why all these **sampling/weighting** schemes are valid?
  - Are some schemes better than others?
  - Are there other valid schemes?
  - Novel theoretical framework for MIS $\Rightarrow$ Generalized MIS[5]

[5] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.

Generalized MIS: Schemes with Replacement

- Example with $N = 3$ proposals. For each $n \in \{1, 2, 3\}$:[6]
  1. simulate $j_n \sim \text{Cat}([1, 2, 3]; [1/3, 1/3, 1/3])$
  2. simulate $\mathbf{x}_n \sim q_{j_n}(\mathbf{x})$
  3. weight $W_n = \frac{\pi(\mathbf{x}_n)}{\varphi_n(\mathbf{x}_n)}$ (several options for φ_n)
- 1. and 2. are equivalent to mixture sampling:

$$\mathbf{x}_n \overset{i.i.d.}{\sim} \psi(\mathbf{x}) = \frac{1}{3}\sum_{n=1}^{N} q_n(\mathbf{x})$$

Available proposals		1 2 3	1 2 3	1 2 3
Sampling	j_n	3	3	1
	$\mathbf{x}_n$	$\mathbf{x}_1 \sim q_3$	$\mathbf{x}_2 \sim q_3$	$\mathbf{x}_3 \sim q_1$
Weighting options $w_n = \frac{\pi(\mathbf{x})}{\varphi_n(\mathbf{x})}$	R1	$\frac{\pi(\mathbf{x}_1)}{q_3(\mathbf{x}_1)}$	$\frac{\pi(\mathbf{x}_2)}{q_3(\mathbf{x}_2)}$	$\frac{\pi(\mathbf{x}_3)}{q_1(\mathbf{x}_3)}$
	R2	$\frac{\pi(\mathbf{x}_1)}{\frac{1}{3}(q_3(\mathbf{x}_1)+q_3(\mathbf{x}_1)+q_1(\mathbf{x}_1))}$	$\frac{\pi(\mathbf{x}_2)}{\frac{1}{3}(q_3(\mathbf{x}_2)+q_3(\mathbf{x}_2)+q_1(\mathbf{x}_2))}$	$\frac{\pi(\mathbf{x}_3)}{\frac{1}{3}(q_3(\mathbf{x}_3)+q_3(\mathbf{x}_3)+q_1(\mathbf{x}_3))}$
	R3	$\frac{\pi(\mathbf{x}_1)}{\frac{1}{3}(q_1(\mathbf{x}_1)+q_2(\mathbf{x}_1)+q_3(\mathbf{x}_1))}$	$\frac{\pi(\mathbf{x}_2)}{\frac{1}{3}(q_1(\mathbf{x}_2)+q_2(\mathbf{x}_2)+q_3(\mathbf{x}_2))}$	$\frac{\pi(\mathbf{x}_3)}{\frac{1}{3}(q_1(\mathbf{x}_3)+q_2(\mathbf{x}_3)+q_3(\mathbf{x}_3))}$

[6] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.

Generalized MIS: Schemes with No Replacement

- Example with $N = 3$ proposals. For each $n \in \{1, 2, 3\}$:[7]
  1. set $j_n = n$
  2. simulate $x_n \sim q_{j_n}(\mathbf{x})$
  3. weight $W_n = \frac{\pi(\mathbf{x}_n)}{\varphi_n(\mathbf{x}_n)}$ (several options for φ_n)
- 1. and 2. can be seen as mixture sampling $\mathbf{x}_n \sim \psi(\mathbf{x}) = \frac{1}{3}\sum_{n=1}^{N} q_n(\mathbf{x})$
  - but not i.i.d.!
  - only possible for mixtures with specific weights

Available proposals		1	2	3
Sampling	j_n	1	2	3
	$\mathbf{x}_n$	$\mathbf{x}_1 \sim q_1$	$\mathbf{x}_2 \sim q_2$	$\mathbf{x}_3 \sim q_3$
Weighting options $w_n = \frac{\pi(\mathbf{x})}{\varphi_n(\mathbf{x})}$	N1	$\frac{\pi(\mathbf{x}_1)}{q_1(\mathbf{x}_1)}$	$\frac{\pi(\mathbf{x}_2)}{q_2(\mathbf{x}_2)}$	$\frac{\pi(\mathbf{x}_3)}{q_3(\mathbf{x}_3)}$
	N3	$\frac{\pi(\mathbf{x}_1)}{\frac{1}{3}(q_1(\mathbf{x}_1)+q_2(\mathbf{x}_1)+q_3(\mathbf{x}_1))}$	$\frac{\pi(\mathbf{x}_2)}{\frac{1}{3}(q_1(\mathbf{x}_2)+q_2(\mathbf{x}_2)+q_3(\mathbf{x}_2))}$	$\frac{\pi(\mathbf{x}_3)}{\frac{1}{3}(q_1(\mathbf{x}_3)+q_2(\mathbf{x}_3)+q_3(\mathbf{x}_3))}$

[7] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.

Generalized MIS: Variance analysis

Theorem.[8] *For any **target distribution** $\pi(\mathbf{x})$, any integrable **function** h, and any **set of proposal densities** $\{q_n(\mathbf{x})\}_{n=1}^N$ such that the variance of the corresponding MIS estimators is finite,*

$$Var(\hat{I}_{\mathtt{N1}}) = Var(\hat{I}_{\mathtt{R1}}) \geq Var(\hat{I}_{\mathtt{R3}}) \geq Var(\hat{I}_{\mathtt{N3}})$$

[8] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.

Generalized MIS: Numerical example

- Simulate M total samples from two proposals:

$$\mathbf{x}_1^{(m)} \sim q_1, \qquad m = 1, ..., M/2$$
$$\mathbf{x}_2^{(m)} \sim q_2, \qquad m = 1, ..., M/2$$

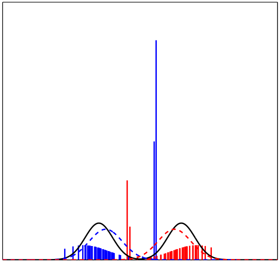
- Weighting:

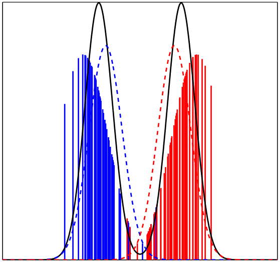
(a) N1

$$w_1^{(m)} = \frac{\pi(\mathbf{x}_1^{(m)})}{q_1(\mathbf{x}_1^{(m)})}$$
$$w_2^{(m)} = \frac{\pi(\mathbf{x}_2^{(m)})}{q_2(\mathbf{x}_2^{(m)})}$$

(b) N3

$$w_1^{(m)} = \frac{\pi(\mathbf{x}_1)}{\frac{1}{2}(q_1(\mathbf{x}_1^{(m)})+q_2(\mathbf{x}_1^{(m)}))}$$
$$w_2^{(m)} = \frac{\pi(\mathbf{x}_2)}{\frac{1}{2}(q_1(\mathbf{x}_2^{(m)})+q_2(\mathbf{x}_2^{(m)}))}$$





MIS Scheme	R1	N1	R2	N2	R3	N3
$\mathrm{Var}(\widehat{Z})$	1847	6874	10285	5474	0.01	0.01

Generalized MIS: Numerical example

- Simulate M total samples from two proposals:

$$\mathbf{x}_1^{(m)} \sim q_1, \qquad m = 1, \dots, M/2$$
$$\mathbf{x}_2^{(m)} \sim q_2, \qquad m = 1, \dots, M/2$$

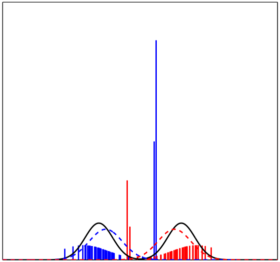
- Weighting:

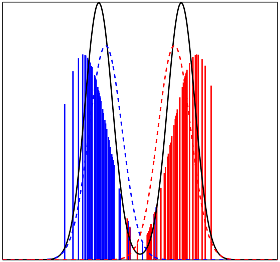
(a) N1

$$w_1^{(m)} = \frac{\pi(\mathbf{x}_1^{(m)})}{q_1(\mathbf{x}_1^{(m)})}$$
$$w_2^{(m)} = \frac{\pi(\mathbf{x}_2^{(m)})}{q_2(\mathbf{x}_2^{(m)})}$$

(b) N3

$$w_1^{(m)} = \frac{\pi(\mathbf{x}_1)}{\frac{1}{2}(q_1(\mathbf{x}_1^{(m)}) + q_2(\mathbf{x}_1^{(m)}))}$$
$$w_2^{(m)} = \frac{\pi(\mathbf{x}_2)}{\frac{1}{2}(q_1(\mathbf{x}_2^{(m)}) + q_2(\mathbf{x}_2^{(m)}))}$$





MIS Scheme	R1	N1	R2	N2	R3	N3
$\mathrm{Var}(\widehat{Z})$	1847	6874	10285	5474	0.01	0.01

Generalized MIS: Numerical example

- Simulate M total samples from two proposals:

$$\mathbf{x}_1^{(m)} \sim q_1, \qquad m = 1, \ldots, M/2$$
$$\mathbf{x}_2^{(m)} \sim q_2, \qquad m = 1, \ldots, M/2$$

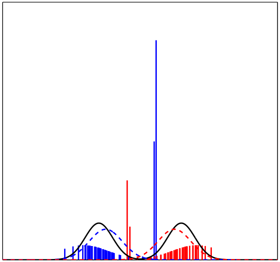
- Weighting:

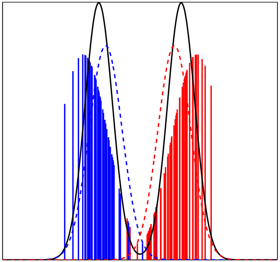
(a) N1

$$w_1^{(m)} = \frac{\pi(\mathbf{x}_1^{(m)})}{q_1(\mathbf{x}_1^{(m)})}$$
$$w_2^{(m)} = \frac{\pi(\mathbf{x}_2^{(m)})}{q_2(\mathbf{x}_2^{(m)})}$$

