Particle Markov Chain Monte Carlo

Christophe Andrieu, Arnaud Doucet & Roman Holenstein
Departments of Computer Science and Statistics
University of British Columbia
Two main classes of Monte Carlo methods are used nowadays for Bayesian computation: **Markov chain Monte Carlo** (MCMC) and **Sequential Monte Carlo** (SMC) aka Particle filters.
Objectives

- Two main classes of Monte Carlo methods are used nowadays for Bayesian computation: **Markov chain Monte Carlo (MCMC)** and **Sequential Monte Carlo (SMC)** aka Particle filters.
- It seems natural to combine MCMC and SMC.
Objectives

- Two main classes of Monte Carlo methods are used nowadays for Bayesian computation: **Markov chain Monte Carlo** (MCMC) and **Sequential Monte Carlo** (SMC) aka Particle filters.
- It seems natural to combine MCMC and SMC.
- We know how to use MCMC within SMC (Gilks & Berzuini, JRSS B, 2001).
Objectives

- Two main classes of Monte Carlo methods are used nowadays for Bayesian computation: **Markov chain Monte Carlo** (MCMC) and **Sequential Monte Carlo** (SMC) aka Particle filters.
- It seems natural to combine MCMC and SMC.
- We know how to use MCMC within SMC (Gilks & Berzuini, JRSS B, 2001).
- We show here how to use SMC within MCMC in a simple and principled way.
Motivating Example: General State-Space Models

Let \( \{X_k\}_{k \geq 1} \) be a Markov process defined by

\[
X_1 \sim \mu(\cdot) \quad \text{and} \quad X_k | (X_{k-1} = x_{k-1}) \sim f(\cdot | x_{k-1}).
\]
Motivating Example: General State-Space Models

Let $\{X_k\}_{k \geq 1}$ be a Markov process defined by

$$X_1 \sim \mu(\cdot) \text{ and } X_k \mid (X_{k-1} = x_{k-1}) \sim f(\cdot \mid x_{k-1}).$$

We only have access to a process $\{Y_k\}_{k \geq 1}$ such that, conditional upon $\{X_k\}_{k \geq 1}$, the observations are statistically independent and

$$Y_k \mid (X_k = x_k) \sim g(\cdot \mid x_k).$$
Bayesian Inference in General State-Space Models

- Given a collection of observations $y_{1:T} := (y_1, ..., y_T)$, we are interested in inference about $X_{1:T} := (X_1, ..., X_T)$.
Given a collection of observations \( y_{1:T} := (y_1, ..., y_T) \), we are interested in inference about \( X_{1:T} := (X_1, ..., X_T) \).

In this Bayesian framework, inference relies on

\[
p (x_{1:T} | y_{1:T}) = \frac{p (x_{1:T}, y_{1:T})}{p (y_{1:T})}
\]

where

\[
p (x_{1:T}, y_{1:T}) = \mu (x_1) \prod_{k=2}^{T} f (x_k | x_{k-1}) \prod_{k=1}^{T} g (y_k | x_k)
\]

\[
\begin{align*}
\text{prior} & \quad \text{likelihood}
\end{align*}
\]
Given a collection of observations \( y_{1:T} := (y_1, ..., y_T) \), we are interested in inference about \( X_{1:T} := (X_1, ..., X_T) \).

In this Bayesian framework, inference relies on

\[
p (x_{1:T} | y_{1:T}) = \frac{p (x_{1:T}, y_{1:T})}{p (y_{1:T})}
\]

where

\[
p (x_{1:T}, y_{1:T}) = \mu (x_1) \prod_{k=2}^{T} f (x_k | x_{k-1}) \prod_{k=1}^{T} g (y_k | x_k)
\]

\[\text{prior} \quad \text{likelihood}\]

Except for a few models including finite state-space HMM and linear Gaussian models, this posterior distribution does not admit a standard form.
Standard MCMC Approaches

- We sample an ergodic Markov chain \( \{X_{1:T}(i)\} \) of invariant distribution \( p(x_{1:T}|y_{1:T}) \) using a Metropolis-Hastings (MH) algorithm; e.g. an independent MH sampler.
We sample an ergodic Markov chain \( \{X_{1:T}(i)\} \) of invariant distribution \( p(X_{1:T} | y_{1:T}) \) using a Metropolis-Hastings (MH) algorithm; e.g. an independent MH sampler.

**At iteration** \( i \)

Sample \( X^*_{1:T} \sim q(X_{1:T}) \).

W.p. \( 1 \land \frac{p(X^*_{1:T}, y_{1:T})}{q(X^*_{1:T})} \frac{q(X_{1:T}(i-1))}{p(X_{1:T}(i-1), y_{1:T})} \) set \( X_{1:T}(i) = X^*_{1:T} \), otherwise set \( X_{1:T}(i) = X_{1:T}(i-1) \).
We sample an ergodic Markov chain \( \{X_{1:T}(i)\} \) of invariant distribution \( p(x_{1:T} \mid y_{1:T}) \) using a Metropolis-Hastings (MH) algorithm; e.g. an independent MH sampler.

**At iteration i**
Sample \( X_{1:T}^* \sim q(x_{1:T}) \).

W.p. \( 1 \wedge \frac{p(X_{1:T}^*, y_{1:T})}{q(X_{1:T}^*)} \frac{q(X_{1:T}(i-1))}{p(X_{1:T}(i-1), y_{1:T})} \) set \( X_{1:T}(i) = X_{1:T}^* \), otherwise set \( X_{1:T}(i) = X_{1:T}(i-1) \).

Very simple but very inefficient.
We sample an ergodic Markov chain $\{X_{1:T}(i)\}$ of invariant distribution $p(x_{1:T}|y_{1:T})$ using a Metropolis-Hastings (MH) algorithm; e.g. an independent MH sampler.

**At iteration $i$**

Sample $X^*_{1:T} \sim q(x_{1:T})$.

W.p. $1 \wedge \frac{p(X^*_{1:T},y_{1:T})}{q(X^*_{1:T})} \frac{q(x_{1:T}(i-1))}{p(x_{1:T}(i-1),y_{1:T})}$ set $X_{1:T}(i) = X^*_{1:T}$, otherwise set $X_{1:T}(i) = X_{1:T}(i-1)$.

Very simple but very inefficient.

