Sequential Monte Carlo Methods for Bayesian Computation

A. Doucet

Kyoto

Sept. 2012

Motivating Example 1: Generic Bayesian Model

- Let X be a vector parameter of interest with an associated prior μ; i.e.

$$X \sim \mu(\cdot).$$

Motivating Example 1: Generic Bayesian Model

- Let X be a vector parameter of interest with an associated prior μ; i.e.

$$X \sim \mu(\cdot).$$

- We observe a realization of y of Y which is assumed to satisfy

$$Y|(X = x) \sim g(\cdot|x);$$

  i.e. the likelihood function is $g(y|x)$.

Motivating Example 1: Generic Bayesian Model

- Let X be a vector parameter of interest with an associated prior μ; i.e.

$$X \sim \mu\left(\cdot\right).$$

- We observe a realization of y of Y which is assumed to satisfy

$$Y|\left(X = x\right) \sim g\left(\cdot| x\right);$$

  i.e. the likelihood function is $g\left(y| x\right)$.
- Bayesian inference on X relies on the posterior of X given $Y = y$:

$$p\left(x| y\right) = \frac{\mu\left(x\right) g\left(y| x\right)}{p\left(y\right)}$$

  where the marginal likelihood/evidence satisfies

$$p\left(y\right) = \int \mu\left(x\right) g\left(y| x\right) dx.$$

Motivating Example 1: Generic Bayesian Model

- Let X be a vector parameter of interest with an associated prior μ; i.e.

$$X \sim \mu(\cdot).$$

- We observe a realization of y of Y which is assumed to satisfy

$$Y|(X = x) \sim g(\cdot|x);$$

  i.e. the likelihood function is $g(y|x)$.
- Bayesian inference on X relies on the posterior of X given $Y = y$:

$$p(x|y) = \frac{\mu(x)\, g(y|x)}{p(y)}$$

  where the marginal likelihood/evidence satisfies

$$p(y) = \int \mu(x)\, g(y|x)\, dx.$$

- "Machine learning" examples: Latent Dirichlet Allocation, (Hiearchical) Dirichlet processes...

Motivating Example 2: State-Space Models

- Let $\{X_t\}_{t\geq 1}$ be a latent/hidden Markov process with

$$X_1 \sim \mu(\cdot) \text{ and } X_t | (X_{t-1} = x) \sim f(\cdot | x).$$

Motivating Example 2: State-Space Models

- Let $\{X_t\}_{t\geq1}$ be a latent/hidden Markov process with
$$X_1 \sim \mu(\cdot) \text{ and } X_t|\left(X_{t-1}=x\right) \sim f\left(\cdot|\,x\right).$$
- Let $\{Y_t\}_{t\geq1}$ be an observation process such that observations are conditionally independent given $\{X_t\}_{t\geq1}$ and
$$Y_t|\left(X_t=x\right) \sim g\left(\cdot|\,x\right).$$

Motivating Example 2: State-Space Models

- Let $\{X_t\}_{t\geq 1}$ be a latent/hidden Markov process with
$$X_1 \sim \mu(\cdot) \text{ and } X_t | (X_{t-1} = x) \sim f(\cdot | x).$$
- Let $\{Y_t\}_{t\geq 1}$ be an observation process such that observations are conditionally independent given $\{X_t\}_{t\geq 1}$ and
$$Y_t | (X_t = x) \sim g(\cdot | x).$$
- Let $z_{i:j} := (z_i, z_{i+1}, \ldots, z_j)$ then Bayesian inference on $X_{1:t}$ relies on the posterior of $X_{1:t}$ given $Y = y_{1:t}$:
$$p(x_{1:t} | y_{1:t}) = \frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})}$$
where the marginal likelihood/evidence satisfies
$$p(y_{1:t}) = \int p(x_{1:t}, y_{1:t})\, dx_{1:t}.$$

Motivating Example 2: State-Space Models

- Let $\{X_t\}_{t\geq 1}$ be a latent/hidden Markov process with

$$X_1 \sim \mu(\cdot) \text{ and } X_t | (X_{t-1} = x) \sim f(\cdot | x).$$

- Let $\{Y_t\}_{t\geq 1}$ be an observation process such that observations are conditionally independent given $\{X_t\}_{t\geq 1}$ and

$$Y_t | (X_t = x) \sim g(\cdot | x).$$

- Let $z_{i:j} := (z_i, z_{i+1}, ..., z_j)$ then Bayesian inference on $X_{1:t}$ relies on the posterior of $X_{1:t}$ given $Y = y_{1:t}$:

$$p(x_{1:t} | y_{1:t}) = \frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})}$$

where the marginal likelihood/evidence satisfies

$$p(y_{1:t}) = \int p(x_{1:t}, y_{1:t}) \, dx_{1:t}.$$

- "Machine learning" examples: Biochemical network models, Dynamic topic models, Neuroscience models etc.

Bayesian Inference and Machine Learning

- Bayesian approaches have been adopted by a large part of the ML community.

Bayesian Inference and Machine Learning

- Bayesian approaches have been adopted by a large part of the ML community.
- Bayesian inference offers a number of attractive advantages over conventional approach

Bayesian Inference and Machine Learning

- Bayesian approaches have been adopted by a large part of the ML community.
- Bayesian inference offers a number of attractive advantages over conventional approach
  - flexibility in constructing complex models from simple parts;

Bayesian Inference and Machine Learning

- Bayesian approaches have been adopted by a large part of the ML community.
- Bayesian inference offers a number of attractive advantages over conventional approach
  - flexibility in constructing complex models from simple parts;
  - the incorporation of prior knowledge is very natural;

Bayesian Inference and Machine Learning

- Bayesian approaches have been adopted by a large part of the ML community.
- Bayesian inference offers a number of attractive advantages over conventional approach
  - flexibility in constructing complex models from simple parts;
  - the incorporation of prior knowledge is very natural;
  - all modelling assumptions are made explicit;

Bayesian Inference and Machine Learning

- Bayesian approaches have been adopted by a large part of the ML community.
- Bayesian inference offers a number of attractive advantages over conventional approach
  - flexibility in constructing complex models from simple parts;
  - the incorporation of prior knowledge is very natural;
  - all modelling assumptions are made explicit;
  - uncertainties over model order;

Bayesian Inference and Machine Learning

- Bayesian approaches have been adopted by a large part of the ML community.
- Bayesian inference offers a number of attractive advantages over conventional approach
  - flexibility in constructing complex models from simple parts;
  - the incorporation of prior knowledge is very natural;
  - all modelling assumptions are made explicit;
  - uncertainties over model order;
  - model parameters and predictions are technically straightforward to compute;

Bayesian Inference and Machine Learning

- Bayesian approaches have been adopted by a large part of the ML community.
- Bayesian inference offers a number of attractive advantages over conventional approach
  - flexibility in constructing complex models from simple parts;
  - the incorporation of prior knowledge is very natural;
  - all modelling assumptions are made explicit;
  - uncertainties over model order;
  - model parameters and predictions are technically straightforward to compute;
- The cost to pay is that approximate inference techniques are necessary to approximate the resulting posterior distributions for all but trivial models.

Approximate Inference Methods

- Gaussian/Laplace approximation, local linearization, Extended Kalman filters.

Approximate Inference Methods

- Gaussian/Laplace approximation, local linearization, Extended Kalman filters.
- Variational methods, density assumed filters.

Approximate Inference Methods

- Gaussian/Laplace approximation, local linearization, Extended Kalman filters.
- Variational methods, density assumed filters.
- Expectation-Propagation.

Approximate Inference Methods

- Gaussian/Laplace approximation, local linearization, Extended Kalman filters.
- Variational methods, density assumed filters.
- Expectation-Propagation.
- Markov chain Monte Carlo (MCMC) methods.

Approximate Inference Methods

- Gaussian/Laplace approximation, local linearization, Extended Kalman filters.
- Variational methods, density assumed filters.
- Expectation-Propagation.
- Markov chain Monte Carlo (MCMC) methods.
- Sequential Monte Carlo (SMC) methods.

Monte Carlo Methods

- Variational and EP methods are computationally cheap but perform functional approximations of the posteriors of interest.

Monte Carlo Methods

- Variational and EP methods are computationally cheap but perform functional approximations of the posteriors of interest.
- Both MCMC and SMC are asymptotically (as you increase computational efforts) bias-free but computationally expensive.

Monte Carlo Methods

- Variational and EP methods are computationally cheap but perform functional approximations of the posteriors of interest.
- Both MCMC and SMC are asymptotically (as you increase computational efforts) bias-free but computationally expensive.
- MCMC are the tools of choice in Bayesian computation for over 20 years whereas SMC have been widely used for 15 years in vision and robotics.

Monte Carlo Methods

- Variational and EP methods are computationally cheap but perform functional approximations of the posteriors of interest.
- Both MCMC and SMC are asymptotically (as you increase computational efforts) bias-free but computationally expensive.
- MCMC are the tools of choice in Bayesian computation for over 20 years whereas SMC have been widely used for 15 years in vision and robotics.
- The development of new methodology combined to the emergence of cheap multicore architectures makes now SMC a powerful alternative/complementary approach to MCMC to address general Bayesian computational problems.

