Introduction to Sequential Monte Carlo and Particle MCMC Methods

Axel Finke

Department of Statistics,
University of Warwick, UK

Oberseminar, Münster: July 10, 2013

Piecewise Deterministic Processes (PDPs)

Ingredients

- **Time-dependent** parameters: marked point process $(\tau_j, \phi_j)_{j \in \mathbb{N} \cup \{0\}}$ with
  - jump times $0 = \tau_0 < \tau_1 < \tau_2 < \ldots$
  - jump sizes $\phi_0, \phi_1, \phi_2, \ldots$
- **Static** parameters: θ.
- Deterministic function: F^θ.

Piecewise Deterministic Processes (PDPs)

Ingredients

- **Time-dependent** parameters: marked point process $(\tau_j, \phi_j)_{j \in \mathbb{N} \cup \{0\}}$ with
  - jump times $0 = \tau_0 < \tau_1 < \tau_2 < \dots$
  - jump sizes $\phi_0, \phi_1, \phi_2, \dots$
- **Static** parameters: θ.
- Deterministic function: F^θ.

Piecewise Deterministic Processes (PDPs)

Ingredients

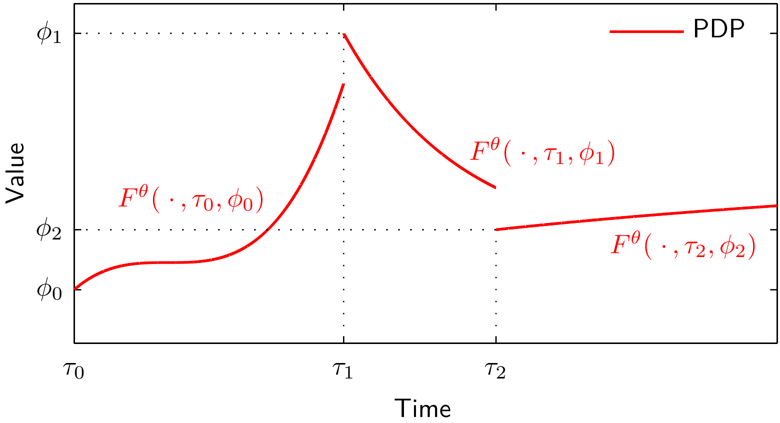
- **Time-dependent** parameters: marked point process $(\tau_j, \phi_j)_{j \in \mathbb{N} \cup \{0\}}$ with
  - jump times $0 = \tau_0 < \tau_1 < \tau_2 < \dots$
  - jump sizes $\phi_0, \phi_1, \phi_2, \dots$
- **Static** parameters: θ.
- Deterministic function: F^θ.

Piecewise Deterministic Processes (PDPs)

Ingredients

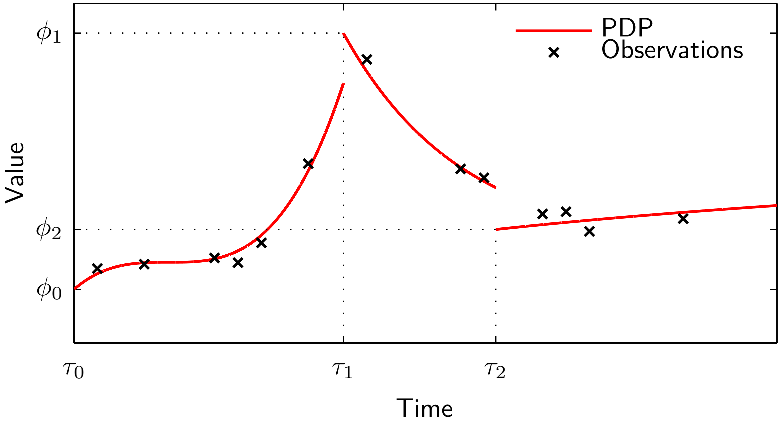
- **Time-dependent** parameters: marked point process $(\tau_j, \phi_j)_{j \in \mathbb{N} \cup \{0\}}$ with
  - jump times $0 = \tau_0 < \tau_1 < \tau_2 < \dots$
  - jump sizes $\phi_0, \phi_1, \phi_2, \dots$
- **Static** parameters: θ.
- Deterministic function: F^θ.

Sketch

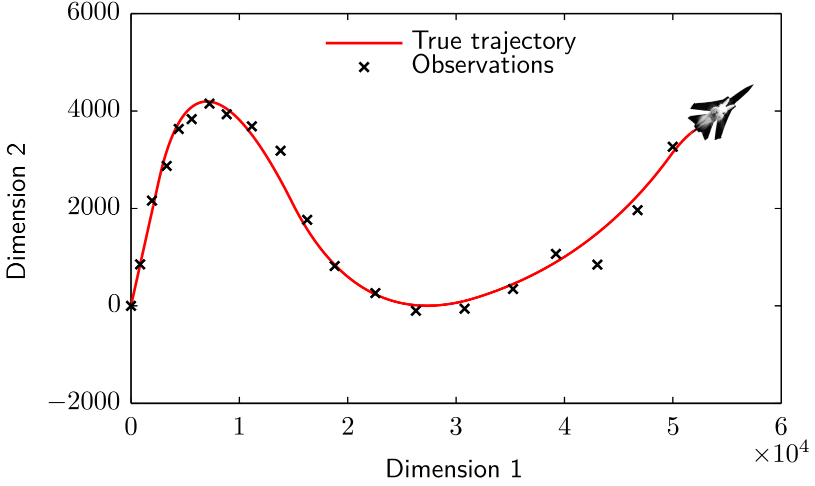


PDP
ϕ_1
ϕ_2
ϕ_0
Value
$F^\theta(\,\cdot\,,\tau_0,\phi_0)$
$F^\theta(\,\cdot\,,\tau_1,\phi_1)$
$F^\theta(\,\cdot\,,\tau_2,\phi_2)$
τ_0
τ_1
τ_2
Time

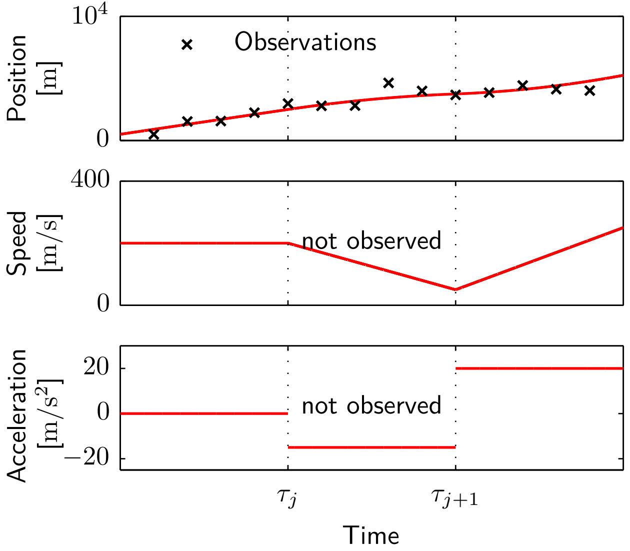
Observations



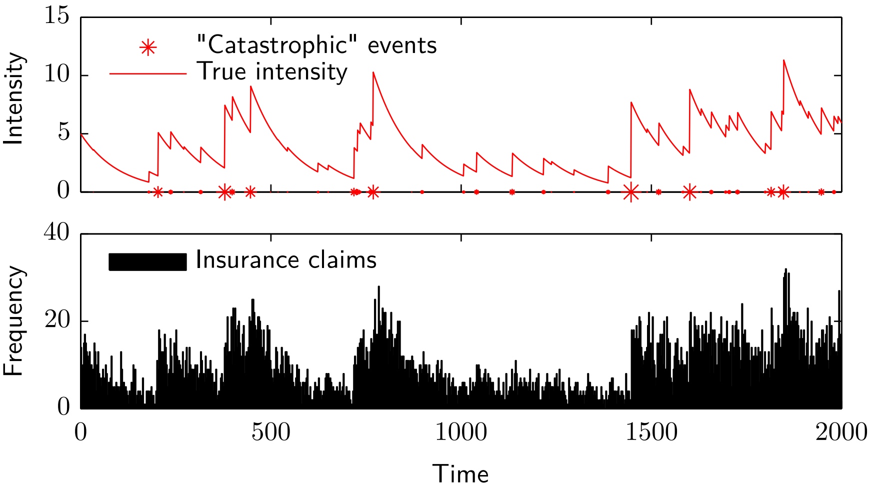
Example I: Object Tracking



Example I: Object Tracking (continued)



Example II: Shot-Noise Cox Process



Inference

- Sequential Monte Carlo filter for PDPs introduced by Whiteley, Johansen & Godsill (2011).
- Efficient methods for estimating θ still missing (though Reversible-Jump MCMC works for simple models)

Inference

- Sequential Monte Carlo filter for PDPs introduced by Whiteley, Johansen & Godsill (2011).
- Efficient methods for estimating θ still missing (though Reversible-Jump MCMC works for simple models)

Motivation

- $X \sim \pi$ with support E.
- $f : E \to \mathbb{R}$ some (π-integrable) function.
- Want to calculate

$$\pi(f) := \int_E f(x)\pi(\mathrm{d}x)$$

$$\left(= \int_E f(x)\pi(x)\,\mathrm{d}x \right)$$

$$= \mathbf{E}[f(X)].$$

Motivation

- $X \sim \pi$ with support E.
- $f : E \to \mathbb{R}$ some (π-integrable) function.
- Want to calculate

$$\pi(f) := \int_E f(x)\pi(\mathrm{d}x)$$
$$\left(= \int_E f(x)\pi(x)\,\mathrm{d}x \right)$$
$$= \mathbf{E}[f(X)].$$

Motivation

- $X \sim \pi$ with support E.
- $f : E \to \mathbb{R}$ some (π-integrable) function.
- Want to calculate

$$\pi(f) := \int_E f(x)\pi(\mathrm{d}x)$$
$$\left(= \int_E f(x)\pi(x)\,\mathrm{d}x \right)$$
$$= \mathbf{E}[f(X)].$$

Motivation, continued

Example

Often, $E \subseteq \mathbb{R}$ and $\pi(x) = p(x|y)$ for some data y, so that

$$\pi(f) = \begin{cases} \mathbf{P}(X \in A | Y = y), & \text{if } f = 1_A \text{ for } A \subseteq E, \\ \mathbf{E}[X^k | Y = y], & \text{if } f = \mathrm{id}^k, \\ \mathbf{var}[X | Y = y], & \text{if } f = [\mathrm{id} - \pi(f)]^2, \\ \vdots & \end{cases}$$

Motivation, continued

Example

Often, $E \subseteq \mathbb{R}$ and $\pi(x) = p(x|y)$ for some data y, so that

$$\pi(f) = \begin{cases} \mathbf{P}(X \in A | Y = y), & \text{if } f = 1_A \text{ for } A \subseteq E, \\ \mathbf{E}[X^k | Y = y], & \text{if } f = \mathrm{id}^k, \\ \mathbf{var}[X | Y = y], & \text{if } f = [\mathrm{id} - \pi(f)]^2, \\ \vdots & \end{cases}$$

Monte Carlo Methods

- **Problem:** analytical evaluation of $\pi(f)$ costly/impossible.
- **Idea:**
  1. construct approximation $\hat{\pi}$ of π.
  2. estimate $\pi(f)$ by

$$\hat{\pi}(f) = \int_E f(x)\,\hat{\pi}(\mathrm{d}x).$$

- *Monte Carlo methods:* construction of $\hat{\pi}$ based on (computer-generated) (pseudo-)random numbers.

Monte Carlo Methods

- **Problem:** analytical evaluation of $\pi(f)$ costly/impossible.
- **Idea:**
  1. construct approximation $\hat{\pi}$ of π.
  2. estimate $\pi(f)$ by

$$\hat{\pi}(f) = \int_E f(x)\,\hat{\pi}(\mathrm{d}x).$$

- *Monte Carlo methods:* construction of $\hat{\pi}$ based on (computer-generated) (pseudo-)random numbers.

Monte Carlo Methods

- **Problem:** analytical evaluation of $\pi(f)$ costly/impossible.
- **Idea:**
  1. construct approximation $\hat{\pi}$ of π.
  2. estimate $\pi(f)$ by

$$\hat{\pi}(f) = \int_E f(x)\,\hat{\pi}(\mathrm{d}x).$$

- *Monte Carlo methods:* construction of $\hat{\pi}$ based on (computer-generated) (pseudo-)random numbers.

Monte Carlo Methods

- **Problem:** analytical evaluation of $\pi(f)$ costly/impossible.
- **Idea:**
  1. construct approximation $\hat{\pi}$ of π.
  2. estimate $\pi(f)$ by

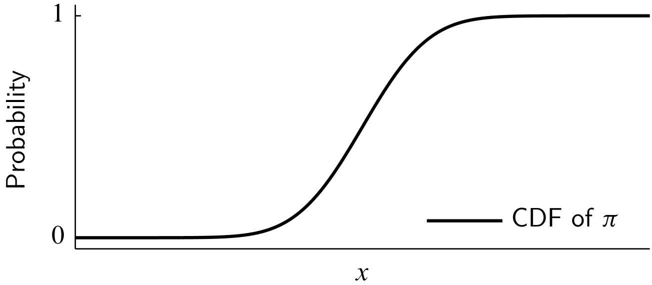
$$\hat{\pi}(f) = \int_E f(x)\,\hat{\pi}(\mathrm{d}x).$$

- *Monte Carlo methods:* construction of $\hat{\pi}$ based on (computer-generated) (pseudo-)random numbers.

Vanilla Monte Carlo

- Sample $X^1, \dots, X^N \overset{\text{IID}}{\sim} \pi$.
- Approximate $\pi(\mathrm{d}x)$ by the empirical measure:

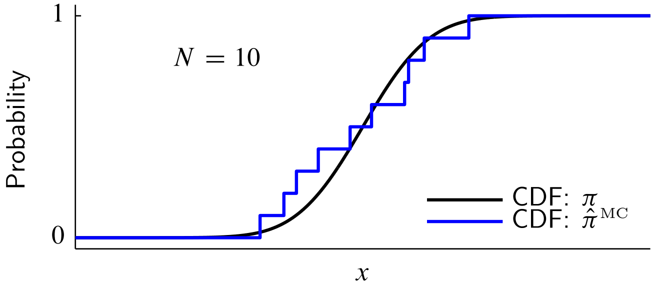
$$\hat{\pi}^{\mathrm{MC}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i}(\mathrm{d}x).$$



Vanilla Monte Carlo

- Sample $X^1, \dots, X^N \overset{\text{IID}}{\sim} \pi$.
- Approximate $\pi(\mathrm{d}x)$ by the empirical measure:

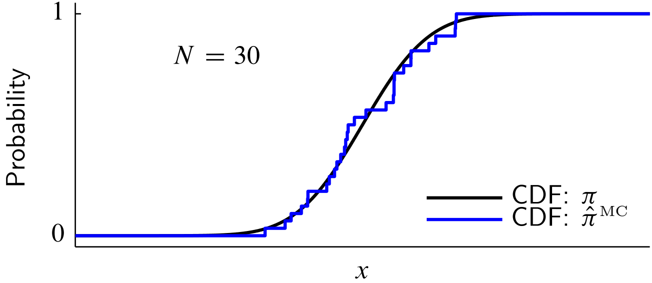
$$\hat{\pi}^{\mathrm{MC}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i}(\mathrm{d}x).$$



Vanilla Monte Carlo

- Sample $X^1, \ldots, X^N \overset{\text{IID}}{\sim} \pi$.
- Approximate $\pi(\mathrm{d}x)$ by the empirical measure:

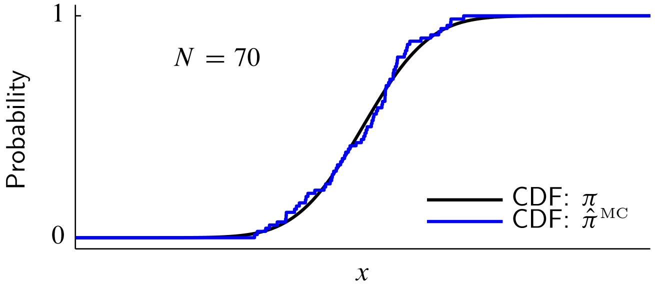
$$\hat{\pi}^{\text{MC}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i}(\mathrm{d}x).$$



Vanilla Monte Carlo

- Sample $X^1, \ldots, X^N \overset{\mathrm{IID}}{\sim} \pi$.
- Approximate $\pi(\mathrm{d}x)$ by the empirical measure:

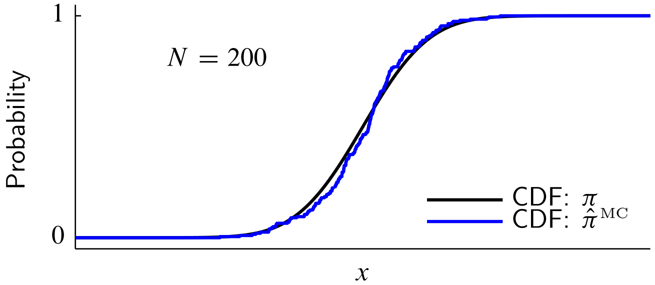
$$\hat{\pi}^{\mathrm{MC}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i}(\mathrm{d}x).$$



Vanilla Monte Carlo

- Sample $X^1, \dots, X^N \overset{\text{IID}}{\sim} \pi$.
- Approximate $\pi(\mathrm{d}x)$ by the empirical measure:

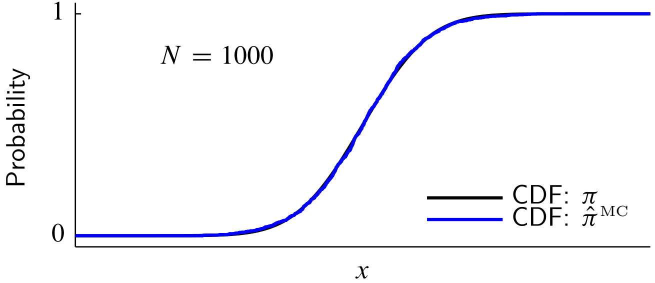
$$\hat{\pi}^{\mathrm{MC}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i}(\mathrm{d}x).$$



Vanilla Monte Carlo

- Sample $X^1, \ldots, X^N \overset{\text{IID}}{\sim} \pi$.
- Approximate $\pi(\mathrm{d}x)$ by the empirical measure:

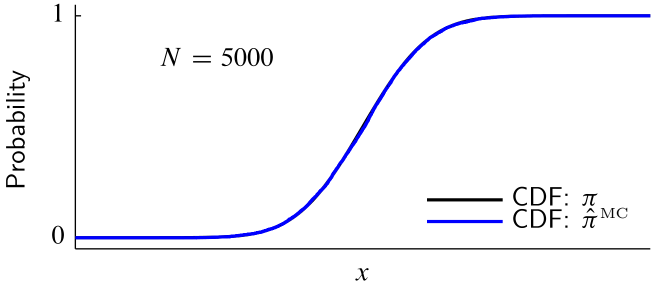
$$\hat{\pi}^{\text{MC}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i}(\mathrm{d}x).$$



Vanilla Monte Carlo

- Sample $X^1, \ldots, X^N \overset{\text{IID}}{\sim} \pi$.
- Approximate $\pi(\mathrm{d}x)$ by the empirical measure:

$$\hat{\pi}^{\mathrm{MC}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i}(\mathrm{d}x).$$



Vanilla Monte Carlo, continued

- Estimate $\pi(f) = \mathbf{E}[f(X)]$ by

$$\hat{\pi}^{\mathrm{MC}}(f) = \int_E f(x)\, \hat{\pi}^{\mathrm{MC}}(\mathrm{d}x) = \frac{1}{N}\sum_{i=1}^{N} f(X^i).$$

- Unbiased and consistent.
- Monte Carlo methods are best viewed as simulation techniques for *approximating measures.*

Vanilla Monte Carlo, continued

- Estimate $\pi(f) = \mathbf{E}[f(X)]$ by

$$\begin{aligned}\hat{\pi}^{\mathrm{MC}}(f) &= \int_E f(x)\,\hat{\pi}^{\mathrm{MC}}(\mathrm{d}x) \\ &= \frac{1}{N}\sum_{i=1}^{N} f(X^i).\end{aligned}$$

- Unbiased and consistent.
- Monte Carlo methods are best viewed as simulation techniques for *approximating measures*.

Vanilla Monte Carlo, continued

- Estimate $\pi(f) = \mathbf{E}[f(X)]$ by

$$\hat{\pi}^{\mathrm{MC}}(f) = \int_E f(x)\, \hat{\pi}^{\mathrm{MC}}(\mathrm{d}x) = \frac{1}{N}\sum_{i=1}^{N} f(X^i).$$

- Unbiased and consistent.
- Monte Carlo methods are best viewed as simulation techniques for *approximating measures.*

Motivation

Setup

Assume that

$$\pi(x) = \frac{\gamma(x)}{Z}$$

with *normalising constant* $Z = \int_E \gamma(x)\,\mathrm{d}x$, but

- we cannot sample from π.
- Z is unknown (i.e. we can evaluate γ but not π)

Example (Bayesian inference)

Let $\pi(x) := p(x|y)$ for some data y, then often,

- we can evaluate $\gamma(x) = p(x, y)$,
- but $Z = p(y) = \int_E p(x, y)\,\mathrm{d}x$ is typically intractable.

Motivation

Setup

Assume that

$$\pi(x) = \frac{\gamma(x)}{Z}$$

with *normalising constant* $Z = \int_E \gamma(x)\,\mathrm{d}x$, but

- we cannot sample from π.
- Z is unknown (i.e. we can evaluate γ but not π)

Example (Bayesian inference)

Let $\pi(x) := p(x|y)$ for some data y, then often,

- we can evaluate $\gamma(x) = p(x, y)$,
- but $Z = p(y) = \int_E p(x, y)\,\mathrm{d}x$ is typically intractable.

Motivation

Setup

Assume that

$$\pi(x) = \frac{\gamma(x)}{Z}$$

with *normalising constant* $Z = \int_E \gamma(x)\,\mathrm{d}x$, but

- we cannot sample from π.
- Z is unknown (i.e. we can evaluate γ but not π)

Example (Bayesian inference)

Let $\pi(x) := p(x|y)$ for some data y, then often,

- we can evaluate $\gamma(x) = p(x, y)$,
- but $Z = p(y) = \int_E p(x, y)\,\mathrm{d}x$ is typically intractable.

Motivation

Setup

Assume that

$$\pi(x) = \frac{\gamma(x)}{Z}$$

with *normalising constant* $Z = \int_E \gamma(x)\,\mathrm{d}x$, but

- we cannot sample from π.
- Z is unknown (i.e. we can evaluate γ but not π)

Example (Bayesian inference)

Let $\pi(x) := p(x|y)$ for some data y, then often,

- we can evaluate $\gamma(x) = p(x, y)$,
- but $Z = p(y) = \int_E p(x, y)\,\mathrm{d}x$ is typically intractable.

Motivation

Setup

Assume that

$$\pi(x) = \frac{\gamma(x)}{Z}$$

with *normalising constant* $Z = \int_E \gamma(x)\,\mathrm{d}x$, but

- we cannot sample from π.
- Z is unknown (i.e. we can evaluate γ but not π)

Example (Bayesian inference)

Let $\pi(x) := p(x|y)$ for some data y, then often,

- we can evaluate $\gamma(x) = p(x, y)$,
- but $Z = p(y) = \int_E p(x, y)\,\mathrm{d}x$ is typically intractable.

Motivation

Setup

Assume that

$$\pi(x) = \frac{\gamma(x)}{Z}$$

with *normalising constant* $Z = \int_E \gamma(x)\,\mathrm{d}x$, but

- we cannot sample from π.
- Z is unknown (i.e. we can evaluate γ but not π)

Example (Bayesian inference)

Let $\pi(x) := p(x|y)$ for some data y, then often,

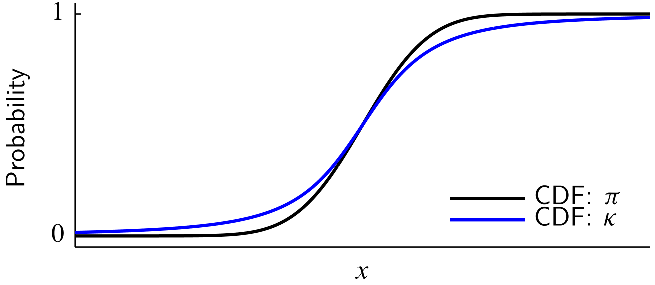
- we can evaluate $\gamma(x) = p(x, y)$,
- but $Z = p(y) = \int_E p(x, y)\,\mathrm{d}x$ is typically intractable.

Importance Sampling

Assume that κ is another distribution s.t. $\pi \ll \kappa$.

1. **Sample** $X^1, \ldots, X^N \overset{\mathrm{IID}}{\sim} \kappa$.
2. **Approximate** π by the *weighted* empirical measure:

$$\hat{\pi}^{\mathrm{IS}}(\mathrm{d}x) := \sum_{i=1}^{N} W^i \delta_{X^i}(\mathrm{d}x).$$

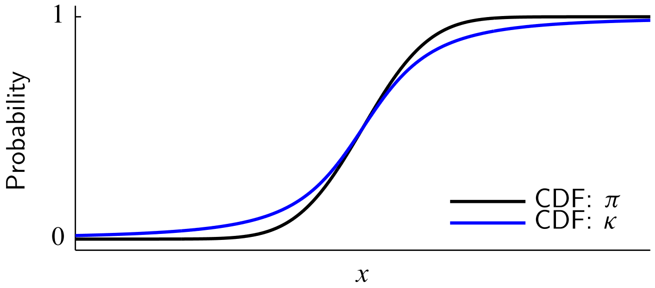


Importance Sampling

Assume that κ is another distribution s.t. $\pi \ll \kappa$.

1. **Sample** $X^1, \ldots, X^N \overset{\text{IID}}{\sim} \kappa$.
2. **Approximate** π by the *weighted* empirical measure:

$$\hat{\pi}^{\text{IS}}(\mathrm{d}x) := \sum_{i=1}^{N} W^i \delta_{X^i}(\mathrm{d}x).$$

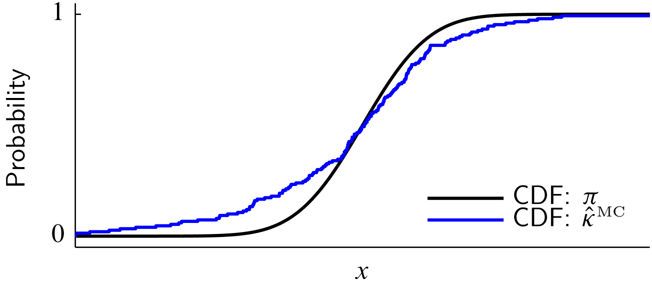


Importance Sampling

Assume that κ is another distribution s.t. $\pi \ll \kappa$.

1. **Sample** $X^1, \ldots, X^N \overset{\text{IID}}{\sim} \kappa$.
2. **Approximate** π by the *weighted* empirical measure:

$$\hat{\pi}^{\text{IS}}(\mathrm{d}x) := \sum_{i=1}^{N} W^i \delta_{X^i}(\mathrm{d}x).$$

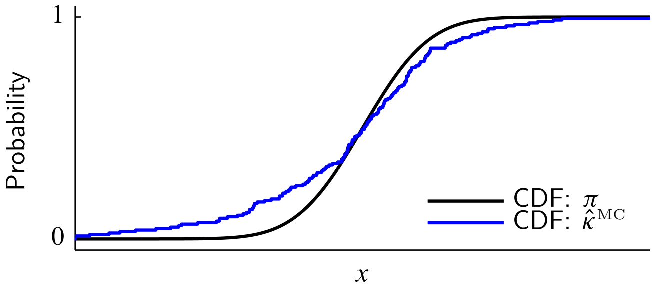


Importance Sampling

Assume that κ is another distribution s.t. $\pi \ll \kappa$.

1. **Sample** $X^1, \ldots, X^N \overset{\text{IID}}{\sim} \kappa$.
2. **Approximate** π by the *weighted* empirical measure:

$$\hat{\pi}^{\text{IS}}(\mathrm{d}x) := \sum_{i=1}^{N} W^i \delta_{X^i}(\mathrm{d}x).$$

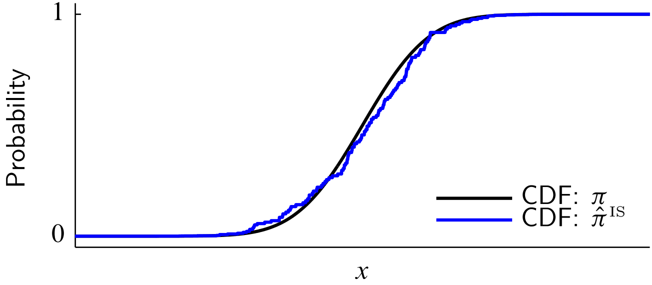


Importance Sampling

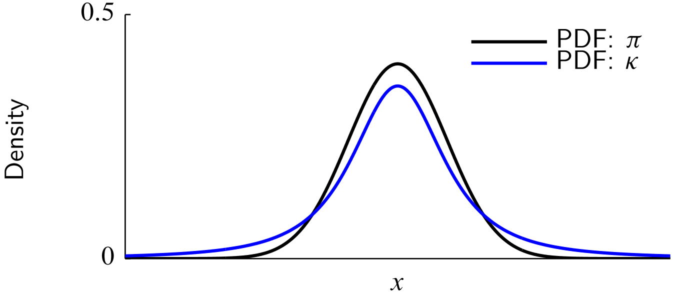
Assume that κ is another distribution s.t. $\pi \ll \kappa$.

1. **Sample** $X^1, \ldots, X^N \overset{\text{IID}}{\sim} \kappa$.
2. **Approximate** π by the *weighted* empirical measure:

$$\hat{\pi}^{\text{IS}}(\mathrm{d}x) := \sum_{i=1}^{N} W^i \delta_{X^i}(\mathrm{d}x).$$



Constructing the Importance Weights



Constructing the Importance Weights, continued

Want to set $W^i \propto \frac{\mathrm{d}\pi}{\mathrm{d}\kappa}(X^i) = \frac{\pi(X^i)}{\kappa(X^i)}$.

Problem: can only evaluate $G(X^i) := \frac{\mathrm{d}\gamma}{\mathrm{d}\kappa}(X^i) = \frac{\gamma(X^i)}{\kappa(X^i)}$.

Solution: approximate γ and Z separately, i.e.

1. approximate $\gamma(\mathrm{d}x)$ by

$$\hat{\gamma}^{\mathrm{IS,u}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=1}^{N} G(X^i) \delta_{X^i}(\mathrm{d}x)$$

2. approximate $Z = \int_E G(x)\kappa(\mathrm{d}x)$ by

$$\hat{Z} := \underbrace{\hat{\kappa}^{\mathrm{MC}}(G)}_{\substack{\text{'vanilla' Monte Carlo} \\ \text{estimate of } \kappa(G)}} = \frac{1}{N} \sum_{i=1}^{N} G(X^i).$$

Constructing the Importance Weights, continued

Want to set $W^i \propto \frac{\mathrm{d}\pi}{\mathrm{d}\kappa}(X^i) = \frac{\pi(X^i)}{\kappa(X^i)}$.

Problem: can only evaluate $G(X^i) := \frac{\mathrm{d}\gamma}{\mathrm{d}\kappa}(X^i) = \frac{\gamma(X^i)}{\kappa(X^i)}$.

Solution: approximate γ and Z separately, i.e.

1. approximate $\gamma(\mathrm{d}x)$ by

$$\hat{\gamma}^{\mathrm{IS,u}}(\mathrm{d}x) := \frac{1}{N}\sum_{i=1}^{N} G(X^i)\delta_{X^i}(\mathrm{d}x)$$

2. approximate $Z = \int_E G(x)\kappa(\mathrm{d}x)$ by

$$\hat{Z} := \underbrace{\hat{\kappa}^{\mathrm{MC}}(G)}_{\substack{\text{'vanilla' Monte Carlo} \\ \text{estimate of } \kappa(G)}} = \frac{1}{N}\sum_{i=1}^{N} G(X^i).$$

Constructing the Importance Weights, continued

Want to set $W^i \propto \frac{\mathrm{d}\pi}{\mathrm{d}\kappa}(X^i) = \frac{\pi(X^i)}{\kappa(X^i)}$.

Problem: can only evaluate $G(X^i) := \frac{\mathrm{d}\gamma}{\mathrm{d}\kappa}(X^i) = \frac{\gamma(X^i)}{\kappa(X^i)}$.

Solution: approximate γ and Z separately, i.e.

1. approximate $\gamma(\mathrm{d}x)$ by

$$\hat{\gamma}^{\mathrm{IS,u}}(\mathrm{d}x) := \frac{1}{N}\sum_{i=1}^{N} G(X^i)\delta_{X^i}(\mathrm{d}x)$$

2. approximate $Z = \int_E G(x)\kappa(\mathrm{d}x)$ by

$$\hat{Z} := \underbrace{\hat{\kappa}^{\mathrm{MC}}(G)}_{\substack{\text{'vanilla' Monte Carlo} \\ \text{estimate of } \kappa(G)}} = \frac{1}{N}\sum_{i=1}^{N} G(X^i).$$

Constructing the Importance Weights, continued

Want to set $W^i \propto \frac{\mathrm{d}\pi}{\mathrm{d}\kappa}(X^i) = \frac{\pi(X^i)}{\kappa(X^i)}$.

Problem: can only evaluate $G(X^i) := \frac{\mathrm{d}\gamma}{\mathrm{d}\kappa}(X^i) = \frac{\gamma(X^i)}{\kappa(X^i)}$.

Solution: approximate γ and Z separately, i.e.

1. approximate $\gamma(\mathrm{d}x)$ by

$$\hat{\gamma}^{\mathrm{IS,u}}(\mathrm{d}x) := \frac{1}{N}\sum_{i=1}^{N} G(X^i)\delta_{X^i}(\mathrm{d}x)$$

2. approximate $Z = \int_E G(x)\kappa(\mathrm{d}x)$ by

$$\hat{Z} := \underbrace{\hat{\kappa}^{\mathrm{MC}}(G)}_{\substack{\text{'vanilla' Monte Carlo} \\ \text{estimate of } \kappa(G)}} = \frac{1}{N}\sum_{i=1}^{N} G(X^i).$$

Constructing the Importance Weights, continued

Want to set $W^i \propto \frac{\mathrm{d}\pi}{\mathrm{d}\kappa}(X^i) = \frac{\pi(X^i)}{\kappa(X^i)}$.