(b) N3

$$w_1^{(m)} = \frac{\pi(\mathbf{x}_1)}{\frac{1}{2}(q_1(\mathbf{x}_1^{(m)})+q_2(\mathbf{x}_1^{(m)}))}$$
$$w_2^{(m)} = \frac{\pi(\mathbf{x}_2)}{\frac{1}{2}(q_1(\mathbf{x}_2^{(m)})+q_2(\mathbf{x}_2^{(m)}))}$$





MIS Scheme	R1	N1	R2	N2	R3	N3
$\mathrm{Var}(\widehat{Z})$	1847	6874	10285	5474	0.01	0.01

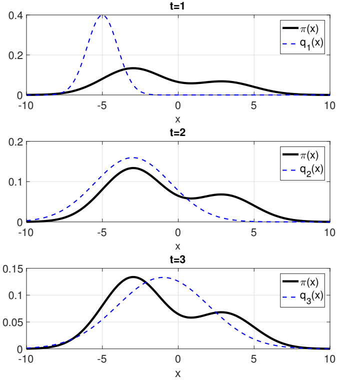
Adaptive Importance Sampling: Basics

- N proposals $\{q_{n,j}(\mathbf{x}|\boldsymbol{\theta}_{n,j})\}_{n=1}^{N}$ adapted over $j = 1, ..., J$ iterations

$$\{q_{n,1}(\mathbf{x}|\boldsymbol{\theta}_{n,1})\}_{n=1}^{N} \rightarrow \{q_{n,2}(\mathbf{x}|\boldsymbol{\theta}_{n,2})\}_{n=1}^{N} \rightarrow ... \rightarrow \{q_{n,J}(\mathbf{x}|\boldsymbol{\theta}_{n,J}\}_{n=1}^{N}$$

- Parametric AIS summarized as:[9]

$$\{\boldsymbol{\theta}_{n,1}\}_{n=1}^{N} \rightarrow \{\boldsymbol{\theta}_{n,2}\}_{n=1}^{N} \rightarrow ... \rightarrow \{\boldsymbol{\theta}_{n,J}\}_{n=1}^{N}$$



[9] M. F. Bugallo et al. "Adaptive importance sampling: The past, the present, and the future". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 60–79.

Adaptive Importance Sampling: Generic Algorithm

Initialization: Choose J, N, K, $\{q_{n,1}\}_{n=1}^N$, and initial parameters $\{\boldsymbol{\theta}_{n,1}\}_{n=1}^N$

For $j = 1, \ldots, J$:

1. **Sampling:** Simulate NK samples as

$$\mathbf{x}_{n,j}^{(k)} \sim q_{n,j}(\mathbf{x}|\boldsymbol{\theta}_{n,j}), \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

2. **Weighting:** Weight the NK samples as [Elvira19]

$$W_{n,j}^{(k)} = \frac{\pi(\mathbf{x}_{n,j}^{(k)})}{\varphi_{n,j}(\mathbf{x}_{n,j}^{(k)})}, \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

3. **Adaptation of the parameters:** Update the proposal parameters

$$\{\boldsymbol{\theta}_{n,j}\}_{n=1}^N \xrightarrow{\text{Adapt}} \{\boldsymbol{\theta}_{n,j+1}\}_{n=1}^N$$

Outputs: NKJ weighted samples, $\{\mathbf{x}_{n,j}^{(k)}, W_{n,j}^{(k)}\}_{n=1,k=1,j=1}^{N,K,J}$

two questions: (1) Weighting scheme?[10] (2) Adaptive procedure of $\boldsymbol{\theta}_{n,j}$?[11]

[10] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.

[11] M. F. Bugallo et al. "Adaptive importance sampling: The past, the present, and the future". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 60–79.

Adaptive Importance Sampling: Generic Algorithm

Initialization: Choose J, N, K, $\{q_{n,1}\}_{n=1}^{N}$, and initial parameters $\{\boldsymbol{\theta}_{n,1}\}_{n=1}^{N}$
For $j = 1, \ldots, J$**:**

1. **Sampling:** Simulate NK samples as

$$\mathbf{x}_{n,j}^{(k)} \sim q_{n,j}(\mathbf{x}|\boldsymbol{\theta}_{n,j}), \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

2. **Weighting:** Weight the NK samples as [Elvira19]

$$W_{n,j}^{(k)} = \frac{\pi(\mathbf{x}_{n,j}^{(k)})}{\varphi_{n,j}(\mathbf{x}_{n,j}^{(k)})}, \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

3. **Adaptation of the parameters:** Update the proposal parameters

$$\{\boldsymbol{\theta}_{n,j}\}_{n=1}^{N} \xrightarrow{\text{Adapt}} \{\boldsymbol{\theta}_{n,j+1}\}_{n=1}^{N}$$

Outputs: NKJ weighted samples, $\{\mathbf{x}_{n,j}^{(k)}, W_{n,j}^{(k)}\}_{n=1,k=1,j=1}^{N,K,J}$

two questions: (1) Weighting scheme?[10] (2) Adaptive procedure of $\boldsymbol{\theta}_{n,j}$?[11]

[10] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.
[11] M. F. Bugallo et al. "Adaptive importance sampling: The past, the present, and the future". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 60–79.

Adaptive Importance Sampling: Generic Algorithm

Initialization: Choose J, N, K, $\{q_{n,1}\}_{n=1}^{N}$, and initial parameters $\{\boldsymbol{\theta}_{n,1}\}_{n=1}^{N}$
For $j = 1, \ldots, J$**:**

1. **Sampling:** Simulate NK samples as

$$\mathbf{x}_{n,j}^{(k)} \sim q_{n,j}(\mathbf{x}|\boldsymbol{\theta}_{n,j}), \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

2. **Weighting:** Weight the NK samples as [Elvira19]

$$W_{n,j}^{(k)} = \frac{\pi(\mathbf{x}_{n,j}^{(k)})}{\varphi_{n,j}(\mathbf{x}_{n,j}^{(k)})}, \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

3. **Adaptation of the parameters:** Update the proposal parameters

$$\{\boldsymbol{\theta}_{n,j}\}_{n=1}^{N} \xrightarrow{\text{Adapt}} \{\boldsymbol{\theta}_{n,j+1}\}_{n=1}^{N}$$

Outputs: NKJ weighted samples, $\{\mathbf{x}_{n,j}^{(k)}, W_{n,j}^{(k)}\}_{n=1,k=1,j=1}^{N,K,J}$

two questions: (1) Weighting scheme?[10] (2) Adaptive procedure of $\boldsymbol{\theta}_{n,j}$?[11]

---

[10] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.

[11] M. F. Bugallo et al. "Adaptive importance sampling: The past, the present, and the future". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 60–79.

Adaptive Importance Sampling: Generic Algorithm

Initialization: Choose J, N, K, $\{q_{n,1}\}_{n=1}^{N}$, and initial parameters $\{\boldsymbol{\theta}_{n,1}\}_{n=1}^{N}$
For $j = 1, \ldots, J$**:**

1. **Sampling:** Simulate NK samples as

$$\mathbf{x}_{n,j}^{(k)} \sim q_{n,j}(\mathbf{x}|\boldsymbol{\theta}_{n,j}), \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

2. **Weighting:** Weight the NK samples as [Elvira19]

$$W_{n,j}^{(k)} = \frac{\pi(\mathbf{x}_{n,j}^{(k)})}{\varphi_{n,j}(\mathbf{x}_{n,j}^{(k)})}, \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

3. **Adaptation of the parameters:** Update the proposal parameters

$$\{\boldsymbol{\theta}_{n,j}\}_{n=1}^{N} \xrightarrow{\text{Adapt}} \{\boldsymbol{\theta}_{n,j+1}\}_{n=1}^{N}$$

Outputs: NKJ weighted samples, $\{\mathbf{x}_{n,j}^{(k)}, W_{n,j}^{(k)}\}_{n=1,k=1,j=1}^{N,K,J}$

two questions: (1) Weighting scheme?[10] (2) Adaptive procedure of $\boldsymbol{\theta}_{n,j}$?[11]

[10] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.

[11] M. F. Bugallo et al. "Adaptive importance sampling: The past, the present, and the future". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 60–79.

Adaptive Importance Sampling: Generic Algorithm

Initialization: Choose J, N, K, $\{q_{n,1}\}_{n=1}^{N}$, and initial parameters $\{\boldsymbol{\theta}_{n,1}\}_{n=1}^{N}$
For $j = 1, \ldots, J$**:**

1. **Sampling:** Simulate NK samples as

$$\mathbf{x}_{n,j}^{(k)} \sim q_{n,j}(\mathbf{x}|\boldsymbol{\theta}_{n,j}), \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

2. **Weighting:** Weight the NK samples as [Elvira19]

$$W_{n,j}^{(k)} = \frac{\pi(\mathbf{x}_{n,j}^{(k)})}{\varphi_{n,j}(\mathbf{x}_{n,j}^{(k)})}, \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.$$

3. **Adaptation of the parameters:** Update the proposal parameters

$$\{\boldsymbol{\theta}_{n,j}\}_{n=1}^{N} \xrightarrow{\text{Adapt}} \{\boldsymbol{\theta}_{n,j+1}\}_{n=1}^{N}$$

Outputs: NKJ weighted samples, $\{\mathbf{x}_{n,j}^{(k)}, W_{n,j}^{(k)}\}_{n=1,k=1,j=1}^{N,K,J}$

two questions: (1) Weighting scheme?[10] (2) Adaptive procedure of $\boldsymbol{\theta}_{n,j}$?[11]

[10] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. "Generalized Multiple Importance Sampling". In: *Statistical Science* 34.1 (2019), pp. 129–155.
[11] M. F. Bugallo et al. "Adaptive importance sampling: The past, the present, and the future". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 60–79.

