Introduction to sequential Monte Carlo (SMC)

Sara Pérez Vieites

Aalto University
Finnish Center for Artificial Intelligence (FCAI)

CPS group meeting

5 November 2024

Index

Index

Problem Setting: Computing Expectations

Problem: Compute the expected value of a variable $Y = f(X)$, where

- the r.v. $X \in D \subseteq \mathbb{R}^d$ has a probability density function $p(x)$
- f is a real-valued function defined over D

$$\mu = \mathbb{E}_{p(x)}[f(X)] = \int_D f(x)p(x)dx. \quad (1)$$

When $p(x)$ is complex or high-dimensional, there is no analytical solution.

Monte Carlo estimator:

$$\widehat{\mu}_{MC} = \frac{1}{N}\sum_{i=1}^{N} f(X_i) \quad \text{where} \quad X_i \sim p(x) \text{ are i.i.d.} \quad (2)$$

Problem Setting: Computing Expectations

Problem: Compute the expected value of a variable $Y = f(X)$, where

- the r.v. $X \in D \subseteq \mathbb{R}^d$ has a probability density function $p(x)$
- f is a real-valued function defined over D

$$\mu = \mathbb{E}_{p(x)}[f(X)] = \int_D f(x)p(x)dx. \tag{1}$$

When $p(x)$ is complex or high-dimensional, there is no analytical solution.

Monte Carlo estimator:

$$\widehat{\mu}_{MC} = \frac{1}{N}\sum_{i=1}^{N} f(X_i) \quad \text{where} \quad X_i \sim p(x) \text{ are i.i.d.} \tag{2}$$

Problem Setting: Computing Expectations

Problem: Compute the expected value of a variable $Y = f(X)$, where

- the r.v. $X \in D \subseteq \mathbb{R}^d$ has a probability density function $p(x)$
- f is a real-valued function defined over D

$$\mu = \mathbb{E}_{p(x)}[f(X)] = \int_D f(x)p(x)dx. \tag{1}$$

When $p(x)$ is complex or high-dimensional, there is no analytical solution.

Monte Carlo estimator:

$$\widehat{\mu}_{MC} = \frac{1}{N}\sum_{i=1}^{N} f(X_i) \quad \text{where} \quad X_i \sim p(x) \text{ are i.i.d.} \tag{2}$$

Importance Sampling

Motivation: When sampling from $p(x)$ directly is inefficient or costly, we use a proposal distribution $q(x)$ to approximate the expectation.

$$\mu = \int_D f(x)p(x)dx = \int_D f(x)\frac{p(x)}{q(x)}q(x)dx \tag{3}$$

IS Estimator:

$$\widehat{\mu}_{IS} = \frac{1}{N}\sum_{i=1}^{N} f(X_i) \underbrace{\frac{p(X_i)}{q(X_i)}}_{\text{imp. weights}}, \quad \text{where} \quad X_i \sim q(x) \text{ are i.i.d.} \tag{4}$$

Note: need $\text{Supp}(p) \subset \text{Supp}(q)$

Importance Sampling

Motivation: When sampling from $p(x)$ directly is inefficient or costly, we use a proposal distribution $q(x)$ to approximate the expectation.

$$\mu = \int_D f(x)p(x)dx = \int_D f(x)\frac{p(x)}{q(x)}q(x)dx \tag{3}$$

IS Estimator:

$$\widehat{\mu}_{IS} = \frac{1}{N}\sum_{i=1}^{N} f(X_i)\underbrace{\frac{p(X_i)}{q(X_i)}}_{\text{imp. weights}}, \quad \text{where} \quad X_i \sim q(x) \text{ are i.i.d.} \tag{4}$$

Note: need $\mathrm{Supp}(p) \subset \mathrm{Supp}(q)$

Error/variance of the estimators

$$\mathrm{MSE}(\widehat{\mu}_{MC}) = \mathbb{E}_{p(x)}\left[(\widehat{\mu}_{MC} - \mu)^2\right] = \frac{\mathrm{Var}_{p(x)}[f(X)]}{N} \quad (5)$$

$$\mathrm{MSE}(\widehat{\mu}_{IS}) = \mathbb{E}_{q(x)}\left[(\widehat{\mu}_{IS} - \mu)^2\right] = \frac{\mathrm{Var}_{q(x)}[w(X)f(X)]}{N} \quad (6)$$

where $w(X) = \frac{p(X)}{q(X)}$.

Key Points:

- The variance of IS depends on the choice of $q(x)$: a good proposal distribution $q(x)$ should closely resemble $p(x)$ in regions where $f(x)$ contributes significantly to the integral.
- **Goal of IS:** To choose $q(x)$ such that

$$\mathrm{Var}_{q(x)}[w(X)f(X)] < \mathrm{Var}_{p(x)}[f(X)]$$

Error/variance of the estimators

$$\mathrm{MSE}(\widehat{\mu}_{MC}) = \mathbb{E}_{p(x)}\left[(\widehat{\mu}_{MC} - \mu)^2\right] = \frac{\mathrm{Var}_{p(x)}[f(X)]}{N} \tag{5}$$

$$\mathrm{MSE}(\widehat{\mu}_{IS}) = \mathbb{E}_{q(x)}\left[(\widehat{\mu}_{IS} - \mu)^2\right] = \frac{\mathrm{Var}_{q(x)}[w(X)f(X)]}{N} \tag{6}$$

where $w(X) = \frac{p(X)}{q(X)}$.