Monte Carlo Methods

- Variational and EP methods are computationally cheap but perform functional approximations of the posteriors of interest.
- Both MCMC and SMC are asymptotically (as you increase computational efforts) bias-free but computationally expensive.
- MCMC are the tools of choice in Bayesian computation for over 20 years whereas SMC have been widely used for 15 years in vision and robotics.
- The development of new methodology combined to the emergence of cheap multicore architectures makes now SMC a powerful alternative/complementary approach to MCMC to address general Bayesian computational problems.
- The aim of these lectures is to provide an introduction to this active research field and discuss some open research problems.

Some References and Resources

- A.D., J.F.G. De Freitas & N.J. Gordon (editors), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag: New York, 2001.

Some References and Resources

- A.D., J.F.G. De Freitas & N.J. Gordon (editors), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag: New York, 2001.
- P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer-Verlag: New York, 2004.

Some References and Resources

- A.D., J.F.G. De Freitas & N.J. Gordon (editors), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag: New York, 2001.
- P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer-Verlag: New York, 2004.
- O. Cappé, E. Moulines & T. Ryden, *Hidden Markov Models*, Springer-Verlag: New York, 2005.

Some References and Resources

- A.D., J.F.G. De Freitas & N.J. Gordon (editors), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag: New York, 2001.
- P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer-Verlag: New York, 2004.
- O. Cappé, E. Moulines & T. Ryden, *Hidden Markov Models*, Springer-Verlag: New York, 2005.

- **Webpage with links to papers and codes**: http://www.stats.ox.ac.uk/~doucet/smc_resources.html

Some References and Resources

- A.D., J.F.G. De Freitas & N.J. Gordon (editors), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag: New York, 2001.
- P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer-Verlag: New York, 2004.
- O. Cappé, E. Moulines & T. Ryden, *Hidden Markov Models*, Springer-Verlag: New York, 2005.
- **Webpage with links to papers and codes**: http://www.stats.ox.ac.uk/~doucet/smc_resources.html
- Thousands of papers on the subject appear every year.

Organization of Lectures

- **State-Space Models** (approx.4 hours)

Organization of Lectures

- **State-Space Models** (approx.4 hours)
  - SMC filtering and smoothing

Organization of Lectures

- **State-Space Models** (approx.4 hours)
  - SMC filtering and smoothing
  - Maximum likelihood parameter inference

Organization of Lectures

- **State-Space Models** (approx.4 hours)
  - SMC filtering and smoothing
  - Maximum likelihood parameter inference
  - Bayesian parameter inference

Organization of Lectures

- **State-Space Models** (approx.4 hours)
  - SMC filtering and smoothing
  - Maximum likelihood parameter inference
  - Bayesian parameter inference
- **Beyond State-Space Models** (approx. 2 hours)

Organization of Lectures

- **State-Space Models** (approx.4 hours)
  - SMC filtering and smoothing
  - Maximum likelihood parameter inference
  - Bayesian parameter inference
- **Beyond State-Space Models** (approx. 2 hours)
  - SMC methods for generic sequence of target distributions

Organization of Lectures

- **State-Space Models** (approx.4 hours)
  - SMC filtering and smoothing
  - Maximum likelihood parameter inference
  - Bayesian parameter inference
- **Beyond State-Space Models** (approx. 2 hours)
  - SMC methods for generic sequence of target distributions
  - SMC samplers.

Organization of Lectures

- **State-Space Models** (approx.4 hours)
  - SMC filtering and smoothing
  - Maximum likelihood parameter inference
  - Bayesian parameter inference
- **Beyond State-Space Models** (approx. 2 hours)
  - SMC methods for generic sequence of target distributions
  - SMC samplers.
  - Approximate Bayesian Computation.

Organization of Lectures

- **State-Space Models** (approx.4 hours)
  - SMC filtering and smoothing
  - Maximum likelihood parameter inference
  - Bayesian parameter inference
- **Beyond State-Space Models** (approx. 2 hours)
  - SMC methods for generic sequence of target distributions
  - SMC samplers.
  - Approximate Bayesian Computation.
  - Optimal design, optimal control.

State-Space Models

- Let $\{X_t\}_{t\geq 1}$ be a latent/hidden $\mathcal{X}$-valued Markov process with

$$X_1 \sim \mu(\cdot) \text{ and } X_t | (X_{t-1} = x) \sim f(\cdot | x).$$

State-Space Models

- Let $\{X_t\}_{t\geq 1}$ be a latent/hidden $\mathcal{X}$-valued Markov process with

$$X_1 \sim \mu(\cdot) \text{ and } X_t|(X_{t-1}=x) \sim f(\cdot|x).$$

- Let $\{Y_t\}_{t\geq 1}$ be an $\mathcal{Y}$-valued Markov observation process such that observations are conditionally independent given $\{X_t\}_{t\geq 1}$ and

$$Y_t|(X_t=x) \sim g(\cdot|x).$$

State-Space Models

- Let $\{X_t\}_{t\geq 1}$ be a latent/hidden $\mathcal{X}$-valued Markov process with

$$X_1 \sim \mu(\cdot) \text{ and } X_t | (X_{t-1} = x) \sim f(\cdot | x).$$

- Let $\{Y_t\}_{t\geq 1}$ be an $\mathcal{Y}$-valued Markov observation process such that observations are conditionally independent given $\{X_t\}_{t\geq 1}$ and

$$Y_t | (X_t = x) \sim g(\cdot | x).$$

- General class of time series models aka Hidden Markov Models (HMM) including

$$X_t = \Psi(X_{t-1}, V_t), \quad Y_t = \Phi(X_t, W_t)$$

where V_t, W_t are two sequences of i.i.d. random variables.

State-Space Models

- Let $\{X_t\}_{t\geq 1}$ be a latent/hidden $\mathcal{X}$-valued Markov process with

$$X_1 \sim \mu(\cdot) \text{ and } X_t | (X_{t-1} = x) \sim f(\cdot | x).$$

- Let $\{Y_t\}_{t\geq 1}$ be an $\mathcal{Y}$-valued Markov observation process such that observations are conditionally independent given $\{X_t\}_{t\geq 1}$ and

$$Y_t | (X_t = x) \sim g(\cdot | x).$$

- General class of time series models aka Hidden Markov Models (HMM) including

$$X_t = \Psi(X_{t-1}, V_t), \; Y_t = \Phi(X_t, W_t)$$

where V_t, W_t are two sequences of i.i.d. random variables.

- **Aim**: Infer $\{X_t\}$ given observations $\{Y_t\}$ on-line or off-line.

State-Space Models

- State-space models are ubiquitous in control, data mining, econometrics, geosciences, system biology etc. Since Jan. 2012, more than 13,500 papers have already appeared (source: Google Scholar).

State-Space Models

- State-space models are ubiquitous in control, data mining, econometrics, geosciences, system biology etc. Since Jan. 2012, more than 13,500 papers have already appeared (source: Google Scholar).
- **Finite State-space HMM**: $\mathcal{X}$ is a finite space, i.e. $\{X_t\}$ is a finite Markov chain

$$Y_t | (X_t = x) \sim g(\cdot | x)$$

State-Space Models

- State-space models are ubiquitous in control, data mining, econometrics, geosciences, system biology etc. Since Jan. 2012, more than 13,500 papers have already appeared (source: Google Scholar).
- **Finite State-space HMM**: $\mathcal{X}$ is a finite space, i.e. $\{X_t\}$ is a finite Markov chain

$$Y_t | (X_t = x) \sim g(\cdot | x)$$

- **Linear Gaussian state-space model**

$$\begin{aligned} X_t &= AX_{t-1} + BV_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I) \\ Y_t &= CX_t + DW_t, \quad W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I) \end{aligned}$$

State-Space Models

- State-space models are ubiquitous in control, data mining, econometrics, geosciences, system biology etc. Since Jan. 2012, more than 13,500 papers have already appeared (source: Google Scholar).
- **Finite State-space HMM**: $\mathcal{X}$ is a finite space, i.e. $\{X_t\}$ is a finite Markov chain

$$Y_t | (X_t = x) \sim g(\cdot | x)$$

- **Linear Gaussian state-space model**

$$\begin{aligned} X_t &= AX_{t-1} + BV_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I) \\ Y_t &= CX_t + DW_t, \quad W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I) \end{aligned}$$

- **Switching Linear Gaussian state-space model:** $X_t = (X_t^1, X_t^2)$ where $\{X_t^1\}$ is a finite Markov chain,

$$\begin{aligned} X_t^2 &= A(X_t^1) X_{t-1}^2 + B(X_t^1) V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I) \\ Y_t &= C(X_t^1) X_t^2 + D(X_t^1) W_t, \quad W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I) \end{aligned}$$

State-Space Models

- **Stochastic Volatility model**

$$
\begin{aligned}
X_t &= \phi X_{t-1} + \sigma V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \\
Y_t &= \beta \exp(X_t/2)\, W_t, \ \ W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)
\end{aligned}
$$

State-Space Models

- **Stochastic Volatility model**

$$
\begin{aligned}
X_t &= \phi X_{t-1} + \sigma V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \\
Y_t &= \beta \exp(X_t/2)\, W_t, \;\; W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)
\end{aligned}
$$