Outline

Importance sampling for sequential inference

- Back to our problem: compute the joint $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ and/or filtering $p(\mathbf{x}_t|\mathbf{y}_{1:t})$:

$$p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})/Z$$

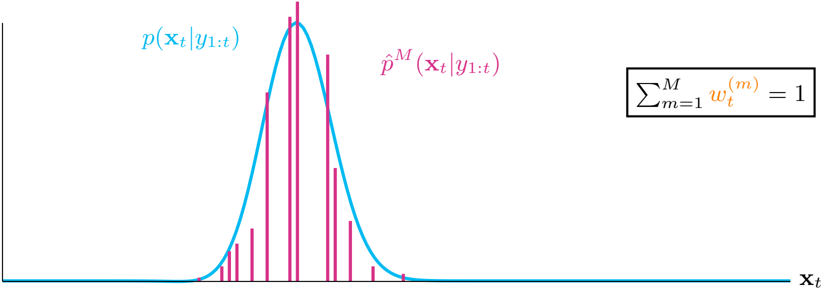
  - Norm. constant: $Z = p(\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) d\mathbf{x}_{1:t}$ cannot be computed
  - Marginalization: $p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{1:t-1}$ cannot be computed
- Importance sampling :
  1. Sample M trajectories $\mathbf{x}_{1:t}^{(m)} \sim q(\mathbf{x}_{1:t})$, $m = 1, ..., M$.
  2. Weight each trajectory $W^{(m)} = \frac{p(\mathbf{x}_{1:t}^{(m)}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t}^{(m)})} \propto \frac{p(\mathbf{x}_{1:t}^{(m)}, \mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t}^{(m)})}$, $m = 1, ..., M$.
- Approximate the joint target with weighted trajectories

$$p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \approx p^M(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = \sum_{m=1}^{M} w^{(m)} \delta_{\mathbf{x}_{1:t}^{(m)}}(\mathbf{x}_{1:t})$$

where $w^{(m)} = \frac{W^{(m)}}{\sum_{j=1}^{M} W^{(m)}}$ are the normalized weights ($\sum_{j=1}^{M} w^{(j)} = 1$).

Particle filtering/sequential Monte Carlo

- Particle filtering/sequential Monte Carlo
- The distributions are approximated by a random measure of M particles and associated normalized weights $\mathcal{X} = \{\mathbf{x}_t^{(m)}, w_t^{(m)}\}_{m=1}^{M}$
  - $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \hat{p}^M(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{m=1}^{M} w_t^{(m)} \delta_{\mathbf{x}_t^{(m)}}(\mathbf{x}_t)$



$\boxed{\sum_{m=1}^{M} w_t^{(m)} = 1}$

Batch importance sampling

- Which proposal $q(\mathbf{x}_{1:t})$ for sampling/simulating the M trajectories $\mathbf{x}_{1:t}^{(n)}$?
  - Easiest choice: $q(\mathbf{x}_{1:t}) = p(\mathbf{x}_{1:t}) \equiv p(\mathbf{x}_1) \cdot p(\mathbf{x}_2|\mathbf{x}_1) \cdot ... \cdot p(\mathbf{x}_t|\mathbf{x}_{t-1})$
    - proposal factorizes (it is a choice)
- Batch procedure of IS with M samples:

for $m = 1$ to M **do**

1. Sample the m-th trajectory **(sampling)**:

$$\begin{aligned} \mathbf{x}_1^{(m)} &\sim p(\mathbf{x}_1^{(m)}) \\ \mathbf{x}_2^{(m)} &\sim p(\mathbf{x}_2|x_1^{(m)}) \\ &\cdots \\ \mathbf{x}_t^{(m)} &\sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}) \end{aligned}$$

$\mathbf{x}_{1:t}^{(m)} = [\mathbf{x}_1^{(m)}, ..., \mathbf{x}_t^{(m)}]$

2. Weight for the m-th trajectory **(weighting)**:

$$W_t^{(m)} = \frac{p(\mathbf{x}_{1:t}^{(m)}, \mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t}^{(m)})} = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}^{(m)})p(\mathbf{x}_{1:t}^{(m)})}{p(\mathbf{x}_{1:t}^{(m)})} = p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}^{(m)})$$

(likelihood evaluated at the m-th trajectory)

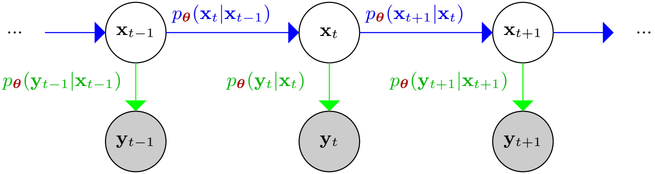
Normalize weights as $w^{(m)} = \frac{W_t^{(m)}}{\sum_{j=1}^{M} W_t^{(j)}}$

end for

$$p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \approx p^M(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = \sum_{m=1}^{M} w^{(m)} \delta_{\mathbf{x}_{1:t}^{(m)}}(\mathbf{x}_{1:t})$$

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx p^M(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{m=1}^{M} w^{(m)} \delta_{\mathbf{x}_t^{(m)}}(\mathbf{x}_{1:t})$$ (Monte Carlo marginalization)

Sequential importance sampling (SIS)



- Due to model structure, joint likelihood factorizes as

$$w_t^{(m)} \propto p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}^{(m)}) = p(\mathbf{y}_1|\mathbf{x}_1^{(m)}) \cdot p(\mathbf{y}_2|\mathbf{x}_2^{(m)}) \cdot \ldots \cdot p(\mathbf{y}_t|\mathbf{x}_t^{(m)})$$

- If we receive $\mathbf{y}_{t+1}$, can we approximate $p(\mathbf{x}_{1:t+1}|\mathbf{y}_{1:t+1})$ without re-processing $\mathbf{y}_{1:t}$?

$$\begin{aligned}
p(\mathbf{x}_{1:t+1}|\mathbf{y}_{1:t+1}) &= \frac{p(\mathbf{y}_{t+1}, \mathbf{x}_{1:t+1}|\mathbf{y}_{1:t})}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})} = \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{1:t+1}, \mathbf{y}_{1:t})}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})} p(\mathbf{x}_{1:t+1}|\mathbf{y}_{1:t}) \quad \text{(Bayes)} \\
&= \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{1:t+1}, \mathbf{y}_{1:t})}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})} p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \quad \text{(factorization)} \\
&= \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{x}_t)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})} p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \quad \text{(model structure)}
\end{aligned}$$

Sequential importance sampling (SIS)

- Sequential importance sampling (SIS):
  - sample and weight sequentially at time t
  - process each observation $\mathbf{y}_t$ without reprocessing $\mathbf{y}_{1:t-1}$

for $t = 2$ to T **do**
  for $m = 1$ to M **do**
    1. Sample the m-th trajectory at time t:

$$\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$$

    $\mathbf{x}_{1:t}^{(m)} = [\mathbf{x}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
    2. Weight for the m-th trajectory:

$$W_t^{(m)} = p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}^{(m)}) = W_{t-1}^{(m)} p(\mathbf{y}_t|\mathbf{x}_t^{(m)})$$

  end for
end for
Normalize weights: $w_T^{(m)} = \frac{W_T^{(m)}}{\sum_{j=1}^{M} W_T^{(j)}}$, $m = 1, ..., M$.
return M trajectories with their weights: $\{\mathbf{x}_{1:T}^{(m)}, w_T^{(m)}\}_{m=1}^{M}$

Quality of the approximation and resampling step

- We can approximate any function f of the desired $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ with the self-normalized estimator:

$$\widetilde{I}_N(f) = \sum_{m=1}^{M} f\left(\mathbf{x}_{1:t}^{(m)}\right) w_t^{(m)}$$

  - Example: mean of the filtering distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is simply the weighted average of the particles at time t.

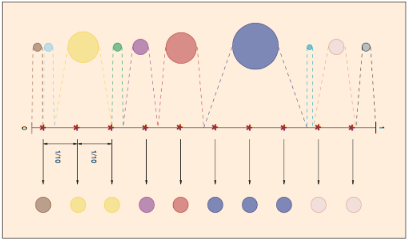
$$\widetilde{I}_N = \sum_{m=1}^{M} \mathbf{x}_t^{(m)} w_t^{(m)}$$

  - The quality of the approximation depends on weights variability:
    - ⋆ if one particle gets a weight ≈ 1 (and the others almost zero), the approximation is very bad.

- **Resampling** step:
  - at each time t we kill bad trajectories with very low weight and replicate good trajectories (before processing future observations)
  - implicit improvement of the marginal proposal for $t+1$

Resampling step

- Resampling step: third step (after 1. Sampling, and 2. Weighting)
  - easy but **necessary** to make the PF work.
- At time t, after computing the weights we replicate good particles and kill bad particles.
  - Urn example: draw i.i.d. M balls (particles) with replacement, with probability equal to the associated weights.
  - More precisely: $\mathcal{X}_t = \{\mathbf{x}_t^{(m)}, w_t^{(m)}\}_{m=1}^{M}$ forms an empirical distribution $p^M(\mathbf{x}_t|\mathbf{y}_{1:t}) \equiv \sum_{j=1}^{M} w_t^{(j)} \delta_{\mathbf{x}_t^{(j)}}(\mathbf{x})$.
    - Sample M particles, $\tilde{\mathbf{x}}_t^{(m)} \sim p^M(\mathbf{x}_t|\mathbf{y}_{1:t})$, $m = 1, \ldots, M$.



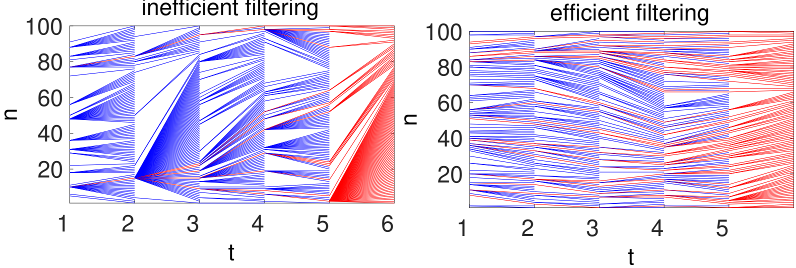
(Source in [12])

[12] P. M. Djuric et al. "Particle filtering". In: *IEEE signal processing magazine* 20.5 (2003), pp. 19–38.

