Controlled sequential Monte Carlo

Jeremy Heng,
Department of Statistics, Harvard University

Joint work with Adrian Bishop (UTS, CSIRO), George Deligiannidis & Arnaud Doucet (Oxford)

Bayesian Computation for High-Dimensional Statistical Models
Institute for Mathematical Sciences, NUS
14 September 2018
Outline

1. State space models
2. Sequential Monte Carlo
3. Controlled sequential Monte Carlo
4. Bayesian parameter inference
5. Extensions and future work
Outline

1. State space models
2. Sequential Monte Carlo
3. Controlled sequential Monte Carlo
4. Bayesian parameter inference
5. Extensions and future work
Latent Markov chain

\[ X_0 \sim \mu, \]
State space models

Latent Markov chain

\[ X_0 \sim \mu, \quad X_t | X_{t-1} \sim f_t(X_{t-1}, \cdot), \quad t \in [1 : T] \]
State space models

Latent Markov chain

\[ X_0 \sim \mu, \quad X_t|X_{t-1} \sim f_t(X_{t-1}, \cdot), \quad t \in [1 : T] \]

Observations

\[ Y_t|X_{0:T} \sim g_t(X_t, \cdot), \quad t \in [0 : T] \]
State space models

Latent Markov chain

\[X_0 \sim \mu_{\theta}, \quad X_t|X_{t-1} \sim f_{t,\theta}(X_{t-1}, \cdot), \quad t \in [1 : T]\]

Observations

\[Y_t|X_{0:T} \sim g_{t,\theta}(X_t, \cdot), \quad t \in [0 : T]\]
<table>
<thead>
<tr>
<th>Google Scholar</th>
<th>&quot;state space models&quot;</th>
<th>Articles</th>
<th>About 72,100 results (0.03 sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Scholar</td>
<td>&quot;hidden markov models&quot;</td>
<td>Articles</td>
<td>About 382,000 results (0.12 sec)</td>
</tr>
</tbody>
</table>
State space models

Market volatility
Percentage points

Source: Bloomberg
State space models

Google Scholar

Articles

Google Scholar

Articles
State space models
Neuroscience example

- Joint work with Demba Ba, Harvard School of Engineering
Neuroscience example

- Joint work with Demba Ba, Harvard School of Engineering
- 3000 measurements $y_t \in \{0, \ldots, 50\}$ collected from a neuroscience experiment (Temereanca et al., 2008)
Neuroscience example

- Observation model

\[ Y_t | X_t \sim \text{Binomial}(50, \kappa(X_t)), \quad \kappa(u) = (1 + \exp(-u))^{-1} \]
Neuroscience example

• Observation model

\[ Y_t | X_t \sim \text{Binomial}(50, \kappa(X_t)), \quad \kappa(u) = (1 + \exp(-u))^{-1} \]

• Latent Markov chain

\[ X_0 \sim \mathcal{N}(0, 1), \quad X_t | X_{t-1} \sim \mathcal{N}(\alpha X_{t-1}, \sigma^2) \]
Neuroscience example

• Observation model

\[ Y_t \mid X_t \sim \text{Binomial}(50, \kappa(X_t)), \quad \kappa(u) = (1 + \exp(-u))^{-1} \]

• Latent Markov chain

\[ X_0 \sim \mathcal{N}(0, 1), \quad X_t \mid X_{t-1} \sim \mathcal{N}(\alpha X_{t-1}, \sigma^2) \]

• Unknown parameters

\[ \theta = (\alpha, \sigma^2) \in [0, 1] \times (0, \infty) \]
Objects of interest

- Online estimation: compute **filtering distributions**

\[ p(x_t | y_{0:t}, \theta), \quad t \geq 0 \]
Objects of interest

- **Online estimation:** compute **filtering distributions**
  
  \[ p(x_t | y_{0:t}, \theta), \quad t \geq 0 \]

- **Offline estimation:** compute **smoothing distribution**
  
  \[ p(x_{0:T} | y_{0:T}, \theta), \quad T \in \mathbb{N} \]

or **marginal likelihood**

\[ p(y_{0:T} | \theta) = \int_{\mathcal{X}^{T+1}} \mu_\theta(x_0) \prod_{t=1}^{T} f_{t,\theta}(x_{t-1}, x_t) \prod_{t=0}^{T} g_{t,\theta}(x_t, y_t) dx_{0:T} \]
Objects of interest

- **Online estimation:** compute **filtering distributions**
  \[
p(x_t | y_{0:t}, \theta), \quad t \geq 0
\]

- **Offline estimation:** compute **smoothing distribution**
  \[
p(x_{0:T} | y_{0:T}, \theta), \quad T \in \mathbb{N}
\]
  or **marginal likelihood**
  \[
p(y_{0:T} | \theta) = \int_{\mathcal{X}^{T+1}} \mu_\theta(x_0) \prod_{t=1}^{T} f_{t,\theta}(x_{t-1}, x_t) \prod_{t=0}^{T} g_{t,\theta}(x_t, y_t) dx_{0:T}
\]

- **Parameter inference**
  \[
  \arg \max_{\theta \in \Theta} p(y_{0:T} | \theta), \quad p(\theta | y_{0:T}) \propto p(\theta) p(y_{0:T} | \theta)
  \]
Outline

1. State space models
2. Sequential Monte Carlo
3. Controlled sequential Monte Carlo
4. Bayesian parameter inference
5. Extensions and future work
Sequential Monte Carlo