- **Biochemical Network model**

$$
\begin{aligned}
&\Pr\left(X^1_{t+dt}{=}x^1_t{+}1, X^2_{t+dt}{=}x^2_t \,\middle|\, x^1_t, x^2_t\right) = \alpha\, x^1_t\, dt + o(dt), \\
&\Pr\left(X^1_{t+dt}{=}x^1_t{-}1, X^2_{t+dt}{=}x^2_t{+}1 \,\middle|\, x^1_t, x^2_t\right) = \beta\, x^1_t\, x^2_t\, dt + o(dt), \\
&\Pr\left(X^1_{t+dt}{=}x^1_t, X^2_{t+dt}{=}x^2_t{-}1 \,\middle|\, x^1_t, x^2_t\right) = \gamma\, x^2_t\, dt + o(dt),
\end{aligned}
$$

with

$$
Y_k = X^1_{k\Delta T} + W_k \text{ with } W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right).
$$

State-Space Models

- **Stochastic Volatility model**

$$\begin{aligned} X_t &= \phi X_{t-1} + \sigma V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \\ Y_t &= \beta \exp(X_t/2)\, W_t, \ \ W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \end{aligned}$$

- **Biochemical Network model**

$$\begin{aligned} &\Pr\left(X^1_{t+dt}{=}x^1_t{+}1, X^2_{t+dt}{=}x^2_t \,\middle|\, x^1_t, x^2_t\right) = \alpha\, x^1_t\, dt + o(dt), \\ &\Pr\left(X^1_{t+dt}{=}x^1_t{-}1, X^2_{t+dt}{=}x^2_t{+}1 \,\middle|\, x^1_t, x^2_t\right) = \beta\, x^1_t\, x^2_t\, dt + o(dt), \\ &\Pr\left(X^1_{t+dt}{=}x^1_t, X^2_{t+dt}{=}x^2_t{-}1 \,\middle|\, x^1_t, x^2_t\right) = \gamma\, x^2_t\, dt + o(dt), \end{aligned}$$

with

$$Y_k = X^1_{k\Delta T} + W_k \text{ with } W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right).$$

- **Nonlinear Diffusion model**

$$\begin{aligned} dX_t &= \alpha(X_t)\, dt + \beta(X_t)\, dV_t, \ \ V_t \text{ Brownian motion} \\ Y_k &= \gamma(X_{k\Delta T}) + W_k, \ W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right). \end{aligned}$$

Inference in State-Space Models

- Given observations $y_{1:t} := (y_1, y_2, \ldots, y_t)$, inference about $X_{1:t} := (X_1, ..., X_t)$ relies on the posterior

$$p(x_{1:t} | y_{1:t}) = \frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})}$$

where

$$p(x_{1:t}, y_{1:t}) = \underbrace{\mu(x_1) \prod_{k=2}^{t} f(x_k | x_{k-1})}_{p(x_{1:t})} \underbrace{\prod_{k=1}^{t} g(y_k | x_k)}_{p(y_{1:t} | x_{1:t})},$$

$$p(y_{1:t}) = \int \cdots \int p(x_{1:t}, y_{1:t}) \, dx_{1:t}$$

Inference in State-Space Models

- Given observations $y_{1:t} := (y_1, y_2, \ldots, y_t)$, inference about $X_{1:t} := (X_1, ..., X_t)$ relies on the posterior

$$p(x_{1:t}|\, y_{1:t}) = \frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})}$$

where

$$p(x_{1:t}, y_{1:t}) = \underbrace{\mu(x_1) \prod_{k=2}^{t} f(x_k|\, x_{k-1})}_{p(x_{1:t})} \underbrace{\prod_{k=1}^{t} g(y_k|\, x_k)}_{p(y_{1:t}|x_{1:t})},$$

$$p(y_{1:t}) = \int \cdots \int p(x_{1:t}, y_{1:t})\, dx_{1:t}$$

- When $\mathcal{X}$ is finite & linear Gaussian models, $\{p(x_t|\, y_{1:t})\}_{t\geq 1}$ can be computed exactly. For non-linear models, approximations are required: EKF, UKF, Gaussian sum filters, etc.

Inference in State-Space Models

- Given observations $y_{1:t} := (y_1, y_2, \ldots, y_t)$, inference about $X_{1:t} := (X_1, ..., X_t)$ relies on the posterior

$$p(x_{1:t} | y_{1:t}) = \frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})}$$

where

$$p(x_{1:t}, y_{1:t}) = \underbrace{\mu(x_1) \prod_{k=2}^{t} f(x_k | x_{k-1})}_{p(x_{1:t})} \underbrace{\prod_{k=1}^{t} g(y_k | x_k)}_{p(y_{1:t} | x_{1:t})},$$

$$p(y_{1:t}) = \int \cdots \int p(x_{1:t}, y_{1:t}) \, dx_{1:t}$$

- When $\mathcal{X}$ is finite & linear Gaussian models, $\{p(x_t | y_{1:t})\}_{t \geq 1}$ can be computed exactly. For non-linear models, approximations are required: EKF, UKF, Gaussian sum filters, etc.
- Approximations of $\{p(x_t | y_{1:t})\}_{t=1}^{T}$ provide approximation of $p(x_{1:T} | y_{1:T})$.

Monte Carlo Methods Basics

- Assume you can generate $X_{1:t}^{(i)} \sim p\left(x_{1:t} \middle| y_{1:t} \right)$ where $i = 1, ..., N$ then MC approximation is

$$\widehat{p}\left(x_{1:t} \middle| y_{1:t} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}} \left(x_{1:t} \right)$$

Monte Carlo Methods Basics

- Assume you can generate $X_{1:t}^{(i)} \sim p\left(x_{1:t} \middle| y_{1:t}\right)$ where $i = 1, ..., N$ then MC approximation is

$$\widehat{p}\left(x_{1:t} \middle| y_{1:t}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right)$$

- *Integration is straightforward.*

$$\begin{aligned}\int \varphi_t\left(x_{1:t}\right) p\left(x_{1:t} \middle| y_{1:t}\right) dx_{1:t} &\approx \int \varphi_t\left(x_{1:t}\right) \widehat{p}\left(x_{1:t} \middle| y_{1:t}\right) dx_{1:t} \\ &= \tfrac{1}{N}\sum_{i=1}^{N} \varphi\left(X_{1:t}^{(i)}\right)\end{aligned}$$

Monte Carlo Methods Basics

- Assume you can generate $X_{1:t}^{(i)} \sim p\left(\left. x_{1:t} \right| y_{1:t} \right)$ where $i = 1, ..., N$ then MC approximation is

$$\widehat{p}\left(\left. x_{1:t} \right| y_{1:t} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}}\left(x_{1:t} \right)$$

- *Integration is straightforward.*

$$\begin{aligned} \int \varphi_t\left(x_{1:t} \right) p\left(\left. x_{1:t} \right| y_{1:t} \right) dx_{1:t} \approx \int \varphi_t\left(x_{1:t} \right) \widehat{p}\left(\left. x_{1:t} \right| y_{1:t} \right) dx_{1:t} \\ = \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \varphi\left(X_{1:t}^{(i)} \right) \end{aligned}$$

- *Marginalization is straightforward.*

$$\widehat{p}\left(\left. x_k \right| y_{1:t} \right) = \int \widehat{p}\left(\left. x_{1:t} \right| y_{1:t} \right) dx_{1:k-1} dx_{k+1:t} = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_k^{(i)}}\left(x_k \right).$$

Monte Carlo Methods Basics

- Assume you can generate $X_{1:t}^{(i)} \sim p\left(x_{1:t} \middle| y_{1:t} \right)$ where $i = 1, ..., N$ then MC approximation is

$$\widehat{p}\left(x_{1:t} \middle| y_{1:t} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}} \left(x_{1:t} \right)$$

- *Integration is straightforward.*

$$\begin{aligned} \int \varphi_t \left(x_{1:t} \right) p\left(x_{1:t} \middle| y_{1:t} \right) dx_{1:t} &\approx \int \varphi_t \left(x_{1:t} \right) \widehat{p}\left(x_{1:t} \middle| y_{1:t} \right) dx_{1:t} \\ &= \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \varphi \left(X_{1:t}^{(i)} \right) \end{aligned}$$

- *Marginalization is straightforward.*

$$\widehat{p}\left(x_k \middle| y_{1:t} \right) = \int \widehat{p}\left(x_{1:t} \middle| y_{1:t} \right) dx_{1:k-1} dx_{k+1:t} = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_k^{(i)}} \left(x_k \right).$$

- **Basic and key property**: $\mathbb{V}\left[\frac{1}{N} \sum_{i=1}^{N} \varphi \left(X_{1:t}^{(i)} \right) \right] = \frac{C\left(t \dim\left(\mathcal{X} \right) \right)}{N}$, i.e. rate of convergence to zero is independent of $\dim\left(\mathcal{X} \right)$ and t.

- **Problem 1**: We cannot typically generate exact samples from $p\left(x_{1:t} \mid y_{1:t}\right)$ for non-linear non-Gaussian models.