Key Points:

- The variance of IS depends on the choice of $q(x)$: a good proposal distribution $q(x)$ should closely resemble $p(x)$ in regions where $f(x)$ contributes significantly to the integral.
- **Goal of IS:** To choose $q(x)$ such that

$$\mathrm{Var}_{q(x)}[w(X)f(X)] < \mathrm{Var}_{p(x)}[f(X)]$$

Error/variance of the estimators

$$\text{MSE}(\widehat{\mu}_{MC}) = \mathbb{E}_{p(x)}\left[(\widehat{\mu}_{MC} - \mu)^2\right] = \frac{\text{Var}_{p(x)}[f(X)]}{N} \tag{5}$$

$$\text{MSE}(\widehat{\mu}_{IS}) = \mathbb{E}_{q(x)}\left[(\widehat{\mu}_{IS} - \mu)^2\right] = \frac{\text{Var}_{q(x)}\left[w(X)f(X)\right]}{N} \tag{6}$$

where $w(X) = \frac{p(X)}{q(X)}$.

Key Points:

- The variance of IS depends on the choice of $q(x)$: a good proposal distribution $q(x)$ should closely resemble $p(x)$ in regions where $f(x)$ contributes significantly to the integral.
- **Goal of IS:** To choose $q(x)$ such that

$$\text{Var}_{q(x)}\left[w(X)f(X)\right] < \text{Var}_{p(x)}[f(X)]$$

Static Setting vs. Dynamical Setting

Static Setting:

- **Goal**: Compute the expectation of a function $f(X)$ under a fixed distribution $p(x)$:

$$\mu = \mathbb{E}_{p(x)}[f(X)] = \int f(x)\, p(x)\, dx.$$

- **Assumption**: The target distribution $p(x)$ is constant (does not change over time).

Dynamical Setting:

- **Goal**: Estimate an evolving expectation over a time-varying distribution $p(x_{0:t}|y_{1:t})$:

$$\mu_t = \mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(X_{0:t})] = \int f(x_{0:t})\, p(x_{0:t}|y_{1:t})\, dx_{0:t}.$$

- **Challenge**: $p(x_{0:t}|y_{1:t})$ changes at each time step t as new data y_t arrives.

Static Setting vs. Dynamical Setting

Static Setting:

- **Goal**: Compute the expectation of a function $f(X)$ under a fixed distribution $p(x)$:

$$\mu = \mathbb{E}_{p(x)}[f(X)] = \int f(x)\, p(x)\, dx.$$

- **Assumption**: The target distribution $p(x)$ is constant (does not change over time).

Dynamical Setting:

- **Goal**: Estimate an evolving expectation over a time-varying distribution $p(x_{0:t}|y_{1:t})$:

$$\mu_t = \mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(X_{0:t})] = \int f(x_{0:t})\, p(x_{0:t}|y_{1:t})\, dx_{0:t}.$$

- **Challenge**: $p(x_{0:t}|y_{1:t})$ changes at each time step t as new data y_t arrives.

Index

Dynamical setting: state-space model (SSM)

We are interested in systems that can be represented by **Markov state-space dynamical models**, where

$$\overbrace{x_t}^{\text{state}} = \overbrace{f(x_{t-1})}^{\text{state transition function}} + \overbrace{v_t}^{\text{state noise}}, \tag{7}$$

$$\underbrace{y_t}_{\text{observation}} = \underbrace{g(x_t)}_{\text{observation function}} + \underbrace{r_t}_{\text{observation noise}}. \tag{8}$$

In terms of relevant **probability density functions (pdfs)**:

- Prior distribution: initial state $x_0 \sim p(x_0)$.
- Transition pdf of the state: $x_t \sim p(x_t|x_{t-1})$ describes the system dynamics over time.
- Conditional pdf of the observation: $y_t \sim p(y_t|x_t)$ relates the observations to the hidden state.

Dynamical setting: state-space model (SSM)

We are interested in systems that can be represented by **Markov state-space dynamical models**, where

$$\overbrace{x_t}^{\text{state}} = \overbrace{f(x_{t-1})}^{\text{state transition function}} + \overbrace{v_t}^{\text{state noise}}, \tag{7}$$

$$\underbrace{y_t}_{\text{observation}} = \underbrace{g(x_t)}_{\text{observation function}} + \underbrace{r_t}_{\text{observation noise}}. \tag{8}$$

In terms of relevant **probability density functions (pdfs)**:

- Prior distribution: initial state $x_0 \sim p(x_0)$.
- Transition pdf of the state: $x_t \sim p(x_t|x_{t-1})$ describes the system dynamics over time.
- Conditional pdf of the observation: $y_t \sim p(y_t|x_t)$ relates the observations to the hidden state.