Problem: can only evaluate $G(X^i) := \frac{\mathrm{d}\gamma}{\mathrm{d}\kappa}(X^i) = \frac{\gamma(X^i)}{\kappa(X^i)}$.

Solution: approximate γ and Z separately, i.e.

1. approximate $\gamma(\mathrm{d}x)$ by

$$\hat{\gamma}^{\mathrm{IS,u}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=1}^{N} G(X^i)\delta_{X^i}(\mathrm{d}x)$$

2. approximate $Z = \int_E G(x)\kappa(\mathrm{d}x)$ by

$$\widehat{Z} := \underbrace{\hat{\kappa}^{\mathrm{MC}}(G)}_{\substack{\text{`vanilla' Monte Carlo} \\ \text{estimate of } \kappa(G)}} = \frac{1}{N} \sum_{i=1}^{N} G(X^i).$$

Constructing the Importance Weights, continued

3. approximate $\pi(\mathrm{d}x)$ by

$$\hat{\pi}^{\mathrm{IS}}(\mathrm{d}x) := \hat{\gamma}^{\mathrm{IS,u}}(\mathrm{d}x)/\hat{Z} = \sum_{i=1}^{N} W^i \delta_{X^i}(\mathrm{d}x),$$

where

$$W^i := \frac{G(X^i)}{\sum_{j=1}^{N} G(X^j)}.$$

Importance sampling yields unbiased (and consistent) estimates of normalising constants!

Constructing the Importance Weights, continued

3. approximate $\pi(\mathrm{d}x)$ by

$$\hat{\pi}^{\mathrm{IS}}(\mathrm{d}x) := \hat{\gamma}^{\mathrm{IS,u}}(\mathrm{d}x)/\hat{Z}$$
$$= \sum_{i=1}^{N} W^i \delta_{X^i}(\mathrm{d}x),$$

where

$$W^i := \frac{G(X^i)}{\sum_{j=1}^{N} G(X^j)}.$$

Importance sampling yields unbiased (and consistent) estimates of normalising constants!

Markov Chain Monte Carlo Methods

Let K be a π-invariant ergodic Markov kernel.

1. **simulate** a Markov chain with transitions K, i.e. sample

$$X^1 \sim \kappa(\cdot),\ X^2 \sim K(\cdot|X^1),\ X^3 \sim K(\cdot|X^2),\ \ldots$$

2. **approximate** $\pi(\mathrm{d}x)$ by

$$\hat{\pi}^{\mathrm{MCMC}}(\mathrm{d}x) := \frac{1}{N} \sum_{i=R+1}^{R+N} \delta_{X^i}(\mathrm{d}x).$$

after a suitable burn-in time R.

Markov Chain Monte Carlo Methods

Let K be a π-invariant ergodic Markov kernel.

1. **simulate** a Markov chain with transitions K, i.e. sample

$$X^1 \sim \kappa(\cdot),\ X^2 \sim K(\cdot|X^1),\ X^3 \sim K(\cdot|X^2),\ \ldots$$

2. **approximate** $\pi(\mathrm{d}x)$ by

$$\hat{\pi}^{\mathrm{MCMC}}(\mathrm{d}x) := \frac{1}{N}\sum_{i=R+1}^{R+N} \delta_{X^i}(\mathrm{d}x).$$

after a suitable burn-in time R.

Markov Chain Monte Carlo Methods

Let K be a π-invariant ergodic Markov kernel.

1. **simulate** a Markov chain with transitions K, i.e. sample

$$X^1 \sim \kappa(\cdot),\ X^2 \sim K(\cdot|X^1),\ X^3 \sim K(\cdot|X^2),\ \ldots$$

2. **approximate** $\pi(\mathrm{d}x)$ by

$$\hat{\pi}^{\mathrm{MCMC}}(\mathrm{d}x) := \frac{1}{N}\sum_{i=R+1}^{R+N} \delta_{X^i}(\mathrm{d}x).$$

after a suitable burn-in time R.

Constructing the Markov Kernel K

Example (Gibbs sampler)

Let E be d-dimensional. The standard Gibbs sampler cycles through all full conditional distributions (under π), i.e.

$$K(\mathrm{d}x^i|x^{i-1}) := \prod_{j=1}^{d} \pi(\mathrm{d}x_j^i|x_{1:j-1}^i, x_{j+1:d}^{i-1})$$

Partially-collapsed Gibbs sampler (Van Dyk & Park, 2008):

- often no need to sample from *full* conditionals.
- instead, sample from conditionals under a *marginal of* π

Constructing the Markov Kernel K

Example (Gibbs sampler)

Let E be d-dimensional. The standard Gibbs sampler cycles through all full conditional distributions (under π), i.e.

$$K(\mathrm{d}x^i|x^{i-1}) := \prod_{j=1}^{d} \pi(\mathrm{d}x_j^i|x_{1:j-1}^i, x_{j+1:d}^{i-1})$$

Partially-collapsed Gibbs sampler (Van Dyk & Park, 2008):

- often no need to sample from *full* conditionals.
- instead, sample from conditionals under a *marginal of* π

Constructing the Markov Kernel K

Example (Gibbs sampler)

Let E be d-dimensional. The standard Gibbs sampler cycles through all full conditional distributions (under π), i.e.

$$K(\mathrm{d}x^i | x^{i-1}) := \prod_{j=1}^{d} \pi(\mathrm{d}x^i_j | x^i_{1:j-1}, x^{i-1}_{j+1:d})$$

Partially-collapsed Gibbs sampler (Van Dyk & Park, 2008):

- often no need to sample from *full* conditionals.
- instead, sample from conditionals under a *marginal of* π

Constructing the Markov Kernel K

Example (Metropolis–Hastings algorithm)

1. **sample** $X^\star \sim Q(\cdot|X^{i-1})$ (where Q is *not* π-invariant)
2. **accept** $X^i := X^\star$ with probability

$$\alpha(X^\star|X^i) := 1 \wedge \frac{\gamma(X^\star)Q(X^i|X^\star)}{\gamma(X^i)Q(X^\star|X^i)},$$

   otherwise, set $X^i := X^{i-1}$.

Thus, K has the form

$$K(\mathrm{d}x^i|x^{i-1}) := \alpha(x^i|x^{i-1})Q(\mathrm{d}x^i|x^{i-1}) + r(x^{i-1})\delta_x(\mathrm{d}x^i),$$

where $r(x) := 1 - \int_E \alpha(z|x)Q(\mathrm{d}z|x)$.

Constructing the Markov Kernel K

Example (Metropolis–Hastings algorithm)

1. **sample** $X^\star \sim Q(\cdot|X^{i-1})$ (where Q is *not* π-invariant)
2. **accept** $X^i := X^\star$ with probability

$$\alpha(X^\star|X^i) := 1 \wedge \frac{\gamma(X^\star)Q(X^i|X^\star)}{\gamma(X^i)Q(X^\star|X^i)},$$

   otherwise, set $X^i := X^{i-1}$.

Thus, K has the form

$$K(\mathrm{d}x^i|x^{i-1}) := \alpha(x^i|x^{i-1})Q(\mathrm{d}x^i|x^{i-1}) + r(x^{i-1})\delta_x(\mathrm{d}x^i),$$

where $r(x) := 1 - \int_E \alpha(z|x)Q(\mathrm{d}z|x)$.

Constructing the Markov Kernel K

Example (Metropolis–Hastings algorithm)

1. **sample** $X^\star \sim Q(\cdot|X^{i-1})$ (where Q is *not* π-invariant)
2. **accept** $X^i := X^\star$ with probability

$$\alpha(X^\star|X^i) := 1 \wedge \frac{\gamma(X^\star)Q(X^i|X^\star)}{\gamma(X^i)Q(X^\star|X^i)},$$

otherwise, set $X^i := X^{i-1}$.

Thus, K has the form

$$K(\mathrm{d}x^i|x^{i-1}) := \alpha(x^i|x^{i-1})Q(\mathrm{d}x^i|x^{i-1}) + r(x^{i-1})\delta_x(\mathrm{d}x^i),$$

where $r(x) := 1 - \int_E \alpha(z|x)Q(\mathrm{d}z|x)$.

State-Space Extension Tricks

- Want to target $\tilde{\pi}(\theta) = \tilde{\gamma}(\theta)/Z$, for $Z > 0$.
- What if we *cannot* evaluate $\tilde{\gamma}(\theta)$? (needed for IS/MCMC).
- **Idea:**
  1. instead, target $\pi(\theta, x) = \gamma(\theta, x)/Z$, s.t.
     i. $\pi(\theta, x)$ admits $\tilde{\pi}(\theta)$ as a marginal,
     ii. $\gamma(\theta, x)$ can be evaluated.
  2. construct IS/MCMC approximation $\hat{\pi}$ of $\pi = \gamma/Z$.
  3. approximate $\tilde{\pi}$ by θ-marginal of $\hat{\pi}$.
- Many names for this: *state-space extension, auxiliary-variable construction, data augmentation, . . .*

State-Space Extension Tricks

- Want to target $\tilde{\pi}(\theta) = \tilde{\gamma}(\theta)/Z$, for $Z > 0$.
- What if we *cannot* evaluate $\tilde{\gamma}(\theta)$? (needed for IS/MCMC).
- **Idea:**
  1. instead, target $\pi(\theta, x) = \gamma(\theta, x)/Z$, s.t.
     i. $\pi(\theta, x)$ admits $\tilde{\pi}(\theta)$ as a marginal,
     ii. $\gamma(\theta, x)$ can be evaluated.
  2. construct IS/MCMC approximation $\hat{\pi}$ of $\pi = \gamma/Z$.
  3. approximate $\tilde{\pi}$ by θ-marginal of $\hat{\pi}$.
- Many names for this: *state-space extension, auxiliary-variable construction, data augmentation, ...*

State-Space Extension Tricks

- Want to target $\tilde{\pi}(\theta) = \tilde{\gamma}(\theta)/Z$, for $Z > 0$.
- What if we *cannot* evaluate $\tilde{\gamma}(\theta)$? (needed for IS/MCMC).
- **Idea:**
  1. instead, target $\pi(\theta, x) = \gamma(\theta, x)/Z$, s.t.
     i. $\pi(\theta, x)$ admits $\tilde{\pi}(\theta)$ as a marginal,
     ii. $\gamma(\theta, x)$ can be evaluated.
  2. construct IS/MCMC approximation $\hat{\pi}$ of $\pi = \gamma/Z$.
  3. approximate $\tilde{\pi}$ by θ-marginal of $\hat{\pi}$.
- Many names for this: *state-space extension, auxiliary-variable construction, data augmentation, ...*

State-Space Extension Tricks

- Want to target $\tilde{\pi}(\theta) = \tilde{\gamma}(\theta)/Z$, for $Z > 0$.
- What if we *cannot* evaluate $\tilde{\gamma}(\theta)$? (needed for IS/MCMC).
- **Idea:**
  1. instead, target $\pi(\theta, x) = \gamma(\theta, x)/Z$, s.t.
     i. $\pi(\theta, x)$ admits $\tilde{\pi}(\theta)$ as a marginal,
     ii. $\gamma(\theta, x)$ can be evaluated.
  2. construct IS/MCMC approximation $\hat{\pi}$ of $\pi = \gamma/Z$.
  3. approximate $\tilde{\pi}$ by θ-marginal of $\hat{\pi}$.
- Many names for this: *state-space extension, auxiliary-variable construction, data augmentation, ...*

State-Space Extension Tricks

- Want to target $\tilde{\pi}(\theta) = \tilde{\gamma}(\theta)/Z$, for $Z > 0$.
- What if we *cannot* evaluate $\tilde{\gamma}(\theta)$? (needed for IS/MCMC).
- **Idea:**
  1. instead, target $\pi(\theta, x) = \gamma(\theta, x)/Z$, s.t.
     i. $\pi(\theta, x)$ admits $\tilde{\pi}(\theta)$ as a marginal,
     ii. $\gamma(\theta, x)$ can be evaluated.
  2. construct IS/MCMC approximation $\hat{\pi}$ of $\pi = \gamma/Z$.
  3. approximate $\tilde{\pi}$ by θ-marginal of $\hat{\pi}$.
- Many names for this: *state-space extension, auxiliary-variable construction, data augmentation, ...*

State-Space Extension Tricks

- Want to target $\tilde{\pi}(\theta) = \tilde{\gamma}(\theta)/Z$, for $Z > 0$.
- What if we *cannot* evaluate $\tilde{\gamma}(\theta)$? (needed for IS/MCMC).
- **Idea:**
  1. instead, target $\pi(\theta, x) = \gamma(\theta, x)/Z$, s.t.
     i. $\pi(\theta, x)$ admits $\tilde{\pi}(\theta)$ as a marginal,
     ii. $\gamma(\theta, x)$ can be evaluated.
  2. construct IS/MCMC approximation $\hat{\pi}$ of $\pi = \gamma/Z$.
  3. approximate $\tilde{\pi}$ by θ-marginal of $\hat{\pi}$.
- Many names for this: *state-space extension, auxiliary-variable construction, data augmentation, ...*

State-Space Extension Tricks

- Want to target $\tilde{\pi}(\theta) = \tilde{\gamma}(\theta)/Z$, for $Z > 0$.
- What if we *cannot* evaluate $\tilde{\gamma}(\theta)$? (needed for IS/MCMC).
- **Idea:**
  1. instead, target $\pi(\theta, x) = \gamma(\theta, x)/Z$, s.t.
     i. $\pi(\theta, x)$ admits $\tilde{\pi}(\theta)$ as a marginal,
     ii. $\gamma(\theta, x)$ can be evaluated.
  2. construct IS/MCMC approximation $\hat{\pi}$ of $\pi = \gamma/Z$.
  3. approximate $\tilde{\pi}$ by θ-marginal of $\hat{\pi}$.
- Many names for this: *state-space extension, auxiliary-variable construction, data augmentation, ...*

State-Space Extension Tricks

- Want to target $\tilde{\pi}(\theta) = \tilde{\gamma}(\theta)/Z$, for $Z > 0$.
- What if we *cannot* evaluate $\tilde{\gamma}(\theta)$? (needed for IS/MCMC).
- **Idea:**
  1. instead, target $\pi(\theta, x) = \gamma(\theta, x)/Z$, s.t.
     i. $\pi(\theta, x)$ admits $\tilde{\pi}(\theta)$ as a marginal,
     ii. $\gamma(\theta, x)$ can be evaluated.
  2. construct IS/MCMC approximation $\hat{\pi}$ of $\pi = \gamma/Z$.
  3. approximate $\tilde{\pi}$ by θ-marginal of $\hat{\pi}$.
- Many names for this: *state-space extension, auxiliary-variable construction, data augmentation, ...*

State-Space Extension Tricks, continued

Example (Hidden Markov model)

$X_0 \sim \mu^\theta$, and for $n \in \mathbb{N}$,

$$X_n \sim f^\theta(\cdot|X_{n-1}),$$
$$Y_n \sim g^\theta(\cdot|X_n),$$

where θ are some 'static' parameters.

Assume we are interested in $\tilde{\pi}(\theta) := p(\theta|y_{1:n})$.

- $\tilde{\gamma}(\theta) = p(\theta, y_{1:n})$ and $Z = p(y_{1:n})$ are intractable.
- but we can evaluate

$$\gamma(\theta, x_{0:n}, y_{1:n}) := p(\theta)\mu^\theta(x_0)\prod_{p=1}^{n} f^\theta(x_p|x_{p-1})g^\theta(y_p|x_p).$$

State-Space Extension Tricks, continued

Example (Hidden Markov model)

$X_0 \sim \mu^\theta$, and for $n \in \mathbb{N}$,

$$X_n \sim f^\theta(\cdot|X_{n-1}),$$
$$Y_n \sim g^\theta(\cdot|X_n),$$

where θ are some 'static' parameters.
Assume we are interested in $\tilde{\pi}(\theta) := p(\theta|y_{1:n})$.

- $\tilde{\gamma}(\theta) = p(\theta, y_{1:n})$ and $Z = p(y_{1:n})$ are intractable.
- but we can evaluate

$$\gamma(\theta, x_{0:n}, y_{1:n}) := p(\theta)\mu^\theta(x_0)\prod_{p=1}^{n} f^\theta(x_p|x_{p-1})g^\theta(y_p|x_p).$$

State-Space Extension Tricks, continued

Example (Hidden Markov model)

$X_0 \sim \mu^\theta$, and for $n \in \mathbb{N}$,

$$X_n \sim f^\theta(\cdot|X_{n-1}),$$
$$Y_n \sim g^\theta(\cdot|X_n),$$

where θ are some 'static' parameters.
Assume we are interested in $\tilde{\pi}(\theta) := p(\theta|y_{1:n})$.

- $\tilde{\gamma}(\theta) = p(\theta, y_{1:n})$ and $Z = p(y_{1:n})$ are intractable.
- but we can evaluate

$$\gamma(\theta, x_{0:n}, y_{1:n}) := p(\theta)\mu^\theta(x_0)\prod_{p=1}^{n} f^\theta(x_p|x_{p-1})g^\theta(y_p|x_p).$$

State-Space Extension Tricks, continued

Example (Hidden Markov model)

$X_0 \sim \mu^\theta$, and for $n \in \mathbb{N}$,

$$X_n \sim f^\theta(\cdot|X_{n-1}),$$
$$Y_n \sim g^\theta(\cdot|X_n),$$

where θ are some 'static' parameters.
Assume we are interested in $\tilde{\pi}(\theta) := p(\theta|y_{1:n})$.