Side effect of the resampling step

- Resampling:
  - reduces particle degeneracy (few samples get most of the probability mass)
  - side effect of introducing path degeneracy (few ancestor particles surviving)
- recall we approximate the joint target with weighted trajectories

$$p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \approx p^M(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = \sum_{m=1}^{M} w^{(m)} \delta_{\mathbf{x}_{1:t}^{(m)}}(\mathbf{x}_{1:t})$$



- Resampling with moderation only when particle degeneracy is severe:
  - Effective sampling size (ESS):[13] $\widehat{\text{ESS}} = \dfrac{1}{\sum_{m=1}^{M} [w_t^{(m)}]^2}$
  - Adaptive resampling: resample only if $\widehat{\text{ESS}} < \gamma$, with $1 \le \gamma \le M$
    - $\gamma = M$: resample always (equiv. BPF)
    - $\gamma = 1$: never resample (equiv. batch IS)

[13] V. Elvira, L. Martino, and C. P. Robert. "Rethinking the effective sample size". In: *International Statistical Review* 90.3 (2022), pp. 525–550.

The bootstrap PF (BPF)

- Bootstrap PF $\equiv$ Sequential importance sampling resampling (SISR)[14]

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

(ii) Recursive step. At time t,

1. **Prediction** (particles propagation): $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$
2. **Update** (weights calculation): compute the normalized weights as $w_t^{(m)} \propto p(\mathbf{y}_t|\mathbf{x}_t^{(m)})$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \text{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_{1:t}^{(m)} = \mathbf{x}_{1:t}^{(i^{(m)})}$ $m = 1, \ldots, M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

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

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

[14] N. J. Gordon, D. J. Salmond, and A. F. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE proceedings F (radar and signal processing)*. Vol. 140. 2. IET. 1993, pp. 107–113.

The bootstrap PF (BPF)

- Bootstrap PF $\equiv$ Sequential importance sampling resampling (SISR)[14]

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

(ii) Recursive step. At time t,

1. **Prediction** (particles propagation): $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t | \widetilde{\mathbf{x}}_{t-1}^{(m)})$
2. **Update** (weights calculation): compute the normalized weights as $w_t^{(m)} \propto p(\mathbf{y}_t | \mathbf{x}_t^{(m)})$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \text{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_{1:t}^{(m)} = \mathbf{x}_{1:t}^{(i^{(m)})}$ $m = 1, \ldots, M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

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

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

[14] N. J. Gordon, D. J. Salmond, and A. F. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE proceedings F (radar and signal processing)*. Vol. 140. 2. IET. 1993, pp. 107–113.

The bootstrap PF (BPF)

- Bootstrap PF ≡ Sequential importance sampling resampling (SISR)[14]

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

(ii) Recursive step. At time t,

1. **Prediction** (particles propagation): $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$
2. **Update** (weights calculation): compute the normalized weights as $w_t^{(m)} \propto p(\mathbf{y}_t|\mathbf{x}_t^{(m)})$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \text{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_{1:t}^{(m)} = \mathbf{x}_{1:t}^{(i^{(m)})}$ $m = 1, \ldots, M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

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

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

[14] N. J. Gordon, D. J. Salmond, and A. F. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE proceedings F (radar and signal processing)*. Vol. 140. 2. IET. 1993, pp. 107–113.

The bootstrap PF (BPF)

- Bootstrap PF $\equiv$ Sequential importance sampling resampling (SISR)[14]

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

(ii) Recursive step. At time t,

1. **Prediction** (particles propagation): $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$
2. **Update** (weights calculation): compute the normalized weights as $w_t^{(m)} \propto p(\mathbf{y}_t|\mathbf{x}_t^{(m)})$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \text{Cat}([1, ..., M]; [w_t^{(1)}, ..., w_t^{(M)}])$, $m = 1, ..., M$
   b) set $\widetilde{\mathbf{x}}_{1:t}^{(m)} = \mathbf{x}_{1:t}^{(i^{(m)})}$ $m = 1, ..., M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

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

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

[14] N. J. Gordon, D. J. Salmond, and A. F. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE proceedings F (radar and signal processing)*. Vol. 140. 2. IET. 1993, pp. 107–113.

The bootstrap PF (BPF)

- Bootstrap PF $\equiv$ Sequential importance sampling resampling (SISR)[14]

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

(ii) Recursive step. At time t,

1. **Prediction** (particles propagation): $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$
2. **Update** (weights calculation): compute the normalized weights as $w_t^{(m)} \propto p(\mathbf{y}_t|\mathbf{x}_t^{(m)})$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \text{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_{1:t}^{(m)} = \mathbf{x}_{1:t}^{(i^{(m)})}$ $m = 1, \ldots, M$

equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

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

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

---

[14] N. J. Gordon, D. J. Salmond, and A. F. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE proceedings F (radar and signal processing)*. Vol. 140. 2. IET. 1993, pp. 107–113.

The bootstrap PF (BPF)

- Bootstrap PF $\equiv$ Sequential importance sampling resampling (SISR)[14]

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

(ii) Recursive step. At time t,

1. **Prediction** (particles propagation): $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t | \widetilde{\mathbf{x}}_{t-1}^{(m)})$
2. **Update** (weights calculation): compute the normalized weights as $w_t^{(m)} \propto p(\mathbf{y}_t | \mathbf{x}_t^{(m)})$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \mathrm{Cat}([1, \dots, M]; [w_t^{(1)}, \dots, w_t^{(M)}])$, $m = 1, \dots, M$
   b) set $\widetilde{\mathbf{x}}_{1:t}^{(m)} = \mathbf{x}_{1:t}^{(i^{(m)})}$ $m = 1, \dots, M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

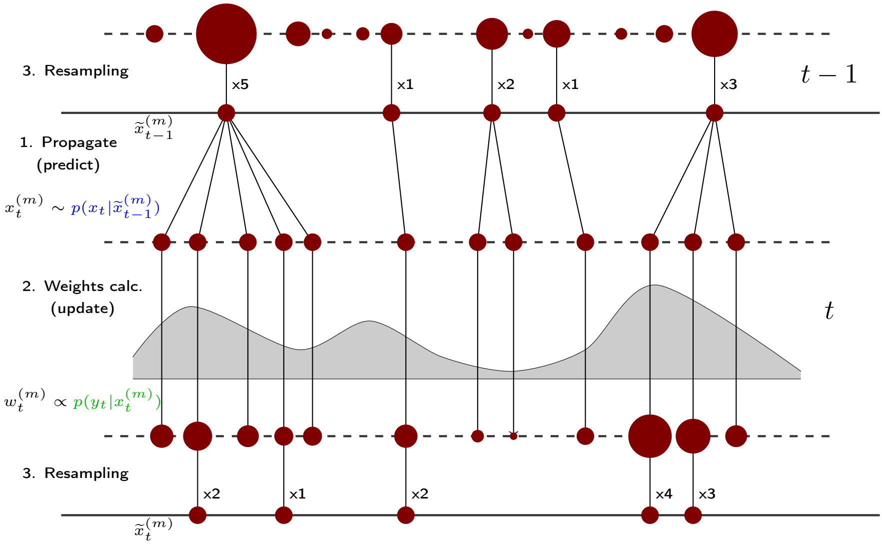
$$\widetilde{\mathbf{x}}_t^{(m)} \sim p^M(\mathbf{x}_t | \mathbf{y}_{1:t}) \equiv \sum_{j=1}^{M} w_t^{(j)} \delta_{\mathbf{x}_t^{(j)}}(\mathbf{x})$$

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

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

[14] N. J. Gordon, D. J. Salmond, and A. F. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE proceedings F (radar and signal processing)*. Vol. 140. 2. IET. 1993, pp. 107–113.

Bootstrap particle filter



Outline

Mini-project: BF for the Lorenz 63 model

- Take the same state-space model as for the Kalman filtering problem (stochastic Lorenz 63, with Euler discretisation and nonlinear observations) and code a standard bootstrap filter with N particles.
- Compare the performance of the bootstrap filter and the non-linear extensions of KF (e.g., CKF): try different values of state noise variance and observational noise variance, gap between observations, and M (number of particles).

Beyond BPF

- The m-th proposal in BPF is the transition kernel $q(\mathbf{x}_t) = p(\mathbf{x}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$
  - note that is conditioned on $\widetilde{\mathbf{x}}_{t-1}^{(m)}$ which comes from a resampling step.
  - proposal of all samples can be interpreted as approximate predictive, since resampling $(t-1)$ + propagation at t = mixture sampling:

$$\mathbf{x}_t^{(m)} \overset{i.i.d.}{\sim} p^M(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \sum_{m=1}^{M} w_t^{(m)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$$

- From IS theory, high variance of the importance weights $\Rightarrow$ low efficiency/accuracy of the filter
- **Intuition** (imprecise): inefficiency in BPF will happen when predictive $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$, thus mixture proposal, differs from filtering distribution:

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

  - equivalent to say that $\mathbf{y}_t$ is **informative**, which happens when:
    - $p(\mathbf{y}_t|\mathbf{x}_t)$ is "peaky" compared to $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ (e.g., very low-variance observation noise)
    - $p(\mathbf{y}_t|\mathbf{x}_t)$ is placed in a "different" area of the space compared to $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ (e.g., outlier observation)
  - in those scenarios, great variability of the weights.