SSM: example

A nonlinear state-space model could be:

$$x_t = \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1 + x_{t-1}^2} + 8\cos(1.2t) + v_t,$$

$$y_t = \frac{x_t^2}{20} + r_t,$$

where:

- $v_t \sim \mathcal{N}(0, \sigma_v^2)$ is the process noise,
- $r_t \sim \mathcal{N}(0, \sigma_r^2)$ is the observation noise.

The associated probability density functions (pdfs) are:

- **State transition pdf**: $p(x_t|x_{t-1}) = \mathcal{N}\left(x_t \,\middle|\, \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t), \sigma_v^2\right)$.
- **Observation pdf**: $p(y_t|x_t) = \mathcal{N}\left(y_t \,\middle|\, \frac{x_t^2}{20}, \sigma_r^2\right)$.

SSM: example

A nonlinear state-space model could be:

$$x_t = \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1 + x_{t-1}^2} + 8\cos(1.2t) + v_t,$$

$$y_t = \frac{x_t^2}{20} + r_t,$$

where:

- $v_t \sim \mathcal{N}(0, \sigma_v^2)$ is the process noise,
- $r_t \sim \mathcal{N}(0, \sigma_r^2)$ is the observation noise.

The associated probability density functions (pdfs) are:

- **State transition pdf**: $p(x_t|x_{t-1}) = \mathcal{N}\left(x_t \,\middle|\, \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t), \sigma_v^2\right)$.
- **Observation pdf**: $p(y_t|x_t) = \mathcal{N}\left(y_t \,\middle|\, \frac{x_t^2}{20}, \sigma_r^2\right)$.

Target Integrals in State-Space Models

We want to compute **expectations over the posterior distribution** $p(x_{0:t}|y_{1:t})$, for example:

$$\mu_t = \mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(X_{0:t})] = \int f(x_{0:t})\, p(x_{0:t}|y_{1:t})\, dx_{0:t}. \tag{9}$$

Main Challenges:

- Sampling from the posterior: Direct sampling from $p(x_{0:t}|y_{1:t})$ is often infeasible, especially in nonlinear or non-Gaussian models.
- Sequential Nature: The posterior evolves with each new observation y_t, requiring a recursive approach.
- Computational Cost: Sequential estimation can require a large number of samples for accuracy, increasing computational demands.

Target Integrals in State-Space Models

We want to compute **expectations over the posterior distribution** $p(x_{0:t}|y_{1:t})$, for example:

$$\mu_t = \mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(X_{0:t})] = \int f(x_{0:t})\, p(x_{0:t}|y_{1:t})\, dx_{0:t}. \tag{9}$$

Main Challenges:

- Sampling from the posterior: Direct sampling from $p(x_{0:t}|y_{1:t})$ is often infeasible, especially in nonlinear or non-Gaussian models.
- Sequential Nature: The posterior evolves with each new observation y_t, requiring a recursive approach.
- Computational Cost: Sequential estimation can require a large number of samples for accuracy, increasing computational demands.

Index

Sequential Importance Sampling (SIS)

We use a proposal distribution $q(x_{0:t})$ and rewrite the posterior in terms of this proposal:

$$\mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(x_t)] = \int f(x_t) \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t})} q(x_{0:t})\, dx_{0:t}.$$

where the **importance weights** are

$$w_t = \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t})}.$$

To make it recursive in time, we rewrite $q(x_{0:t}) = q(x_t|x_{0:t-1})q(x_{0:t-1})$, and:

$$p(x_{0:t}|y_{1:t}) = \frac{p(y_t|x_t)\, p(x_t|x_{t-1})}{p(y_t|y_{1:t-1})} \underbrace{p(x_{0:t-1}|y_{1:t-1})}_{\text{Posterior at } t-1}.$$

using Bayes' theorem.

Remember Bayes' theorem: $p(A, B) = p(A|B)p(B) = p(B|A)p(A)$

Sequential Importance Sampling (SIS)

We use a proposal distribution $q(x_{0:t})$ and rewrite the posterior in terms of this proposal:

$$\mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(x_t)] = \int f(x_t) \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t})} q(x_{0:t})\, dx_{0:t}.$$

where the **importance weights** are

$$w_t = \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t})}.$$

To make it recursive in time, we rewrite $q(x_{0:t}) = q(x_t|x_{0:t-1})q(x_{0:t-1})$, and:

$$p(x_{0:t}|y_{1:t}) = \frac{p(y_t|x_t)\, p(x_t|x_{t-1})}{p(y_t|y_{1:t-1})} \underbrace{p(x_{0:t-1}|y_{1:t-1})}_{\text{Posterior at } t-1}.$$

using Bayes' theorem.