- Sequential Monte Carlo (SMC) aka bootstrap particle filter (BPF) recursively simulates an interacting particle system of size $N$

$$(X^1_t, \ldots, X^N_t), \quad t \in [0 : T]$$

Sequential Monte Carlo

• Sequential Monte Carlo (SMC) aka bootstrap particle filter (BPF) recursively simulates an interacting particle system of size $N$

$$(X^1_t, \ldots, X^N_t), \quad t \in [0 : T]$$

• Unbiased and consistent marginal likelihood estimator

$$\hat{p}(y_0:T | \theta) = \prod_{t=0}^{T} \left\{ \frac{1}{N} \sum_{n=1}^{N} g_{t,\theta}(X^n_t, y_t) \right\}$$

Sequential Monte Carlo

- Sequential Monte Carlo (SMC) aka bootstrap particle filter (BPF) recursively simulates an interacting particle system of size $N$

$$(X^1_t, \ldots, X^N_t), \quad t \in [0 : T]$$

- Unbiased and consistent marginal likelihood estimator

$$\hat{p}(y_{0:T} | \theta) = \prod_{t=0}^{T} \left\{ \frac{1}{N} \sum_{n=1}^{N} g_{t,\theta}(X^n_t, y_t) \right\}$$

- Consistent approximation of smoothing distribution

$$\frac{1}{N} \sum_{n=1}^{N} \varphi(X^n_{0:T}) \rightarrow \int \varphi(x_{0:T}) p(x_{0:T} | y_{0:T}, \theta) dx_{0:T}$$

as $N \rightarrow \infty$

For time $t = 0$ and particle $n \in [1 : N]$

sample $X^n_0 \sim \mu_\theta$
For time $t = 0$ and particle $n \in [1 : N]$

$$\text{weight } W_0^n \propto g_{0,\theta}(X_0^n, y_0)$$
For time $t = 0$ and particle $n \in [1 : N]$

sample ancestor $A_0^n \sim \mathcal{R} \left( W_0^1, \ldots, W_0^N \right)$, resampled particle: $X_0^{A_0^n}$
For time $t = 1$ and particle $n \in [1 : N]$

$$\text{sample } X_1^n \sim f_{1,\theta}(X_0^A_0, \cdot)$$
For time $t = 1$ and particle $n \in [1 : N]$

$$\text{weight } W_1^n \propto g_{1,\theta}(X_1^n, y_1)$$
For time $t = 1$ and particle $n \in [1 : N]$

sample ancestor $A_1^n \sim \mathcal{R} \left( \mathcal{W}_1^1, \ldots, \mathcal{W}_1^N \right)$, resampled particle: $X_1^{A_1^n}$
Repeat for time $t \in [2 : T]$. 
Repeat for time $t \in [2 : T]$. Note this is for a given $\theta$!
Cannot run Metropolis-Hastings algorithm to sample from

\[ p(\theta|y_0:T) \propto p(\theta)p(y_0:T|\theta) \]

For iteration \( i \geq 1 \)

1. Sample \( \theta^* \sim q(\theta^{(i-1)}, \cdot) \)

2. With probability

\[
\min \left\{1, \frac{p(\theta^*)p(y_0:T|\theta^*)q(\theta^*, \theta^{(i-1)})}{p(\theta^{(i-1)})p(y_0:T|\theta^{(i-1)})q(\theta^{(i-1)}, \theta^*)} \right\}
\]

set \( \theta^{(i)} = \theta^* \), otherwise set \( \theta^{(i)} = \theta^{(i-1)} \)
Particle marginal Metropolis-Hastings (PMMH)

For iteration $i \geq 1$

1. Sample $\theta^* \sim q(\theta^{(i-1)}, .)$
2. Compute $\hat{p}(y_0:T|\theta^*)$ with SMC

2. With probability

$$\min \left\{ 1, \frac{p(\theta^*) \hat{p}(y_0:T|\theta^*) q(\theta^*, \theta^{(i-1)})}{p(\theta^{(i-1)}) \hat{p}(y_0:T|\theta^{(i-1)}) q(\theta^{(i-1)}, \theta^*)} \right\}$$

set $\theta^{(i)} = \theta^*$ and $\hat{p}(y_0:T|\theta^{(i)}) = \hat{p}(y_0:T|\theta^*)$, otherwise set $\theta^{(i)} = \theta^{(i-1)}$ and $\hat{p}(y_0:T|\theta^{(i)}) = \hat{p}(y_0:T|\theta^{(i-1)})$

Particle marginal Metropolis-Hastings (PMMH)

- Exact approximation:

\[
\frac{1}{m} \sum_{i=1}^{m} \varphi(\theta^{(i)}) \rightarrow \int_{\Theta} \varphi(\theta)p(\theta|y_{0:T})d\theta
\]

as \( m \rightarrow \infty \), for any \( N \geq 1 \)
Particle marginal Metropolis-Hastings (PMMH)

- **Exact approximation:**

  \[
  \frac{1}{m} \sum_{i=1}^{m} \varphi(\theta^{(i)}) \to \int_{\Theta} \varphi(\theta) p(\theta | y_{0:T}) d\theta
  \]

  as \( m \to \infty \), for any \( N \geq 1 \)

- **Performance depends on the variance of \( \hat{\rho}(y_{0:T} | \theta) \)**

Particle marginal Metropolis-Hastings (PMMH)

- Exact approximation:

\[ \frac{1}{m} \sum_{i=1}^{m} \varphi(\theta^{(i)}) \rightarrow \int_{\Theta} \varphi(\theta) p(\theta | y_{0:T}) d\theta \]

as \( m \to \infty \), for any \( N \geq 1 \)

- Performance depends on the variance of \( \hat{p}(y_{0:T}|\theta) \)


- Account for computational cost to optimize efficiency

Particle marginal Metropolis-Hastings (PMMH)