- $\tilde{\gamma}(\theta) = p(\theta, y_{1:n})$ and $Z = p(y_{1:n})$ are intractable.
- but we can evaluate

$$\gamma(\theta, x_{0:n}, y_{1:n}) := p(\theta)\mu^\theta(x_0)\prod_{p=1}^{n} f^\theta(x_p|x_{p-1})g^\theta(y_p|x_p).$$

Target distributions

Setup

Want to target a sequence of *related* distributions $(\pi_n^\theta(x_{1:n}))_{n\in\mathbb{N}}$ which

- are defined on spaces $(E_n)_{n\in\mathbb{N}}$ of *increasing* dimension [will be relaxed later],
- have unknown normalising constants $(Z_n^\theta)_{n\in\mathbb{N}}$.

θ known [will be relaxed later].

Problem

IS/MCMC require new algorithm *for each* $n \in \mathbb{N}$.

Target distributions

Setup

Want to target a sequence of *related* distributions $(\pi_n^\theta(x_{1:n}))_{n\in\mathbb{N}}$ which

- are defined on spaces $(E_n)_{n\in\mathbb{N}}$ of *increasing* dimension [will be relaxed later],
- have unknown normalising constants $(Z_n^\theta)_{n\in\mathbb{N}}$.

θ known [will be relaxed later].

Problem

IS/MCMC require new algorithm *for each* $n \in \mathbb{N}$.

Target distributions

Setup

Want to target a sequence of *related* distributions $(\pi_n^\theta(x_{1:n}))_{n\in\mathbb{N}}$ which

- are defined on spaces $(E_n)_{n\in\mathbb{N}}$ of *increasing* dimension [will be relaxed later],
- have unknown normalising constants $(Z_n^\theta)_{n\in\mathbb{N}}$.

θ known [will be relaxed later].

Problem

IS/MCMC require new algorithm *for each* $n \in \mathbb{N}$.

Target distributions

Setup

Want to target a sequence of *related* distributions $(\pi_n^\theta(x_{1:n}))_{n\in\mathbb{N}}$ which

- are defined on spaces $(E_n)_{n\in\mathbb{N}}$ of *increasing* dimension [will be relaxed later],
- have unknown normalising constants $(Z_n^\theta)_{n\in\mathbb{N}}$.

θ known [will be relaxed later].

Problem

IS/MCMC require new algorithm *for each* $n \in \mathbb{N}$.

Target distributions

Setup

Want to target a sequence of *related* distributions $(\pi_n^\theta(x_{1:n}))_{n\in\mathbb{N}}$ which

- are defined on spaces $(E_n)_{n\in\mathbb{N}}$ of *increasing* dimension [will be relaxed later],
- have unknown normalising constants $(Z_n^\theta)_{n\in\mathbb{N}}$.

θ known [will be relaxed later].

Problem

IS/MCMC require new algorithm *for each $n \in \mathbb{N}$.*

Sequential Monte Carlo Methods

Sequential Monte Carlo (SMC): propagate weighted samples ('particles') $(X_{1:n}^i, W_n^{\theta,i})_{i\in\{1,\ldots,N\}}$ to construct

$$\hat{\pi}_n^\theta(\mathrm{d}x_{1:n}) := \sum_{i=1}^{N} W_n^{\theta,i}\delta_{X_{1:n}^i}(\mathrm{d}x_{1:n}).$$

SMC Algorithm

At time n, given $(X_{1:n-1}^i, W_{n-1}^{\theta,i})_{i\in\{1,\ldots,N\}}$,

1. **sample** $X_n^i \sim K_n^\theta(\cdot|X_{1:n-1}^i)$
   - this approximates $\pi_{n-1}^\theta(\mathrm{d}x_{1:n-1})K_n^\theta(\mathrm{d}x_n|x_{1:n-1})$,
2. **re-weight** particle paths $(X_{1:n}^i)_{i\in\{1,\ldots,N\}}$
   - to approximate $\pi_n^\theta(\mathrm{d}x_{1:n})$,
3. **resample**: reset weights, after
   - "multiplying" particles with large weights,
   - "killing off" paths with small weights.

Sequential Monte Carlo Methods

Sequential Monte Carlo (SMC): propagate weighted samples ('particles') $(X^i_{1:n}, W^{\theta,i}_n)_{i\in\{1,\dots,N\}}$ to construct

$$\hat{\pi}^\theta_n(\mathrm{d}x_{1:n}) := \sum_{i=1}^N W^{\theta,i}_n \delta_{X^i_{1:n}}(\mathrm{d}x_{1:n}).$$

SMC Algorithm

At time n, given $(X^i_{1:n-1}, W^{\theta,i}_{n-1})_{i\in\{1,\dots,N\}}$,

1. **sample** $X^i_n \sim K^\theta_n(\cdot|X^i_{1:n-1})$
   - this approximates $\pi^\theta_{n-1}(\mathrm{d}x_{1:n-1})K^\theta_n(\mathrm{d}x_n|x_{1:n-1})$,
2. **re-weight** particle paths $(X^i_{1:n})_{i\in\{1,\dots,N\}}$
   - to approximate $\pi^\theta_n(\mathrm{d}x_{1:n})$,
3. **resample**: reset weights, after
   - "multiplying" particles with large weights,
   - "killing off" paths with small weights.

Sequential Monte Carlo Methods

Sequential Monte Carlo (SMC): propagate weighted samples ('particles') $(X^i_{1:n}, W^{\theta,i}_n)_{i\in\{1,\dots,N\}}$ to construct

$$\hat{\pi}^\theta_n(\mathrm{d}x_{1:n}) := \sum_{i=1}^N W^{\theta,i}_n \delta_{X^i_{1:n}}(\mathrm{d}x_{1:n}).$$

SMC Algorithm

At time n, given $(X^i_{1:n-1}, W^{\theta,i}_{n-1})_{i\in\{1,\dots,N\}}$,

1. **sample** $X^i_n \sim K^\theta_n(\,\cdot\,|X^i_{1:n-1})$
   - this approximates $\pi^\theta_{n-1}(\mathrm{d}x_{1:n-1})K^\theta_n(\mathrm{d}x_n|x_{1:n-1})$,
2. **re-weight** particle paths $(X^i_{1:n})_{i\in\{1,\dots,N\}}$
   - to approximate $\pi^\theta_n(\mathrm{d}x_{1:n})$,
3. **resample**: reset weights, after
   - "multiplying" particles with large weights,
   - "killing off" paths with small weights.

Sequential Monte Carlo Methods

Sequential Monte Carlo (SMC): propagate weighted samples ('particles') $(X_{1:n}^i, W_n^{\theta,i})_{i\in\{1,\dots,N\}}$ to construct

$$\hat{\pi}_n^\theta(\mathrm{d}x_{1:n}) := \sum_{i=1}^N W_n^{\theta,i}\delta_{X_{1:n}^i}(\mathrm{d}x_{1:n}).$$

SMC Algorithm

At time n, given $(X_{1:n-1}^i, W_{n-1}^{\theta,i})_{i\in\{1,\dots,N\}}$,

1. **sample** $X_n^i \sim K_n^\theta(\,\cdot\,|X_{1:n-1}^i)$
   - this approximates $\pi_{n-1}^\theta(\mathrm{d}x_{1:n-1})K_n^\theta(\mathrm{d}x_n|x_{1:n-1})$,
2. **re-weight** particle paths $(X_{1:n}^i)_{i\in\{1,\dots,N\}}$
   - to approximate $\pi_n^\theta(\mathrm{d}x_{1:n})$,
3. **resample**: reset weights, after
   - "multiplying" particles with large weights,
   - "killing off" paths with small weights.

Sequential Monte Carlo Methods, continued

Particle filters: SMC methods applied to the filtering problem.

Almost all SMC methods, e.g.

- auxiliary particle filters (Pitt & Shepard, 1999),
- block sampling (Doucet, Briers & Sénécal, 2006),
- resample–move algorithms (Gilks & Berzuini, 2001),
- SMC samplers (Del Moral, Doucet & Jasra, 2006),
- discrete state-space particle filters (Fearnhead, 1998),
- ...

are special cases of this algorithm!

Sequential Monte Carlo Methods, continued

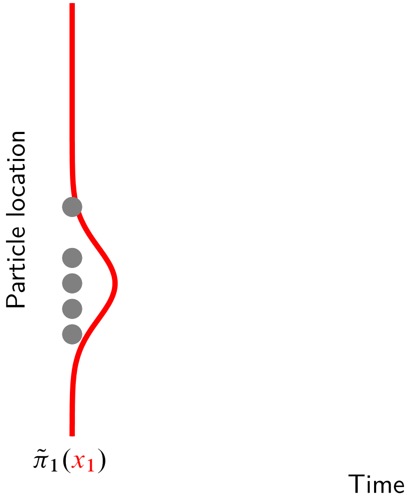
Particle filters: SMC methods applied to the filtering problem.

Almost all SMC methods, e.g.

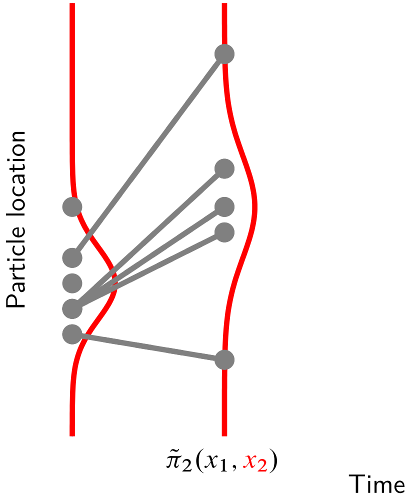
- auxiliary particle filters (Pitt & Shepard, 1999),
- block sampling (Doucet, Briers & Sénécal, 2006),
- resample–move algorithms (Gilks & Berzuini, 2001),
- SMC samplers (Del Moral, Doucet & Jasra, 2006),
- discrete state-space particle filters (Fearnhead, 1998),
- ...

are special cases of this algorithm!

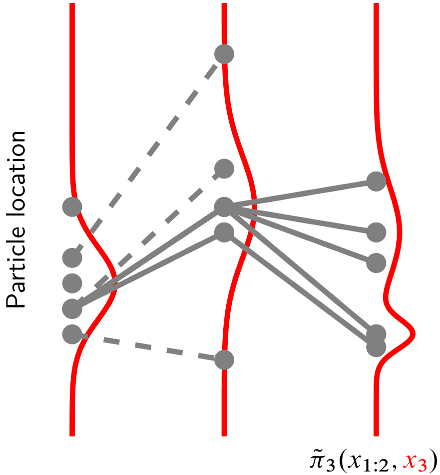
SMC: Sketch



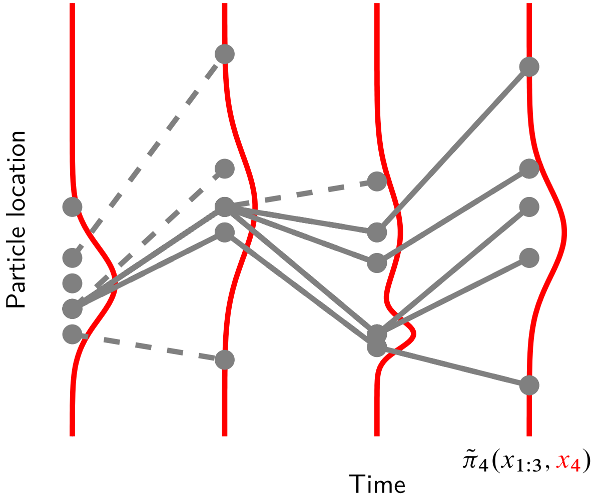
SMC: Sketch



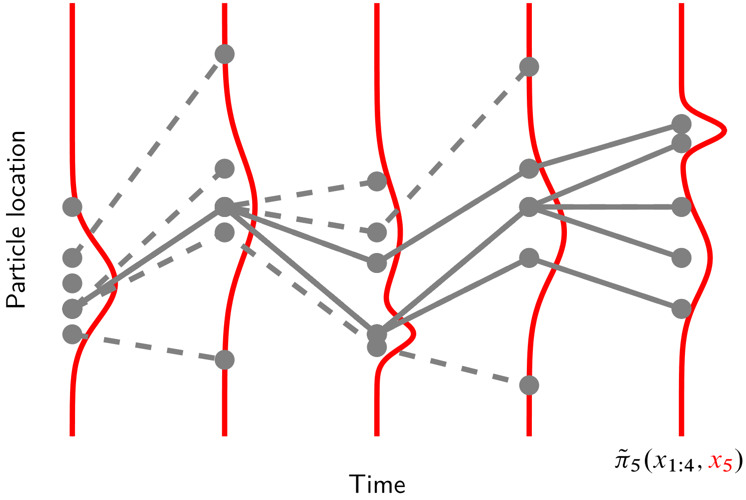
SMC: Sketch



SMC: Sketch



SMC: Sketch



Sample Degeneracy

- Also known as *sample impoverishment*, or *path degeneracy.*
- Eventually, all particles share a common ancestor.
- Thus, SMC methods rely on an ergodicity ('forgetting') property of $(\pi_n^\theta)_{n\in\mathbb{N}}$

Sample Degeneracy

- Also known as *sample impoverishment*, or *path degeneracy.*
- Eventually, all particles share a common ancestor.
- Thus, SMC methods rely on an ergodicity ('forgetting') property of $(\pi_n^\theta)_{n\in\mathbb{N}}$

Sample Degeneracy

- Also known as *sample impoverishment*, or *path degeneracy.*
- Eventually, all particles share a common ancestor.
- Thus, SMC methods rely on an ergodicity ('forgetting') property of $(\pi_n^\theta)_{n\in\mathbb{N}}$

Sample Degeneracy, continued

Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,

  $\longrightarrow$ approximated by a diverse set of particles.

- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,

  $\longrightarrow$ often approximated by a single particle.

Sample Degeneracy, continued

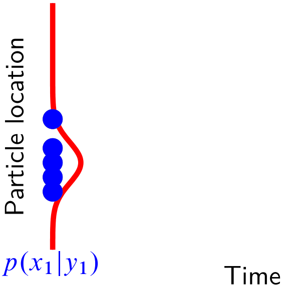
Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,

  $\longrightarrow$ approximated by a diverse set of particles.
- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,

  $\longrightarrow$ often approximated by a single particle.

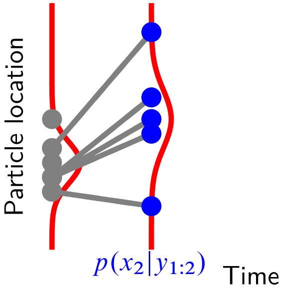


Sample Degeneracy, continued

Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,
  $\longrightarrow$ approximated by a diverse set of particles.
- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,
  $\longrightarrow$ often approximated by a single particle.

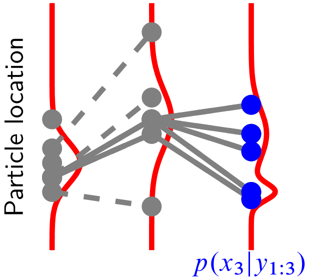


Sample Degeneracy, continued

Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,
  $\longrightarrow$ approximated by a diverse set of particles.
- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,
  $\longrightarrow$ often approximated by a single particle.



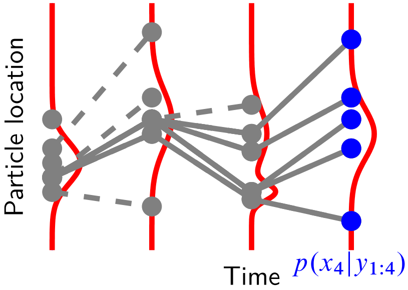
Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,

  $\longrightarrow$ approximated by a diverse set of particles.
- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,

  $\longrightarrow$ often approximated by a single particle.



Sample Degeneracy, continued

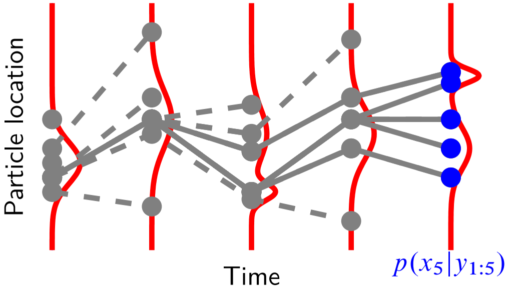
Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,

  $\longrightarrow$ approximated by a diverse set of particles.
- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,

  $\longrightarrow$ often approximated by a single particle.



Sample Degeneracy, continued

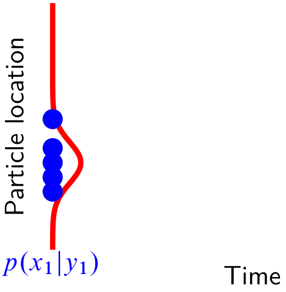
Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,

  $\longrightarrow$ approximated by a diverse set of particles.
- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,

  $\longrightarrow$ often approximated by a single particle.

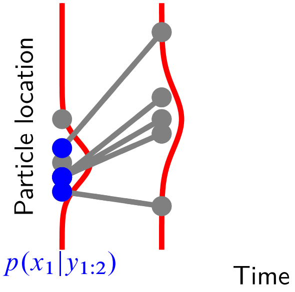


Sample Degeneracy, continued

Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,
  $\longrightarrow$ approximated by a diverse set of particles.
- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,
  $\longrightarrow$ often approximated by a single particle.