For good performance, we need to select $q(x_{1:T}) \approx p(x_{1:T}|y_{1:T})$ but for highly nonlinear non-Gaussian models it is essentially impossible as soon as $T$ is large.
Standard practice consists of building an MCMC kernel updating each component $X_k$ individually: this typically leads to slow mixing algorithms.
Common Approaches and Limitations

- Standard practice consists of building an MCMC kernel updating each component $X_k$ individually: this typically leads to *slow mixing algorithms*.

- Many complex models (e.g. some Lévy-driven volatility models) are such that it is only possible to sample forward from the prior but impossible to evaluate it pointwise: how could you use MCMC for such models?
Sequential Monte Carlo aka Particle Filters

- SMC methods provide an alternative way to approximate $p(x_{1:T} \mid y_{1:T})$ and to compute $p(y_{1:T})$.  

  To sample from $p(x_{1:T} \mid y_{1:T})$, SMC proceeds sequentially by first approximating $p(x_{1} \mid y_{1})$ at time 1 then $p(x_{1:2} \mid y_{1:2})$ at time 2 and so on.  

  SMC methods approximate the distributions of interest via a cloud of $N$ particles which are propagated using Importance Sampling and Resampling steps.
SMC methods provide an alternative way to approximate $p(x_{1:T}|y_{1:T})$ and to compute $p(y_{1:T})$.

To sample from $p(x_{1:T}|y_{1:T})$, SMC proceeds sequentially by first approximating $p(x_1|y_1)$ at time 1 then $p(x_{1:2}|y_{1:2})$ at time 2 and so on.
SMC methods provide an alternative way to approximate $p(x_{1:T} | y_{1:T})$ and to compute $p(y_{1:T})$.

To sample from $p(x_{1:T} | y_{1:T})$, SMC proceeds sequentially by first approximating $p(x_{1} | y_{1})$ at time 1 then $p(x_{1:2} | y_{1:2})$ at time 2 and so on.

SMC methods approximate the distributions of interest via a cloud of $N$ particles which are propagated using *Importance Sampling* and *Resampling* steps.
Vanilla SMC - The Bootstrap Filter

At time 1

- Sample $X_1^{(i)} \sim \mu(x_1)$ and compute $W_1^{(i)} \propto g(y_1 | X_1^{(i)})$, 
  $\sum_{i=1}^{N} W_1^{(i)} = 1$.

At time $k$, $k > 1$
Vanilla SMC - The Bootstrap Filter

At time 1

- Sample $X_1^{(i)} \sim \mu(x_1)$ and compute $W_1^{(i)} \propto g\left(y_1 | X_1^{(i)} \right)$, $\sum_{i=1}^N W_1^{(i)} = 1$.
- Resample $N$ times from $\tilde{p}(x_1 | y_1) = \sum_{i=1}^N W_1^{(i)} \delta_{X_1^{(i)}}(x_1)$.

At time $k$, $k > 1$
Vanilla SMC - The Bootstrap Filter

At time 1

- Sample $X_1^{(i)} \sim \mu(x_1)$ and compute $W_1^{(i)} \propto g(y_1 | X_1^{(i)})$, $\sum_{i=1}^{N} W_1^{(i)} = 1$.
- Resample $N$ times from $\hat{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X_1^{(i)}}(x_1)$.

At time $k$, $k > 1$

- Sample $X_k^{(i)} \sim f(x_k | X_{k-1}^{(i)})$, set $X_{1:k}^{(i)} = (X_{1:k-1}^{(i)}, X_k^{(i)})$ and compute $W_k^{(i)} \propto g(y_k | X_k^{(i)})$, $\sum_{i=1}^{N} W_k^{(i)} = 1$. 
At time 1

- Sample $X_1^{(i)} \sim \mu (x_1)$ and compute $W_1^{(i)} \propto g \left( y_1 \mid X_1^{(i)} \right)$, $\sum_{i=1}^{N} W_1^{(i)} = 1$.
- Resample $N$ times from $\hat{p} (x_1 \mid y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X_1^{(i)}} (x_1)$.

At time $k$, $k > 1$

- Sample $X_k^{(i)} \sim \pi \left( x_k \mid X_{k-1}^{(i)} \right)$, set $X_{1:k}^{(i)} = \left( X_{1:k-1}^{(i)}, X_k^{(i)} \right)$ and compute $W_k^{(i)} \propto g \left( y_k \mid X_k^{(i)} \right)$, $\sum_{i=1}^{N} W_k^{(i)} = 1$.
- Resample $N$ times from $\hat{p} (x_{1:k} \mid y_{1:k}) = \sum_{i=1}^{N} W_k^{(i)} \delta_{X_{1:k}^{(i)}} (x_{1:k})$. 
At time $T$, we obtain the following approximation of the posterior of interest $p(x_{1:T}|y_{1:T})$

$$
\hat{p}(x_{1:T}|y_{1:T}) = \sum_{i=1}^{N} W_{T}^{(i)} \delta_{x_{1:T}^{(i)}}(x_{1:T})
$$

and an approximation of $p(y_{1:T})$ given by

$$
\hat{p}(y_{1:T}) = \hat{p}(y_{1}) \prod_{k=2}^{T} \hat{p}(y_{k}|y_{1:k-1}) = \prod_{k=1}^{T} \left( \frac{1}{N} \sum_{i=1}^{N} g(y_{k}|X_{k}^{(i)}) \right).
$$
At time $T$, we obtain the following approximation of the posterior of interest $p(x_{1:T} \mid y_{1:T})$

$$
\hat{p}(x_{1:T} \mid y_{1:T}) = \sum_{i=1}^{N} W_{T}^{(i)} \delta_{X_{1:T}^{(i)}}(x_{1:T})
$$

and an approximation of $p(y_{1:T})$ given by

$$
\hat{p}(y_{1:T}) = \hat{p}(y_{1}) \prod_{k=2}^{T} \hat{p}(y_{k} \mid y_{1:k-1}) = \prod_{k=1}^{T} \left( \frac{1}{N} \sum_{i=1}^{N} g(y_{k} \mid X_{k}^{(i)}) \right).
$$

These approximations are asymptotically (i.e. $N \rightarrow \infty$) consistent under very weak assumptions.
Under *mixing assumptions* (Del Moral, 2004), we have

\[ \| \mathcal{L}(X_{1:T}) - p(x_{1:T} \mid y_{1:T}) \|_{tv} \leq C \frac{T}{N} \]

where \( X_{1:T} \sim \hat{p}(x_{1:T} \mid y_{1:T}) \).
Some Theoretical Results

- Under *mixing assumptions* (Del Moral, 2004), we have

\[ \| \text{law} (X_{1:T}) - p (x_{1:T} | y_{1:T}) \|_{tv} \leq C \frac{T}{N} \]

where \( X_{1:T} \sim \hat{p} (x_{1:T} | y_{1:T}) \).