Remember Bayes' theorem: $p(A,B) = p(A|B)p(B) = p(B|A)p(A)$

Sequential Importance Sampling (SIS)

That changes the integral to

$$\begin{aligned} \mu_t &= \mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(X_{0:t})] \qquad (10) \\ &= \int f(x_{0:t}) \underbrace{\frac{p(y_t|x_t)\, p(x_t|x_{t-1})}{\underbrace{p(y_t|y_{1:t-1})}_{\text{normalizing constant}} \quad q(x_t|x_{0:t-1})} \underbrace{\frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1})}}_{w_{t-1}}}_{w_t} q(x_{0:t})\, dx_{0:t}. \end{aligned}$$

Recursive decomposition of weights: The weights are given recursively as (as a function of the trajectory):

$$w_t(x_{0:t}) = \frac{p(y_t|x_t)\, p(x_t|x_{t-1})}{p(y_t|y_{1:t-1})q(x_t|x_{0:t-1})} w_{t-1}(x_{0:t-1}). \qquad (11)$$

Sequential Importance Sampling (SIS)

That changes the integral to

$$\begin{aligned}
\mu_t &= \mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(X_{0:t})] \qquad (10)\\
&= \int f(x_{0:t}) \underbrace{\frac{p(y_t|x_t)\,p(x_t|x_{t-1})}{\underbrace{p(y_t|y_{1:t-1})}_{\text{normalizing constant}}\; q(x_t|x_{0:t-1})} \underbrace{\frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1})}}_{w_{t-1}}}_{w_t} q(x_{0:t})\, dx_{0:t}.
\end{aligned}$$

Recursive decomposition of weights: The weights are given recursively as (as a function of the trajectory):

$$w_t(x_{0:t}) = \frac{p(y_t|x_t)\,p(x_t|x_{t-1})}{p(y_t|y_{1:t-1})q(x_t|x_{0:t-1})} w_{t-1}(x_{0:t-1}). \qquad (11)$$

An aside: Self-Normalized Importance Sampling (SNIS)

Problem: The normalizing constant $p(y_t|y_{1:t-1})$
Solution: We also apply importance sampling, reusing the same samples

$$
\begin{aligned}
p(y_t|y_{1:t-1}) &= \int p(y_t|x_t)p(x_t|x_{t-1})p(x_{0:t-1}|y_{1:t-1})dx_{0:t} \qquad (12) \\
&= \int \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_t|x_{0:t-1})} \frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1})} q(x_{0:t})dx_{0:t} \\
&\simeq \frac{1}{N}\sum_{i=1}^{N} \frac{p(y_t|x_t^{(i)})\, p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)})} w_{t-1}^{(i)} \quad \text{for } x_{0:t}^{(i)} \sim q(x_{0:t})
\end{aligned}
$$

An aside: Self-Normalized Importance Sampling (SNIS)

Problem: The normalizing constant $p(y_t|y_{1:t-1})$
Solution: We also apply importance sampling, reusing the same samples

$$
\begin{aligned}
p(y_t|y_{1:t-1}) &= \int p(y_t|x_t)p(x_t|x_{t-1})p(x_{0:t-1}|y_{1:t-1})dx_{0:t} \qquad (12)\\
&= \int \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_t|x_{0:t-1})}\frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1})}q(x_{0:t})dx_{0:t}\\
&\simeq \frac{1}{N}\sum_{i=1}^{N}\frac{p(y_t|x_t^{(i)})\,p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)})}w_{t-1}^{(i)} \quad \text{for } x_{0:t}^{(i)} \sim q(x_{0:t})
\end{aligned}
$$

An aside: Self-Normalized Importance Sampling (SNIS)

Self-normalized weights (for $i = 1, \ldots, N$ samples):

$$w_t(x_{0:t}) \quad = \quad \frac{\frac{p(y_t|x_t)\, p(x_t|x_{t-1})}{q(x_t|x_{0:t-1})} w_{t-1}}{p(y_t|y_{1:t-1})}, \tag{13}$$

$$w_t(x_{0:t}^{(i)}) = w_t^{(i)} \quad \simeq \quad \frac{\frac{1}{N} \frac{p(y_t|x_t^{(i)})\, p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)})} w_{t-1}^{(i)}}{\frac{1}{N} \sum_{j=1}^N \frac{p(y_t|x_t^{(j)})\, p(x_t^{(j)}|x_{t-1}^{(j)})}{q(x_t^{(j)}|x_{t-1}^{(j)})} w_{t-1}^{(j)}} = \frac{\bar{w}_t^{(i)}}{\sum_{j=1}^N \bar{w}_t^{(j)}} \tag{14}$$

Note that $\sum_{i=1}^N w_t^{(i)} = 1$, and $w_t^{(i)} \in [0, 1], \forall i$.

Result: The normalization removes the need for $p(y_t|y_{1:t-1})$, allowing us to approximate expectations over the posterior without knowing the exact normalizing constant.