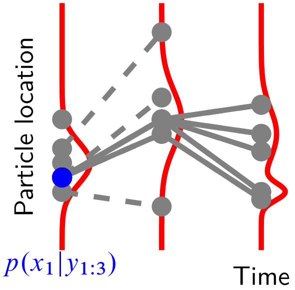


Sample Degeneracy, continued

Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,
  $\longrightarrow$ approximated by a diverse set of particles.
- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,
  $\longrightarrow$ often approximated by a single particle.



Sample Degeneracy, continued

Hidden Markov model, continued

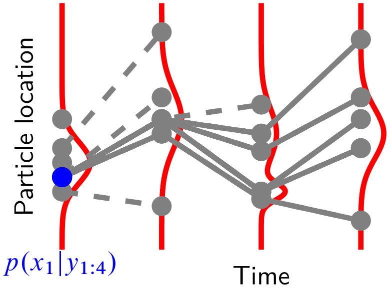
Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,

  $\longrightarrow$ approximated by a diverse set of particles.

- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,

  $\longrightarrow$ often approximated by a single particle.

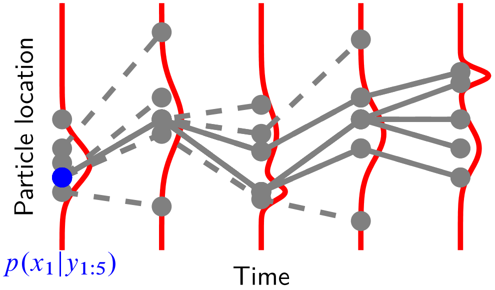


Sample Degeneracy, continued

Hidden Markov model, continued

Let $\pi_n^\theta(x_{1:n}) := p(x_{1:n}|\theta, y_{1:n})$ then we are often interested in

- **filtering** distributions: $\pi_n^\theta(x_n) = p(x_n|\theta, y_{1:n})$,
  $\longrightarrow$ approximated by a diverse set of particles.
- **smoothing** distributions: $\pi_n^\theta(x_k) = p(x_k|\theta, y_{1:n})$,
  $\longrightarrow$ often approximated by a single particle.



Alleviating the Degeneracy

- Use residual/stratified/systematic resampling
  $\longrightarrow$ avoid multinomial resampling!
- Only resample when necessary.
- Devise better proposal kernels K_n^θ.
  - e.g. avoid $K_n^\theta(\cdot|x_{n-1}) := f^\theta(\cdot|x_{n-1})$ in HMMs ('bootstrap' filter),
  - better: make use of y_n when sampling X_n^i.

Alleviating the Degeneracy

- Use residual/stratified/systematic resampling
  $\longrightarrow$ avoid multinomial resampling!
- Only resample when necessary.
- Devise better proposal kernels K_n^θ.
  - e.g. avoid $K_n^\theta(\cdot|x_{n-1}) := f^\theta(\cdot|x_{n-1})$ in HMMs ('bootstrap' filter),
  - better: make use of y_n when sampling X_n^i.

Alleviating the Degeneracy

- Use residual/stratified/systematic resampling
  $\longrightarrow$ avoid multinomial resampling!
- Only resample when necessary.
- Devise better proposal kernels K_n^θ.
  - e.g. avoid $K_n^\theta(\cdot|x_{n-1}) := f^\theta(\cdot|x_{n-1})$ in HMMs ('bootstrap' filter),
  - better: make use of y_n when sampling X_n^i.

Alleviating the Degeneracy

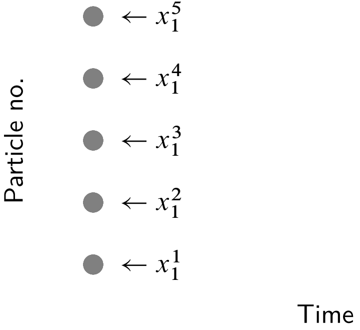
- Use residual/stratified/systematic resampling
  $\longrightarrow$ avoid multinomial resampling!
- Only resample when necessary.
- Devise better proposal kernels K_n^θ.
  - e.g. avoid $K_n^\theta(\cdot|x_{n-1}) := f^\theta(\cdot|x_{n-1})$ in HMMs ('bootstrap' filter),
  - better: make use of y_n when sampling X_n^i.

Alleviating the Degeneracy

- Use residual/stratified/systematic resampling
  $\longrightarrow$ avoid multinomial resampling!
- Only resample when necessary.
- Devise better proposal kernels K_n^θ.
  - e.g. avoid $K_n^\theta(\cdot|x_{n-1}) := f^\theta(\cdot|x_{n-1})$ in HMMs ('bootstrap' filter),
  - better: make use of y_n when sampling X_n^i.

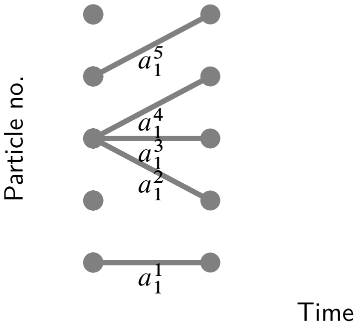
Particle Lineages

$$\psi_n^\theta(\overbrace{\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}}^{\substack{\text{all random variables generated} \\ \text{by the SMC algorithm}}}) := \left[\prod_{i=1}^{N} K_1^\theta(x_1^i)\right] \times \left[\prod_{p=2}^{n} r^\theta(\mathbf{a}_{p-1}|\mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) \prod_{i=1}^{N} K_p^\theta(x_p^i | x_{1:p-1}^{a_{p-1}^i})\right]$$



Particle Lineages

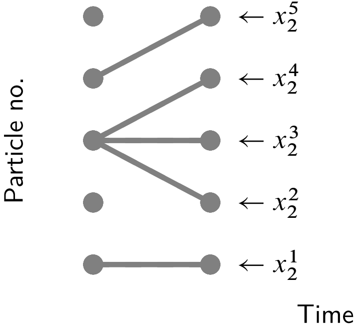
$$\psi_n^\theta(\overbrace{\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}}^{\text{all random variables generated by the SMC algorithm}}) := \left[\prod_{i=1}^{N} K_1^\theta(x_1^i)\right]$$
$$\times \left[\prod_{p=2}^{n} r^\theta(\mathbf{a}_{p-1}|\mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) \prod_{i=1}^{N} K_p^\theta(x_p^i|x_{1:p-1}^{a_{p-1}^i})\right]$$



Particle Lineages

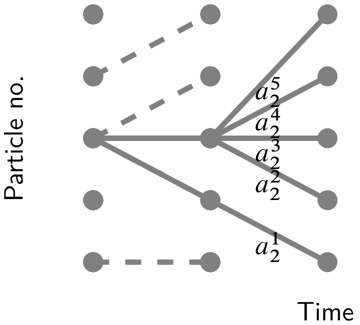
$$\psi_n^\theta(\overbrace{\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}}^{\text{all random variables generated by the SMC algorithm}}) := \left[\prod_{i=1}^{N} K_1^\theta(x_1^i)\right]$$

$$\times \left[\prod_{p=2}^{n} r^\theta(\mathbf{a}_{p-1}|\mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) \prod_{i=1}^{N} K_p^\theta(x_p^i|x_{1:p-1}^{a_{p-1}^i})\right]$$



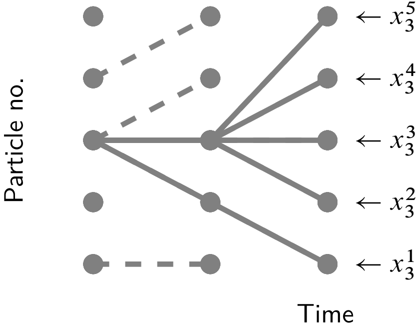
Particle Lineages

$$\psi_n^\theta(\overbrace{\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}}^{\text{all random variables generated by the SMC algorithm}}) := \left[\prod_{i=1}^{N} K_1^\theta(x_1^i)\right]$$
$$\times \left[\prod_{p=2}^{n} r^\theta(\mathbf{a}_{p-1}|\mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) \prod_{i=1}^{N} K_p^\theta(x_p^i|x_{1:p-1}^{a_{p-1}^i})\right]$$



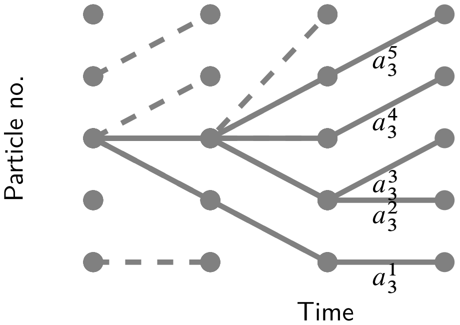
Particle Lineages

$$\psi_n^\theta(\overbrace{\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}}^{\text{all random variables generated by the SMC algorithm}}) := \left[\prod_{i=1}^{N} K_1^\theta(x_1^i)\right] \times \left[\prod_{p=2}^{n} r^\theta(\mathbf{a}_{p-1}|\mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) \prod_{i=1}^{N} K_p^\theta(x_p^i|x_{1:p-1}^{a_{p-1}^i})\right]$$



Particle Lineages

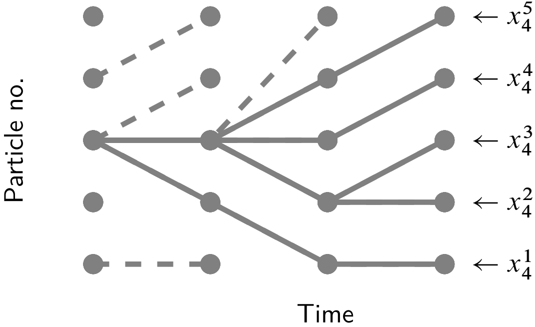
$$\psi_n^\theta(\overbrace{\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}}^{\text{all random variables generated by the SMC algorithm}}) := \left[\prod_{i=1}^{N} K_1^\theta(x_1^i)\right] \times \left[\prod_{p=2}^{n} r^\theta(\mathbf{a}_{p-1}|\mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) \prod_{i=1}^{N} K_p^\theta(x_p^i|x_{1:p-1}^{a_{p-1}^i})\right]$$



Particle Lineages

$$\psi_n^\theta(\overbrace{\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}}^{\text{all random variables generated by the SMC algorithm}}) := \left[\prod_{i=1}^N K_1^\theta(x_1^i)\right]$$

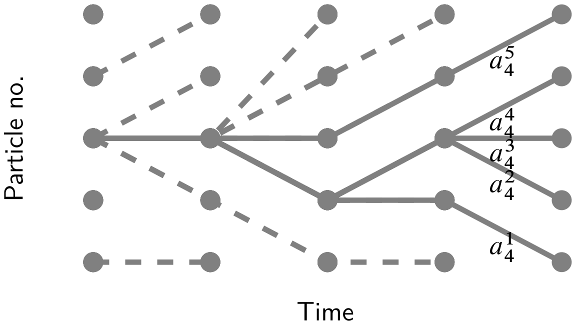
$$\times \left[\prod_{p=2}^n r^\theta(\mathbf{a}_{p-1}|\mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) \prod_{i=1}^N K_p^\theta(x_p^i|x_{1:p-1}^{a_{p-1}^i})\right]$$



Particle Lineages

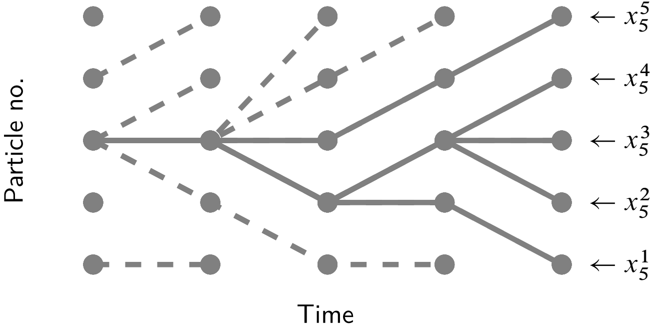
$$\psi_n^\theta(\overbrace{\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}}^{\text{all random variables generated by the SMC algorithm}}) := \left[\prod_{i=1}^{N} K_1^\theta(x_1^i)\right]$$

$$\times \left[\prod_{p=2}^{n} r^\theta(\mathbf{a}_{p-1}|\mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) \prod_{i=1}^{N} K_p^\theta(x_p^i|x_{1:p-1}^{a_{p-1}^i})\right]$$



Particle Lineages

$$\psi_n^\theta(\overbrace{\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}}^{\text{all random variables generated by the SMC algorithm}}) := \left[\prod_{i=1}^{N} K_1^\theta(x_1^i)\right] \times \left[\prod_{p=2}^{n} r^\theta(\mathbf{a}_{p-1}|\mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) \prod_{i=1}^{N} K_p^\theta(x_p^i | x_{1:p-1}^{a_{p-1}^i})\right]$$



Interpretation as Importance Sampling

- SMC methods are just (standard!) importance sampling on a suitably extended space.
- Hence, SMC methods yield an unbiased estimate of the normalising constant Z_n^θ, which is given by

$$\widehat{Z}_n^\theta(\mathbf{X}_{1:n}, \mathbf{A}_{1:n-1}) = \prod_{p=1}^{n} \Bigg[\frac{1}{N} \sum_{i=1}^{N} \underbrace{G_p^\theta(X_{1:p}^i)}_{\substack{i\text{th unnormalised} \\ \text{incremental weight at time } n}} \Bigg].$$

Interpretation as Importance Sampling

- SMC methods are just (standard!) importance sampling on a suitably extended space.
- Hence, SMC methods yield an unbiased estimate of the normalising constant Z_n^θ, which is given by

$$\widehat{Z}_n^\theta(\mathbf{X}_{1:n}, \mathbf{A}_{1:n-1}) = \prod_{p=1}^{n} \left[\frac{1}{N} \sum_{i=1}^{N} \underbrace{G_p^\theta(X_{1:p}^i)}_{\substack{i\text{th unnormalised} \\ \text{incremental weight at time } n}} \right].$$

Motivation

What if target distributions $(\tilde{\pi}_n)_{n\in\mathbb{N}}$ are defined on spaces $(\widetilde{E}_n)_{n\in\mathbb{N}}$ of *non-increasing* dimension?

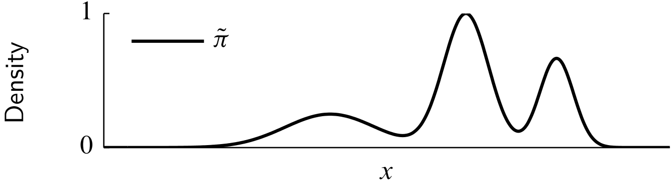
Motivation, continued

Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n}[\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \ldots, P$, where
  - we can easily sample from μ_1,
  - $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1.$



Motivation, continued

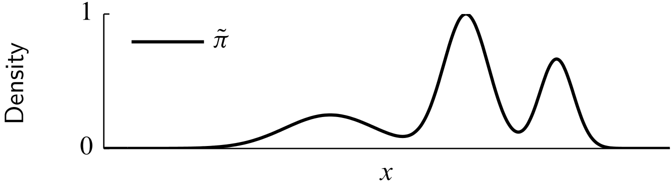
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n}[\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \ldots, P$, where

- we can easily sample from μ_1,
- $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



Motivation, continued

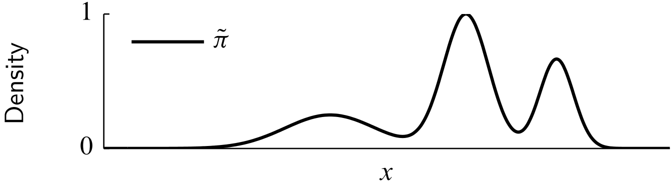
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n}[\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \dots, P$, where

- we can easily sample from μ_1,
- $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



Motivation, continued

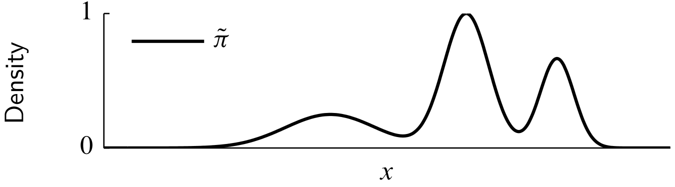
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n}[\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \ldots, P$, where

- we can easily sample from μ_1,
- $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



Motivation, continued

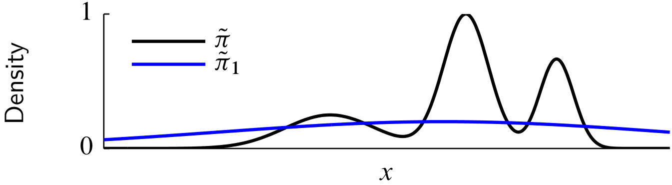
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n}[\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \ldots, P$, where

- we can easily sample from μ_1,
- $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



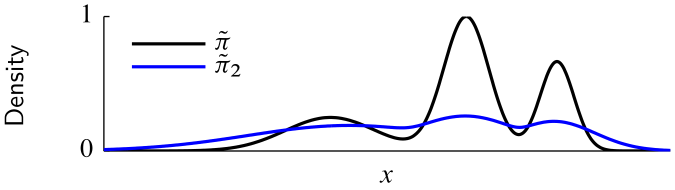
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n}[\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \ldots, P$, where

- we can easily sample from μ_1,
- $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