- Under *mixing assumptions* (Chopin, 2004; Del Moral, 2004) we also have

\[ \frac{\mathbb{V} [\hat{p} (y_{1:T})]}{p^2 (y_{1:T})} \leq D \frac{T}{N}. \]
Some Theoretical Results

- Under *mixing assumptions* (Del Moral, 2004), we have
  \[ \| \mathcal{L} \text{aw} (X_{1:T}) - p (x_{1:T} \mid y_{1:T}) \|_{tv} \leq C \frac{T}{N} \]

  where \( X_{1:T} \sim \hat{p} (x_{1:T} \mid y_{1:T}) \).

- Under *mixing assumptions* (Chopin, 2004; Del Moral, 2004) we also have
  \[ \frac{\mathbb{V} \left[ \hat{p} (y_{1:T}) \right]}{p^2 (y_{1:T})} \leq D \frac{T}{N}. \]

- Loosely speaking, the performance of SMC only degrade linearly with time instead of exponentially for naive approaches.
‘Idea’: Use the output of our SMC algorithm to define ‘good’ high-dimensional proposal distributions for MCMC.

---

Problem:

\[ X_1:T \sim b_p(x_1:T \mid y_1:T) \]

Then

\[ L_{aw}(X_1:T) = q(x_1:T) = E \sum_{i=1}^{\infty} W(i) T \delta_{X(i)}(x_1:T) !. \]

However, \( q(x_1:T) \) does not admit a closed-form and so we cannot use directly the MH algorithm to correct for the discrepancy between \( q(x_1:T) \) and \( p(x_1:T \mid y_1:T) \). Hence the need for a new methodology....
‘Idea’: Use the output of our SMC algorithm to define ‘good’ high-dimensional proposal distributions for MCMC.

Problem: Consider \( X_{1:T} \sim \hat{p}(x_{1:T} \mid y_{1:T}) \) then

\[
\text{Law} (X_{1:T}) := q(x_{1:T}) = E \left( \sum_{i=1}^{N} W_T^{(i)} \delta_{X_{1:T}^{(i)}} (x_{1:T}) \right).
\]

However, \( q(x_{1:T}) \) does not admit a closed-form and so we cannot use directly the MH algorithm to correct for the discrepancy between \( q(x_{1:T}) \) and \( p(x_{1:T} \mid y_{1:T}) \).
‘Idea’: Use the output of our SMC algorithm to define ‘good’ high-dimensional proposal distributions for MCMC.

**Problem:** Consider $X_{1:T} \sim \hat{p}(x_{1:T} | y_{1:T})$ then

$$\text{Law} (X_{1:T}) := q(x_{1:T}) = \mathbb{E} \left( \sum_{i=1}^{N} W_T^{(i)} \delta_{X_{1:T}^{(i)}} (x_{1:T}) \right).$$

However, $q(x_{1:T})$ does not admit a closed-form and so we cannot use directly the MH algorithm to correct for the discrepancy between $q(x_{1:T})$ and $p(x_{1:T} | y_{1:T})$.

Hence the need for a new methodology....
At iteration 1

- Run an SMC algorithm to obtain \( \hat{p}^{(1)}(x_{1:T} | y_{1:T}) \) and \( \hat{p}^{(1)}(y_{1:T}) \).

At iteration \( i; i \geq 2 \)
At iteration 1

- Run an SMC algorithm to obtain \( \hat{p}^{(1)} (x_{1:T} | y_{1:T}) \) and \( \hat{p}^{(1)} (y_{1:T}) \).
- Sample \( X_{1:T} (1) \sim \hat{p}^{(1)} (x_{1:T} | y_{1:T}) \).

At iteration \( i; i \geq 2 \)
At iteration 1

- Run an SMC algorithm to obtain $\hat{p}^{(1)}(x_{1:T} | y_{1:T})$ and $\hat{p}^{(1)}(y_{1:T})$.
- Sample $X_{1:T}(1) \sim \hat{p}^{(1)}(x_{1:T} | y_{1:T})$.

At iteration $i; i \geq 2$

- Run an SMC algorithm to obtain $\hat{p}^*(x_{1:T} | y_{1:T})$ and $\hat{p}^*(y_{1:T})$. 
Particle MH Sampler

**At iteration 1**

- Run an SMC algorithm to obtain \( \hat{p}^{(1)} (x_{1:T} | y_{1:T}) \) and \( \hat{p}^{(1)} (y_{1:T}) \).
- Sample \( X_{1:T} (1) \sim \hat{p}^{(1)} (x_{1:T} | y_{1:T}) \).

**At iteration \( i; \ i \geq 2 \)**

- Run an SMC algorithm to obtain \( \hat{p}^* (x_{1:T} | y_{1:T}) \) and \( \hat{p}^* (y_{1:T}) \).
- Sample \( X^{*}_{1:T} \sim \hat{p}^* (x_{1:T} | y_{1:T}) \).
Particle MH Sampler

At iteration 1

- Run an SMC algorithm to obtain $\hat{p}^{(1)}(x_{1:T}|y_{1:T})$ and $\hat{p}^{(1)}(y_{1:T})$.
- Sample $X_{1:T}(1) \sim \hat{p}^{(1)}(x_{1:T}|y_{1:T})$.