An adapted PF with generic proposal

- A good proposal must include knowledge about $\mathbf{y}_t$
  - avoid particles being sampled into regions of the state space which are unlikely in light of that observation
- Generic sampling:

$$\mathbf{x}_{1:t}^{(m)} \sim q(\mathbf{x}_{1:t}) = \prod_{k=1}^{t} q(\mathbf{x}_k; \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k})$$

  with trajectory weight:

$$W_t^{(m)} = \frac{p(\mathbf{x}_{1:t}^{(m)}, \mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t}^{(m)})} = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}^{(m)})p(\mathbf{x}_{1:t}^{(m)})}{q(\mathbf{x}_{1:t}^{(m)})} = \frac{\prod_{k=1}^{t} p(\mathbf{y}_k|\mathbf{x}_k^{(m)})p(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})}{\prod_{k=1}^{t} q(\mathbf{x}_k^{(m)}; \mathbf{x}_{1:k-1}^{(m)}, \mathbf{y}_{1:k})}$$

- possible proposal choices:
  - BPF (online-oriented) uninformative:

$$\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t; \mathbf{x}_{1:t-1}^{(m)}, \mathbf{y}_{1:t}) = q(\mathbf{x}_t; \mathbf{x}_{t-1}^{(m)}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$$

  - still convenient (online-oriented) but more informative:

$$\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t; \mathbf{x}_{1:t-1}^{(m)}, \mathbf{y}_{1:t}) = q(\mathbf{x}_t; \mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t)$$

An adapted PF with generic proposal

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

(ii) Recursive step. At time t,

1. **Sampling/propagation**: $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t; \widetilde{\mathbf{x}}_{t-1}^{(m)}, \mathbf{y}_t)$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
2. **Weights calculation**: compute the normalized weights as $w_t^{(m)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\widetilde{\mathbf{x}}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)};\widetilde{\mathbf{x}}_{t-1}^{(m)},\mathbf{y}_t)}$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \text{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_t^{(m)} = \mathbf{x}_t^{(i^{(m)})}$, $m = 1, \ldots, M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

An adapted PF with generic proposal

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

(ii) Recursive step. At time t,

1. **Sampling/propagation**: $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t; \widetilde{\mathbf{x}}_{t-1}^{(m)}, \mathbf{y}_t)$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
2. **Weights calculation**: compute the normalized weights as $w_t^{(m)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\widetilde{\mathbf{x}}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)};\widetilde{\mathbf{x}}_{t-1}^{(m)},\mathbf{y}_t)}$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \text{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_t^{(m)} = \mathbf{x}_t^{(i^{(m)})}$, $m = 1, \ldots, M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

An adapted PF with generic proposal

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

(ii) Recursive step. At time t,

1. **Sampling/propagation**: $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t; \widetilde{\mathbf{x}}_{t-1}^{(m)}, \mathbf{y}_t)$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
2. **Weights calculation**: compute the normalized weights as $w_t^{(m)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\widetilde{\mathbf{x}}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)};\widetilde{\mathbf{x}}_{t-1}^{(m)},\mathbf{y}_t)}$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \mathrm{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_t^{(m)} = \mathbf{x}_t^{(i^{(m)})}$, $m = 1, \ldots, M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

An adapted PF with generic proposal

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

(ii) Recursive step. At time t,

1. **Sampling/propagation**: $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t; \widetilde{\mathbf{x}}_{t-1}^{(m)}, \mathbf{y}_t)$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
2. **Weights calculation**: compute the normalized weights as $w_t^{(m)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\widetilde{\mathbf{x}}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)};\widetilde{\mathbf{x}}_{t-1}^{(m)},\mathbf{y}_t)}$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \text{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_t^{(m)} = \mathbf{x}_t^{(i^{(m)})}$, $m = 1, \ldots, M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

An adapted PF with generic proposal

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

(ii) Recursive step. At time t,

1. **Sampling/propagation**: $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t; \widetilde{\mathbf{x}}_{t-1}^{(m)}, \mathbf{y}_t)$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
2. **Weights calculation**: compute the normalized weights as $w_t^{(m)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\widetilde{\mathbf{x}}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)};\widetilde{\mathbf{x}}_{t-1}^{(m)},\mathbf{y}_t)}$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \mathrm{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_t^{(m)} = \mathbf{x}_t^{(i^{(m)})}$, $m = 1, \ldots, M$

equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

An adapted PF with generic proposal

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

(ii) Recursive step. At time t,

1. **Sampling/propagation**: $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t; \widetilde{\mathbf{x}}_{t-1}^{(m)}, \mathbf{y}_t)$ associated to trajectory $\mathbf{x}_{1:t}^{(m)} = [\widetilde{\mathbf{x}}_{1:t-1}^{(m)}, \mathbf{x}_t^{(m)}]$
2. **Weights calculation**: compute the normalized weights as $w_t^{(m)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\widetilde{\mathbf{x}}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)};\widetilde{\mathbf{x}}_{t-1}^{(m)},\mathbf{y}_t)}$
3. Multinomial **resampling**
   a) simulate $i^{(m)} \sim \text{Cat}([1, \ldots, M]; [w_t^{(1)}, \ldots, w_t^{(M)}])$, $m = 1, \ldots, M$
   b) set $\widetilde{\mathbf{x}}_t^{(m)} = \mathbf{x}_t^{(i^{(m)})}$, $m = 1, \ldots, M$

   equivalent to simulate M i.i.d. samples from the approx. filtering dist.

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

In search of better filters

- **Proposal choice** to minimize variance of

$$w_t^{(m)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\widetilde{\mathbf{x}}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)};\widetilde{\mathbf{x}}_{t-1}^{(m)},\mathbf{y}_t)}$$

  - "optimal" kernel: $q(\mathbf{x}_t;\widetilde{\mathbf{x}}_{t-1}^{(m)},\mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{y}_t,\widetilde{\mathbf{x}}_{t-1}^{(m)})$
    - proportional to te numerator $p(\mathbf{y}_t,\mathbf{x}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$
    - intractable kernel
    - reduces the variance of each weight (constant) but still weights are different across them:

$$W_t^{(m)} = p(\mathbf{y}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$$

- Even if available, the kernel does not solve all the problems: the resampling remains blind to the new observation $\mathbf{y}_t$
  - **Goal**: we would like to modify the resampling weights to replicate trajectories at $t-1$ that will perform better in t

In search of better filters: trajectory perspective

- Recap: BPF and adapted PF proceed in the following order:
  1. Resampling $(t-1)$: resample trajectories $\widetilde{\mathbf{x}}_{1:t-1}^{(n)}$ ($\mathbf{y}_t$ is not used in adapted PF nor in BPF)
     - recall: equivalent to simulating at $t-1$ as

$$\widetilde{\mathbf{x}}_{1:t-1}^{(m)} \sim p^M(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}) \equiv \sum_{j=1}^{M} w_{t-1}^{(j)} \delta_{\mathbf{x}_{1:t-1}^{(j)}}(\mathbf{x}_{1:t-1})$$

  2. Sampling (t): propagate from $\widetilde{\mathbf{x}}_{t-1}^{(m)}$ to $\mathbf{x}_t^{(m)}$ ($\mathbf{y}_t$ is used in adapted PF but not used in BPF)
  3. Weighting (t): Bayesian update ($\mathbf{y}_t$ is used)
- Is it possible to use $\mathbf{y}_t$ at resampling?

In search of better filters: trajectory perspective

- More precisely, resample trajectories from $p^M(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t})$ instead of $p^M(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1})$?

$$
\begin{aligned}
p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t}) &= \int p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})d\mathbf{x}_t \\
&= \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_t \\
&= p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}) \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})d\mathbf{x}_t \\
&\approx \left[\sum_{j=1}^{M} w_{t-1}^{(j)}\delta_{\mathbf{x}_{1:t-1}^{(j)}}(\mathbf{x}_{1:t-1})\right] \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})d\mathbf{x}_t \\
&= \sum_{j=1}^{M} w_{t-1}^{(j)}\delta_{\mathbf{x}_{1:t-1}^{(j)}}(\mathbf{x}_{1:t-1}) \underbrace{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)})d\mathbf{x}_t}_{v_t^{(j)}=p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(j)})} \\
&= \sum_{j=1}^{M} \underbrace{w_{t-1}^{(j)}v_t^{(j)}}_{\lambda_t^{(j)}} \delta_{\mathbf{x}_{1:t-1}^{(j)}}(\mathbf{x}_{1:t-1})
\end{aligned}
$$

- $v_t^{(j)}$ is intractable $\Rightarrow$ cheap approximation: $v_t^{(j)} \approx p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(j)})$
  - justification $= p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)}) \approx \delta_{\bar{\mathbf{x}}_t^{(j)}}(\mathbf{x}_t)$, where $\bar{\mathbf{x}}_t^{(j)} = \mathrm{E}_{p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)})}[\mathbf{x}_t]$

In search of better filters: trajectory perspective

New approach:

1. Resampling $(t-1)$: resample trajectories $\widetilde{\mathbf{x}}_{1:t-1}^{(m)}$ ($\mathbf{y}_t$ is now used)
2. Sampling (t): simulate from $q(\mathbf{x}_t; \widetilde{\mathbf{x}}_{t-1}, \mathbf{y}_t)$ ($\mathbf{y}_t$ potentially used)
   - trajectory proposal is now

$$q(\mathbf{x}_t; \mathbf{x}_{1:t-1}) = q(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{y}_t) \sum_{j=1}^{M} \lambda_t^{(j)} \delta_{\mathbf{x}_{1:t-1}^{(j)}}(\mathbf{x}_{1:t-1})$$

   and sampling as

   a) simulate $i^{(m)} \sim \text{Cat}([1, \dots, M]; [\lambda_t^{(1)}, \dots, \lambda_t^{(M)}])$, $m = 1, \dots, M$
   b) simulate $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t; \mathbf{x}_{t-1}^{(i^{(m)})}, \mathbf{y}_t)$, $m = 1, \dots, M$

3. Weighting (t) in the joint space, using $\mathbf{x}_{1:t-1}^{(j)}$ fulfills

$$q(\mathbf{x}_{1:t-1}^{(j)}; \mathbf{y}_{1:t}) = \frac{\lambda_t^{(j)}}{w_t^{(j)}} p^M(\mathbf{x}_{1:t-1}^{(j)} | \mathbf{y}_{1:t-1}),$$

$$w_t(\mathbf{x}_{1:t}^{(m)}) = \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) p(\mathbf{x}_t^m | \mathbf{x}_{t-1}^{(i^{(m)})}) p(\mathbf{x}_{1:t-1}^{(i^{(m)})} | \mathbf{y}_{1:t-1})}{q(\mathbf{x}_t^{(m)}; \mathbf{x}_{1:t-1}^{(i^{(m)})}, \mathbf{y}_t) q(\mathbf{x}_{1:t-1}^{(i^{(m)})}; \mathbf{y}_{1:t})}$$

$$= \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) p(\mathbf{x}_t^m | \mathbf{x}_{t-1}^{(i^{(m)})}) p(\mathbf{x}_{1:t-1}^{(i^{(m)})} | \mathbf{y}_{1:t-1})}{q(\mathbf{x}_t^{(m)}; \mathbf{x}_{1:t-1}^{(i^{(m)})}, \mathbf{y}_t) \frac{\lambda_t^{(i^{(m)})}}{w_t^{(i^{(m)})}} p^M(\mathbf{x}_{1:t-1}^{(i^{(m)})} | \mathbf{y}_{1:t-1})}$$

$$\approx \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) p(\mathbf{x}_t^m | \mathbf{x}_{t-1}^{(i^{(m)})})}{q(\mathbf{x}_t^{(m)}; \mathbf{x}_{1:t-1}^{(i^{(m)})}, \mathbf{y}_t) p(\mathbf{y}_t | \mu_t^{(i^{(m)})})}$$

Auxiliary PF (APF)

- Proposed in [15] as an alternative to BPF
  - APF improves sometimes the performance of BPF, but **not always.**
  - it attempts to sample in better areas in light of the new observation $\mathbf{y}_t$

(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

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

[15] M. K. Pitt and N. Shephard. "Filtering via simulation: Auxiliary particle filters". In: *Journal of the American statistical association* 94.446 (1999), pp. 590–599.