An aside: Self-Normalized Importance Sampling (SNIS)

Self-normalized weights (for $i = 1, \ldots, N$ samples):

$$w_t(x_{0:t}) \quad = \quad \frac{\frac{p(y_t|x_t)\, p(x_t|x_{t-1})}{q(x_t|x_{0:t-1})} w_{t-1}}{p(y_t|y_{1:t-1})}, \tag{13}$$

$$w_t(x_{0:t}^{(i)}) = w_t^{(i)} \quad \simeq \quad \frac{\frac{1}{N} \frac{p(y_t|x_t^{(i)})\, p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)})} w_{t-1}^{(i)}}{\frac{1}{N} \sum_{j=1}^N \frac{p(y_t|x_t^{(j)})\, p(x_t^{(j)}|x_{t-1}^{(j)})}{q(x_t^{(j)}|x_{t-1}^{(j)})} w_{t-1}^{(j)}} = \frac{\bar{w}_t^{(i)}}{\sum_{j=1}^N \bar{w}_t^{(j)}} \tag{14}$$

Note that $\sum_{i=1}^N w_t^{(i)} = 1$, and $w_t^{(i)} \in [0, 1], \forall i$.

Result: The normalization removes the need for $p(y_t|y_{1:t-1})$, allowing us to approximate expectations over the posterior without knowing the exact normalizing constant.

An aside: Self-Normalized Importance Sampling (SNIS)

Self-normalized weights (for $i = 1, \ldots, N$ samples):

$$w_t(x_{0:t}) = \frac{\frac{p(y_t|x_t)\,p(x_t|x_{t-1})}{q(x_t|x_{0:t-1})} w_{t-1}}{p(y_t|y_{1:t-1})}, \tag{13}$$

$$w_t(x_{0:t}^{(i)}) = w_t^{(i)} \simeq \frac{\frac{1}{N}\frac{p(y_t|x_t^{(i)})\,p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)})} w_{t-1}^{(i)}}{\frac{1}{N}\sum_{j=1}^{N}\frac{p(y_t|x_t^{(j)})\,p(x_t^{(j)}|x_{t-1}^{(j)})}{q(x_t^{(j)}|x_{t-1}^{(j)})} w_{t-1}^{(j)}} = \frac{\bar{w}_t^{(i)}}{\sum_{j=1}^{N}\bar{w}_t^{(j)}} \tag{14}$$

Note that $\sum_{i=1}^{N} w_t^{(i)} = 1$, and $w_t^{(i)} \in [0,1], \forall i$.

Result: The normalization removes the need for $p(y_t|y_{1:t-1})$, allowing us to approximate expectations over the posterior without knowing the exact normalizing constant.

An aside: Self-Normalized Importance Sampling (SNIS)

Self-normalized weights (for $i = 1, \ldots, N$ samples):

$$w_t(x_{0:t}) = \frac{\frac{p(y_t|x_t)\, p(x_t|x_{t-1})}{q(x_t|x_{0:t-1})} w_{t-1}}{p(y_t|y_{1:t-1})}, \tag{13}$$

$$w_t(x_{0:t}^{(i)}) = w_t^{(i)} \simeq \frac{\frac{1}{N} \frac{p(y_t|x_t^{(i)})\, p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)})} w_{t-1}^{(i)}}{\frac{1}{N} \sum_{j=1}^{N} \frac{p(y_t|x_t^{(j)})\, p(x_t^{(j)}|x_{t-1}^{(j)})}{q(x_t^{(j)}|x_{t-1}^{(j)})} w_{t-1}^{(j)}} = \frac{\bar{w}_t^{(i)}}{\sum_{j=1}^{N} \bar{w}_t^{(j)}} \tag{14}$$

Note that $\sum_{i=1}^{N} w_t^{(i)} = 1$, and $w_t^{(i)} \in [0, 1], \forall i$.

Result: The normalization removes the need for $p(y_t|y_{1:t-1})$, allowing us to approximate expectations over the posterior without knowing the exact normalizing constant.

So... Sequential Importance Sampling (SIS)

$$\mu_t = \mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(X_{0:t})] \tag{15}$$

$$= \int f(x_{0:t}) \underbrace{\frac{p(y_t|x_t)\, p(x_t|x_{t-1})}{\underbrace{p(y_t|y_{1:t-1})}_{\text{normalizing constant}}\; q(x_t|x_{0:t-1})} \underbrace{\frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1})}}_{w_{t-1}}}_{w_t} q(x_{0:t})\, dx_{0:t}.$$

We can approximate the integral

$$\widehat{\mu}_t \simeq \frac{1}{N} \sum_{i=1}^{N} f(x_{0:t}^{(i)}) \frac{\bar{w}_t^{(i)}}{\sum_{j=1}^{N} \bar{w}_t^{(j)}} \qquad \text{for } x_{0:t}^{(i)} \sim q(x_{0:t}) \tag{16}$$

where the **unnormalized weights** are

$$\bar{w}_t^{(i)} = \frac{p(y_t|x_t^{(i)})\, p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)})} w_{t-1}^{(i)}.$$