- Exact approximation:

\[
\frac{1}{m} \sum_{i=1}^{m} \varphi(\theta^{(i)}) \rightarrow \int_{\Theta} \varphi(\theta)p(\theta|y_{0:T})d\theta
\]

as \( m \to \infty \), for any \( N \geq 1 \)

- Performance depends on the variance of \( \hat{p}(y_{0:T}|\theta) \)


- Account for computational cost to optimize efficiency


- Proposed method lowers variance of \( \hat{p}(y_{0:T}|\theta) \) at fixed cost
Relative variance of log-marginal likelihood estimator

$$\sigma^2 \mapsto \text{Var} \left[ \frac{\log \hat{p}(y_0:T | (\alpha, \sigma^2))}{\log p(y_0:T | (\alpha, \sigma^2))} \right], \quad \text{fix } \alpha = 0.99$$

with $N = 1024$
Neuroscience example: SMC performance

Relative variance of log-marginal likelihood estimator

\[ \sigma^2 \mapsto \text{Var} \left[ \frac{\log \hat{p}(y_{0:T} | (\alpha, \sigma^2))}{\log p(y_{0:T} | (\alpha, \sigma^2))} \right], \quad \text{fix } \alpha = 0.99 \]

with \( N = 2048 \)
Neuroscience example: SMC performance

Relative variance of log-marginal likelihood estimator

$$\sigma^2 \mapsto \text{Var} \left[ \frac{\log \hat{p}(y_{0:T} | (\alpha, \sigma^2))}{\log p(y_{0:T} | (\alpha, \sigma^2))} \right], \quad \text{fix } \alpha = 0.99$$

with \( N = 5529 \)
Neuroscience example: SMC performance

Relative variance of log-marginal likelihood estimator

\[ \sigma^2 \mapsto \text{Var} \left[ \frac{\log \hat{p}(y_{0:T} | (\alpha, \sigma^2))}{\log p(y_{0:T} | (\alpha, \sigma^2))} \right], \quad \text{fix} \ \alpha = 0.99 \]

with \( N = 5529 \) vs. controlled SMC
SMC weights samples from model dynamics

\[ X_t | X_{t-1} \sim \mathcal{N}(\alpha X_{t-1}, \sigma^2) \]

without taking observations into account
Mismatch between dynamics and observations

SMC weights samples from model dynamics

\[ X_t \mid X_{t-1} \sim \mathcal{N}(\alpha X_{t-1}, \sigma^2) \]

without taking observations into account
Mismatch between dynamics and observations

SMC weights samples from model dynamics

\[ X_t | X_{t-1} \sim \mathcal{N}(\alpha X_{t-1}, \sigma^2) \]

without taking observations into account
Mismatch between dynamics and observations

SMC weights samples from model dynamics

\[
X_t | X_{t-1} \sim \mathcal{N}(\alpha X_{t-1}, \sigma^2)
\]

without taking observations into account
1. State space models
2. Sequential Monte Carlo
3. Controlled sequential Monte Carlo
4. Bayesian parameter inference
5. Extensions and future work
Optimal dynamics

- Fix $\theta$ and suppress notational dependency
Optimal dynamics

- Fix $\theta$ and suppress notational dependency
- Sampling from smoothing distribution $p(x_0:T|y_0:T)$

$$X_0 \sim p(x_0|y_0:T), \quad X_t|X_{t-1} \sim p(x_t|x_{t-1}, y_t:T), \quad t \in [1:T]$$
Optimal dynamics

- Fix $\theta$ and suppress notational dependency
- Sampling from smoothing distribution $p(x_0: T \mid y_0: T)$

\[ X_0 \sim p(x_0 \mid y_0: T), \quad X_t \mid X_{t-1} \sim p(x_t \mid x_{t-1}, y_t: T), \quad t \in [1 : T] \]

- Define **backward information filter** $\psi^*_t(x_t) = p(y_t: T \mid x_t)$, then

\[
p(x_0 \mid y_0: T) = \frac{\mu(x_0) \psi^*_0(x_0)}{\mu(\psi^*_0)}
\]

with $\mu(\psi^*_0) = \int_{\mathcal{X}} \psi^*_0(x_0) \mu(x_0) dx_0$, and

\[
p(x_t \mid x_{t-1}, y_t: T) = \frac{f_t(x_{t-1}, x_t) \psi^*_t(x_t)}{f_t(\psi^*_t)(x_{t-1})}
\]

with $f_t(\psi^*_t)(x_{t-1}) = \int_{\mathcal{X}} \psi^*_t(x_t) f_t(x_{t-1}, x_t) dx_t$
Controlled state space model

- Given a **policy**

\[ \psi = (\psi_0, \ldots, \psi_T) \]

i.e. positive and bounded functions
Controlled state space model

- Given a **policy**

\[ \psi = (\psi_0, \ldots, \psi_T) \]

i.e. positive and bounded functions

- Construct **\( \psi \)-controlled dynamics**

\[ X_0 \sim \mu^\psi, \quad X_t | X_{t-1} \sim f_t^\psi(X_{t-1}, \cdot), \quad t \in [1 : T] \]

where

\[ \mu^\psi(x_0) = \frac{\mu(x_0) \psi_0(x_0)}{\mu(\psi_0)}, \quad f_t^\psi(x_{t-1}, x_t) = \frac{f_t(x_{t-1}, x_t) \psi_t(x_t)}{f_t(\psi_t)(x_{t-1})} \]
Controlled state space model

- Given a policy

\[ \psi = (\psi_0, \ldots, \psi_T) \]

i.e. positive and bounded functions

- Construct \( \psi \)-controlled dynamics

\[ X_0 \sim \mu^\psi, \quad X_t|X_{t-1} \sim f^\psi_t(X_{t-1}, \cdot), \quad t \in [1 : T] \]

where

\[ \mu^\psi(x_0) = \frac{\mu(x_0)\psi_0(x_0)}{\mu(\psi_0)}, \quad f^\psi_t(x_{t-1}, x_t) = \frac{f_t(x_{t-1}, x_t)\psi_t(x_t)}{f_t(\psi_t)(x_{t-1})} \]