At iteration $i; i \geq 2$

- Run an SMC algorithm to obtain $\hat{p}^*(x_{1:T}|y_{1:T})$ and $\hat{p}^*(y_{1:T})$.
- Sample $X^*_{1:T} \sim \hat{p}^*(x_{1:T}|y_{1:T})$.
- With probability

$$1 \wedge \frac{\hat{p}^*(y_{1:T})}{\hat{p}^{(i-1)}(y_{1:T})}$$

set $X_{1:T}(i) = X^*_{1:T}$ and $\hat{p}^{(i)}(y_{1:T}) = \hat{p}^*(y_{1:T})$, otherwise set $X_{1:T}(i) = X_{1:T}(i-1)$ and $\hat{p}^{(i)}(y_{1:T}) = \hat{p}^{(i-1)}(y_{1:T})$. 
**Proposition.** For any \( N \geq 1 \) the PMH sampler is an independent MH sampler defined on the extended space 
\[
\{1, \ldots, N\} \times \mathcal{X}^{TN} \times \{1, \ldots, N\}^{(T-1)N}
\]
with a target density 
\[
\tilde{p} \left( k, x_{1:1}^{1:N}, \ldots, x_{1:T}^{1:N}, i_{1}^{1:N}, \ldots, i_{T-1}^{1:N} \right)
\]
which admits a conditional distribution for 
\[
X_{1:T}^{K} = (X_{1}^{I_{1}^{K}}, X_{2}^{I_{2}^{K}}, \ldots, X_{T-1}^{I_{T-1}^{K}}, X_{T}^{K})
\]
equal to 
\[
p \left( x_{1:T} \mid y_{1:T} \right).
\]
\( \{I_{1}^{K}, I_{2}^{K}, \ldots, I_{T-1}^{K}\} \) corresponds to the ancestral lineage of the path 
\(X_{1:T}^{K} \).
How to sample from

\[ \tilde{p} \left( k, x_{1:T}, i_{1:T-1} \right) \]
How to sample from

\[ \tilde{p} \left( k, x_{1:1}^{1:N}, \ldots, x_{T}^{1:N}, i_{1:1}^{1:N}, \ldots, i_{T-1}^{1:N} \right) \]

Sample \((I_{1}^{K}, I_{2}^{K}, \ldots, I_{T-1}^{K}, K)\) from a uniform distribution on \(\{1, \ldots, N\}^T\).
Structure of the Artificial Extended Target Distribution

- How to sample from
  \[ \tilde{p} \left( k, x_{1:T}^1, \ldots, x_{1:T}^N, i_{1:T}^1, \ldots, i_{1:T-1}^N \right) \]

- Sample \((I_1^K, I_2^K, \ldots, I_{T-1}^K, K)\) from a uniform distribution on \(\{1, \ldots, N\}^T\).

- Sample \(X_{1:T}^K = (X_{1}^{I_1^K}, X_{2}^{I_2^K}, \ldots, X_{T-1}^{I_{T-1}^K}, X_T^K)\) from \(p \left( x_{1:T} \mid y_{1:T} \right)\). (We do not know how to do this, this is why we use MH).
Structure of the Artificial Extended Target Distribution

- How to sample from

\[ \tilde{p} \left( k, x_1^{1:N}, ..., x_T^{1:N}, i_1^{1:N}, ..., i_{T-1}^{1:N} \right) \]

- Sample \((I_K^1, I_K^2, ..., I_K^{T-1}, K)\) from a uniform distribution on \(\{1, ..., N\}^T\).

- Sample \(X_{1:T}^K = (X_1^{I_1^K}, X_2^{I_2^K}, ..., X_{T-1}^{I_{T-1}^K}, X_T^K)\) from \(p(\mathbf{x}_{1:T} | y_{1:T})\). (We do not know how to do this, this is why we use MH).

- Run a conditional SMC algorithm compatible with \(X_{1:T}^K\) and its ancestral lineage \((I_1^K, I_2^K, ..., I_{T-1}^K, K)\).
**Conditional SMC Algorithm**

*At time 1*

- For \( i \neq I_1^K \), sample \( X_1^{(i)} \sim \mu(x_1) \) and compute \( W_1^{(i)} \propto g \left( y_1 | X_1^{(i)} \right) \), \( \sum_{i=1}^N W_1^{(i)} = 1 \).

*At time n, n > 1*
Conditional SMC Algorithm

At time 1

- For \( i \neq I^K_1 \), sample \( X^{(i)}_1 \sim \mu (x_1) \) and compute \( W_1^{(i)} \propto g \left( y_1 | X^{(i)}_1 \right) \), \( \sum_{i=1}^{N} W_1^{(i)} = 1 \).
- Keep \( X^{(I^K_1)}_1 \) and resample \( N - 1 \) times from \( \hat{p} (x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X^{(i)}_1} (x_1) \).

At time \( n, n > 1 \)
Conditional SMC Algorithm

At time 1

- For $i \neq I_1^K$, sample $X_1^{(i)} \sim \mu(x_1)$ and compute $W_1^{(i)} \propto g(y_1 | X_1^{(i)})$, $\sum_{i=1}^{N} W_1^{(i)} = 1$.
- Keep $X_1^{(I_1^K)}$ and resample $N - 1$ times from $\hat{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X_1^{(i)}}(x_1)$.

At time $n, n > 1$

- For $i \neq I_n^K$, sample $X_n^{(i)} \sim f(x_n | X_{n-1}^{(i)})$, set $X_{1:n}^{(i)} = (X_{1:n-1}^{(i)}, X_n^{(i)})$ and compute $W_n^{(i)} \propto g(y_n | X_n^{(i)})$, $\sum_{i=1}^{N} W_n^{(i)} = 1$. 

Conditional SMC Algorithm

At time 1

- For $i \neq I_1^K$, sample $X_1^{(i)} \sim \mu(x_1)$ and compute $W_1^{(i)} \propto g(y_1 | X_1^{(i)})$, $\sum_{i=1}^{N} W_1^{(i)} = 1$.
- Keep $X_1^{(I_1^K)}$ and resample $N - 1$ times from $\hat{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X_1^{(i)}}(x_1)$.