So... Sequential Importance Sampling (SIS)

$$\mu_t = \mathbb{E}_{p(x_{0:t}|y_{1:t})}[f(X_{0:t})] \tag{15}$$

$$= \int f(x_{0:t}) \underbrace{\frac{p(y_t|x_t)\, p(x_t|x_{t-1})}{\underbrace{p(y_t|y_{1:t-1})}_{\text{normalizing constant}}\; q(x_t|x_{0:t-1})} \underbrace{\frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1})}}_{w_{t-1}}}_{w_t} q(x_{0:t})\, dx_{0:t}.$$

We can approximate the integral

$$\widehat{\mu}_t \simeq \frac{1}{N} \sum_{i=1}^{N} f(x_{0:t}^{(i)}) \frac{\bar{w}_t^{(i)}}{\sum_{j=1}^{N} \bar{w}_t^{(j)}} \qquad \text{for } x_{0:t}^{(i)} \sim q(x_{0:t}) \tag{16}$$

where the **unnormalized weights** are

$$\bar{w}_t^{(i)} = \frac{p(y_t|x_t^{(i)})\, p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)})} w_{t-1}^{(i)}.$$

So... Sequential Importance Sampling (SIS)

Algorithm: (at each time step t)

1. Sample $x_t^{(i)} \sim q(x_t | x_{t-1}^{(i)})$, for $i = 1, \ldots, N$.
2. Compute unnormalized weights

$$\bar{w}_t^{(i)} = \frac{p(y_t | x_t^{(i)})\, p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i)} | x_{0:t-1}^{(i)})} w_{t-1}^{(i)}.$$

3. Normalize the weights

$$w_t^{(i)} = \frac{\bar{w}_t^{(i)}}{\sum_{k=1}^{N} \bar{w}_t^{(k)}}.$$

4. Now we can estimate:

$$\widehat{\mu}_t = \frac{1}{N} \sum_{i=1}^{N} f(x_{0:t}^{(i)}) w_t^{(i)}.$$

So... Sequential Importance Sampling (SIS)

Algorithm: (at each time step t)

1. Sample $x_t^{(i)} \sim q(x_t|x_{t-1}^{(i)})$, for $i = 1, \ldots, N$.
2. Compute unnormalized weights

$$\bar{w}_t^{(i)} = \frac{p(y_t|x_t^{(i)})\, p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)})} w_{t-1}^{(i)}.$$

3. Normalize the weights

$$w_t^{(i)} = \frac{\bar{w}_t^{(i)}}{\sum_{k=1}^{N} \bar{w}_t^{(k)}}.$$

4. Now we can estimate:

$$\widehat{\mu}_t = \frac{1}{N} \sum_{i=1}^{N} f(x_{0:t}^{(i)}) w_t^{(i)}.$$

So... Sequential Importance Sampling (SIS)

Algorithm: (at each time step t)

1. Sample $x_t^{(i)} \sim q(x_t|x_{t-1}^{(i)})$, for $i = 1, \ldots, N$.
2. Compute unnormalized weights

$$\bar{w}_t^{(i)} = \frac{p(y_t|x_t^{(i)})\, p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)})} w_{t-1}^{(i)}.$$

3. Normalize the weights

$$w_t^{(i)} = \frac{\bar{w}_t^{(i)}}{\sum_{k=1}^{N} \bar{w}_t^{(k)}}.$$

4. Now we can estimate:

$$\widehat{\mu}_t = \frac{1}{N} \sum_{i=1}^{N} f(x_{0:t}^{(i)}) w_t^{(i)}.$$

So... Sequential Importance Sampling (SIS)

Algorithm: (at each time step t)

1. Sample $x_t^{(i)} \sim q(x_t | x_{t-1}^{(i)})$, for $i = 1, \ldots, N$.
2. Compute unnormalized weights

$$\bar{w}_t^{(i)} = \frac{p(y_t | x_t^{(i)})\, p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i)} | x_{0:t-1}^{(i)})} w_{t-1}^{(i)}.$$

3. Normalize the weights

$$w_t^{(i)} = \frac{\bar{w}_t^{(i)}}{\sum_{k=1}^{N} \bar{w}_t^{(k)}}.$$

4. Now we can estimate:

$$\widehat{\mu}_t = \frac{1}{N} \sum_{i=1}^{N} f(x_{0:t}^{(i)}) w_t^{(i)}.$$

SIS, we have a problem

We use the samples $\{x_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ to approximate integrals with respect to $p(x_{0:t}|y_{1:t})$, such that

$$\widehat{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{0:t}^{(i)}} \tag{17}$$

Problem: Weight Degeneracy
As the weights are updated recursively, they involve products of the previous weights.

Result: Because $w_t^{(i)} < 1$, most weights go to zero as t grows — meaning only a few particles contribute significantly, while the others become negligible, leading to **weight degeneracy**.

SIS, we have a problem

We use the samples $\{x_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^{N}$ to approximate integrals with respect to $p(x_{0:t}|y_{1:t})$, such that

$$\widehat{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{0:t}^{(i)}} \tag{17}$$

Problem: Weight Degeneracy
As the weights are updated recursively, they involve products of the previous weights.