- Introducing \( \psi \)-controlled observation model

\[ Y_t|X_0:T \sim g^\psi_t(X_t, \cdot), \quad t \in [0 : T] \]

gives a \( \psi \)-controlled state space model
Controlled state space model

- Define **controlled observation densities** \((g_0^\psi, \ldots, g_T^\psi)\) so that

\[
p_\psi(x_0:T|y_0:T) = p(x_0:T|y_0:T), \quad p_\psi(y_0:T) = p(y_0:T)
\]
Controlled state space model

- Define **controlled observation densities** \((g_0^\psi, \ldots, g_T^\psi)\) so that

\[
p^\psi(x_0:T | y_0:T) = p(x_0:T | y_0:T), \quad p^\psi(y_0:T) = p(y_0:T)
\]

- Achieved with

\[
g_0^\psi(x_0, y_0) = \frac{\mu(\psi_0) g_0(x_0, y_0) f_1(\psi_1)(x_0)}{\psi_0(x_0)},
\]

\[
g_t^\psi(x_t, y_t) = \frac{g_t(x_t, y_t) f_{t+1}(\psi_{t+1})(x_t)}{\psi_t(x_t)}, \quad t \in [1 : T - 1],
\]

\[
g_T^\psi(x_T, y_T) = \frac{g_T(x_T, y_T)}{\psi_T(x_T)}
\]
Controlled state space model

- Define **controlled observation densities** \((g_{\psi}^0, \ldots, g_{\psi}^T)\) so that

\[
p_{\psi}(x_0:T \mid y_0:T) = p(x_0:T \mid y_0:T), \quad p_{\psi}(y_0:T) = p(y_0:T)
\]

- Achieved with

\[
g_{\psi}^0(x_0, y_0) = \frac{\mu(\psi_0)g_0(x_0, y_0)f_1(\psi_1)(x_0)}{\psi_0(x_0)},
\]

\[
g_{\psi}^t(x_t, y_t) = \frac{g_t(x_t, y_t)f_{t+1}(\psi_{t+1})(x_t)}{\psi_t(x_t)}, \quad t \in [1 : T - 1],
\]

\[
g_{\psi}^T(x_T, y_T) = \frac{g_T(x_T, y_T)}{\psi_T(x_T)}
\]

- Requirements on policy \(\psi\)
  - Evaluating \(g_{\psi}^t\) tractable
Controlled state space model

- Define **controlled observation densities** $\left( g_0^\psi, \ldots, g_T^\psi \right)$ so that

$$p^\psi(x_0:T | y_0:T) = p(x_0:T | y_0:T), \quad p^\psi(y_0:T) = p(y_0:T)$$

- Achieved with

$$g_0^\psi(x_0, y_0) = \frac{\mu(\psi_0)g_0(x_0, y_0)f_1(\psi_1)(x_0)}{\psi_0(x_0)},$$

$$g_t^\psi(x_t, y_t) = \frac{g_t(x_t, y_t)f_{t+1}(\psi_{t+1})(x_t)}{\psi_t(x_t)}, \quad t \in [1 : T - 1],$$

$$g_T^\psi(x_T, y_T) = \frac{g_T(x_T, y_T)}{\psi_T(x_T)}$$

- Requirements on policy $\psi$
  - Evaluating $g_t^\psi$ tractable
  - Sampling $\mu^\psi$ and $f_t^\psi$ feasible
Gaussian policy for neuroscience example

\[ \psi_t(x_t) = \exp \left( -a_t x_t^2 - b_t x_t - c_t \right), \quad (a_t, b_t, c_t) \in \mathbb{R}^3 \]
• Gaussian policy for neuroscience example

\[ \psi_t(x_t) = \exp\left(-a_t x_t^2 - b_t x_t - c_t\right), \quad (a_t, b_t, c_t) \in \mathbb{R}^3 \]

• Both requirements satisfied since

\[ \mu^\psi = \mathcal{N}(-k_0 b_0, k_0), \quad f^\psi_t(x_{t-1}, \cdot) = \mathcal{N}(k_t\{\alpha \sigma^{-2} x_{t-1} - b_t\}, k_t) \]

with \( k_0 = (1 + 2a_0)^{-1} \) and \( k_t = (\sigma^{-2} + 2a_t)^{-1} \)
• Construct \( \psi \text{-controlled SMC} \) as SMC applied to \( \psi \text{-controlled} \) state space model

\[
X_0 \sim \mu^\psi, \quad X_t | X_{t-1} \sim f_t^\psi(X_{t-1}, \cdot), \quad t \in [1 : T]
\]
\[
Y_t | X_0 : T \sim g_t^\psi(X_t, \cdot), \quad t \in [0 : T]
\]

Unbiased and consistent marginal likelihood estimator

\[
\hat{p}_\psi(y_0 : T) = \prod_{t=0}^T \left\{ \frac{1}{N} \sum_{n=1}^N g_t^\psi(X_n, y_t) \right\}
\]

Consistent approximation of smoothing distribution

\[
\frac{1}{N} \sum_{n=1}^N \phi(X_n^0 : T) \to \int \phi(x^0 : T) p(x^0 : T | y_0 : T) \, dx^0 : T \quad \text{as} \quad N \to \infty
\]
Controlled SMC