Monte Carlo Methods

- **Problem 1**: We cannot typically generate exact samples from $p(x_{1:t}|y_{1:t})$ for non-linear non-Gaussian models.
- **Problem 2**: Even if we could, algorithms to generate samples from $p(x_{1:t}|y_{1:t})$ will have at least complexity $\mathcal{O}(t)$.

Monte Carlo Methods

- **Problem 1**: We cannot typically generate exact samples from $p(x_{1:t}|\, y_{1:t})$ for non-linear non-Gaussian models.
- **Problem 2**: Even if we could, algorithms to generate samples from $p(x_{1:t}|\, y_{1:t})$ will have at least complexity $\mathcal{O}(t)$.
- Typical solution to problem 1 is to generate approximate samples using MCMC methods but these methods are not recursive.

Monte Carlo Methods

- **Problem 1**: We cannot typically generate exact samples from $p(x_{1:t}|\, y_{1:t})$ for non-linear non-Gaussian models.
- **Problem 2**: Even if we could, algorithms to generate samples from $p(x_{1:t}|\, y_{1:t})$ will have at least complexity $\mathcal{O}(t)$.
- Typical solution to problem 1 is to generate approximate samples using MCMC methods but these methods are not recursive.
- **SMC Methods** solves *partially* Problem 1 and Problem 2 by breaking the problem of sampling from $p(x_{1:t}|\, y_{1:t})$ into a collection of simpler subproblems. First approximate $p(x_1|\, y_1)$ and $p(y_1)$ at time 1, then $p(x_{1:2}|\, y_{1:2})$ and $p(y_{1:2})$ at time 2 and so on.

Monte Carlo Methods

- **Problem 1**: We cannot typically generate exact samples from $p(x_{1:t}|y_{1:t})$ for non-linear non-Gaussian models.
- **Problem 2**: Even if we could, algorithms to generate samples from $p(x_{1:t}|y_{1:t})$ will have at least complexity $\mathcal{O}(t)$.
- Typical solution to problem 1 is to generate approximate samples using MCMC methods but these methods are not recursive.
- **SMC Methods** solves *partially* Problem 1 and Problem 2 by breaking the problem of sampling from $p(x_{1:t}|y_{1:t})$ into a collection of simpler subproblems. First approximate $p(x_1|y_1)$ and $p(y_1)$ at time 1, then $p(x_{1:2}|y_{1:2})$ and $p(y_{1:2})$ at time 2 and so on.
- Each target distribution is approximated by a cloud of random samples termed *particles* evolving according to *importance sampling* and *resampling* steps.

Standard Bayesian Recursion

- In most textbooks, you will find the following recursion for $\{p(x_t | y_{1:t})\}_{t \geq 1}$.

Standard Bayesian Recursion

- In most textbooks, you will find the following recursion for $\{p(x_t|\, y_{1:t})\}_{t\geq 1}$.
- **Prediction step**

$$\begin{aligned} p(x_t|\, y_{1:t-1}) &= \int p(x_{t-1}, x_t|\, y_{1:t-1})\, dx_{t-1} \\ &= \int p(x_t|\, y_{1:t-1}, x_{t-1})\, p(x_{t-1}|\, y_{1:t-1})\, dx_{t-1} \\ &= \int f(x_t|\, x_{t-1})\, p(x_{t-1}|\, y_{1:t-1})\, dx_{t-1}. \end{aligned}$$

Standard Bayesian Recursion

- In most textbooks, you will find the following recursion for $\{p(x_t|y_{1:t})\}_{t\geq 1}$.
- **Prediction step**

$$\begin{aligned} p(x_t|y_{1:t-1}) &= \int p(x_{t-1}, x_t|y_{1:t-1})\,dx_{t-1} \\ &= \int p(x_t|y_{1:t-1}, x_{t-1})\,p(x_{t-1}|y_{1:t-1})\,dx_{t-1} \\ &= \int f(x_t|x_{t-1})\,p(x_{t-1}|y_{1:t-1})\,dx_{t-1}. \end{aligned}$$

- **Bayes Updating step**

$$p(x_t|y_{1:t}) = \frac{g(y_t|x_t)\;\; p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}$$

where

$$p(y_t|y_{1:t-1}) = \int g(y_t|x_t)\;\; p(x_t|y_{1:t-1})\,dx_t$$

Standard Bayesian Recursion

- In most textbooks, you will find the following recursion for $\{p(x_t | y_{1:t})\}_{t\geq 1}$.
- **Prediction step**

$$\begin{aligned} p(x_t | y_{1:t-1}) &= \int p(x_{t-1}, x_t | y_{1:t-1})\, dx_{t-1} \\ &= \int p(x_t | y_{1:t-1}, x_{t-1})\, p(x_{t-1} | y_{1:t-1})\, dx_{t-1} \\ &= \int f(x_t | x_{t-1})\, p(x_{t-1} | y_{1:t-1})\, dx_{t-1}. \end{aligned}$$

- **Bayes Updating step**

$$p(x_t | y_{1:t}) = \frac{g(y_t | x_t)\ p(x_t | y_{1:t-1})}{p(y_t | y_{1:t-1})}$$

where

$$p(y_t | y_{1:t-1}) = \int g(y_t | x_t)\ p(x_t | y_{1:t-1})\, dx_t$$

- This is the recursion implemented by Wonham and Kalman filters...

Bayesian Recursion on Path Space

- SMC approximate directly $\{p(x_{1:t}|\, y_{1:t})\}_{t\geq 1}$ not $\{p(x_t|\, y_{1:t})\}_{t\geq 1}$ and relies on

$$p(x_{1:t}|\, y_{1:t}) = \frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})} = \frac{g(y_t|\, x_t)\; f(x_t|\, x_{t-1})}{p(y_t|\, y_{1:t-1})} \frac{p(x_{1:t-1}, y_{1:t-1})}{p(y_{1:t-1})}$$

$$= \frac{g(y_t|\, x_t)\; \overbrace{f(x_t|\, x_{t-1})\, p(x_{1:t-1}|\, y_{1:t-1})}^{\text{predictive } p(x_{1:t}|y_{1:t-1})}}{p(y_t|\, y_{1:t-1})}$$

where

$$p(y_t|\, y_{1:t-1}) = \int g(y_t|\, x_t)\;\; p(x_{1:t}|\, y_{1:t-1})\, dx_{1:t}$$

Bayesian Recursion on Path Space

- SMC approximate directly $\{p(x_{1:t}|y_{1:t})\}_{t\geq 1}$ not $\{p(x_t|y_{1:t})\}_{t\geq 1}$ and relies on

$$p(x_{1:t}|y_{1:t}) = \frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})} = \frac{g(y_t|x_t)\ f(x_t|x_{t-1})}{p(y_t|y_{1:t-1})}\frac{p(x_{1:t-1}, y_{1:t-1})}{p(y_{1:t-1})}$$

$$= \frac{g(y_t|x_t)\ \overbrace{f(x_t|x_{t-1})\,p(x_{1:t-1}|y_{1:t-1})}^{\text{predictive } p(x_{1:t}|y_{1:t-1})}}{p(y_t|y_{1:t-1})}$$

where

$$p(y_t|y_{1:t-1}) = \int g(y_t|x_t)\ p(x_{1:t}|y_{1:t-1})\,dx_{1:t}$$

- This can be alternatively written as

Prediction $\quad p(x_{1:t}|y_{1:t-1}) = f(x_t|x_{t-1})\,p(x_{1:t-1}|y_{1:t-1})$,

Update $\quad p(x_{1:t}|y_{1:t}) = \frac{g(y_t|x_t)p(x_{1:t}|y_{1:t-1})}{p(y_t|y_{1:t-1})}$.

Bayesian Recursion on Path Space

- SMC approximate directly $\{p(x_{1:t}|y_{1:t})\}_{t\geq 1}$ not $\{p(x_t|y_{1:t})\}_{t\geq 1}$ and relies on

$$p(x_{1:t}|y_{1:t}) = \frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})} = \frac{g(y_t|x_t)\ f(x_t|x_{t-1})}{p(y_t|y_{1:t-1})} \frac{p(x_{1:t-1}, y_{1:t-1})}{p(y_{1:t-1})}$$

$$= \frac{g(y_t|x_t)\ \overbrace{f(x_t|x_{t-1})\, p(x_{1:t-1}|y_{1:t-1})}^{\text{predictive } p(x_{1:t}|y_{1:t-1})}}{p(y_t|y_{1:t-1})}$$

where

$$p(y_t|y_{1:t-1}) = \int g(y_t|x_t)\ p(x_{1:t}|y_{1:t-1})\, dx_{1:t}$$

- This can be alternatively written as

Prediction $p(x_{1:t}|y_{1:t-1}) = f(x_t|x_{t-1})\, p(x_{1:t-1}|y_{1:t-1})$,

Update $p(x_{1:t}|y_{1:t}) = \frac{g(y_t|x_t)p(x_{1:t}|y_{1:t-1})}{p(y_t|y_{1:t-1})}$.

- SMC is a simple and natural simulation-based implementation of this recursion.