Motivation, continued

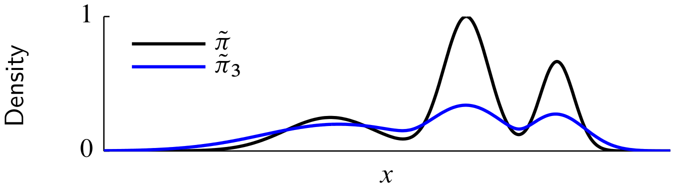
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n}[\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \ldots, P$, where

- we can easily sample from μ_1,
- $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



Motivation, continued

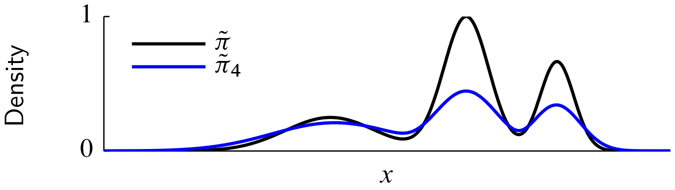
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n}[\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \ldots, P$, where

- we can easily sample from μ_1,
- $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



Motivation, continued

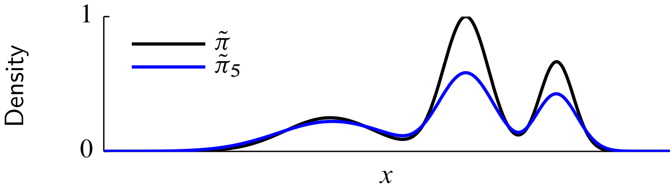
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n} [\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \dots, P$, where

- we can easily sample from μ_1,
- $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



Motivation, continued

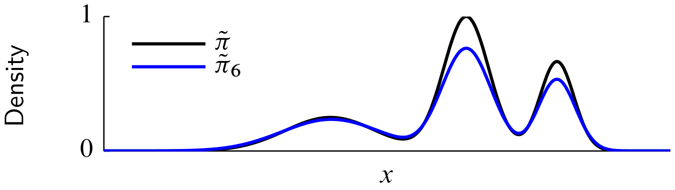
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n} [\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \ldots, P$, where

  - we can easily sample from μ_1,
  - $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



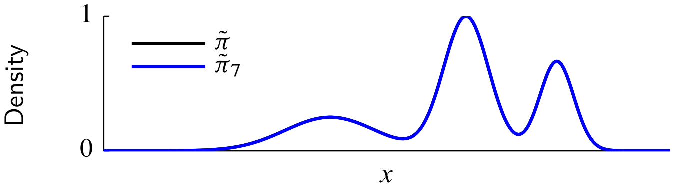
Example (Annealing):

- Want to target complicated distribution η on a space E.
- **Idea:** use SMC methods to target *bridging distributions*

$$\tilde{\pi}_n(x) \propto \tilde{\gamma}_n(x) := [\eta(x)]^{\phi_n}[\mu_1(x)]^{1-\phi_n},$$

for $n = 1, \ldots, P$, where

- we can easily sample from μ_1,
- $0 = \phi_1 < \phi_2 < \cdots < \phi_{P-1} < \phi_P = 1$.



SMC Samplers

- **Problem:** evaluating the weights difficult/impossible.
- **Solution:** target a sequence of extended distributions $(\pi_n)_{n\in\mathbb{N}}$ s.t.
  1. π_n admits $\tilde{\pi}_n$ as a marginal,
  2. $(\pi_n)_{n\in\mathbb{N}}$ are defined on spaces of *increasing* dimension.

SMC Samplers (Del Moral, Doucet & Jasra, 2006)

Use *'backward' Markov kernels* $L_{n-1}(x_{n-1}|x_n)$, so that

$$\pi_n(x_{1:n}) := \tilde{\pi}_n(x_n) \prod_{p=1}^{n} L_{p-1}(x_{p-1}|x_p).$$

SMC Samplers

- **Problem:** evaluating the weights difficult/impossible.
- **Solution:** target a sequence of extended distributions $(\pi_n)_{n \in \mathbb{N}}$ s.t.
  1. π_n admits $\tilde{\pi}_n$ as a marginal,
  2. $(\pi_n)_{n \in \mathbb{N}}$ are defined on spaces of *increasing* dimension.

SMC Samplers (Del Moral, Doucet & Jasra, 2006)

Use *'backward' Markov kernels* $L_{n-1}(x_{n-1}|x_n)$, so that

$$\pi_n(x_{1:n}) := \tilde{\pi}_n(x_n) \prod_{p=1}^{n} L_{p-1}(x_{p-1}|x_p).$$

SMC Samplers

- **Problem:** evaluating the weights difficult/impossible.
- **Solution:** target a sequence of extended distributions $(\pi_n)_{n\in\mathbb{N}}$ s.t.
    1. π_n admits $\tilde{\pi}_n$ as a marginal,
    2. $(\pi_n)_{n\in\mathbb{N}}$ are defined on spaces of *increasing* dimension.

SMC Samplers (Del Moral, Doucet & Jasra, 2006)

Use *'backward' Markov kernels* $L_{n-1}(x_{n-1}|x_n)$, so that

$$\pi_n(x_{1:n}) := \tilde{\pi}_n(x_n) \prod_{p=1}^{n} L_{p-1}(x_{p-1}|x_p).$$

Aim

- Now: want to approximate

$$\pi_P(\theta, x_{1:P}) = \frac{\gamma_P(\theta, x_{1:P})}{Z}$$

or its marginal $\pi_P(\theta)$.

Example

If $\pi_P(\theta, x_{1:P}) = p(\theta, x_{1:P} | y_{1:P})$ for observations $y_{1:P}$, then

- $\gamma_P(\theta, x_{1:P}) = p(\theta, x_{1:P}, y_{1:P})$,
- $Z = p(y_{1:P})$.

Aim

- Now: want to approximate

$$\pi_P(\theta, x_{1:P}) = \frac{\gamma_P(\theta, x_{1:P})}{Z}$$

or its marginal $\pi_P(\theta)$.

Example

If $\pi_P(\theta, x_{1:P}) = p(\theta, x_{1:P} | y_{1:P})$ for observations $y_{1:P}$, then

- $\gamma_P(\theta, x_{1:P}) = p(\theta, x_{1:P}, y_{1:P})$,
- $Z = p(y_{1:P})$.

Aim

- Now: want to approximate

$$\pi_P(\theta, x_{1:P}) = \frac{\gamma_P(\theta, x_{1:P})}{Z}$$

or its marginal $\pi_P(\theta)$.

Example

If $\pi_P(\theta, x_{1:P}) = p(\theta, x_{1:P}|y_{1:P})$ for observations $y_{1:P}$, then

- $\gamma_P(\theta, x_{1:P}) = p(\theta, x_{1:P}, y_{1:P})$,
- $Z = p(y_{1:P})$.

Aim

- Now: want to approximate

$$\pi_P(\theta, x_{1:P}) = \frac{\gamma_P(\theta, x_{1:P})}{Z}$$

or its marginal $\pi_P(\theta)$.

Example

If $\pi_P(\theta, x_{1:P}) = p(\theta, x_{1:P}|y_{1:P})$ for observations $y_{1:P}$, then

- $\gamma_P(\theta, x_{1:P}) = p(\theta, x_{1:P}, y_{1:P})$,
- $Z = p(y_{1:P})$.

Ideal MCMC Algorithms

Ideal Metropolis–Hastings Algorithm

Given $(\theta, X_{1:P})$,

1. propose new values $(\theta^\star, X^\star_{1:P})$,
2. accept with some probability.

BUT: difficult to design good proposals for $X^\star_{1:P}$.

- **Idea:** use SMC approximation $\hat{\pi}^{\theta^\star}$ as proposal for $X^\star_{1:P}$.
- **Problem:** proposal density is intractable.

  $\Longrightarrow$ cannot evaluate acceptance probability.

Ideal MCMC Algorithms

Ideal Metropolis–Hastings Algorithm

Given $(\theta, X_{1:P})$,

1. propose new values $(\theta^\star, X^\star_{1:P})$,
2. accept with some probability.

BUT: difficult to design good proposals for $X^\star_{1:P}$.

- **Idea:** use SMC approximation $\hat{\pi}^{\theta^\star}$ as proposal for $X^\star_{1:P}$.
- **Problem:** proposal density is intractable.

  $\Longrightarrow$ cannot evaluate acceptance probability.

Ideal MCMC Algorithms

Ideal Metropolis–Hastings Algorithm

Given $(\theta, X_{1:P})$,

1. propose new values $(\theta^\star, X^\star_{1:P})$,
2. accept with some probability.

BUT: difficult to design good proposals for $X^\star_{1:P}$.

- **Idea:** use SMC approximation $\hat{\pi}^{\theta^\star}$ as proposal for $X^\star_{1:P}$.
- **Problem:** proposal density is intractable.

$\implies$ cannot evaluate acceptance probability.

Ideal MCMC Algorithms

Ideal Metropolis–Hastings Algorithm

Given $(\theta, X_{1:P})$,

1. propose new values $(\theta^\star, X^\star_{1:P})$,
2. accept with some probability.

BUT: difficult to design good proposals for $X^\star_{1:P}$.

- **Idea:** use SMC approximation $\hat{\pi}^{\theta^\star}$ as proposal for $X^\star_{1:P}$.
- **Problem:** proposal density is intractable.

  $\Longrightarrow$ cannot evaluate acceptance probability.

Ideal MCMC Algorithms

Ideal Metropolis–Hastings Algorithm

Given $(\theta, X_{1:P})$,

1. propose new values $(\theta^\star, X^\star_{1:P})$,
2. accept with some probability.

BUT: difficult to design good proposals for $X^\star_{1:P}$.

- **Idea:** use SMC approximation $\hat{\pi}^{\theta^\star}$ as proposal for $X^\star_{1:P}$.
- **Problem:** proposal density is intractable.

  $\Longrightarrow$ cannot evaluate acceptance probability.

Ideal MCMC Algorithms

Ideal Metropolis–Hastings Algorithm

Given $(\theta, X_{1:P})$,

1. propose new values $(\theta^\star, X^\star_{1:P})$,
2. accept with some probability.

BUT: difficult to design good proposals for $X^\star_{1:P}$.

- **Idea:** use SMC approximation $\hat{\pi}^{\theta^\star}$ as proposal for $X^\star_{1:P}$.
- **Problem:** proposal density is intractable.

  $\Longrightarrow$ cannot evaluate acceptance probability.

Ideal MCMC Algorithms, continued

Ideal Gibbs sampler:

Given $(\theta, X_{1:P})$, sample from

1. $\pi_P(\theta | x_{1:P})$,
2. $\pi_P(x_{1:P} | \theta) = \pi_P^\theta(x_{1:P})$,

BUT: cannot sample from π_P^θ.

- **Idea:** sample from SMC approximation $\hat{\pi}^\theta$ of π_P^θ.
- **Problem:** $\hat{\pi}_P^\theta \neq \pi_P^\theta$.

Ideal MCMC Algorithms, continued

Ideal Gibbs sampler:

Given $(\theta, X_{1:P})$, sample from

1. $\pi_P(\theta | x_{1:P})$,
2. $\pi_P(x_{1:P} | \theta) = \pi_P^\theta(x_{1:P})$,

BUT: cannot sample from π_P^θ.

- **Idea:** sample from SMC approximation $\hat{\pi}^\theta$ of π_P^θ.
- **Problem:** $\hat{\pi}_P^\theta \neq \pi_P^\theta$.

Ideal MCMC Algorithms, continued

Ideal Gibbs sampler:

Given $(\theta, X_{1:P})$, sample from

1. $\pi_P(\theta|x_{1:P})$,
2. $\pi_P(x_{1:P}|\theta) = \pi_P^\theta(x_{1:P})$,

BUT: cannot sample from π_P^θ.

- **Idea:** sample from SMC approximation $\hat{\pi}^\theta$ of π_P^θ.
- **Problem:** $\hat{\pi}_P^\theta \neq \pi_P^\theta$.

Ideal MCMC Algorithms, continued

Ideal Gibbs sampler:

Given $(\theta, X_{1:P})$, sample from

1. $\pi_P(\theta | x_{1:P})$,
2. $\pi_P(x_{1:P} | \theta) = \pi_P^\theta(x_{1:P})$,

BUT: cannot sample from π_P^θ.

- **Idea:** sample from SMC approximation $\hat{\pi}^\theta$ of π_P^θ.
- **Problem:** $\hat{\pi}_P^\theta \neq \pi_P^\theta$.

Ideal MCMC Algorithms, continued

Ideal Gibbs sampler:

Given $(\theta, X_{1:P})$, sample from

1. $\pi_P(\theta | x_{1:P})$,
2. $\pi_P(x_{1:P} | \theta) = \pi_P^\theta(x_{1:P})$,

BUT: cannot sample from π_P^θ.

- **Idea:** sample from SMC approximation $\hat{\pi}^\theta$ of π_P^θ.
- **Problem:** $\hat{\pi}_P^\theta \neq \pi_P^\theta$.

Extended Target Distribution

Particle MCMC Methods

- Andrieu, Doucet & Holenstein (2010).
- exact MCMC methods, i.e.
  - Metropolis–Hastings algorithm,
  - Gibbs sampler

  targeting an *extended* distribution.
- this distribution includes all random variables generated by an SMC algorithm, i.e. $(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})$ (and more).
- basis: *Pseudo-Marginal* approach,
  - Andrieu & Roberts (2009),
  - permits IS within MCMC,
  - and SMC can be interpreted as standard IS (on a suitably extended space).

Extended Target Distribution

Particle MCMC Methods

- Andrieu, Doucet & Holenstein (2010).
- exact MCMC methods, i.e.
  - Metropolis–Hastings algorithm,
  - Gibbs sampler

  targeting an *extended* distribution.
- this distribution includes all random variables generated by an SMC algorithm, i.e. $(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})$ (and more).
- basis: *Pseudo-Marginal* approach,
  - Andrieu & Roberts (2009),
  - permits IS within MCMC,
  - and SMC can be interpreted as standard IS (on a suitably extended space).

Extended Target Distribution

Particle MCMC Methods

- Andrieu, Doucet & Holenstein (2010).
- exact MCMC methods, i.e.
  - Metropolis–Hastings algorithm,
  - Gibbs sampler

  targeting an *extended* distribution.
- this distribution includes all random variables generated by an SMC algorithm, i.e. $(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})$ (and more).
- basis: *Pseudo-Marginal* approach,
  - Andrieu & Roberts (2009),
  - permits IS within MCMC,
  - and SMC can be interpreted as standard IS
    (on a suitably extended space).

Extended Target Distribution

Particle MCMC Methods

- Andrieu, Doucet & Holenstein (2010).
- exact MCMC methods, i.e.
  - Metropolis–Hastings algorithm,
  - Gibbs sampler

  targeting an *extended* distribution.
- this distribution includes all random variables generated by an SMC algorithm, i.e. $(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})$ (and more).
- basis: *Pseudo-Marginal* approach,
  - Andrieu & Roberts (2009),
  - permits IS within MCMC,
  - and SMC can be interpreted as standard IS (on a suitably extended space).

Extended Target Distribution

Particle MCMC Methods

- Andrieu, Doucet & Holenstein (2010).
- exact MCMC methods, i.e.
  - Metropolis–Hastings algorithm,
  - Gibbs sampler

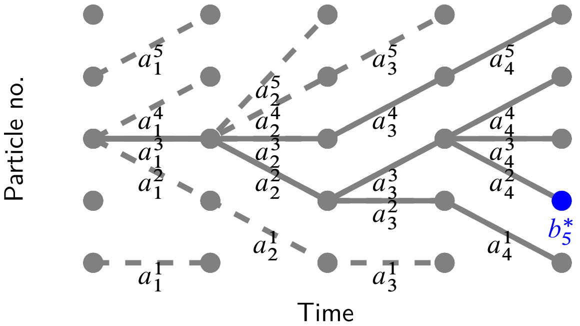
  targeting an *extended* distribution.
- this distribution includes all random variables generated by an SMC algorithm, i.e. $(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})$ (and more).
- basis: *Pseudo-Marginal* approach,
  - Andrieu & Roberts (2009),
  - permits IS within MCMC,
  - and SMC can be interpreted as standard IS (on a suitably extended space).