Result: Because $w_t^{(i)} < 1$, most weights go to zero as t grows — meaning only a few particles contribute significantly, while the others become negligible, leading to **weight degeneracy**.

SIS, we have a problem

We use the samples $\{x_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ to approximate integrals with respect to $p(x_{0:t}|y_{1:t})$, such that

$$\widehat{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{0:t}^{(i)}} \tag{17}$$

Problem: Weight Degeneracy

As the weights are updated recursively, they involve products of the previous weights.

Result: Because $w_t^{(i)} < 1$, most weights go to zero as t grows — meaning only a few particles contribute significantly, while the others become negligible, leading to **weight degeneracy**.

SIS, we have a problem

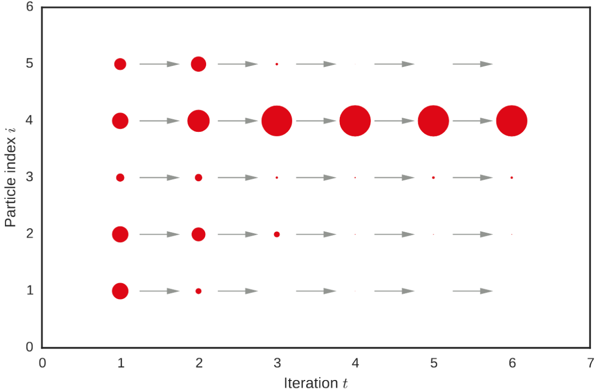

Figure: Weight degeneracy.

Figure from: Naesseth, C. A., Lindsten, F., & Schön, T. B. (2019). *Elements of sequential Monte Carlo.* Foundations and Trends in Machine Learning, 12(3), 307-392.

Index

Resampling in SIS

Goal: Reduce weight degeneracy by discarding low-weight particles and replicating high-weight ones. But the new set of particles needs to represent the same pdf such that

$$\widehat{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{0:t}^{(i)}} = \sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta_{\tilde{x}_{0:t}^{(i)}} \quad (18)$$

How?

- Resampling can be seen as a multinomial sampling process.
- We draw N particles with replacement from the existing set, where each particle $x_{0:t}^{(i)}$ is selected with probability proportional to its weight $w_t^{(i)}$.

Result: Particles with higher weights are more likely to be chosen multiple times, while those with lower weights may be removed.

Multinomial Resampling Steps

Given particles and weights $\{x_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$:

1. **Draw Indices:** Sample N indices $I^{(1)}, I^{(2)}, \ldots, I^{(N)}$ from the discrete distribution defined by $\{w_t^{(1)}, \ldots, w_t^{(N)}\}$:

$$I^{(j)} \sim \text{Discrete}(w_t^{(1)}, w_t^{(2)}, \ldots, w_t^{(N)})$$

2. **Generate the resampled set** $\{\tilde{x}_{0:t}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^N$: For each $j = 1, \ldots, N$, set:

$$\tilde{x}_{0:t}^{(j)} = x_{0:t}^{(I^{(j)})} \quad \text{and} \quad \tilde{w}_t^{(j)} = \frac{1}{N}.$$

Multinomial Resampling

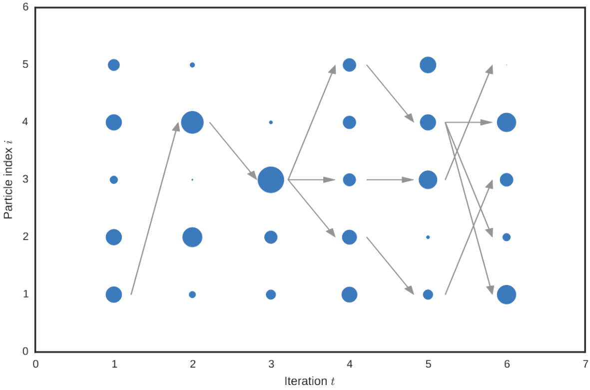

Figure: Resampling.

Figure from: Naesseth, C. A., Lindsten, F., & Schön, T. B. (2019). *Elements of sequential Monte Carlo.* Foundations and Trends in Machine Learning, 12(3), 307-392.

Sequential Importance Resampling (SIR)

Algorithm: (at each time step t)

1. **Sample:** Draw samples $x_t^{(i)} \sim q(x_t | x_{t-1}^{(i)})$ for $i = 1, \ldots, N$.
2. **Compute Weights:**

$$\bar{w}_t^{(i)} = \frac{p(y_t | x_t^{(i)})\, p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i)} | x_{0:t-1}^{(i)})}\, w_{t-1}^{(i)} \quad \text{and} \quad w_t^{(i)} = \frac{\bar{w}_t^{(i)}}{\sum_{k=1}^{N} \bar{w}_t^{(k)}}$$