Monte Carlo Implementation of Prediction Step

- Assume you have at time $t-1$

$$\widehat{p}\left(x_{1:t-1}\middle|\, y_{1:t-1}\right)=\frac{1}{N}\sum_{i=1}^{N}\delta_{X_{1:t-1}^{(i)}}\left(x_{1:t-1}\right).$$

Monte Carlo Implementation of Prediction Step

- Assume you have at time $t-1$

$$\widehat{p}\left(x_{1:t-1}\middle|\, y_{1:t-1}\right) = \frac{1}{N}\sum_{i=1}^{N}\delta_{X_{1:t-1}^{(i)}}\left(x_{1:t-1}\right).$$

- By sampling $\widetilde{X}_t^{(i)} \sim f\left(x_t \middle|\, X_{t-1}^{(i)}\right)$ and setting $\widetilde{X}_{1:t}^{(i)} = \left(X_{1:t-1}^{(i)}, \widetilde{X}_t^{(i)}\right)$ then

$$\widehat{p}\left(x_{1:t}\middle|\, y_{1:t-1}\right) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right).$$

Monte Carlo Implementation of Prediction Step

- Assume you have at time $t-1$

$$\widehat{p}\left(x_{1:t-1} \middle| y_{1:t-1} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:t-1}^{(i)}} \left(x_{1:t-1} \right).$$

- By sampling $\widetilde{X}_t^{(i)} \sim f\left(x_t \middle| X_{t-1}^{(i)} \right)$ and setting $\widetilde{X}_{1:t}^{(i)} = \left(X_{1:t-1}^{(i)}, \widetilde{X}_t^{(i)} \right)$ then

$$\widehat{p}\left(x_{1:t} \middle| y_{1:t-1} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\widetilde{X}_{1:t}^{(i)}} \left(x_{1:t} \right).$$

- Sampling from $f\left(x_t \middle| x_{t-1} \right)$ is usually straightforward and can be done even if $f\left(x_t \middle| x_{t-1} \right)$ does not admit any analytical expression; e.g. biochemical network models.

Importance Sampling Implementation of Updating Step

- Our target at time t is

$$p\left(x_{1:t} \middle| y_{1:t}\right) = \frac{g\left(y_t \middle| x_t\right) p\left(x_{1:t} \middle| y_{1:t-1}\right)}{p\left(y_t \middle| y_{1:t-1}\right)}$$

so by substituting $\widehat{p}\left(x_{1:t} \middle| y_{1:t-1}\right)$ to $p\left(x_{1:t} \middle| y_{1:t-1}\right)$ we obtain

$$\begin{aligned}\widehat{p}\left(y_t \middle| y_{1:t-1}\right) &= \int g\left(y_t \middle| x_t\right) \widehat{p}\left(x_{1:t} \middle| y_{1:t-1}\right) dx_{1:t} \\ &= \frac{1}{N} \sum_{i=1}^{N} g\left(y_t \middle| \widetilde{X}_t^{(i)}\right).\end{aligned}$$

Importance Sampling Implementation of Updating Step

- Our target at time t is

$$p\left(x_{1:t}\middle|\, y_{1:t}\right) = \frac{g\left(y_t\middle|\, x_t\right) p\left(x_{1:t}\middle|\, y_{1:t-1}\right)}{p\left(y_t\middle|\, y_{1:t-1}\right)}$$

so by substituting $\widehat{p}\left(x_{1:t}\middle|\, y_{1:t-1}\right)$ to $p\left(x_{1:t}\middle|\, y_{1:t-1}\right)$ we obtain

$$\begin{aligned}
\widehat{p}\left(y_t\middle|\, y_{1:t-1}\right) &= \int g\left(y_t\middle|\, x_t\right) \widehat{p}\left(x_{1:t}\middle|\, y_{1:t-1}\right) dx_{1:t} \\
&= \frac{1}{N} \sum_{i=1}^{N} g\left(y_t\middle|\, \widetilde{X}_t^{(i)}\right).
\end{aligned}$$

- We now have

$$\widetilde{p}\left(x_{1:t}\middle|\, y_{1:t}\right) = \frac{g\left(y_t\middle|\, x_t\right) \widehat{p}\left(x_{1:t}\middle|\, y_{1:t-1}\right)}{\widehat{p}\left(y_t\middle|\, y_{1:t-1}\right)} = \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right).$$

with $W_t^{(i)} \propto g\left(y_t\middle|\, \widetilde{X}_t^{(i)}\right)$, $\sum_{i=1}^{N} W_t^{(i)} = 1$.

Multinomial Resampling

- We have a "weighted" approximation $\widetilde{p}(x_{1:t}|y_{1:t})$ of $p(x_{1:t}|y_{1:t})$

$$\widetilde{p}(x_{1:t}|y_{1:t}) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}(x_{1:t}).$$

Multinomial Resampling

- We have a "weighted" approximation $\widetilde{p}\left(x_{1:t}\middle|\,y_{1:t}\right)$ of $p\left(x_{1:t}\middle|\,y_{1:t}\right)$

$$\widetilde{p}\left(x_{1:t}\middle|\,y_{1:t}\right)=\sum_{i=1}^{N}W_t^{(i)}\delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right).$$

- To obtain N samples $X_{1:t}^{(i)}$ approximately distributed according to $p\left(x_{1:t}\middle|\,y_{1:t}\right)$, resample N times with replacement

$$X_{1:t}^{(i)}\sim\widetilde{p}\left(x_{1:t}\middle|\,y_{1:t}\right)$$

to obtain

$$\widehat{p}\left(x_{1:t}\middle|\,y_{1:t}\right)=\frac{1}{N}\sum_{i=1}^{N}\delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right)=\sum_{i=1}^{N}\frac{N_t^{(i)}}{N}\delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right)$$

where $\left\{N_t^{(i)}\right\}$ follow a multinomial with $\mathbb{E}\left[N_t^{(i)}\right]=NW_t^{(i)}$, $\mathbb{V}\left[N_t^{(1)}\right]=NW_t^{(i)}\left(1-W_t^{(i)}\right)$.

Multinomial Resampling

- We have a "weighted" approximation $\widetilde{p}\left(x_{1:t} \mid y_{1:t}\right)$ of $p\left(x_{1:t} \mid y_{1:t}\right)$

$$\widetilde{p}\left(x_{1:t} \mid y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right).$$

- To obtain N samples $X_{1:t}^{(i)}$ approximately distributed according to $p\left(x_{1:t} \mid y_{1:t}\right)$, resample N times with replacement

$$X_{1:t}^{(i)} \sim \widetilde{p}\left(x_{1:t} \mid y_{1:t}\right)$$

to obtain

$$\widehat{p}\left(x_{1:t} \mid y_{1:t}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right) = \sum_{i=1}^{N} \frac{N_t^{(i)}}{N} \delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right)$$

where $\left\{N_t^{(i)}\right\}$ follow a multinomial with $\mathbb{E}\left[N_t^{(i)}\right] = N W_t^{(i)}$, $\mathbb{V}\left[N_t^{(1)}\right] = N W_t^{(i)}\left(1 - W_t^{(i)}\right)$.

- This can be achieved in $\mathcal{O}(N)$.

Vanilla SMC: Bootstrap Filter (Gordon et al., 1993)

At time $t = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \quad W_1^{(i)} \propto g\left(y_1 \middle| \widetilde{X}_1^{(i)}\right).$$

Vanilla SMC: Bootstrap Filter (Gordon et al., 1993)

At time $t = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \quad W_1^{(i)} \propto g\left(y_1 | \widetilde{X}_1^{(i)}\right).$$

- Resample $X_1^{(i)} \sim \widetilde{p}(x_1 | y_1)$ to obtain $\widehat{p}(x_1 | y_1) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_1^{(i)}}(x_1)$.

Vanilla SMC: Bootstrap Filter (Gordon et al., 1993)

At time $t = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \quad W_1^{(i)} \propto g\left(y_1 | \widetilde{X}_1^{(i)}\right).$$

- Resample $X_1^{(i)} \sim \widetilde{p}(x_1 | y_1)$ to obtain $\widehat{p}(x_1 | y_1) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_1^{(i)}}(x_1)$.

Vanilla SMC: Bootstrap Filter (Gordon et al., 1993)

At time $t = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \quad W_1^{(i)} \propto g\left(y_1 \middle| \widetilde{X}_1^{(i)}\right).$$

- Resample $X_1^{(i)} \sim \widetilde{p}(x_1 | y_1)$ to obtain $\widehat{p}(x_1 | y_1) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_1^{(i)}}(x_1)$.

At time $t \geq 2$

- Sample $\widetilde{X}_t^{(i)} \sim f\left(x_t \middle| X_{t-1}^{(i)}\right)$, set $\widetilde{X}_{1:t}^{(i)} = \left(X_{1:t-1}^{(i)}, \widetilde{X}_t^{(i)}\right)$ and

$$\widetilde{p}(x_{1:t} | y_{1:t}) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}(x_{1:t}), \quad W_t^{(i)} \propto g\left(y_t \middle| \widetilde{X}_t^{(i)}\right).$$

Vanilla SMC: Bootstrap Filter (Gordon et al., 1993)

At time $t = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1|y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \ W_1^{(i)} \propto g\left(y_1|\widetilde{X}_1^{(i)}\right).$$

- Resample $X_1^{(i)} \sim \widetilde{p}(x_1|y_1)$ to obtain $\widehat{p}(x_1|y_1) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_1^{(i)}}(x_1)$.