Extended Target Distribution, continued

Parametrisation I:

$$\bar{\pi}_P(\theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}, b_P^*)$$

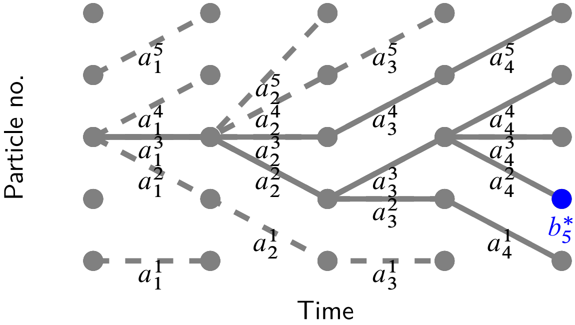
$$\propto p(\theta) \underbrace{w_P^{\theta, b_P^*}}_{\substack{\text{weight of} \\ b_P^* \text{th path}}} \underbrace{\widehat{Z}_P^{\theta}(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})}_{\substack{\text{SMC estimate of} \\ \text{normalising constant}}} \underbrace{\psi_P^{\theta}(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})}_{\text{SMC algorithm}} .$$



Reparametrisation

Using $b_n^* = a_n^{b_{n+1}^*}$ for $n \in \{1, \ldots, P-1\}$,

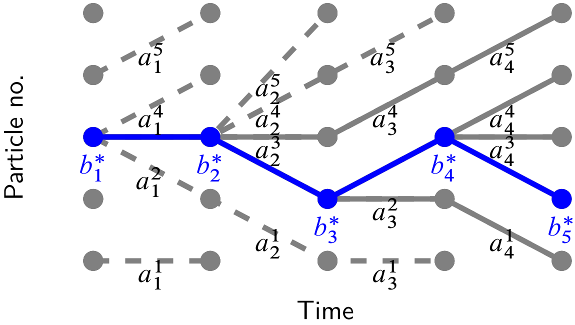
$$\underbrace{(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}, b_P^*)}_{\text{Parametrisation I}} \longleftrightarrow \underbrace{(\mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*}, x_{1:P}^*, b_{1:P}^*)}_{\text{Parametrisation II}} .$$



Reparametrisation

Using $b_n^* = a_n^{b_{n+1}^*}$ for $n \in \{1, \ldots, P-1\}$,

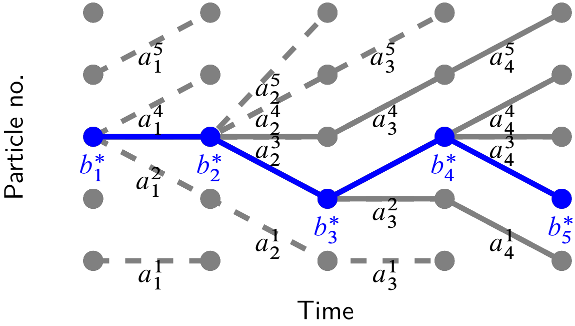
$$\underbrace{(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}, b_P^*)}_{\text{Parametrisation I}} \longleftrightarrow \underbrace{(\mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*}, x_{1:P}^*, b_{1:P}^*)}_{\text{Parametrisation II}}.$$



Extended Target Distribution

Parametrisation II:

$$\bar{\pi}_P(\theta, \mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*}, x_{1:P}^{*}, b_{1:P}^{*}) = \underbrace{\pi_P(\theta, x_{1:P}^{*})}_{\text{'actual' target}} \underbrace{p(b_{1:P}^{*})}_{\text{marginal of } b_{1:P}^{*}} \underbrace{\psi_P^{\theta}(\mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*} \| x_{1:P}^{*}, b_{1:P}^{*})}_{\text{'conditional' SMC algorithm}}.$$



PMMH Algorithm

A Metropolis–Hastings Algorithm Targeting $\bar{\pi}_P$

- **Notation:** $b := b_P^*$ and $\xi := (\theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}, b)$.
- **Proposal kernel:**

$$Q(\xi^\star|\xi) := T(\theta^\star|\theta)\psi_P^{\theta^\star}(\mathbf{x}^\star_{1:P}, \mathbf{a}^\star_{1:P-1})w_P^{\theta^\star, b^\star},$$

- **Acceptance probability** (using Parametrisation I):

$$\begin{aligned}\alpha(\xi^\star|\xi) &:= 1 \wedge \frac{\bar{\pi}_P(\xi^\star)Q(\xi|\xi^\star)}{\bar{\pi}_P(\xi)Q(\xi^\star|\xi)} \\ &= 1 \wedge \frac{p(\theta^\star)}{p(\theta)}\frac{\widehat{Z}_P^{\theta^\star}(\mathbf{x}^\star_{1:P}, \mathbf{a}^\star_{1:P-1})}{\widehat{Z}_P^{\theta}(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})}\frac{T(\theta|\theta^\star)}{T(\theta^\star|\theta)}.\end{aligned}$$

- Special case of the GIMH algorithm (Andrieu & Roberts, 2009)

PMMH Algorithm, continued

- efficiency crucially depends on SMC estimate of Z_P^θ
- usually, $\mathbf{var}[\widehat{Z}_P^\theta(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})]$ **grows linearly** in P.
- need to increase N at least linearly with P.

  $\longrightarrow$ otherwise: low acceptance rate.

PMMH Algorithm, continued

- efficiency crucially depends on SMC estimate of Z_P^θ
- usually, $\mathbf{var}[\widehat{Z}_P^\theta(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})]$ **grows linearly** in P.
- need to increase N at least linearly with P.

  $\longrightarrow$ otherwise: low acceptance rate.

PMMH Algorithm, continued

- efficiency crucially depends on SMC estimate of Z_P^θ
- usually, $\mathbf{var}[\widehat{Z}_P^\theta(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})]$ **grows linearly** in P.
- need to increase N at least linearly with P.

  $\longrightarrow$ otherwise: low acceptance rate.

Alternative "justification" of PMMH/GIMH

- only want to approximate

$$\int \pi_P(\theta, x_{1:P}) \mathrm{d}x_{1:P} = \pi_P(\theta) \propto \gamma_P(\theta).$$

- $p(\theta) Z_P^\theta = \gamma_P(\theta)$, so that

$$\mathbf{E}\big[p(\theta) \widehat{Z}_P^\theta(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})\big] = \gamma_P(\theta).$$

- View as MH algorithm with approximation

$$\frac{\gamma_P(\theta^\star) T(\theta|\theta^\star)}{\gamma_P(\theta) T(\theta^\star|\theta)} \approx \frac{p(\theta^\star)}{p(\theta)} \frac{\widehat{Z}_P^{\theta^\star}(\mathbf{x}_{1:P}^\star, \mathbf{a}_{1:P-1}^\star)}{\widehat{Z}_P^\theta(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})} \frac{T(\theta|\theta^\star)}{T(\theta^\star|\theta)}.$$

Alternative "justification" of PMMH/GIMH

- only want to approximate

$$\int \pi_P(\theta, x_{1:P})\mathrm{d}x_{1:P} = \pi_P(\theta) \propto \gamma_P(\theta).$$

- $p(\theta)Z_P^\theta = \gamma_P(\theta)$, so that

$$\mathbf{E}\big[p(\theta)\widehat{Z}_P^\theta(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})\big] = \gamma_P(\theta).$$

- View as MH algorithm with approximation

$$\frac{\gamma_P(\theta^\star)T(\theta|\theta^\star)}{\gamma_P(\theta)T(\theta^\star|\theta)} \approx \frac{p(\theta^\star)}{p(\theta)}\frac{\widehat{Z}_P^{\theta^\star}(\mathbf{x}_{1:P}^\star, \mathbf{a}_{1:P-1}^\star)}{\widehat{Z}_P^\theta(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})}\frac{T(\theta|\theta^\star)}{T(\theta^\star|\theta)}.$$

Alternative "justification" of PMMH/GIMH

- only want to approximate

$$\int \pi_P(\theta, x_{1:P}) \mathrm{d}x_{1:P} = \pi_P(\theta) \propto \gamma_P(\theta).$$

- $p(\theta) Z_P^\theta = \gamma_P(\theta)$, so that

$$\mathbf{E}\big[p(\theta) \widehat{Z}_P^\theta(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1}) \big] = \gamma_P(\theta).$$

- View as MH algorithm with approximation

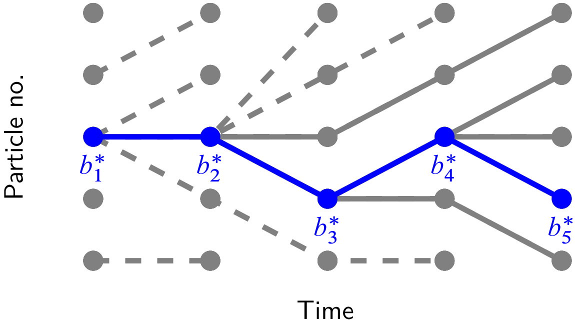
$$\frac{\gamma_P(\theta^\star) T(\theta|\theta^\star)}{\gamma_P(\theta) T(\theta^\star|\theta)} \approx \frac{p(\theta^\star)}{p(\theta)} \frac{\widehat{Z}_P^{\theta^\star}(\mathbf{x}^\star_{1:P}, \mathbf{a}^\star_{1:P-1})}{\widehat{Z}_P^\theta(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})} \frac{T(\theta|\theta^\star)}{T(\theta^\star|\theta)}.$$

Particle Gibbs Sampler

A Gibbs sampler targeting $\bar{\pi}_P$

Recall that

$$\bar{\pi}_P(\theta, \mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*}, x_{1:P}^{*}, b_{1:P}^{*}) = \underbrace{\pi_P(\theta, x_{1:P}^{*})}_{\text{'actual' target}} \underbrace{p(b_{1:P}^{*})}_{\text{distribution of indices of } x_{1:P}^{*}} \underbrace{\psi_P^{\theta}(\mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*} \| x_{1:P}^{*}, b_{1:P}^{*})}_{\text{'conditional' SMC algorithm}}.$$



Particle Gibbs Sweep

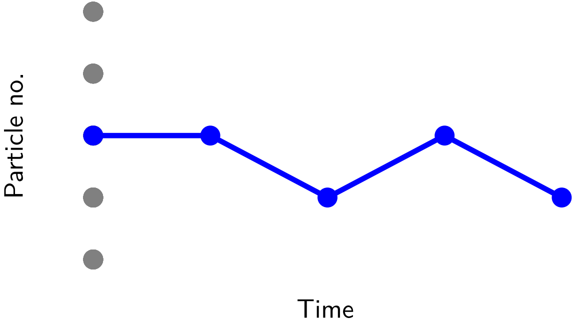
Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^{\theta}_{P}(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_P | \theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})$ [via Parametrisation I].

Particle Gibbs Sweep

Sample from:

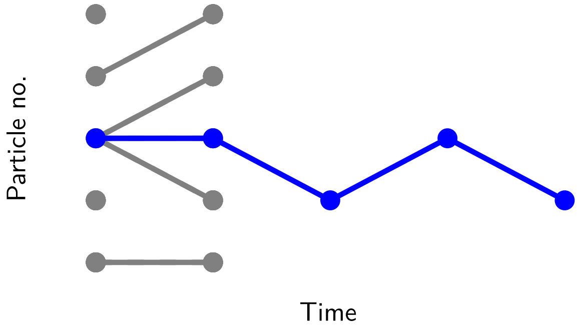
1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_P | \theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})$ [via Parametrisation I].



Particle Gibbs Sweep

Sample from:

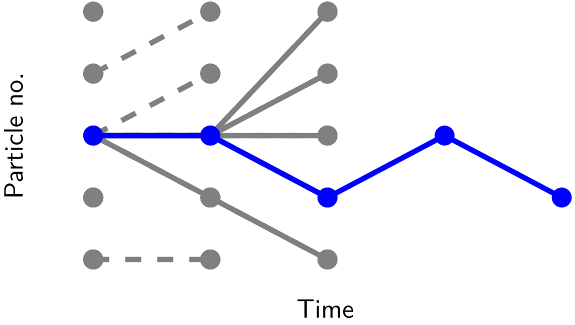
1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via ‘conditional’ SMC],
3. $\bar{\pi}_P(b^*_P | \theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})$ [via Parametrisation I].



Particle Gibbs Sweep

Sample from:

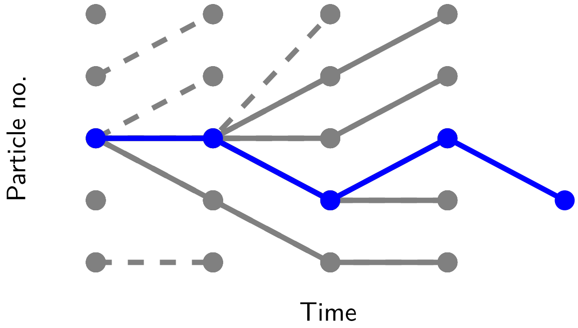
1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via ‘conditional’ SMC],
3. $\bar{\pi}_P(b^*_P | \theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})$ [via Parametrisation I].



Particle Gibbs Sweep

Sample from:

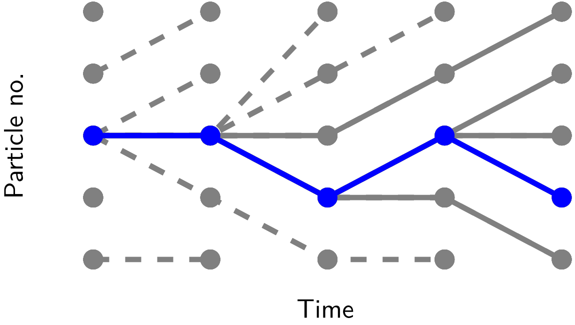
1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_P | \theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})$ [via Parametrisation I].



Particle Gibbs Sweep

Sample from:

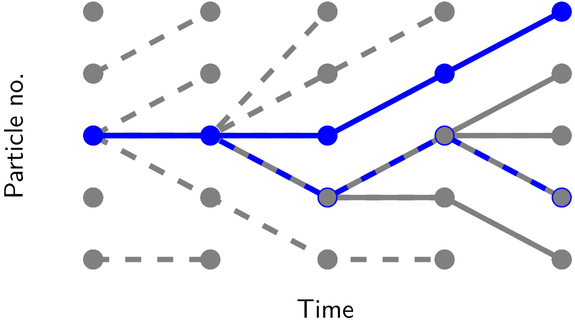
1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_P | \theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})$ [via Parametrisation I].



Particle Gibbs Sweep

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_P | \theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})$ [via Parametrisation I].



Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

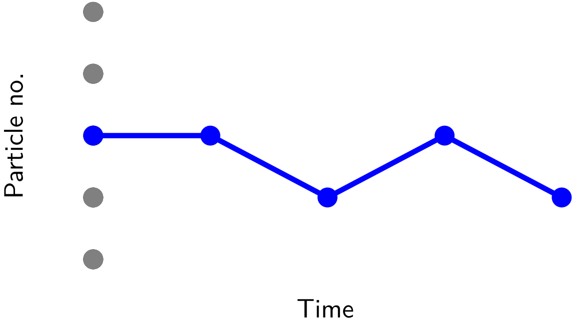
1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

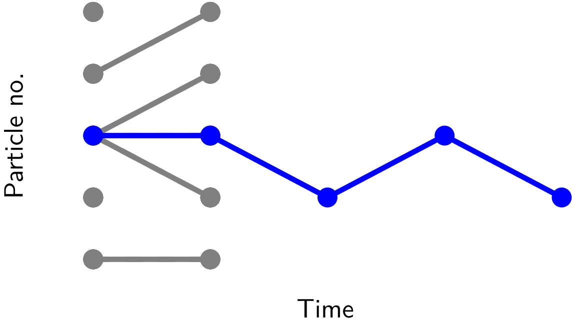


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

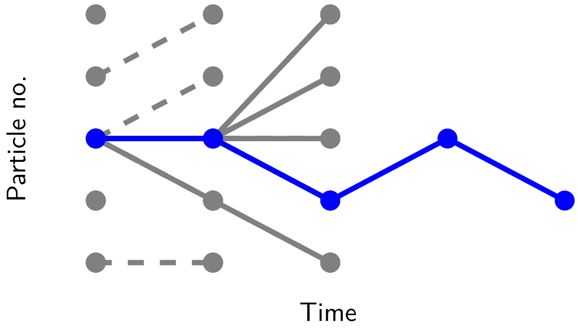


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

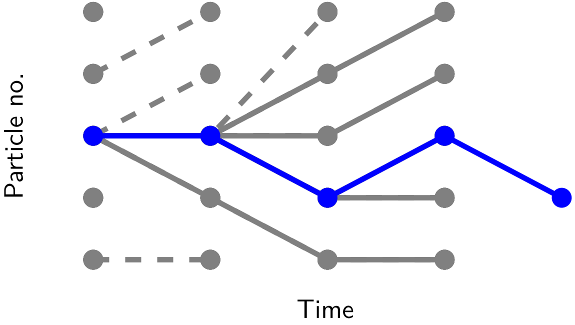


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta|x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_n|\theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