3. **Estimate:**

$$\widehat{\mu}_t = \sum_{i=1}^{N} f(x_{0:t}^{(i)})\, w_t^{(i)}$$

4. **Resampling Step:**
   - Draw N indices $I^{(j)} \sim \text{Discrete}(w_t^{(1)}, \ldots, w_t^{(N)})$
   - Set $\tilde{x}_{0:t}^{(j)} = x_{0:t}^{(I^{(j)})}$ and $\tilde{w}_t^{(j)} = \frac{1}{N}$

Sequential Importance Resampling (SIR)

Algorithm: (at each time step t)

1. **Sample:** Draw samples $x_t^{(i)} \sim q(x_t | x_{t-1}^{(i)})$ for $i = 1, \dots, N$.
2. **Compute Weights:**

$$\bar{w}_t^{(i)} = \frac{p(y_t|x_t^{(i)})\, p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)})}\, w_{t-1}^{(i)} \quad \text{and} \quad w_t^{(i)} = \frac{\bar{w}_t^{(i)}}{\sum_{k=1}^N \bar{w}_t^{(k)}}$$

3. **Estimate:**

$$\widehat{\mu}_t = \sum_{i=1}^N f(x_{0:t}^{(i)}) w_t^{(i)}$$

4. **Resampling Step:**
   - Draw N indices $I^{(j)} \sim \text{Discrete}(w_t^{(1)}, \dots, w_t^{(N)})$
   - Set $\tilde{x}_{0:t}^{(j)} = x_{0:t}^{(I^{(j)})}$ and $\tilde{w}_t^{(j)} = \frac{1}{N}$

References

- Wills, A. G., & Schön, T. B. (2023). Sequential monte carlo: A unified review. Annual Review of Control, Robotics, and Autonomous Systems, 6(1), 159-182.

- Book focused on **SSMs, IS, sequential Monte Carlo and MCMC** (warning: french notation).

  Chopin, N., & Papaspiliopoulos, O. (2020). An introduction to sequential Monte Carlo (Vol. 4). Berlin: Springer.
  Link to the pdf.

- Book about **Monte Carlo, importance sampling**, MCMC, and QMC.

  Owen, A. B. (2013). Monte Carlo theory, methods and examples.
  Link to the pdf.

- Book about Bayesian inference, **Kalman filters**, and a bit of sequential Monte Carlo (particle filters).

  Särkkä, S., & Svensson, L. (2023). Bayesian filtering and smoothing (Vol. 17). Cambridge university press.
  Link to the pdf.

Thank you!

References

- Wills, A. G., & Schön, T. B. (2023). Sequential monte carlo: A unified review. Annual Review of Control, Robotics, and Autonomous Systems, 6(1), 159-182.

- Book focused on **SSMs, IS, sequential Monte Carlo and MCMC** (warning: french notation).

  Chopin, N., & Papaspiliopoulos, O. (2020). An introduction to sequential Monte Carlo (Vol. 4). Berlin: Springer.
  Link to the pdf.

- Book about **Monte Carlo, importance sampling**, MCMC, and QMC.

  Owen, A. B. (2013). Monte Carlo theory, methods and examples.
  Link to the pdf.

- Book about Bayesian inference, **Kalman filters**, and a bit of sequential Monte Carlo (particle filters).

  Särkkä, S., & Svensson, L. (2023). Bayesian filtering and smoothing (Vol. 17). Cambridge university press.
  Link to the pdf.

Thank you!

References

- Wills, A. G., & Schön, T. B. (2023). Sequential monte carlo: A unified review. Annual Review of Control, Robotics, and Autonomous Systems, 6(1), 159-182.

- Book focused on **SSMs, IS, sequential Monte Carlo and MCMC** (warning: french notation).

  Chopin, N., & Papaspiliopoulos, O. (2020). An introduction to sequential Monte Carlo (Vol. 4). Berlin: Springer.
  Link to the pdf.

- Book about **Monte Carlo, importance sampling**, MCMC, and QMC.

  Owen, A. B. (2013). Monte Carlo theory, methods and examples.
  Link to the pdf.

- Book about Bayesian inference, **Kalman filters**, and a bit of sequential Monte Carlo (particle filters).

  Särkkä, S., & Svensson, L. (2023). Bayesian filtering and smoothing (Vol. 17). Cambridge university press.
  Link to the pdf.

Thank you!

References

- Wills, A. G., & Schön, T. B. (2023). Sequential monte carlo: A unified review. Annual Review of Control, Robotics, and Autonomous Systems, 6(1), 159-182.

- Book focused on **SSMs, IS, sequential Monte Carlo and MCMC** (warning: french notation).

  Chopin, N., & Papaspiliopoulos, O. (2020). An introduction to sequential Monte Carlo (Vol. 4). Berlin: Springer.
  Link to the pdf.

- Book about **Monte Carlo, importance sampling**, MCMC, and QMC.

  Owen, A. B. (2013). Monte Carlo theory, methods and examples.
  Link to the pdf.

- Book about Bayesian inference, **Kalman filters**, and a bit of sequential Monte Carlo (particle filters).

  Särkkä, S., & Svensson, L. (2023). Bayesian filtering and smoothing (Vol. 17). Cambridge university press.
  Link to the pdf.

Thank you!