At time $n$, $n > 1$

- For $i \neq I_n^K$, sample $X_n^{(i)} \sim f(x_n | X_{n-1}^{(i)})$, set $X_{1:n}^{(i)} = (X_{1:n-1}^{(i)}, X_n^{(i)})$ and compute $W_n^{(i)} \propto g(y_n | X_n^{(i)})$, $\sum_{i=1}^{N} W_n^{(i)} = 1$.
- Keep $X_{1:n}^{(I_n^K)}$ and resample $N - 1$ times from $\hat{p}(x_{1:k} | y_{1:k}) = \sum_{i=1}^{N} W_k^{(i)} \delta_{X_{1:k}^{(i)}}(x_{1:k})$. 
The ‘standard’ estimate of $\int f(x_{1:T}) p(x_{1:T} \mid y_{1:T}) \, dx_{1:T}$ for $L$ MCMC iterations is

$$\frac{1}{L} \sum_{i=1}^{L} f(X_{1:T}(i)).$$
The ‘standard’ estimate of \( \int f(x_1:T)p(x_1:T|y_1:T)dx_1:T \) for \( L \) MCMC iterations is

\[
\frac{1}{L} \sum_{i=1}^{L} f(X_{1:T}(i)).
\]

We generate \( N \) particles at each iteration \( i \) of the MCMC algorithm to decide whether to accept or reject one single candidate. This seems terribly wasteful.
Waste Recycling

- The ‘standard’ estimate of \( \int f(x_{1:T}) p(x_{1:T} | y_{1:T}) \, dx_{1:T} \) for \( L \) MCMC iterations is

\[
\frac{1}{L} \sum_{i=1}^{L} f(X_{1:T}(i)).
\]

- We generate \( N \) particles at each iteration \( i \) of the MCMC algorithm to decide whether to accept or reject one single candidate. This seems terribly wasteful.

- It is possible to use all the particles by considering the estimate

\[
\frac{1}{L} \sum_{i=1}^{L} \left( \sum_{k=1}^{N} W_T^{(k)}(i) f(X_{1:T}^{(k)}(i)) \right).
\]
The ‘standard’ estimate of \( \int f(x_{1:T}) p(x_{1:T} \mid y_{1:T}) \, dx_{1:T} \) for \( L \) MCMC iterations is

\[
\frac{1}{L} \sum_{i=1}^{L} f(X_{1:T}(i)).
\]

We generate \( N \) particles at each iteration \( i \) of the MCMC algorithm to decide whether to accept or reject one single candidate. This seems terribly wasteful.

It is possible to use all the particles by considering the estimate

\[
\frac{1}{L} \sum_{i=1}^{L} \left( \sum_{k=1}^{N} \mathcal{W}_{T}^{(k)}(i) f(X_{1:T}^{(k)}(i)) \right).
\]

We can even recycle the populations of particles rejected.
Extensions

- If $T$ is too large compared to the number $N$ of particles, then the SMC approximation might be poor and results in an inefficient MH algorithm.
Extensions

- If $T$ is too large compared to the number $N$ of particles, then the SMC approximation might be poor and results in an inefficient MH algorithm.
- We can use a mixture/composition of PMH transition probabilities that update subblocks of the form $X_{a:b}$ for $1 \leq a < b \leq T$, effectively targeting conditional distributions of the type $p\left(X_{a:b}|X_{a-1}, X_{b+1}, Y_{a:b}\right)$. 

Extensions

- If $T$ is too large compared to the number $N$ of particles, then the SMC approximation might be poor and results in an inefficient MH algorithm.
- We can use a mixture/composition of PMH transition probabilities that update subblocks of the form $X_{a:b}$ for $1 \leq a < b \leq T$, effectively targeting conditional distributions of the type $p(X_{a:b} | X_{a-1}, X_{b+1}, Y_{a:b})$.
- Given we know the exact structure of the extended target distribution, we can rejuvenate some of the particles conditional upon the others for computational purposes.
Consider the following model

\[ X_k = \frac{1}{2} X_{k-1} + 25 \frac{X_{k-1}}{1 + X_{k-1}^2} + 8 \cos 1.2k + V_k, \]
\[ Y_k = \frac{X_k^2}{20} + W_k \]

where \( V_k \sim \mathcal{N}(0, \sigma_v^2) \), \( W_k \sim \mathcal{N}(0, \sigma_w^2) \) and \( X_1 \sim \mathcal{N}(0, 5^2) \).
Consider the following model

\[ X_k = \frac{1}{2} X_{k-1} + 25 \frac{X_{k-1}}{1 + X_{k-1}^2} + 8 \cos 1.2k + V_k, \]

\[ Y_k = \frac{X_k^2}{20} + W_k \]

where \( V_k \sim \mathcal{N}(0, \sigma_v^2) \), \( W_k \sim \mathcal{N}(0, \sigma_w^2) \) and \( X_1 \sim \mathcal{N}(0, 5^2) \).

We first compare PMCMC to CBMC (Frenkel et al., 1992) for \( \sigma_v^2 = 10 \) and \( \sigma_w^2 = 0.5 \) in terms of acceptance rate of the proposal for various \( N \) and \( T \). Note that PMCMC is marginally more expensive to implement.
Acceptance probabilities for PMH (left) and CBMC (right) as a function of $T$ and $N$. Each point was computed using 250,000 runs.
In numerous scenarios, \( \mu(x) \), \( f(x' | x) \) and \( g(y | x) \) are not known exactly but depend on an unknown parameter \( \theta \) and we note \( \mu_\theta(x) \), \( f_\theta(x' | x) \) and \( g_\theta(y | x) \).
A More Realistic Problem

- In numerous scenarios, $\mu(x)$, $f(x'|x)$ and $g(y|x)$ are not known exactly but depend on an unknown parameter $\theta$ and we note $\mu_{\theta}(x)$, $f_{\theta}(x'|x)$ and $g_{\theta}(y|x)$.

- We set a prior $p(\theta)$ on $\theta$ and we are interested in

$$p(\theta, x_{1:T}|y_{1:T}) = \frac{p(x_{1:T}, y_{1:T}|\theta)p(\theta)}{p(y_{1:T})}$$
A Block Gibbs Sampler

To sample from $p(\theta, x_{1:T} | y_{1:T})$, an MCMC strategy consists of using the following block Gibbs sampler.

- Sample $X_{1:T}(i) \sim p(x_{1:T} | y_{1:T}, \theta(i))$.
- Sample $\theta(i) \sim p(\theta | x_{1:T}(i), y_{1:T})$.

Problems: We do not know how to sample from $p(\theta, x_{1:T} | y_{1:T})$ so we typically use instead an MH one-at-a-time algorithm.
To sample from $p(\theta, x_{1:T} | y_{1:T})$, an MCMC strategy consists of using the following block Gibbs sampler.

- **At iteration $i$**
  - Sample $X_{1:T}(i) \sim p(x_{1:T} | y_{1:T}, \theta(i-1))$.
  - Sample $\theta(i) \sim p(\theta | y_{1:T}, X_{1:T}(i))$. 