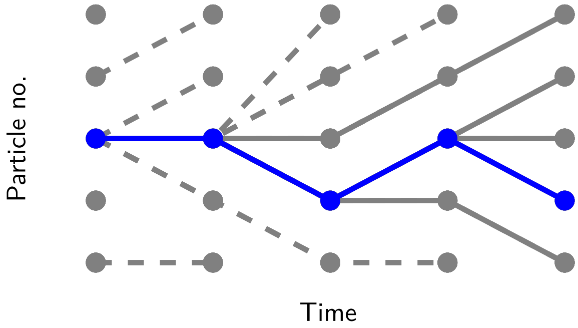


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

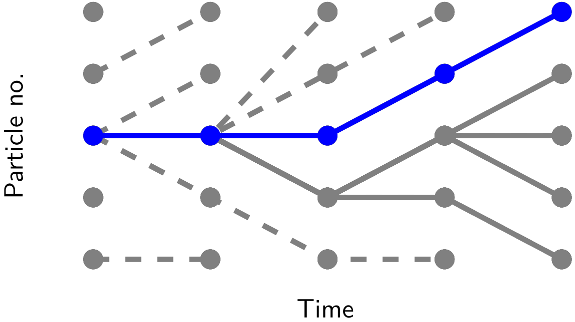


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

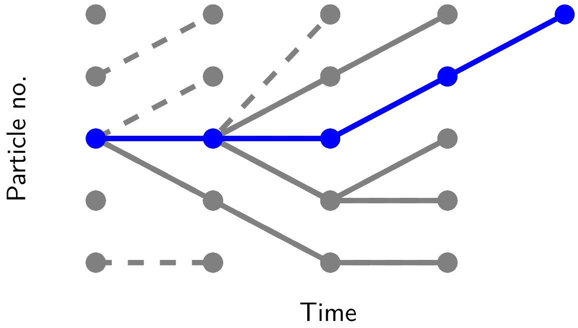


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

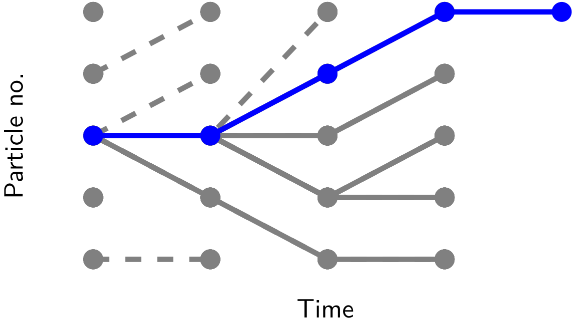


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

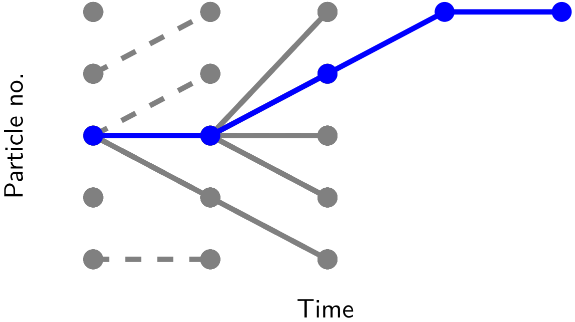


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

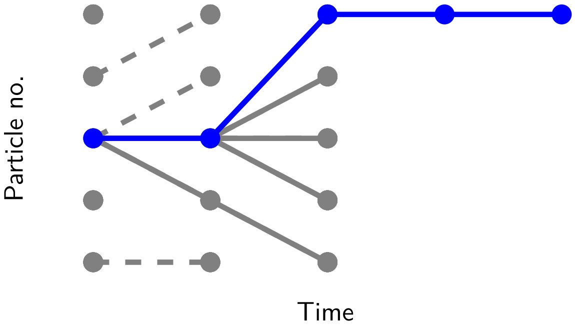


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

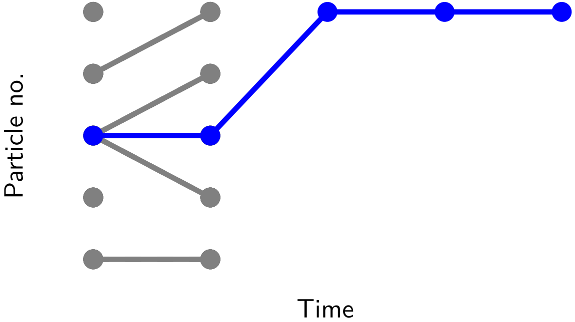


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

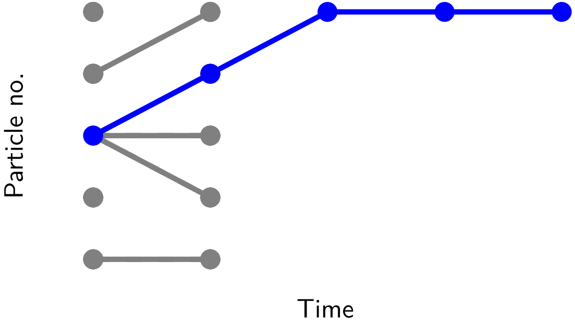


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta|x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_n|\theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

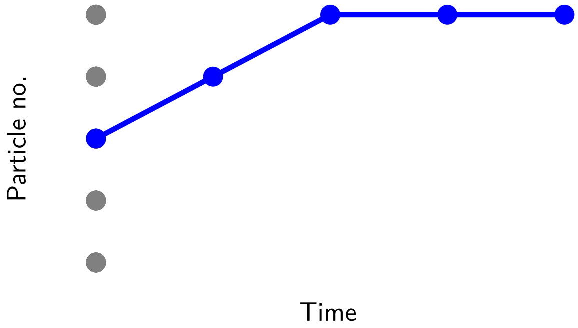


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta|x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^{\theta}_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1}\|x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\overline{\pi}_P(b^*_n|\theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.

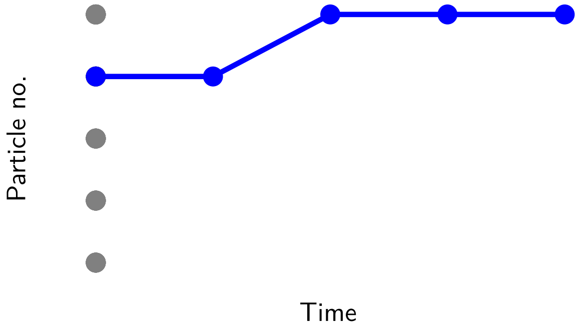


Particle Gibbs with Backward Sampling

Whiteley (2010)

Sample from:

1. $\pi_P(\theta | x^*_{1:P})$ [e.g. via a Metropolis–Hastings step],
2. $\psi^\theta_P(\mathbf{x}^{-*}_{1:P}, \mathbf{a}^{-*}_{1:P-1} \| x^*_{1:P}, b^*_{1:P})$ [via 'conditional' SMC],
3. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$ for $n = P, \ldots, 1$.



Particle Gibbs with Ancestor Sampling

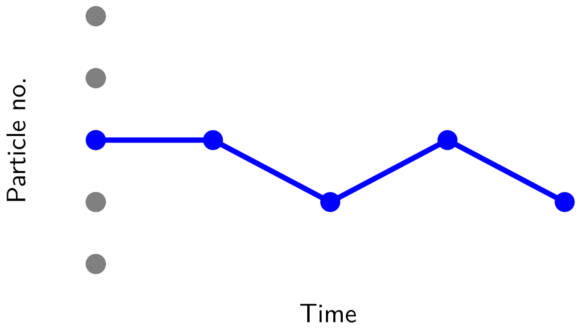
Lindsten, Jordan & Schön (2012)

1. sample from $\pi_P(\theta|x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b_n^*}, \mathbf{a}_{n-1}^{-b_n^*} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\overline{\pi}_P(b^*_n|\theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.

Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

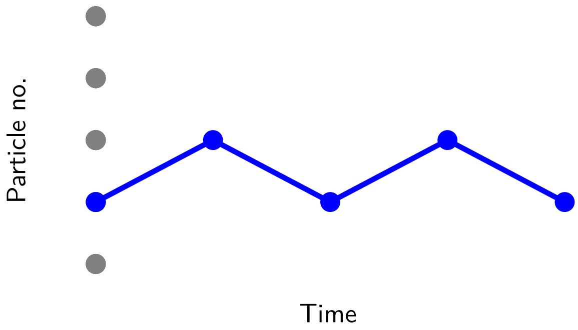
1. sample from $\pi_P(\theta|x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b^*_n}, \mathbf{a}_{n-1}^{-b^*_n} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\overline{\pi}_P(b^*_n|\theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

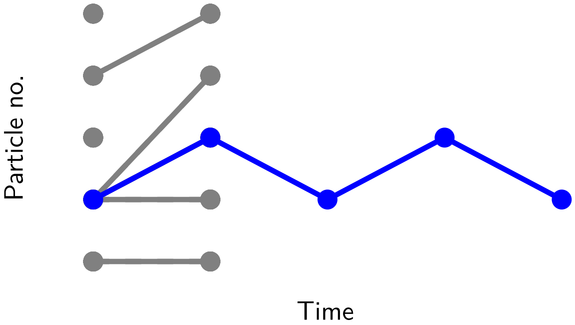
1. sample from $\pi_P(\theta | x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b^*_n}, \mathbf{a}_{n-1}^{-b^*_n} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\overline{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

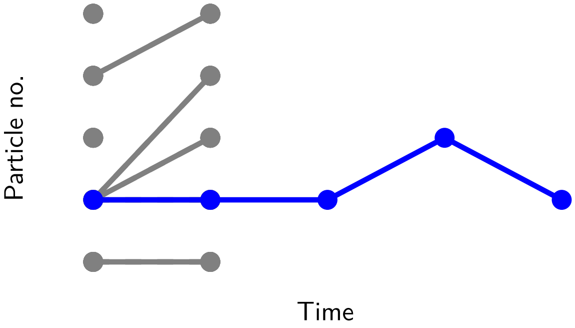
1. sample from $\pi_P(\theta | x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b^*_n}, \mathbf{a}_{n-1}^{-b^*_n} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\overline{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

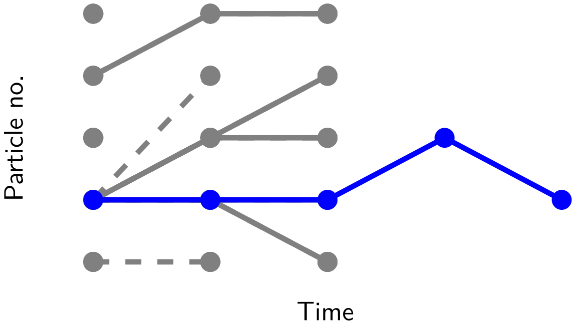
1. sample from $\pi_P(\theta|x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b_n^*}, \mathbf{a}_{n-1}^{-b_n^*} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\bar{\pi}_P(b^*_n|\theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

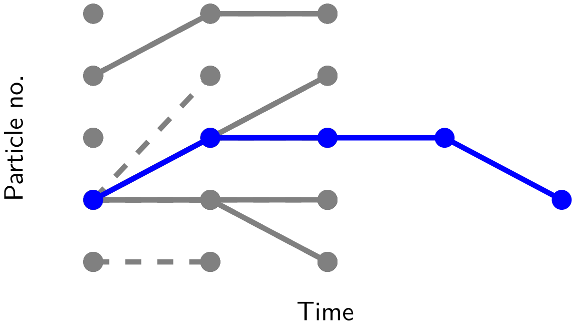
1. sample from $\pi_P(\theta | x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b^*_n}, \mathbf{a}_{n-1}^{-b^*_n} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\overline{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

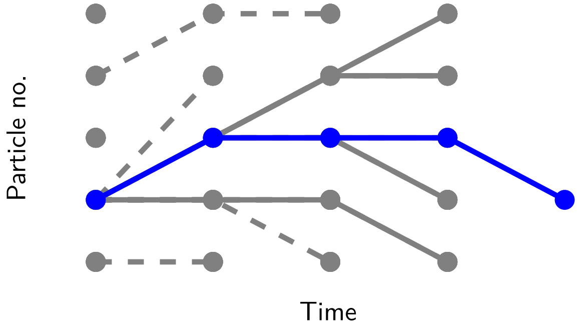
1. sample from $\pi_P(\theta|x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b^*_n}, \mathbf{a}_{n-1}^{-b^*_n} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\bar{\pi}_P(b^*_n|\theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

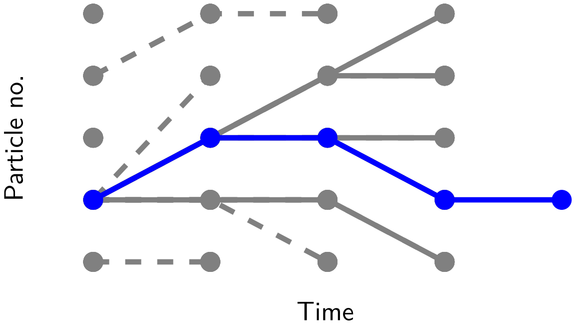
1. sample from $\pi_P(\theta | x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b^*_n}, \mathbf{a}_{n-1}^{-b^*_n} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

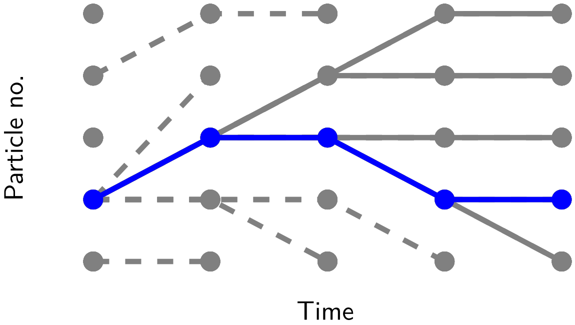
1. sample from $\pi_P(\theta | x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^{\theta}_{P}(\mathbf{x}_n^{-b_n^*}, \mathbf{a}_{n-1}^{-b_n^*} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\bar{\pi}_P(b_n^* | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

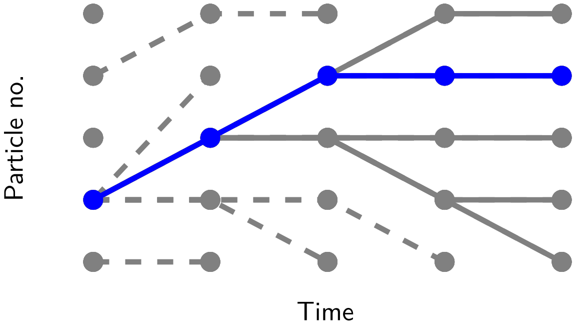
1. sample from $\pi_P(\theta | x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b^*_n}, \mathbf{a}_{n-1}^{-b^*_n} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



Particle Gibbs with Ancestor Sampling

Lindsten, Jordan & Schön (2012)

1. sample from $\pi_P(\theta | x^*_{1:P})$,
2. for $n = 1, \ldots, P$, sample from
   i. $\psi^\theta_P(\mathbf{x}_n^{-b^*_n}, \mathbf{a}_{n-1}^{-b^*_n} \| \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, x^*_{n:P}, b^*_{n:P})$,
   ii. $\bar{\pi}_P(b^*_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x^*_{n+1:P}, b^*_{n+1:P})$.



SMC^2

Chopin, Jacob & Papaspiliopoulos, (2013)

- Particle MCMC methods can only target a *single* distribution $\pi_P(\theta, x_{1:P})$.
- How to approximate $(\pi_n(\theta, x_{1:n}))_{n \in \mathbb{N}}$ (e.g. if observations arrive sequentially in time)?
- Idea: instead of MCMC, use SMC to target (a marginal of) the extended distribution $\overline{\pi}_n(\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, b_n^*)$.
- Can be interpreted as nested SMC algorithms – SMC within SMC, i.e. each particle has its own SMC algorithm.

SMC^2

Chopin, Jacob & Papaspiliopoulos, (2013)

- Particle MCMC methods can only target a *single* distribution $\pi_P(\theta, x_{1:P})$.
- How to approximate $(\pi_n(\theta, x_{1:n}))_{n \in \mathbb{N}}$ (e.g. if observations arrive sequentially in time)?
- Idea: instead of MCMC, use SMC to target (a marginal of) the extended distribution $\overline{\pi}_n(\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, b_n^*)$.
- Can be interpreted as nested SMC algorithms – SMC within SMC, i.e. each particle has its own SMC algorithm.

SMC^2

Chopin, Jacob & Papaspiliopoulos, (2013)

- Particle MCMC methods can only target a *single* distribution $\pi_P(\theta, x_{1:P})$.
- How to approximate $(\pi_n(\theta, x_{1:n}))_{n\in\mathbb{N}}$ (e.g. if observations arrive sequentially in time)?
- Idea: instead of MCMC, use SMC to target (a marginal of) the extended distribution $\overline{\pi}_n(\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, b_n^*)$.
- Can be interpreted as nested SMC algorithms – SMC within SMC, i.e. each particle has its own SMC algorithm.

SMC^2

Chopin, Jacob & Papaspiliopoulos, (2013)

- Particle MCMC methods can only target a *single* distribution $\pi_P(\theta, x_{1:P})$.
- How to approximate $(\pi_n(\theta, x_{1:n}))_{n\in\mathbb{N}}$ (e.g. if observations arrive sequentially in time)?
- Idea: instead of MCMC, use SMC to target (a marginal of) the extended distribution $\overline{\pi}_n(\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, b_n^*)$.
- Can be interpreted as nested SMC algorithms – SMC within SMC, i.e. each particle has its own SMC algorithm.

Literature

- C Andrieu, GO Roberts, (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *ANN STAT*, 37(2), 697–725.
- C Andrieu, A Doucet, R Holenstein (2010). Particle Markov chain Monte Carlo methods. *J ROY STAT SOC B*, 72(3), 269–342.
- N Chopin, PE Jacob, O Papaspiliopoulos, (2013). SMC^2: an efficient algorithm for sequential analysis of state space models. *J ROY STAT SOC B*, 75(3), 397–426.
- P Del Moral, A Doucet, A Jasra, (2006). Sequential Monte Carlo samplers. *J ROY STAT SOC B*, 68(3), 411–436.
- F Lindsten, MI Jordan, T Schön, (2012). Ancestor Sampling for Particle Gibbs. *NIPS*.
- DA Van Dyk, T Park (2006). Partially collapsed Gibbs samplers. *JASA*, 103(482), 790–796.
- N Whiteley (2010). Contribution to the discussion on ‘Particle Markov chain Monte Carlo methods’ by C Andrieu, A Doucet, R Holenstein. *J ROY STAT SOC B*, 72(3), 306–307.
- N Whiteley, AM Johansen, S Godsill (2011). Monte Carlo filtering of piecewise deterministic processes. *J COMPUT GRAPH STAT*, 20(1), 119–139.