- Construct \( \psi \)-controlled SMC as SMC applied to \( \psi \)-controlled state space model

\[
X_0 \sim \mu^\psi, \quad X_t | X_{t-1} \sim f^\psi_t (X_{t-1}, \cdot), \quad t \in [1 : T] \\
Y_t | X_0 : T \sim g^\psi_t (X_t, \cdot), \quad t \in [0 : T]
\]

- Unbiased and consistent marginal likelihood estimator

\[
\hat{p}^\psi (y_0 : T) = \prod_{t=0}^{T} \left\{ \frac{1}{N} \sum_{n=1}^{N} g^\psi_t (X^*_n, y_t) \right\}
\]
Controlled SMC

- **Construct $\psi$-controlled SMC** as SMC applied to $\psi$-controlled state space model

  \[
  X_0 \sim \mu^\psi, \quad X_t | X_{t-1} \sim f_t^\psi(X_{t-1}, \cdot), \quad t \in [1 : T]
  \]

  \[
  Y_t | X_0 : T \sim g_t^\psi(X_t, \cdot), \quad t \in [0 : T]
  \]

- Unbiased and consistent marginal likelihood estimator

  \[
  \hat{p}^\psi(y_0 : T) = \prod_{t=0}^{T} \left\{ \frac{1}{N} \sum_{n=1}^{N} g_t^\psi(X_t^n, y_t) \right\}
  \]

- Consistent approximation of smoothing distribution

  \[
  \frac{1}{N} \sum_{n=1}^{N} \varphi(X_0^n : T) \to \int \varphi(x_0 : T) p(x_0 : T | y_0 : T) dx_0 : T
  \]

  as $N \to \infty$
Optimal policy

- Policy \( \psi^*_t(x_t) = p(y_{t:T}|x_t) \) is optimal since \( \psi^* \)-controlled SMC gives
Optimal policy

- Policy $\psi^*_t(x_t) = p(y_{t:T}|x_t)$ is optimal since $\psi^*$-controlled SMC gives
- $\psi^*$-controlled SMC gives **independent samples** from smoothing distribution
  
  $$X_{0:T}^n \sim p(x_{0:T}|y_{0:T}), \quad n \in [1:N],$$

  and **zero variance** estimator of marginal likelihood for any $N \geq 1$
  
  $$\hat{p}^{\psi^*}(y_{0:T}) = p(y_{0:T})$$
Optimal policy

- Policy \( \psi^*_t(x_t) = p(y_{t:T} | x_t) \) is optimal since \( \psi^* \)-controlled SMC gives
- \( \psi^* \)-controlled SMC gives independent samples from smoothing distribution
  \[ X_{0:T}^n \sim p(x_{0:T} | y_{0:T}), \quad n \in [1 : N], \]
  and zero variance estimator of marginal likelihood for any \( N \geq 1 \)
  \[ \hat{p}^\psi^*(y_{0:T}) = p(y_{0:T}) \]
- (Proposition 1) Optimal policy satisfies backward recursion
  \[ \psi^*_T(x_T) = g_T(x_T, y_T), \]
  \[ \psi^*_t(x_t) = g_t(x_t, y_t)f_{t+1}(\psi^*_{t+1})(x_t), \quad t \in [T - 1 : 0] \]
Optimal policy

- Policy $\psi^*_t(x_t) = p(y_t:T|x_t)$ is optimal since $\psi^*$-controlled SMC gives
- $\psi^*$-controlled SMC gives **independent samples** from smoothing distribution

$$X_{0:T}^n \sim p(x_{0:T}|y_{0:T}), \quad n \in [1:N],$$

and **zero variance** estimator of marginal likelihood for any $N \geq 1$

$$\hat{\psi^*}(y_{0:T}) = p(y_{0:T})$$

- (Proposition 1) Optimal policy satisfies **backward recursion**

$$\psi^*_T(x_T) = g_T(x_T, y_T),$$

$$\psi^*_t(x_t) = g_t(x_t, y_t)f_{t+1}(\psi^*_{t+1})(x_t), \quad t \in [T-1:0]$$

- Backward recursion typically intractable but can be approximated
Connection to optimal control

- $V^*_t = -\log \psi^*_t$ are the optimal **value** functions of the Kullback-Leibler control problem

$$\inf_{\psi \in \Psi} \text{KL} \left( p^\psi(x_0:T) \middle| p(x_0:T|y_0:T) \right)$$
Connection to optimal control

• $V^*_t = -\log \psi^*_t$ are the optimal value functions of the Kullback-Leibler control problem:

$$\inf_{\psi \in \Psi} \text{KL} \left( p^\psi(x_0: T) \Bigg| p(x_0: T | y_0: T) \right)$$

• Connection useful for methodology and analysis

Tsitsiklis & Van Roy (2001). *IEEE Transactions on Neural Networks.*
Connection to optimal control

- $V^*_t = -\log \psi^*_t$ are the optimal **value** functions of the **Kullback-Leibler** control problem

\[
\inf_{\psi \in \Psi} \text{KL} \left( p^\psi(x_0:T) \left| \right. p(x_0:T | y_0:T) \right)
\]

- Connection useful for **methodology** and **analysis**

  Tsitsiklis & Van Roy (2001). *IEEE Transactions on Neural Networks.*

- Methods to approximate backward recursion are known as **approximate dynamic programming** (ADP) for finite horizon control problems
Approximate dynamic programming

- First run standard SMC to get \((X^n_0, \ldots, X^n_T), n \in [1 : N]\)
Approximate dynamic programming

- First run standard SMC to get \((X^n_0, \ldots, X^n_T), n \in [1 : N]\)
- For time \(T\), approximate

\[
\psi^*_T(x_T) = g_T(x_T, y_T)
\]

by \textbf{least squares}

\[
\hat{\psi}_T = \arg \min_{\xi \in \mathcal{F}} \sum_{n=1}^N \{ \log \xi(X^n_T) - \log g_T(X^n_T, y_T) \}^2
\]
Approximate dynamic programming