Outline

Mini-project

- Assume, again, the stochastic Lorenz 63 model, time-discretised using an Euler-Maruyama scheme:

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

  where $Z_{i,n} \sim \mathcal{N}(0,1)$, the state is $X_n = [X_{1,n}, X_{2,n}, X_{3,n}]^\top$ and the parameters are $(s,r,b) = \left(10, 28, \frac{8}{3}\right)$. You may 'play around' with the value of σ as in the previous mini-projects (start with $\sigma = \frac{1}{2}$).
- Assume linear observations, namely $Y_n = X_{1,n} + \sigma_u U_n$, where $U_n \sim \mathcal{N}(0,1)$, every B discrete time steps (e.g., $B = 40$). Again, you may try different values of the parameter σ_u (you may start with $\sigma_u = 2$).

Mini-project

- For the state space model in the previous slide, code a standard SIR algorithm and a SIR algorithm with a Gaussian proposal.
  Try different configurations of the model noise parameters (σ_u, σ, M) and plot the ground-truth signals and their estimates with both filters. Reference values:
  $\sigma_u = 2, \sigma = 1, B = 40$ and $M = 50$ particles in the SIR algorithm.
  The simulation should also return estimation errors (e.g., average square errors over time) for the two filters.

Outline

Revisiting standard filters from the MIS perspective

- Both BPF and APF (and other filters) use several proposals at each t
- e.g., BPF proposal is $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$
  - potentially M different values $\widetilde{\mathbf{x}}_{t-1}^{(m)}$, i.e., M proposals.
  - at least two views:
    1. each sample $\mathbf{x}_t^{(m)}$ is simulated from $p(\mathbf{x}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$
    2. all samples are i.i.d. samples from the mixture

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

- Is it possible to re-interpret BPF and other filters from a MIS perspective?

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

$$\begin{aligned} w_t^{(m)} &\propto \frac{p(\mathbf{x}_t^{(m)} | \mathbf{y}_{1:t})}{\psi_t(\mathbf{x}_t^{(m)})} \propto \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) p(\mathbf{x}_t^{(m)} | \mathbf{y}_{1:t-1})}{\psi_t(\mathbf{x}_t^{(m)})} \\ &\approx \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(j)})}{\psi_t(\mathbf{x}_t^{(m)})} = \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M} \lambda_t^{(j)} p(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(j)})} \end{aligned} \quad (2)$$

- Two questions:[16]
  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 $p(\mathbf{x}_t | \mathbf{y}_{1:t})$ $\Rightarrow$ AIS
  2. Approximate $w_t^{(m)}$ in (2) to derive BPF, APF. and other/new filters?

[16] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric. "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]". 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

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

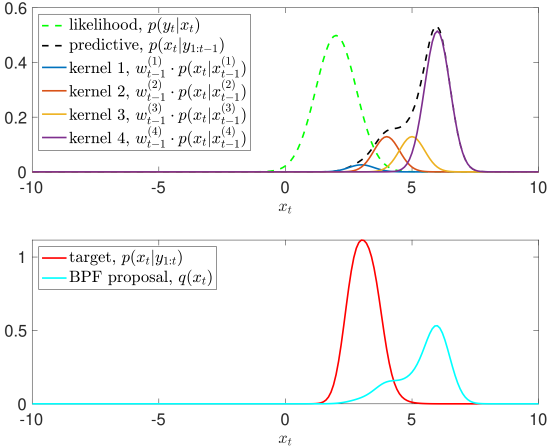
$$\approx \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\psi_t(\mathbf{x}_t^{(m)})} = \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M} w_{t-1}^{(j)}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

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

- Remark: the BPF proposal matches just the prior of the numerator.[17]

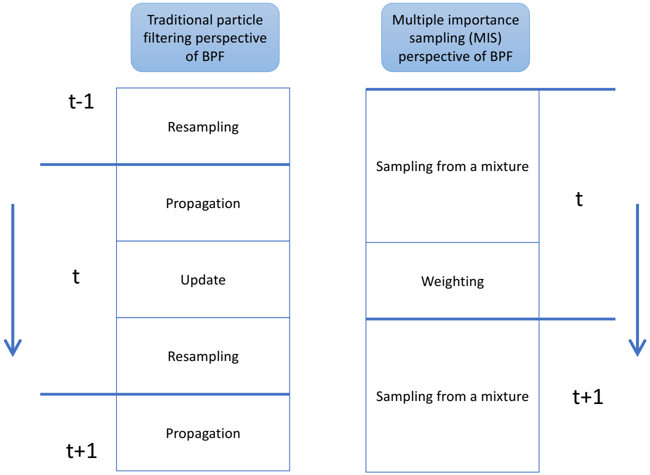
[17] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric. "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]". In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 145–152.

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

BPF from the MIS perspective



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 mixture kernel is amplified by the likelihood eval. 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

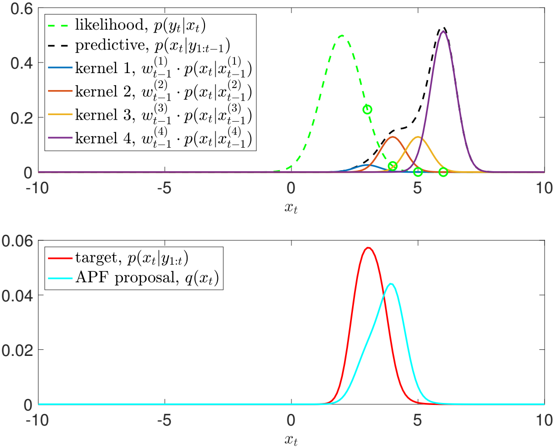
$$w_t^{(m)} \propto \frac{p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t})}{\psi_t(\mathbf{x}_t^{(m)})} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t-1})}{\sum_{j=1}^{M} \lambda_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})} \approx \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})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

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

- Remark:[18]
  - 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).

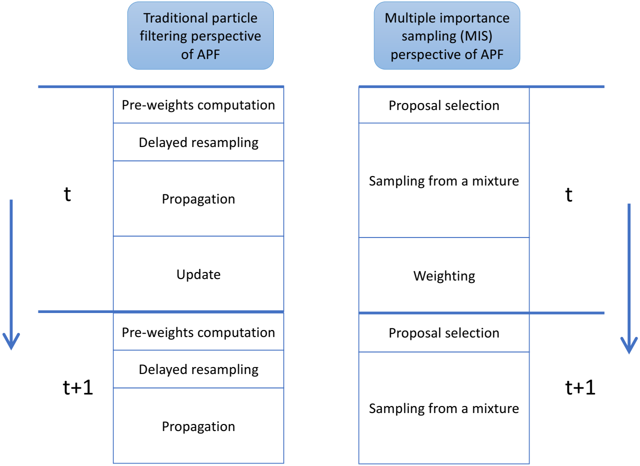
[18] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric. "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]". In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 145–152.

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:[19] Based on this MIS interpretation, we improve the APF
  - MIS perspective: 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**: the "amplification" $\lambda_t^{(j)}$ of j-th kernel, takes into account where all other kernels are placed (unlike APF)
    - 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)
  - Connection to marginal PFs [20]

[19] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djurić. "In search for improved auxiliary particle filters". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE. 2018, pp. 1637–1641.

[20] M. Klaas, N. De Freitas, and A. Doucet. "Toward practical N2 Monte Carlo: The marginal particle filter". In: *arXiv preprint arXiv:1207.1396* (2012).

Optimized APF (OAPF)

- OAPF:[21][22] $K < M$ kernels form the mixture proposal
  - Optimized λ via non-negative least squares (NNLS) by taking the squared distance between target and mixture proposal at the E evaluation points

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \|\mathbf{Q}\boldsymbol{\lambda} - \widetilde{\boldsymbol{\pi}}\|_2^2 \quad \text{subject to} : \boldsymbol{\lambda} \in \mathbb{R}^K_{\geq 0}.$$

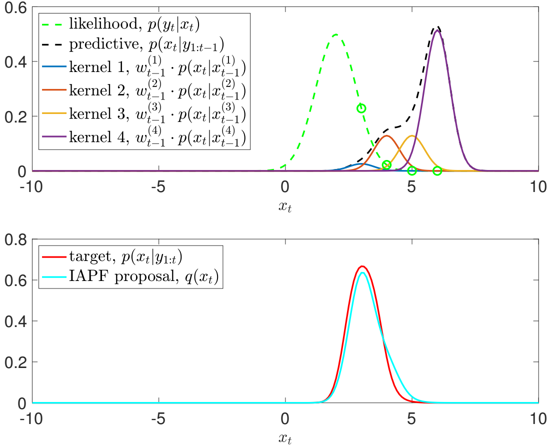
- In IAPF and OAPF, IS weights are *exact*:

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

[21] N. Branchini and V. Elvira. "Optimized auxiliary particle filters". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 1289–1299.