At time $t \geq 2$

- Sample $\widetilde{X}_t^{(i)} \sim f\left(x_t|X_{t-1}^{(i)}\right)$, set $\widetilde{X}_{1:t}^{(i)} = \left(X_{1:t-1}^{(i)}, \widetilde{X}_t^{(i)}\right)$ and

$$\widetilde{p}(x_{1:t}|y_{1:t}) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}(x_{1:t}), \ W_t^{(i)} \propto g\left(y_t|\widetilde{X}_t^{(i)}\right).$$

- Resample $X_{1:t}^{(i)} \sim \widetilde{p}(x_{1:t}|y_{1:t})$ to obtain $\widehat{p}(x_{1:t}|y_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}}(x_{1:t})$.

SMC Output

- At time t, we get

$$\widetilde{p}\left(x_{1:t} \middle| y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right),$$

$$\widehat{p}\left(x_{1:t} \middle| y_{1:t}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right).$$

SMC Output

- At time t, we get

$$\widetilde{p}\left(x_{1:t} \middle| y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right),$$

$$\widehat{p}\left(x_{1:t} \middle| y_{1:t}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right).$$

- The marginal likelihood estimate is given by

$$\widehat{p}\left(y_{1:t}\right) = \prod_{k=1}^{t} \widehat{p}\left(y_k \middle| y_{1:k-1}\right) = \prod_{k=1}^{t} \left(\frac{1}{N} \sum_{i=1}^{N} g\left(y_k \middle| \widetilde{X}_k^{(i)}\right)\right).$$

SMC Output

- At time t, we get

$$\begin{aligned}\widetilde{p}\left(x_{1:t} \middle| y_{1:t}\right) &= \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right), \\ \widehat{p}\left(x_{1:t} \middle| y_{1:t}\right) &= \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right).\end{aligned}$$

- The marginal likelihood estimate is given by

$$\widehat{p}\left(y_{1:t}\right) = \prod_{k=1}^{t} \widehat{p}\left(y_k \middle| y_{1:k-1}\right) = \prod_{k=1}^{t} \left(\frac{1}{N} \sum_{i=1}^{N} g\left(y_k \middle| \widetilde{X}_k^{(i)}\right) \right).$$

- Computational complexity is $\mathcal{O}(N)$ at each time step and memory requirements $\mathcal{O}(tN)$.

SMC Output

- At time t, we get

$$\widetilde{p}\left(x_{1:t}\middle|\, y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right),$$

$$\widehat{p}\left(x_{1:t}\middle|\, y_{1:t}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right).$$

- The marginal likelihood estimate is given by

$$\widehat{p}\left(y_{1:t}\right) = \prod_{k=1}^{t} \widehat{p}\left(y_k\middle|\, y_{1:k-1}\right) = \prod_{k=1}^{t}\left(\frac{1}{N}\sum_{i=1}^{N} g\left(y_k\middle|\, \widetilde{X}_k^{(i)}\right)\right).$$

- Computational complexity is $\mathcal{O}\left(N\right)$ at each time step and memory requirements $\mathcal{O}\left(tN\right)$.
- If we are only interested in $p\left(x_t\middle|\, y_{1:t}\right)$ or $p\left(s_t\left(x_{1:t}\right)\middle|\, y_{1:t}\right)$ where $s_t\left(x_{1:t}\right) = \Psi_t\left(x_t, s_{t-1}\left(x_{1:t-1}\right)\right)$ - e.g. $s_t\left(x_{1:t}\right) = \sum_{k=1}^{t} x_k^2$ - is fixed-dimensional then memory requirements $\mathcal{O}\left(N\right)$.

SMC on Path-Space - figures by Olivier Cappė

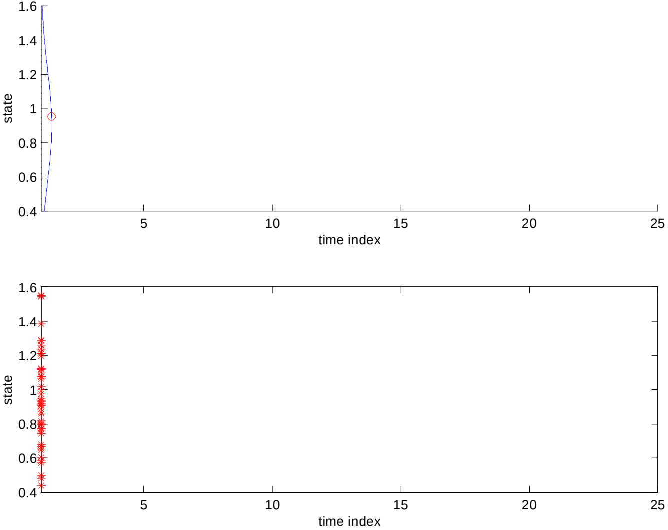


Figure: $p\left(x_1 \mid y_1\right)$ and $\widehat{\mathbb{E}}\left[X_1 \mid y_1\right]$ (top) and particle approximation of $p\left(x_1 \mid y_1\right)$ (bottom)

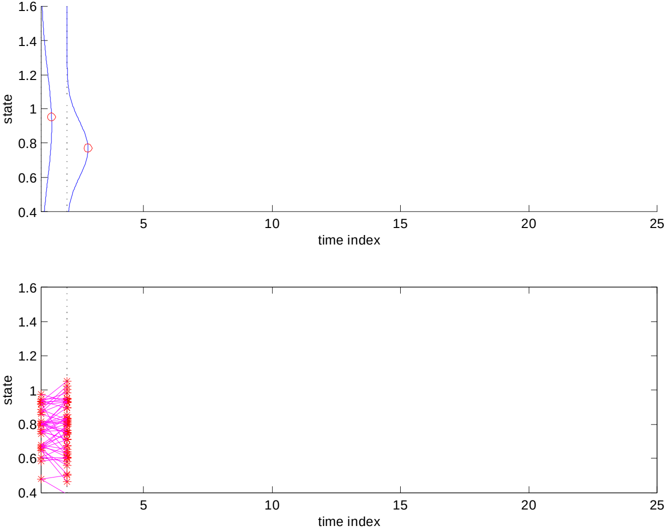


Figure: $p\left(x_1 \mid y_1\right)$, $p\left(x_2 \mid y_{1:2}\right)$ and $\widehat{\mathbb{E}}\left[X_1 \mid y_1\right]$, $\widehat{\mathbb{E}}\left[X_2 \mid y_{1:2}\right]$ (top) and particle approximation of $p\left(x_{1:2} \mid y_{1:2}\right)$ (bottom)

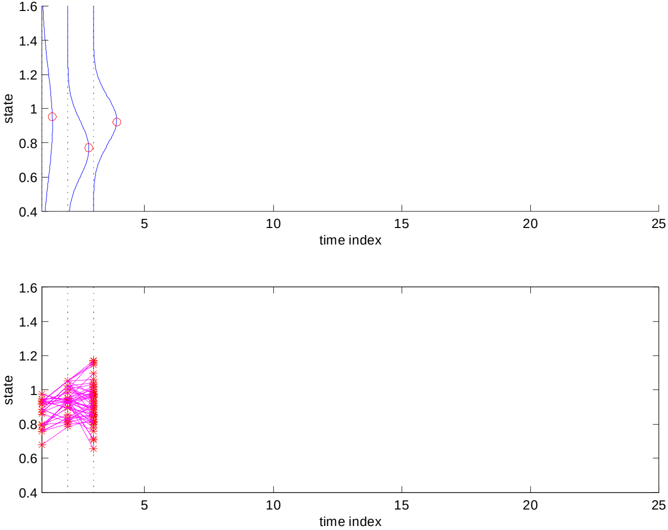


Figure: $p\left(x_t \mid y_{1:t}\right)$ and $\widehat{\mathbb{E}}\left[X_t \mid y_{1:t}\right]$ for $t = 1, 2, 3$ (top) and particle approximation of $p\left(x_{1:3} \mid y_{1:3}\right)$ (bottom)

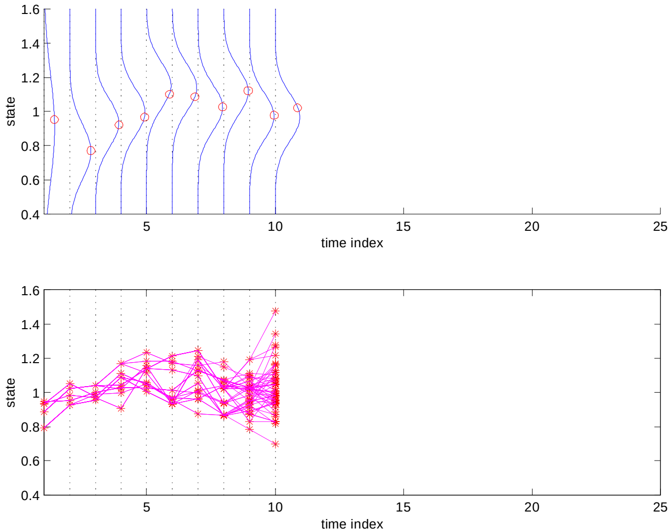
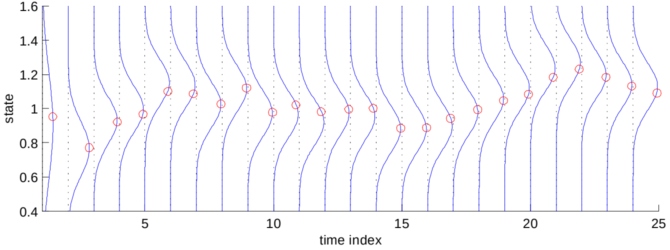