Problems: We do not know how to sample from $p(x_{1:T} | y_{1:T}, \theta)$ so we typically use instead an MH one-at-a-time algorithm.
To sample from $p(\theta, x_{1:T} | y_{1:T})$, an MCMC strategy consists of using the following block Gibbs sampler.

**At iteration $i$**

Sample $X_{1:T}(i) \sim p(x_{1:T} | y_{1:T}, \theta(i - 1))$.
Sample $\theta(i) \sim p(\theta | y_{1:T}, X_{1:T}(i))$.

**Problems:** We do not know how to sample from $p(x_{1:T} | y_{1:T}, \theta)$ so we typically use instead an MH one-at-a time algorithm.
At iteration 1

- Set $\theta(1)$ and $X_{1:T}(1)$ randomly, denote $I(1)$ its ancestral lineage.

At iteration $i; i \geq 2$
Particle Gibbs Sampler

At iteration 1
- Set $\theta(1)$ and $X_{1:T}(1)$ randomly, denote $I(1)$ its ancestral lineage.

At iteration $i; i \geq 2$
- Sample $\theta(i) \sim p(\theta|X_{1:T}(i-1), y_{1:T})$. 
Particle Gibbs Sampler

At iteration 1

- Set $\theta(1)$ and $X_{1:T}(1)$ randomly, denote $I(1)$ its ancestral lineage.

At iteration $i; i \geq 2$

- Sample $\theta(i) \sim p(\theta | X_{1:T}(i-1), y_{1:T})$.
- Run a conditional SMC algorithm for $\theta(i)$ consistent with $X_{1:T}(i-1), I(i-1)$. 
Particle Gibbs Sampler

**At iteration 1**

- Set $\theta^{(1)}$ and $X_{1:T}^{(1)}$ randomly, denote $I^{(1)}$ its ancestral lineage.

**At iteration $i$; $i \geq 2$**

- Sample $\theta^{(i)} \sim p(\theta|X_{1:T}^{(i-1)}, y_{1:T})$.
- Run a conditional SMC algorithm for $\theta^{(i)}$ consistent with $X_{1:T}^{(i-1)}, I^{(i-1)}$.
- Sample $X_{1:T}^{(i)}$ from the resulting approximation and denote $I^{(i)}$ its ancestral lineage.
**Proposition.** Assume the ‘ideal’ Gibbs sampler is irreducible and aperiodic then for $N \geq 2$ the particle Gibbs sampler is irreducible and aperiodic with a target density on the extended space

$\Theta \times \{1, \ldots, N\} \times \mathcal{X}^{TN} \times \{1, \ldots, N\}^{(T-1)N}$

$$\tilde{p} \left( k, x_1^{1:N}, \ldots, x_T^{1:N}, i_1^{1:N}, \ldots, i_{T-1}^{1:N}, \theta \right)$$

which admits a conditional distribution for

$$(X^K_{1:T}, \theta) = (X_1^{I^K_1}, X_2^{I^K_2}, \ldots, X_{T-1}^{I^K_{T-1}}, X_T^{I^K_T}, \theta)$$

equal to $p \left( x_{1:T}, \theta | y_{1:T} \right)$.
We now consider the case where $\theta = (\sigma^2_v, \sigma^2_w)$ is unknown and random with vague inverse-Gamma priors.
We now consider the case where $\theta = (\sigma_v^2, \sigma_w^2)$ is unknown and random with vague inverse-Gamma priors.

We compare the Particle Gibbs sampler to an MH one-at-a time using the same proposal distribution.
We now consider the case where \( \theta = (\sigma^2_v, \sigma^2_w) \) is unknown and random with vague inverse-Gamma priors.

We compare the Particle Gibbs sampler to an MH one-at-a time using the same proposal distribution.

We update \( N \) times the latent process in the MH one-at-a time before sampling \( \theta \) to perform a reasonably fair comparison with PMCMC.
• We now consider the case where $\theta = (\sigma_v^2, \sigma_w^2)$ is unknown and random with vague inverse-Gamma priors.

• We compare the Particle Gibbs sampler to an MH one-at-a-time using the same proposal distribution.

• We update $N$ times the latent process in the MH one-at-a-time before sampling $\theta$ to perform a reasonably fair comparison with PMCMC.

• In this example, the MH one-at-a-time does not provide reliable results and is very sensitive to initialization.
Autocorrelation of the output of the PG and MH algorithms.
Estimates of $p(\sigma_V | y_{1:T})$ (left) and $p(\sigma_W | y_{1:T})$ (right) for particle Gibbs (top) and Gibbs (bottom)
The Gibbs sampler (and hence the particle Gibbs sampler) will be mixing slowly if $X_{1:T}$ and $\theta$ are highly correlated.

Problem: We do not know $p(y_{1:T} \mid \theta) = \int p(x_{1:T}, y_{1:T} \mid \theta) \, dx_{1:T}$ analytically.
The Gibbs sampler (and hence the particle Gibbs sampler) will be mixing slowly if $X_{1:T}$ and $\theta$ are highly correlated.

To sample from $p(\theta|y_{1:T})$, an MCMC strategy consists of using the following marginal MH algorithm which integrates out $X_{1:T}$.

\[ \text{At iteration } i \]

\[ \text{Sample } \theta \sim q(\theta|\theta(i-1)). \]

\[ \text{W.p. } 1 \]

\[ \frac{p(y_{1:T}|\theta(i))}{p(y_{1:T}|	heta(i-1))} \frac{q(\theta(i)|\theta(i-1))}{q(\theta(i-1)|\theta(i))}. \]

\[ \text{set } \theta(i) = \theta(i), \text{ otherwise set } \theta(i) = \theta(i-1). \]
The Gibbs sampler (and hence the particle Gibbs sampler) will be mixing slowly if \(X_{1:T}\) and \(\theta\) are highly correlated.

To sample from \(p(\theta \mid y_{1:T})\), an MCMC strategy consists of using the following marginal MH algorithm which integrates out \(X_{1:T}\).