[22] N. Branchini and V. Elvira. "An adaptive mixture view of particle filters". In: *Foundations of Data Science* (2024), pp. 0–0.

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.**[23] Select the MIS proposal of the form

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

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

$$w_t^{(m)} = ?$$

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

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

[23] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric. "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]". In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 145–152.

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.**[23] Select the MIS proposal of the form

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

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

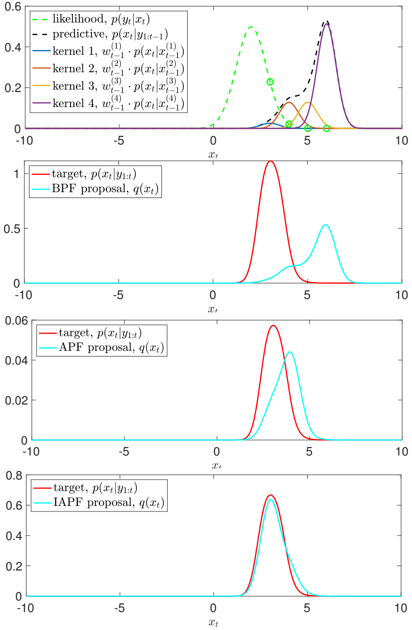
$$w_t^{(m)} = ?$$

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

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

[23] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric. "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]". In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 145–152.

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**

Outline

Filtering in high-dimension spaces

- All methods, and Monte Carlo is not an exception, suffer from the course of dimensionality (in this case, when d_x is high)
- In some models, some of the hidden variables can be integrated (Rao-blackwellized PFs):[24]
  - implicit dimensionality reduction in the latent space
  - lower variance of PF estimators, compared to working in the original space
- Another approach is to partition the space and run several filters in parallel (Multiple PFs):[25]
  - implicit dimensionality reduction in the latent space
  - works well when the dimensions of each subset only interact within each subset
  - more research is needed

[24] K. Murphy and S. Russell. "Rao-Blackwellised particle filtering for dynamic Bayesian networks". In: *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 499–515.

[25] P. M. Djuric, T. Lu, and M. F. Bugallo. "Multiple particle filtering". In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*. Vol. 3. IEEE. 2007, pp. III–1181.

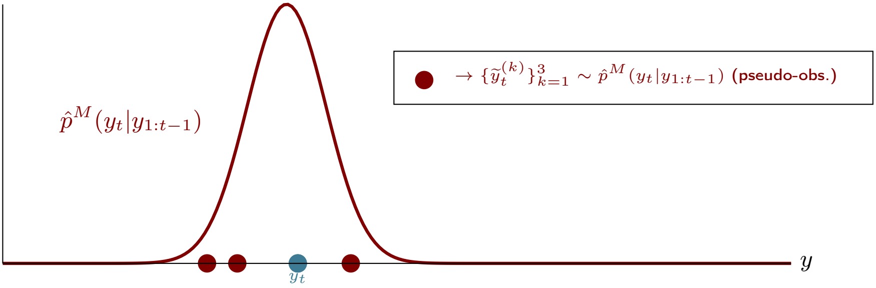
Convergence assessment and adapting N in PF

- **Goal**: in **real time** and for **any SSM**:
  1. **evaluate the convergence**, related to the quality of the approximation and
  2. **adapt the number of particles**[26],
  3. with theoretical guarantees[27]
- **Intuition**: check whether the received observations "make sense" with the approximated predictive distributions
- **Challenge**:
  - at each time step just one observation y_t available
  - the predictive $\hat{p}^M(y_t|y_{1:t-1})$ is evolving with time
- **Proposed method**: At each time step t
  - Generate K fictitious observations $\widetilde{y}_t^{(k)}$ from $\hat{p}^M(y_t|y_{1:t-1})$
    1. $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\widetilde{\mathbf{x}}_{t-1}^{(m)})$ (prediction step of BPF, **for free**)
    2. $\widetilde{y}_t^{(k)} \sim \frac{1}{M}\sum_{m=1}^{M} p(y_t|\mathbf{x}_t^{(m)})$, $k = 1, ..., K$ (**cheap** $K << M$)
  - Compare them with the actual observation y_t.
    - Implicitly, we compare $\hat{p}^M(y_t|y_{1:t-1})$ and $p(y_t|y_{1:t-1})$

[26] V. Elvira, J. Míguez, and P. M. Djurić. "Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment". In: *IEEE Transactions on Signal Processing* 65.7 (2016), pp. 1781–1794.

[27] V. Elvira, J. Miguez, and P. M. Djurić. "On the performance of particle filters with adaptive number of particles". In: *Statistics and Computing* 31 (2021), pp. 1–18.

Ordering observation and fictitious observations

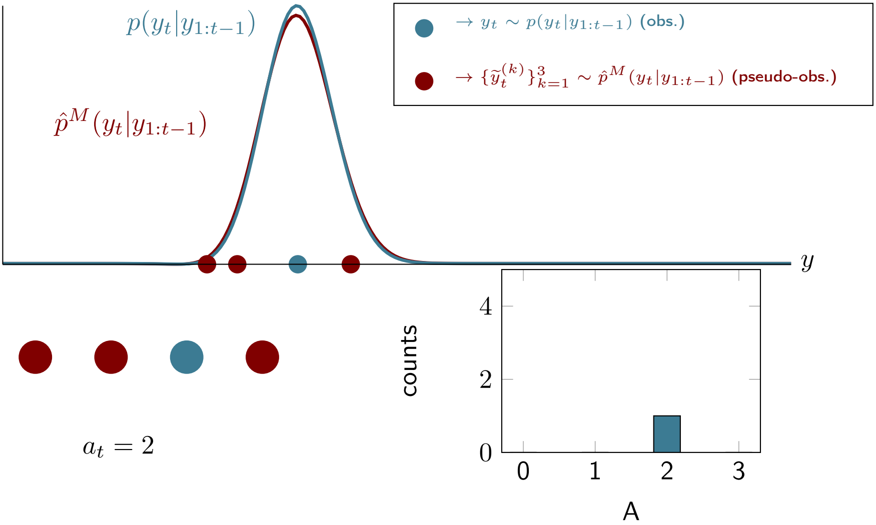


- A_t: number of fictitious observations, $\{\widetilde{y}_t^{(k)}\}_{k=1}^3$, smaller than y_t