- First run standard SMC to get \((X^n_0, \ldots, X^n_T), n \in [1 : N]\)
- For time \(T\), approximate

\[
\psi^*_T(x_T) = g_T(x_T, y_T)
\]

by least squares

\[
\hat{\psi}_T = \arg \min_{\xi \in F} \sum_{n=1}^N \{ \log \xi(X^n_T) - \log g_T(X^n_T, y_T) \}^2
\]

- For \(t \in [T - 1 : 0]\), approximate

\[
\psi^*_t(x_t) = g_t(x_t, y_t)f_{t+1}(\psi^*_{t+1})(x_t)
\]

by least squares and \(\psi^*_{t+1} \approx \hat{\psi}_{t+1}\)

\[
\hat{\psi}_t = \arg \min_{\xi \in F} \sum_{n=1}^N \{ \log \xi(X^n_t) - \log g_t(X^n_t, y_t) - \log f_{t+1}(\hat{\psi}_{t+1})(X^n_t) \}^2
\]

Approximate dynamic programming

- (Proposition 3) Error bounds

\[ \mathbb{E}\|\hat{\psi}_t - \psi^*_t\| \leq \sum_{s=t}^{T} C_{t-1,s-1} e^N_s \]

where \( C_{t,s} \) are **stability constants** and \( e^N_t \) are **least squares errors**
Approximate dynamic programming

- (Proposition 3) Error bounds

\[ \mathbb{E} \| \hat{\psi}_t - \psi^*_t \| \leq \sum_{s=t}^{T} C_{t-1,s-1} e^N_s \]

where \( C_{t,s} \) are \textit{stability constants} and \( e^N_t \) are \textit{least squares errors}

- (Theorem 1) As \( N \to \infty \), \( \hat{\psi} \) converges to \textit{idealized ADP}

\[ \tilde{\psi}_T = \arg \min_{\xi \in \mathcal{F}} \mathbb{E} \left[ \{ \log \xi(X_T) - \log g_T(X_T, y_T) \}^2 \right] , \]

\[ \tilde{\psi}_t = \arg \min_{\xi \in \mathcal{F}} \mathbb{E} \left[ \{ \log \xi(X_t) - \log g_t(X_t, y_t) - \log f_{t+1}(\tilde{\psi}_{t+1})(X_t) \}^2 \right] \]
Marginal likelihood estimates of $\hat{\psi}$-controlled SMC with $N = 128$ ($\alpha = 0.99, \sigma^2 = 0.11$)

Variance reduction $\approx 22$ times
Neuroscience example: controlled SMC

Marginal likelihood estimates of $\hat{\psi}$-controlled SMC with $N = 128$ ($\alpha = 0.99, \sigma^2 = 0.11$)

Variance reduction $\approx 22$ times

We can do better!
Current policy $\hat{\psi}$ defines the dynamics

$$X_0 \sim \mu^{\hat{\psi}}, \quad X_t|X_{t-1} \sim f^{\hat{\psi}}_t(X_{t-1}, \cdot), \quad t \in [1 : T]$$
• **Current policy** $\hat{\psi}$ defines the dynamics

$$X_0 \sim \mu^{\hat{\psi}}, \quad X_t|X_{t-1} \sim f_t^{\hat{\psi}}(X_{t-1}, \cdot), \quad t \in [1 : T]$$

• Further control these dynamics with policy $\phi = (\phi_0, \ldots \phi_T)$

$$X_0 \sim \left(\mu^{\hat{\psi}}\right)^\phi, \quad X_t|X_{t-1} \sim \left(f_t^{\hat{\psi}}\right)^\phi(X_{t-1}, \cdot), \quad t \in [1 : T]$$
• **Current policy** $\hat{\psi}$ defines the dynamics

$$X_0 \sim \mu^{\hat{\psi}}, \quad X_t|X_{t-1} \sim f_t^{\hat{\psi}}(X_{t-1}, \cdot), \quad t \in [1 : T]$$

• Further control these dynamics with policy $\phi = (\phi_0, \ldots, \phi_T)$

$$X_0 \sim \left(\mu^{\hat{\psi}}\right)^{\phi}, \quad X_t|X_{t-1} \sim \left(f_t^{\hat{\psi}}\right)^{\phi}(X_{t-1}, \cdot), \quad t \in [1 : T]$$

• Equivalent to controlling model dynamics $\mu$ and $f_t$ with policy

$$\hat{\psi} \cdot \phi = (\hat{\psi}_0 \cdot \phi_0, \ldots, \hat{\psi}_T \cdot \phi_T)$$
• (Proposition 1) **Optimal refinement** of $\phi^*$ current policy $\hat{\psi}$

\[
\phi^*_T(x_T) = g^T_T(x_T, y_T),
\]
\[
\phi^*_t(x_t) = g^T_t(x_t, y_t)f^T_{t+1}(\phi^*_{t+1})(x_t), \quad t \in [T-1:0]
\]
Policy refinement

- (Proposition 1) **Optimal refinement** of $\phi^*$ current policy $\hat{\psi}$

\[
\phi_T^*(x_T) = g_T^*(x_T, y_T),
\]

\[
\phi_t^*(x_t) = g_t^*(x_t, y_t)f_{t+1}^*(\phi_{t+1}^*)(x_t), \quad t \in [T-1:0]
\]