At iteration \(i\)

- Sample \(\theta^* \sim q(\theta \mid \theta(i-1))\).
- W.p. \(1 \wedge \frac{p(y_{1:T} \mid \theta^*)p(\theta^*)}{q(\theta^* \mid \theta(i-1))p(y_{1:T} \mid \theta(i-1))p(\theta(i-1))} \frac{q(\theta(i-1) \mid \theta^*)}{p(\theta(i-1) \mid \theta(i-1)p(\theta(i-1))}, \) set \(\theta(i) = \theta^*\), otherwise set \(\theta(i) = \theta(i-1)\).
A Marginal Metropolis-Hastings Algorithm

The Gibbs sampler (and hence the particle Gibbs sampler) will be mixing slowly if \( X_{1:T} \) and \( \theta \) are highly correlated.

To sample from \( p (\theta | y_{1:T}) \), an MCMC strategy consists of using the following marginal MH algorithm which integrates out \( X_{1:T} \).

At iteration \( i \)
Sample \( \theta^* \sim q (\theta | \theta (i-1)) \).
W.p. \( 1 \wedge \frac{p(y_{1:T} | \theta^*)p(\theta^*)}{q(\theta^* | \theta(i-1))} \frac{q(\theta(i-1) | \theta^*)}{p(y_{1:T} | \theta(i-1))p(\theta(i-1))} \) set \( \theta (i) = \theta^* \), otherwise set \( \theta (i) = \theta (i-1) \).

Problem: We do not know \( p (y_{1:T} | \theta) = \int p (x_{1:T}, y_{1:T} | \theta) dx_{1:T} \) analytically.
Particle Marginal MH Sampler

At iteration 1

- Set $\theta(1)$ and run an SMC algorithm to obtain $\hat{p}(y_{1:T} \mid \theta(1))$.

At iteration $i; i \geq 1$
Particle Marginal MH Sampler

At iteration 1

- Set $\theta^{(1)}$ and run an SMC algorithm to obtain $\hat{p}(y_{1:T} | \theta^{(1)})$.

At iteration $i; i \geq 1$

- Sample $\theta^* \sim q(\theta | \theta^{(i-1)})$ and run an SMC algorithm to obtain $\hat{p}(y_{1:T} | \theta^*)$. 

Particle Marginal MH Sampler

At iteration 1
- Set $\theta(1)$ and run an SMC algorithm to obtain $\hat{p}(y_{1:T}|\theta(1))$.

At iteration $i; i \geq 1$
- Sample $\theta^* \sim q(\theta|\theta(i-1))$ and run an SMC algorithm to obtain $\hat{p}(y_{1:T}|\theta^*)$.
- With probability
  $$1 \wedge \frac{\hat{p}(y_{1:T}|\theta^*) p(\theta^*)}{\hat{p}(y_{1:T}|\theta(i-1)) p(\theta(i-1))} \frac{q(\theta(i-1)|\theta^*)}{q(\theta^*|\theta(i-1))}$$
  set $\theta(i) = \theta^*$, otherwise set $\theta(i) = \theta(i-1)$. 
This algorithm has been proposed independently in economics (Fernandez-Villaverde & Rubio-Ramirez, Rev. Economic Studies, 2007).
This algorithm has been proposed independently in economics (Fernandez-Villaverde & Rubio-Ramirez, Rev. Economic Studies, 2007).

The obvious question is what is the invariant distribution of this MCMC algorithm?
Proposition. Assume the ‘ideal’ MH sampler is irreducible and aperiodic then for $N \geq 1$ the particle marginal MH sampler is irreducible and aperiodic with a target density on the extended space $\Theta \times \{1, \ldots, N\} \times \mathcal{X}^{TN} \times \{1, \ldots, N\}^{(T-1)N}$

$$\tilde{p} \left( k, x_1^{1:N}, \ldots, x_T^{1:N}, i_1^{1:N}, \ldots, i_{T-1}^{1:N}, \theta \right)$$

which admits as a marginal the target distribution $p ( \theta | y_{1:T} )$. 
This methodology is by no means limited to state-space models and in particular there is no need for any Markovian assumption.
This methodology is by no means limited to state-space models and in particular there is no need for any Markovian assumption.

Wherever SMC have been used, PMCMC can be used: self-avoiding random walks, contingency tables, mixture models etc.
Extensions

- This methodology is by no means limited to state-space models and in particular there is no need for any Markovian assumption.
- Wherever SMC have been used, PMCMC can be used: self-avoiding random walks, contingency tables, mixture models etc.
- Sophisticated SMC algorithms can also be used including ‘clever’ importance distributions and resampling schemes.
Extensions

- This methodology is by no means limited to state-space models and in particular there is no need for any Markovian assumption.
- Wherever SMC have been used, PMCMC can be used: self-avoiding random walks, contingency tables, mixture models etc.
- Sophisticated SMC algorithms can also be used including ‘clever’ importance distributions and resampling schemes.
- Theoretical results of (Andrieu & Roberts, Ann. Stats, to appear) can be extended to this setup.
Consider the following hierarchical model

\[ G \sim DP(\alpha, G_0), \]

\[ U_n | G \sim i.i.d. G, \quad Y_n | U_n \sim g_{U_n}(\cdot) \]

where \( DP(\alpha, G_0) \) is a Dirichlet process of base measure \( G_0 \) and scale parameter \( \alpha \).
Consider the following hierarchical model

\[ G \sim DP(\alpha, G_0), \]

\[ U_n \mid G \overset{i.i.d.}{\sim} G, \quad Y_n \mid U_n \sim g_{U_n}(\cdot) \]

where \( DP(\alpha, G_0) \) is a Dirichlet process of base measure \( G_0 \) and scale parameter \( \alpha \).

By integrating out \( G \), we can equivalently reformulate the model by introducing a vector of cluster labels \( X_{1:n} \) which satisfy

\[
\Pr(X_n = j \mid x_{1:n-1}) = \begin{cases} 
m_{n-1}^j / (n - 1 + \alpha) & \text{for } j = 1, \ldots, k_{n-1} \\
\alpha / (n - 1 + \alpha) & j = k_{n-1} + 1
\end{cases}
\]

where \( k_{n-1} \) is the number of clusters in the assignment \( x_{1:n-1} \) and \( m_{n-1}^j \) is the number of observations that \( x_{1:n-1} \) assigns to cluster \( j \).
Consider the following hierarchical model

\[ \mathbb{G} \sim DP(\alpha, \mathbb{G}_0), \]

\[ U_n \mid \mathbb{G} \text{ i.i.d. } \mathbb{G}, \quad Y_n \mid U_n \sim g_{U_n}(\cdot) \]

where \( DP(\alpha, \mathbb{G}_0) \) is a Dirichlet process of base measure \( \mathbb{G}_0 \) and scale parameter \( \alpha \).