Figure: $p\left(x_t \mid y_{1:t}\right)$ and $\widehat{\mathbb{E}}\left[X_t \mid y_{1:t}\right]$ for $t = 1, ..., 10$ (top) and particle approximation of $p\left(x_{1:10} \mid y_{1:10}\right)$ (bottom)



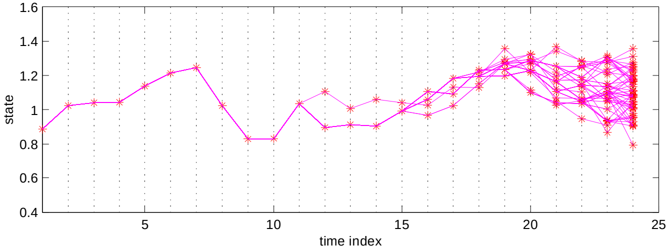


Figure: $p(x_t | y_{1:t})$ and $\widehat{\mathbb{E}}[X_t | y_{1:t}]$ for $t = 1, ..., 24$ (top) and particle approximation of $p(x_{1:24} | y_{1:24})$ (bottom)

Remarks

- Empirically this SMC strategy performs well in terms of estimating the marginals $\{p(x_t | y_{1:t})\}_{t \geq 1}$. This is what is only necessary in many applications thankfully.

Remarks

- Empirically this SMC strategy performs well in terms of estimating the marginals $\{p(x_t | y_{1:t})\}_{t \geq 1}$. This is what is only necessary in many applications thankfully.
- However, the joint distribution $p(x_{1:t} | y_{1:t})$ is poorly estimated when t is large; i.e. we have in the previous example

$$\widehat{p}(x_{1:11} | y_{1:24}) = \delta_{X^*_{1:11}}(x_{1:11}).$$

Remarks

- Empirically this SMC strategy performs well in terms of estimating the marginals $\{p(x_t | y_{1:t})\}_{t\geq 1}$. This is what is only necessary in many applications thankfully.
- However, the joint distribution $p(x_{1:t} | y_{1:t})$ is poorly estimated when t is large; i.e. we have in the previous example

$$\widehat{p}(x_{1:11} | y_{1:24}) = \delta_{X^*_{1:11}}(x_{1:11}).$$

- **Degeneracy problem**. For any N and any k, there exists $t(k, N)$ such that for any $t \geq t(k, N)$

$$\widehat{p}(x_{1:k} | y_{1:t}) = \delta_{X^*_{1:k}}(x_{1:k});$$

$\widehat{p}(x_{1:t} | y_{1:t})$ is an unreliable approximation of $p(x_{1:t} | y_{1:t})$ as $t \nearrow$.

Another Illustration of the Degeneracy Phenomenon

- For the linear Gaussian state-space model described before, we can compute exactly S_t / t where

$$S_t = \int \left(\sum_{k=1}^{t} x_k^2 \right) p\left(x_{1:t} \middle| y_{1:t} \right) dx_{1:t}$$

using Kalman techniques.

Another Illustration of the Degeneracy Phenomenon

- For the linear Gaussian state-space model described before, we can compute exactly S_t / t where

$$S_t = \int \left(\sum_{k=1}^{t} x_k^2 \right) p\left(x_{1:t} \middle| y_{1:t} \right) dx_{1:t}$$

using Kalman techniques.

- We compute the SMC estimate of this quantity using $\widehat{S}_t / t$ where

$$\widehat{S}_t = \int \left(\sum_{k=1}^{t} x_k^2 \right) \widehat{p}\left(x_{1:t} \middle| y_{1:t} \right) dx_{1:t}$$

can be computed sequentially.

Another Illustration of the Degeneracy Phenomenon

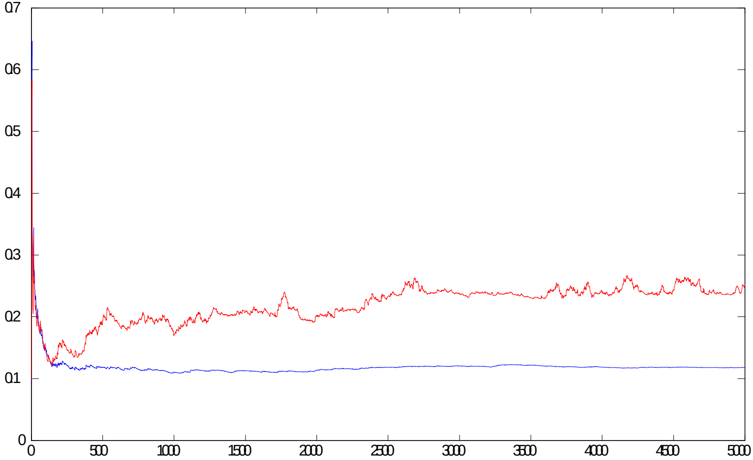


Figure: S_t / t obtained through the Kalman smoother (blue) and its SMC estimate $\widehat{S}_t / t$ (red).

Some Convergence Results for SMC

- Numerous convergence results for SMC are available; see (Del Moral, 2004).

Some Convergence Results for SMC

- Numerous convergence results for SMC are available; see (Del Moral, 2004).
- Let $\varphi_t : \mathcal{X}^t \rightarrow \mathbb{R}$ and consider

$$\overline{\varphi}_t = \int \varphi_t(x_{1:t})\, p(x_{1:t} | y_{1:t})\, dx_{1:t},$$

$$\widehat{\varphi}_t = \int \varphi_t(x_{1:t})\, \widehat{p}(x_{1:t} | y_{1:t})\, dx_{1:t} = \frac{1}{N}\sum_{i=1}^{N} \varphi_t\left(X_{1:t}^{(i)}\right).$$

Some Convergence Results for SMC

- Numerous convergence results for SMC are available; see (Del Moral, 2004).
- Let $\varphi_t : \mathcal{X}^t \to \mathbb{R}$ and consider

$$\overline{\varphi}_t = \int \varphi_t(x_{1:t})\, p(x_{1:t}|\, y_{1:t})\, dx_{1:t},$$

$$\widehat{\varphi}_t = \int \varphi_t(x_{1:t})\, \widehat{p}(x_{1:t}|\, y_{1:t})\, dx_{1:t} = \frac{1}{N}\sum_{i=1}^{N} \varphi_t\left(X_{1:t}^{(i)}\right).$$

- We can prove that for any bounded function φ and any $p \geq 1$

$$\mathbb{E}\left[\left|\widehat{\varphi}_t - \overline{\varphi}_t\right|^p\right]^{1/p} \leq \frac{B(t)\, c(p)\, \|\varphi\|_\infty}{\sqrt{N}},$$

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_t - \overline{\varphi}_t\right) \Rightarrow \mathcal{N}\left(0, \sigma_t^2\right).$$

Some Convergence Results for SMC

- Numerous convergence results for SMC are available; see (Del Moral, 2004).
- Let $\varphi_t : \mathcal{X}^t \rightarrow \mathbb{R}$ and consider

$$\overline{\varphi}_t = \int \varphi_t(x_{1:t})\, p(x_{1:t}|\, y_{1:t})\, dx_{1:t},$$

$$\widehat{\varphi}_t = \int \varphi_t(x_{1:t})\, \widehat{p}(x_{1:t}|\, y_{1:t})\, dx_{1:t} = \frac{1}{N}\sum_{i=1}^{N} \varphi_t\left(X_{1:t}^{(i)}\right).$$

- We can prove that for any bounded function φ and any $p \geq 1$

$$\mathbb{E}\left[|\widehat{\varphi}_t - \overline{\varphi}_t|^p\right]^{1/p} \leq \frac{B(t)\, c(p)\, \|\varphi\|_\infty}{\sqrt{N}},$$

$$\lim_{N\rightarrow\infty} \sqrt{N}\left(\widehat{\varphi}_t - \overline{\varphi}_t\right) \Rightarrow \mathcal{N}\left(0, \sigma_t^2\right).$$

- **Very weak results**: $B(t)$ and σ_t^2 can increase with t and will for a path-dependent $\varphi_t(x_{1:t})$ as the degeneracy problem suggests.