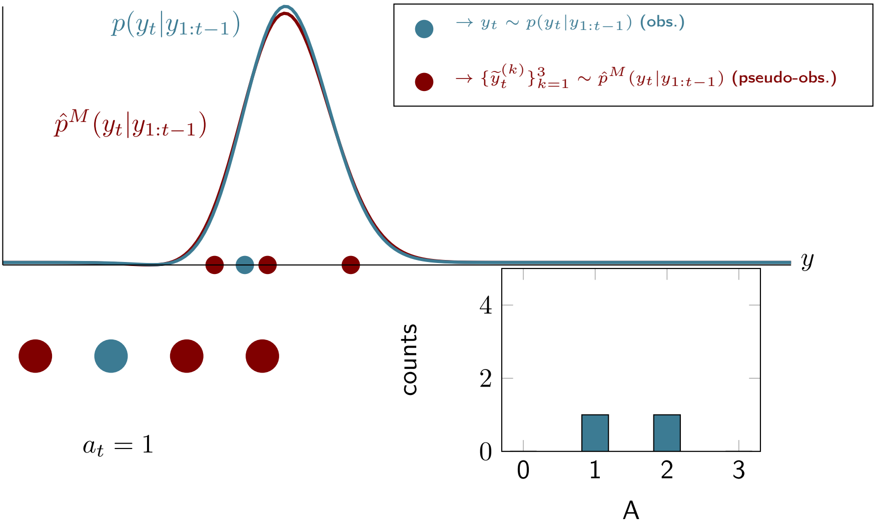
$a_t = 2$

- We can iteratively compute a_t

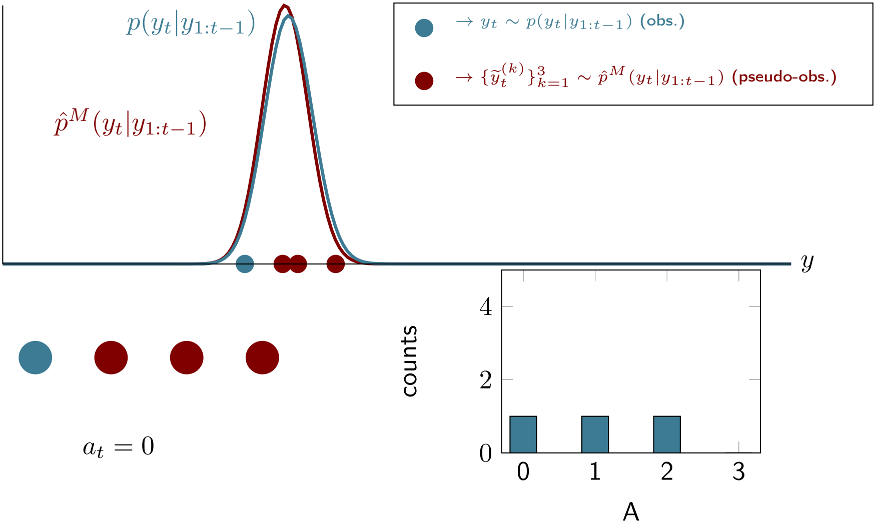
Good approximation, $t = 1$



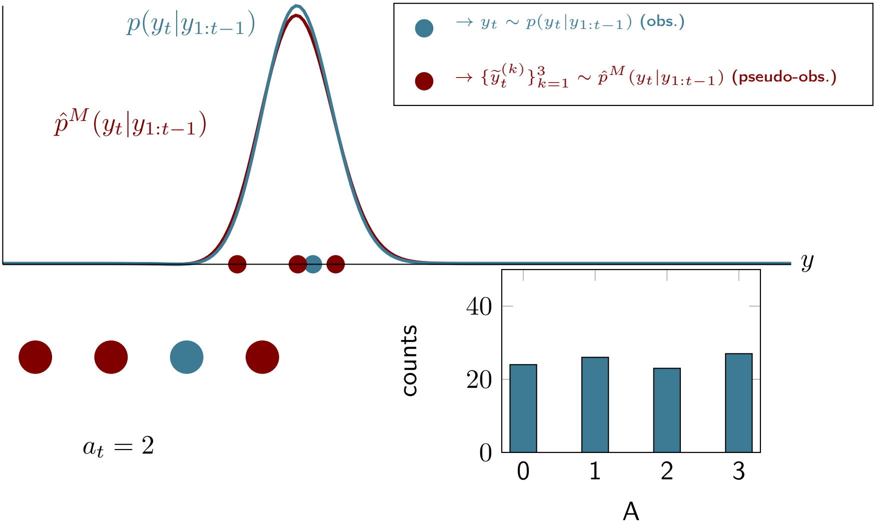
Good approximation, $t = 2$



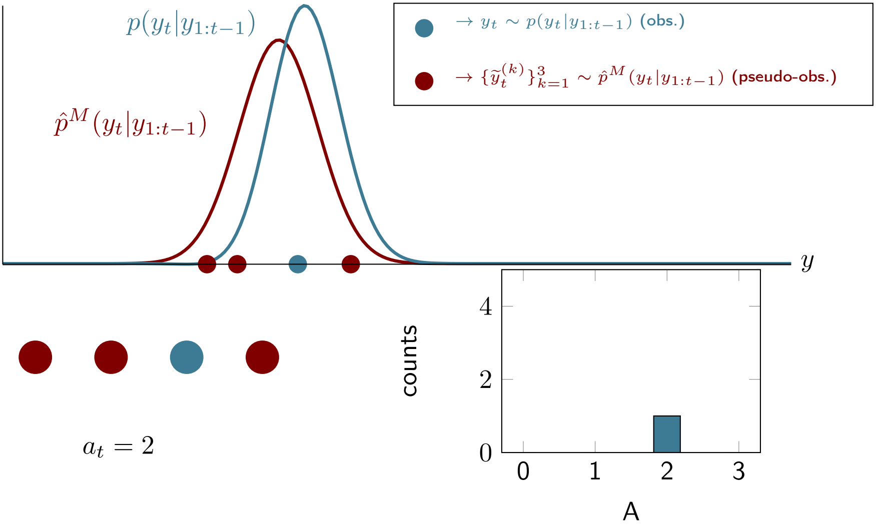
Good approximation, $t = 3$



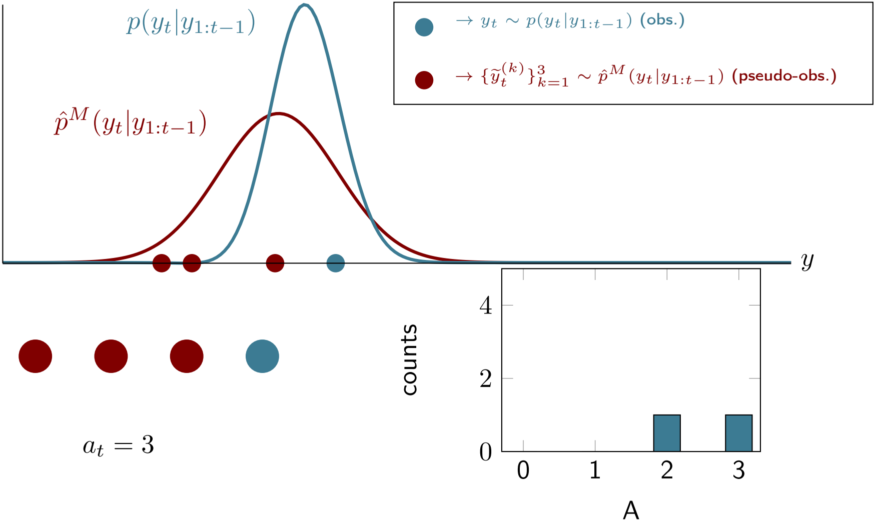
Good approximation, $t = 100$



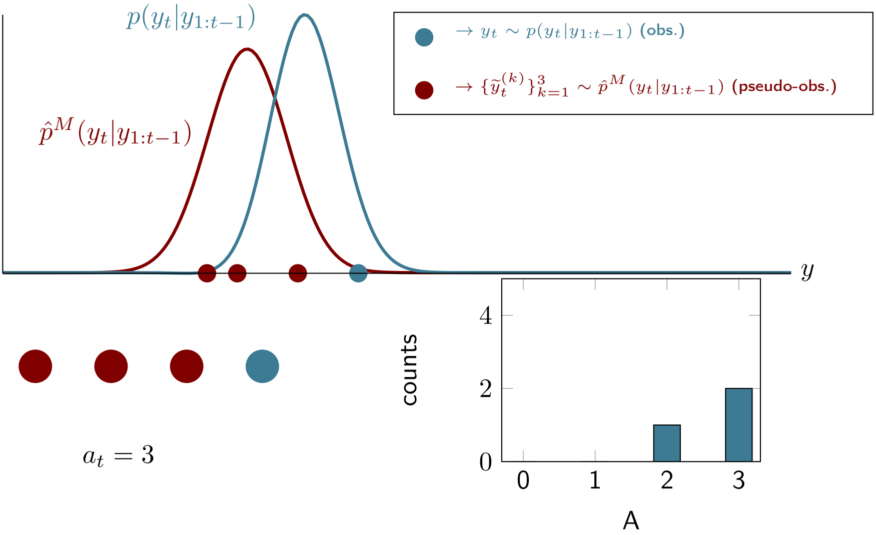
Bad approximation, $t = 1$



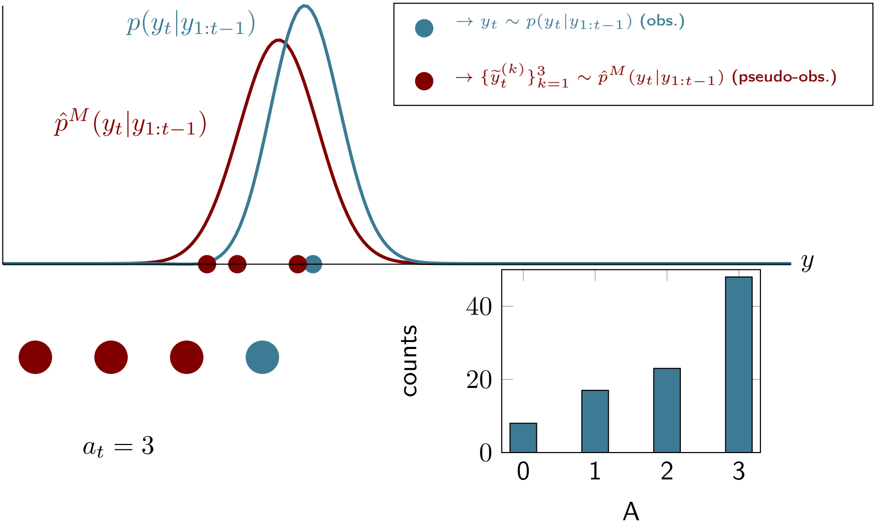
Bad approximation, $t = 2$



Bad approximation, $t = 3$



Bad approximation, $t = 100$



Methodology summary and properties

- **Methodology**: At each time step:
  - simulate $\tilde{y}_t^{(k)} \sim \hat{p}^M(y_t|y_{1:t-1})$, $\qquad k = 1, ..., K$
  - build the r.v. $A_{K,t} := |\mathcal{A}_{K,t}| \in \{0, 1, ..., K\}$, where $\mathcal{A}_{K,t} := \{y \in \{\tilde{y}_t^{(k)}\}_{k=1}^K : y < y_t\}$
- **Properties**: Under the hypothesis of **perfect approximation**:
  - $\mathcal{J}_t := \{y_t, \tilde{y}_t^{(1)}, \ldots, \tilde{y}_t^{(K)}\}$ is a set of i.i.d. samples from a **common** continuous probability distribution $p_t(y_t)$, then:

Proposition 1: *the pmf of the r.v. $A_{K,t}$ is **uniform**:*

$$\mathbb{Q}_K(n) = \frac{1}{K+1}, \qquad n = 0, ..., K.$$

Proposition 2: *the r.v.'s A_{K,t_1} and A_{K,t_2} are **independent**, $\forall t_1, t_2 \in \mathbb{N}$ with $t_1 \neq t_2$.*

- **Invariant wrt the state space model!**

Theoretical results

- **Theoretical analysis**:
  - **convergence** of the predictive pdf of the observations:[28]

$$\lim_{M\to\infty}\left(f,\hat{p}^M(y_t|y_{1:t-1})\right)=\left(f,p(y_t|y_{1:t-1})\right) \qquad \text{a.s.},$$

    with explicit **convergence rate**
    - extends the existing results of pointwise convergence of $\hat{p}^M(y_t|y_{1:t-1})$ to $\hat{p}(y_t|y_{1:t-1})$
    - holds for **multidimensional** observations
    - key for the statistical analysis of $A_{K,t}$
  - **convergence** of the p.m.f. of $A_{K,t}$ to a discrete uniform distribution

$$\frac{1}{K+1}-\varepsilon_M\leq\mathbb{Q}_K(n)\leq\frac{1}{K+1}+\varepsilon_M, \qquad n=0,\ldots,K,$$

    with $\lim_{M\to\infty}\varepsilon_M=0$ a.s.
- Uniformity of the statistic $A_{K,t}$ (with K fictitious observations) equivalent to $\hat{p}^M(y_t|y_{1:t-1})$ and $\hat{p}(y_t|y_{1:t-1})$ matching K moments.[29]

---

[28] V. Elvira, J. Míguez, and P. M. Djurić. "Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment". In: *IEEE Transactions on Signal Processing* 65.7 (2016), pp. 1781–1794.

[29] V. Elvira, J. Miguez, and P. M. Djurić. "On the performance of particle filters with adaptive number of particles". In: *Statistics and Computing* 31 (2021), pp. 1–18.