By integrating out \( \mathbb{G} \), we can equivalently reformulate the model by introducing a vector of cluster labels \( X_{1:n} \) which satisfy

\[
\Pr(X_n = j \mid x_{1:n-1}) = \begin{cases} 
m^j_{n-1} / (n - 1 + \alpha) & \text{for } j = 1, \ldots, k_{n-1} \\ 
\alpha / (n - 1 + \alpha) & j = k_{n-1} + 1
\end{cases}
\]

where \( k_{n-1} \) is the number of clusters in the assignment \( x_{1:n-1} \) and \( m^j_{n-1} \) is the number of observations that \( x_{1:n-1} \) assigns to cluster \( j \).

Additionally we have

\[ \theta_k \text{ i.i.d. } \mathbb{G}_0 \text{ and } Y_n \mid \theta_{X_n} \sim g_{\theta_{X_n}}(\cdot). \]
We focus here on the following Gaussian mixture model for real-valued observations where \( \theta = (\mu, \sigma^2) \) and

\[
G_0 (\mu, \sigma^2) = IG (\sigma^2; a, b) N (\mu; \eta, \tau \sigma^2) .
\]

and for the scale parameter, we use

\[
\alpha \sim G (c, d) .
\]
We focus here on the following Gaussian mixture model for real-valued observations where $\theta = (\mu, \sigma^2)$ and

$$G_0(\mu, \sigma^2) = IG(\sigma^2; a, b) \mathcal{N}(\mu; \eta, \tau \sigma^2).$$

and for the scale parameter, we use

$$\alpha \sim \mathcal{G}(c, d).$$

We want to estimate $p(x_{1:T}, \theta_{1:kT}, \alpha | y_{1:T})$ using particle MCMC.
We focus here on the following Gaussian mixture model for real-valued observations where $\theta = (\mu, \sigma^2)$ and

$$G_0 (\mu, \sigma^2) = IG (\sigma^2 ; a, b) \mathcal{N} (\mu; \eta, \tau \sigma^2) .$$

and for the scale parameter, we use

$$\alpha \sim \mathcal{G} (c, d) .$$

We want to estimate $p (x_1:T, \theta_1:k_T, \alpha | y_1:T)$ using particle MCMC.

We applied our algorithm to a large simulated dataset of 10,000 real-valued observations generated from the Dirichlet process Gaussian mixture model with $\alpha = 1$. 
Average acceptance rate of various versions of the PMH algorithm for a varying number of data points and $N = 100$
Autocorrelation function for parameter $\alpha$ (left) and estimate of $p(\alpha | y_{1:T})$ (right). The vertical line corresponds to the true value.
The main reason why it works is because (Del Moral, 2004)

$$\mathbb{E} [\hat{p}(y_{1:T} | \theta)] = p(y_{1:T} | \theta).$$

More generally, whenever you have a unknown quantity you can simply plug a positive unbiased estimate of it in your MCMC: the invariant distribution is 'exact'. Such ideas are implicitly used in Moller et al. (2004) for unknown normalizing constants, Approximate Bayesian Computation, the random weight PF of Fearnhead et al. (2008) etc. This idea 'works' in practice if the variance of your estimate is small. Particle Gibbs samplers of such schemes can be developed.
The Other Side of The Story

- The main reason why it works is because (Del Moral, 2004)

$$\mathbb{E} [\hat{p} (y_1:T | \theta)] = p (y_1:T | \theta).$$

- More generally, whenever you have an unknown quantity you can simply plug a positive unbiased estimate of it in your MCMC: the invariant distribution is ‘exact’.
The main reason why it works is because (Del Moral, 2004)

\[ \mathbb{E}[\hat{p}(y_{1:T} | \theta)] = p(y_{1:T} | \theta). \]

More generally, whenever you have an unknown quantity you can simply plug a positive unbiased estimate of it in your MCMC: the invariant distribution is ‘exact’.

Such ideas are implicitly used in Moller et al. (2004) for unknown normalizing constants, Approximate Bayesian Computation, the random weight PF of Fearnhead et al. (2008) etc.
The main reason why it works is because (Del Moral, 2004)

\[ \mathbb{E} [\hat{p} (y_{1:T} | \theta)] = p (y_{1:T} | \theta). \]

More generally, whenever you have an unknown quantity you can simply plug a positive unbiased estimate of it in your MCMC: the invariant distribution is ‘exact’.

Such ideas are implicitly used in Moller et al. (2004) for unknown normalizing constants, Approximate Bayesian Computation, the random weight PF of Fearnhead et al. (2008) etc.

This idea ‘works’ in practice if the variance of your estimate is small.
The main reason why it works is because (Del Moral, 2004)

\[ \mathbb{E} [\hat{p} (y_{1:T} | \theta)] = p (y_{1:T} | \theta). \]

More generally, whenever you have an unknown quantity you can simply plug a positive unbiased estimate of it in your MCMC: the invariant distribution is ‘exact’.

Such ideas are implicitly used in Moller et al. (2004) for unknown normalizing constants, Approximate Bayesian Computation, the random weight PF of Fearnhead et al. (2008) etc.

This idea ‘works’ in practice if the variance of your estimate is small.

Particle Gibbs samplers of such schemes can be developed.
PMCMC methods allow us to design ‘good’ high dimensional proposals based only on low dimensional proposals.
PMCMC methods allow us to design ‘good’ high dimensional proposals based only on low dimensional proposals.

PMCMC allow us to perform Bayesian inference for models for which only forward simulation is possible.
PMCMC methods allow us to design ‘good’ high dimensional proposals based only on low dimensional proposals.

PMCMC allow us to perform Bayesian inference for models for which only forward simulation is possible.

Extension to ‘dependent’ proposals in the spirit of MTM (Liu et al., JASA 2000) is feasible.
PMCMC methods allow us to design ‘good’ high dimensional proposals based only on low dimensional proposals.
PMCMC allow us to perform Bayesian inference for models for which only forward simulation is possible.
Extension to ‘dependent’ proposals in the spirit of MTM (Liu et al., JASA 2000) is feasible.
Many open questions...