- Optimal policy $\psi^* = \hat{\psi} \cdot \phi^*$
• (Proposition 1) **Optimal refinement** of $\phi^*$ current policy $\hat{\psi}$

$$
\phi^*_T(x_T) = g^\hat{\psi}_T(x_T, y_T),
$$

$$
\phi^*_t(x_t) = g^\hat{\psi}_t(x_t, y_t)f^\hat{\psi}_{t+1}(\phi^*_{t+1})(x_t), \quad t \in [T - 1 : 0]
$$

• Optimal policy $\psi^* = \hat{\psi} \cdot \phi^*$

• Approximate backward recursion to obtain $\hat{\phi} \approx \phi^*$
  - using particles from $\hat{\psi}$-controlled SMC
  - same function class $\mathcal{F}$
Policy refinement

- (Proposition 1) **Optimal refinement** of $\phi^*$ current policy $\hat{\psi}\\n\newline
\phi^*_T(x_T) = g_T\hat{\psi}(x_T, y_T), \\
\phi^*_t(x_t) = g_t\hat{\psi}(x_t, y_t)f_{t+1}(\phi^*_{t+1})(x_t), \quad t \in [T - 1 : 0]\\n\newline
- Optimal policy $\psi^* = \hat{\psi} \cdot \phi^*$
- Approximate backward recursion to obtain $\hat{\phi} \approx \phi^*$
  - using particles from $\hat{\psi}$-controlled SMC
  - same function class $\mathcal{F}$
- Run controlled SMC with **refined policy** $\hat{\psi} \cdot \hat{\phi}$
Neuroscience example: controlled SMC

Marginal likelihood estimates of controlled SMC iteration 2 with $N = 128$ ($\alpha = 0.99, \sigma^2 = 0.11$)

Further variance reduction $\approx 24$ times

![Box plot showing marginal likelihood estimates for iterations 0, 1, and 2 with reduced variance.](image-url)
Neuroscience example: controlled SMC

Marginal likelihood estimates of controlled SMC iteration 3 with $N = 128$ ($\alpha = 0.99$, $\sigma^2 = 0.11$)

Further variance reduction $\approx 1.3$ times
Coefficients \((a^i_t, b^i_t)\) of Gaussian approximation estimated at iteration \(i \geq 1\)
Effect of policy refinement

- **Residual** from ADP when fitting $\hat{\psi}$

$$\varepsilon_t(x_t) = \log \hat{\psi}_t(x_t) - \log g(x_t, y_t) - \log f_{t+1}(\hat{\psi}_{t+1})(x_t)$$
Effect of policy refinement

- **Residual** from ADP when fitting $\hat{\psi}$

$$\varepsilon_t(x_t) = \log \hat{\psi}_t(x_t) - \log g(x_t, y_t) - \log f_{t+1}(\hat{\psi}_{t+1})(x_t)$$

- Performance of $\hat{\psi}$-controlled SMC related to $\|\varepsilon_t\|$
Effect of policy refinement

- **Residual** from ADP when fitting $\hat{\psi}$

  \[ \varepsilon_t(x_t) = \log \hat{\psi}_t(x_t) - \log g(x_t, y_t) - \log f_{t+1}(\hat{\psi}_{t+1})(x_t) \]

- Performance of $\hat{\psi}$-controlled SMC related to $\|\varepsilon_t\|

- Next ADP refinement **re-fits residual** (like $L^2$-boosting)

\[ \hat{\phi}_t = \arg \min_{\xi \in \mathcal{F}} \sum_{n=1}^{N} \left\{ \log \xi(X^n_t) - \varepsilon_t(X^n_t) - \log f_{t+1}^{\hat{\psi}}(\hat{\phi}_{t+1})(X^n_t) \right\}^2 \]

Effect of policy refinement

- **Residual** from ADP when fitting $\hat{\psi}$

$$
\varepsilon_t(x_t) = \log \hat{\psi}_t(x_t) - \log g(x_t, y_t) - \log f_{t+1}(\hat{\psi}_{t+1})(x_t)
$$

- Performance of $\hat{\psi}$-controlled SMC related to $\|\varepsilon_t\|$

- Next ADP refinement **re-fits residual** (like $L^2$-boosting)

$$
\hat{\phi}_t = \arg\min_{\xi \in \mathcal{F}} \sum_{n=1}^{N} \left\{ \log \xi(X_t^n) - \varepsilon_t(X_t^n) - \log f_{t+1}^\psi(\hat{\phi}_{t+1})(X_t^n) \right\}^2
$$

Effect of policy refinement

- **Residual** from ADP when fitting $\hat{\psi}$

$$\varepsilon_t(x_t) = \log \hat{\psi}_t(x_t) - \log g(x_t, y_t) - \log f_{t+1}(\hat{\psi}_{t+1})(x_t)$$

- Performance of $\hat{\psi}$-controlled SMC related to $\|\varepsilon_t\|$ 

- Next ADP refinement **re-fits residual** (like $L^2$-boosting)

$$\hat{\phi}_t = \arg \min_{\xi \in \mathcal{F}} \sum_{n=1}^{N} \left\{ \log \xi(X_t^n) - \varepsilon_t(X_t^n) - \log f_{t+1}^{\hat{\psi}}(\hat{\phi}_{t+1})(X_t^n) \right\}^2$$


This explains previous plots!
Effect of policy refinement

Coefficients of policy at time 0 over 30 iterations

\begin{align*}
\alpha_0(t) &\quad 0 \quad 5 \quad 10 \quad 15 \quad 20 \quad 25 \quad 30 \\
\beta_0(t) &\quad 0 \quad 5 \quad 10 \quad 15 \quad 20 \quad 25 \quad 30 \\
\end{align*}

(Theorem 2) Under regularity conditions

- Policy refinement generates a Markov chain on policy space
- Converges to unique stationary distribution
- Characterization of stationary distribution as $N \to \infty$
Effect of policy refinement

Coefficients of policy at time 0 over 30 iterations

(Theorem 2) Under regularity conditions

- Policy refinement generates a Markov chain on policy space
Effect of policy refinement

Coefficients of policy at time 0 over 30 iterations

(Theorem 2) Under regularity conditions
- Policy refinement generates a Markov chain on policy space
- Converges to unique stationary distribution
Effect of policy refinement

Coefficients of policy at time 0 over 30 iterations

(Theorem 2) Under regularity conditions

- Policy refinement generates a **Markov chain on policy space**
- Converges to unique **stationary distribution**
- Characterization of stationary distribution as $N \to \infty$
Outline

1. State space models
2. Sequential Monte Carlo
3. Controlled sequential Monte Carlo
4. Bayesian parameter inference
5. Extensions and future work
Neuroscience example: PMMH performance

Trace plots of particle marginal Metropolis-Hastings chain

BPF: \( N = 5529 \)
cSMC: \( N = 128 \), 3 refinements
Each iteration takes 2-3 seconds
Neuroscience example: PMMH

Autocorrelation function of particle marginal Metropolis-Hastings chain

100,000 iterations with BPF (3 days)
≈ 20,000 iterations with cSMC (1/2 day)
Neuroscience example: PMMH

Autocorrelation function of particle marginal Metropolis-Hastings chain

100,000 iterations with BPF (3 days)
≈ 20,000 iterations with cSMC (1/2 day)
≈ 2,000 iterations with cSMC + parallel computation (1.5 hours)

1. State space models
2. Sequential Monte Carlo
3. Controlled sequential Monte Carlo
4. Bayesian parameter inference
5. Extensions and future work
Extension to **static models**

\[ \pi_t(\theta) \propto p(\theta)p(y|\theta)^{\lambda_t} \]

where \(0 = \lambda_0 < \cdots < \lambda_T = 1\)
Static models

- **Extension to** static models

\[ \pi_t(\theta) \propto p(\theta)p(y|\theta)^{\lambda_t} \]

where \( 0 = \lambda_0 < \cdots < \lambda_T = 1 \)

- **SMC samplers** introduces potential

\[ G_t(\theta_{t-1}, \theta_t) = \frac{\pi_t(\theta_t)L_{t-1}(\theta_t, \theta_{t-1})}{\pi_{t-1}(\theta_{t-1})M_t(\theta_{t-1}, \theta_t)} \]

Del Moral, Doucet & Jasra (2006). *JRSSB.*
Static models

- **Extension to static models**

\[ \pi_t(\theta) \propto p(\theta)p(y|\theta)^{\lambda_t} \]

where \( 0 = \lambda_0 < \cdots < \lambda_T = 1 \)

- **SMC samplers** introduces potential

\[ G_t(\theta_{t-1}, \theta_t) = \frac{\pi_t(\theta_t)L_{t-1}(\theta_t, \theta_{t-1})}{\pi_{t-1}(\theta_{t-1})M_t(\theta_{t-1}, \theta_t)} \]


- **Annealed importance sampling**

\[ M_t \text{ is } \pi_t \text{-invariant, } L_{t-1} \text{ is reversal of } M_t \implies G_t(\theta_{t-1}) = \frac{\pi_t}{\pi_{t-1}}(\theta_{t-1}) \]

Jarzynski (1997); Neal (2001); Chopin (2004).
Static models

• Extension to static models

\[ \pi_t(\theta) \propto p(\theta)p(y|\theta)^{\lambda_t} \]

where \( 0 = \lambda_0 < \cdots < \lambda_T = 1 \)

• SMC samplers introduces potential

\[ G_t(\theta_{t-1}, \theta_t) = \frac{\pi_t(\theta_t) L_{t-1}(\theta_t, \theta_{t-1})}{\pi_{t-1}(\theta_{t-1}) M_t(\theta_{t-1}, \theta_t)} \]


• Annealed importance sampling

\( M_t \) is \( \pi_t \)-invariant, \( L_{t-1} \) is reversal of \( M_t \) \( \implies G_t(\theta_{t-1}) = \frac{\pi_t}{\pi_{t-1}}(\theta_{t-1}) \)

Jarzynski (1997); Neal (2001); Chopin (2004).

• We consider \( M_t \) as unadjusted Langevin algorithm (ULA) and

\( L_{t-1}(\theta_t, \theta_{t-1}) = M_t(\theta_t, \theta_{t-1}) \)
Optimally controlled SMC gives independent samples from $p(\theta | y)$ and a zero variance estimator of $p(y)$.
Optimally controlled SMC gives independent samples from $p(\theta \mid y)$ and a zero variance estimator of $p(y)$.

Policy for Cox point process model

$$\psi_t(\theta_{t-1}, \theta_t) = \exp \left\{ -\theta_t^T A_t \theta_t - \theta_t^T b_t - c_t + (\lambda_t - \lambda_{t-1}) \log p(y \mid \theta_{t-1}) \right\}$$

for $A_t \in \mathbb{R}^{d \times d}$ diagonal, $b_t \in \mathbb{R}^d$, $c_t \in \mathbb{R}$.
Static models

Model evidence estimates of Cox point process model in 900 dimensions

Variance reduction $\approx 370$ times compared to AIS
Concluding remarks

- **Extensions:**
  - Online filtering
  - Relax requirements on policies


MATLAB code: https://github.com/jeremyhengjm/controlledSMC
Concluding remarks

- Extensions:
  - Online filtering
  - Relax requirements on policies

- Controlled sequential Monte Carlo.
Concluding remarks

• Extensions:
  – Online filtering
  – Relax requirements on policies

• Controlled sequential Monte Carlo.

• MATLAB code:
  https://github.com/jeremyhengjm/controlledSMC