Stronger Convergence Results

- Assume the following **exponentially stability assumption**: For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_t \middle| y_{2:t}, X_1 = x_1 \right) - p\left(x_t \middle| y_{2:t}, X_1 = x_1' \right) \right| dx_t \leq \alpha^t \text{ for } 0 \leq \alpha < 1.$$

Stronger Convergence Results

- Assume the following **exponentially stability assumption**: For any x_1, x_1'

$$\frac{1}{2}\int \left|p\left(x_t|\, y_{2:t}, X_1 = x_1\right) - p\left(x_t|\, y_{2:t}, X_1 = x_1'\right)\right| dx_t \leq \alpha^t \text{ for } 0 \leq \alpha < 1.$$

- **Marginal distribution**. For $\varphi_t\left(x_{1:t}\right) = \varphi\left(x_{t-L:t}\right)$, there exists $B_1, B_2 < \infty$ s.t.

$$\mathbb{E}\left[\left|\widehat{\varphi}_t - \overline{\varphi}_t\right|^p\right]^{1/p} \leq \frac{B_1\ c\left(p\right)\ \left\|\varphi\right\|_\infty}{\sqrt{N}},$$

$$\lim_{N\to\infty}\sqrt{N}\left(\widehat{\varphi}_t - \overline{\varphi}_t\right) \Rightarrow \mathcal{N}\left(0, \sigma_t^2\right) \text{ where } \sigma_t^2 \leq B_2,$$

i.e. there is no accumulation of numerical errors over time.

Stronger Convergence Results

- Assume the following **exponentially stability assumption**: For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_t \middle| y_{2:t}, X_1 = x_1 \right) - p\left(x_t \middle| y_{2:t}, X_1 = x_1' \right) \right| dx_t \leq \alpha^t \text{ for } 0 \leq \alpha < 1.$$

- **Marginal distribution**. For $\varphi_t\left(x_{1:t}\right) = \varphi\left(x_{t-L:t}\right)$, there exists $B_1, B_2 < \infty$ s.t.

$$\mathbb{E}\left[\left|\widehat{\varphi}_t - \overline{\varphi}_t\right|^p\right]^{1/p} \leq \frac{B_1 \ c\left(p\right) \ \left\|\varphi\right\|_\infty}{\sqrt{N}},$$

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_t - \overline{\varphi}_t\right) \Rightarrow \mathcal{N}\left(0, \sigma_t^2\right) \text{ where } \sigma_t^2 \leq B_2,$$

  i.e. there is no accumulation of numerical errors over time.

- **L1 distance.** If $\overline{p}\left(x_{1:t} \middle| y_{1:t} \right) = \mathbb{E}\left(\widehat{p}\left(x_{1:t} \middle| y_{1:t} \right)\right)$, there exists $B_3 < \infty$ s.t.

$$\int \left| \overline{p}\left(x_{1:t} \middle| y_{1:t} \right) - p\left(x_{1:t} \middle| y_{1:t} \right) \right| dx_{1:t} \leq \frac{B_3 \ t}{N};$$

  i.e. the bias only increases in t.

Stronger Convergence Results

- **Unbiasedness**. The marginal likelihood estimate is unbiased

$$\mathbb{E}\left(\widehat{p}\left(y_{1:t}\right)\right) = p\left(y_{1:t}\right).$$

Stronger Convergence Results

- **Unbiasedness**. The marginal likelihood estimate is unbiased

$$\mathbb{E}\left(\widehat{p}\left(y_{1:t}\right)\right) = p\left(y_{1:t}\right).$$

- **Relative Variance Bound**. There exists $B_4 < \infty$

$$\mathbb{E}\left(\left(\frac{\widehat{p}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)} - 1\right)^2\right) \leq \frac{B_4\ t}{N}$$

Stronger Convergence Results

- **Unbiasedness**. The marginal likelihood estimate is unbiased

$$\mathbb{E}\left(\widehat{p}\left(y_{1:t}\right)\right)=p\left(y_{1:t}\right).$$

- **Relative Variance Bound**. There exists $B_4<\infty$

$$\mathbb{E}\left(\left(\frac{\widehat{p}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)}-1\right)^2\right)\leq\frac{B_4\ t}{N}$$

- **Central Limit Theorem**. There exists $B_5<\infty$ s.t.

$$\lim_{N\rightarrow\infty}\sqrt{N}\left(\log\widehat{p}\left(y_{1:t}\right)-\log p\left(y_{1:t}\right)\right)\Rightarrow\mathcal{N}\left(0,\overline{\sigma}_t^2\right)\text{ with }\overline{\sigma}_t^2\leq B_5\ t.$$

Basic Idea Used to Establish Uniform Lp Bounds

- We denote

$$\eta_k(x_k) = p(x_k | y_{1:k-1})$$

and

$$\widehat{\eta}_k(x_k) = \widehat{p}(x_k | y_{1:k-1})$$

its particle approximation.

Basic Idea Used to Establish Uniform Lp Bounds

- We denote
$$\eta_k(x_k) = p(x_k | y_{1:k-1})$$
and
$$\widehat{\eta}_k(x_k) = \widehat{p}(x_k | y_{1:k-1})$$
its particle approximation.
- Let $\Phi_{k,t}$ be the measure-valued mapping such that
$$\eta_t = \Phi_{k,t}(\eta_k),$$
which satifies
$$\Phi_{k,t}(\eta_k)(x_t) = \int \underbrace{\frac{\eta_k(x_k) . p(y_{k:t-1} | x_k)}{\int \eta_k(x_k) p(y_{k:t-1} | x_k) dx_k}}_{p(x_k | y_{1:t-1})} p(x_t | x_k, y_{k+1:t-1}) dx_k.$$

Key Decomposition Formula

$$
\begin{array}{ccccccc}
\eta_1 & \rightarrow & \eta_2 = \Phi_{1,2}(\eta_1) & \rightarrow & \cdots & \rightarrow & \eta_t = \Phi_{1,t}(\eta_1) \\
\Downarrow & & & & & & \\
\widehat{\eta}_1 & \rightarrow & \Phi_{1,2}(\widehat{\eta}_1) & \rightarrow & \cdots & \rightarrow & \Phi_{1,t}(\widehat{\eta}_1) \\
 & & \Downarrow & & & & \\
 & & \widehat{\eta}_2 & \rightarrow & \cdots & \rightarrow & \Phi_{2,t}(\widehat{\eta}_2) \\
 & & & & \Downarrow & & \\
 & & & & \widehat{\eta}_{t-1} & \rightarrow & \Phi_{t-1,t}(\widehat{\eta}_{t-1}) \\
 & & & & & & \Downarrow \\
 & & & & & & \widehat{\eta}_t
\end{array}
$$

- Decomposition of the error

$$\widehat{\eta}_t - \eta_t = \sum_{k=1}^{t} \left[\Phi_{k,t}(\widehat{\eta}_k) - \Phi_{k,t}\left(\Phi_{k-1,k}(\widehat{\eta}_{k-1})\right) \right]$$

Stability Properties

- We have

$$p\left(x_t \mid x_k, y_{k+1:t-1}\right) = \int p\left(x_{k+1:t} \mid x_k, y_{k+1:t-1}\right) dx_{k+1:t-1}$$

where

$$p\left(x_{k+1:t} \mid x_k, y_{k+1:t-1}\right) = \prod_{m=k+1}^{t} p\left(x_m \mid x_{m-1}, y_{m:t-1}\right)$$

Stability Properties

- We have

$$p\left(x_t \mid x_k, y_{k+1:t-1}\right) = \int p\left(x_{k+1:t} \mid x_k, y_{k+1:t-1}\right) dx_{k+1:t-1}$$

where

$$p\left(x_{k+1:t} \mid x_k, y_{k+1:t-1}\right) = \prod_{m=k+1}^{t} p\left(x_m \mid x_{m-1}, y_{m:t-1}\right)$$

- To summarize, we have

$$\begin{aligned}\Phi_{k,t}\left(\eta_k\right)\left(x_t\right) &= \int \underbrace{\frac{\eta_k\left(x_k\right).p\left(y_{k:t-1} \mid x_k\right)}{\int \eta_k\left(x_k\right) p\left(y_{k:t-1} \mid x_k\right) dx_k}}_{p\left(x_k \mid y_{1:t-1}\right)} \\ &\quad \times \prod_{m=k+1}^{t} p\left(x_m \mid x_{m-1}, y_{m:t-1}\right) dx_{k:t-1}\end{aligned}$$

Stability Properties

- Assume there exists $\epsilon > 0$ s.t. for any x, x'

$$\epsilon^{-1}\nu\left(x'\right) \geq f\left(x' \middle| x\right) \geq \epsilon\nu\left(x'\right)$$

and for any y, x,

$$0 < \underline{g} \leq g\left(y \middle| x\right) \leq \overline{g} < \infty$$

then there exists $0 \leq \lambda < 1$

$$\frac{1}{2}\int \left|\Phi_{k,k+t}\left(\eta\right)\left(x\right) - \Phi_{k,k+t}\left(\eta'\right)\left(x\right)\right| dx \leq \lambda^{t}$$

Stability Properties

- Assume there exists $\epsilon > 0$ s.t. for any x, x'

$$\epsilon^{-1}\nu\left(x'\right) \geq f\left(x' \middle| x\right) \geq \epsilon\nu\left(x'\right)$$

  and for any y, x,

$$0 < \underline{g} \leq g\left(y \middle| x\right) \leq \overline{g} < \infty$$

  then there exists $0 \leq \lambda < 1$

$$\frac{1}{2}\int \left|\Phi_{k,k+t}\left(\eta\right)\left(x\right) - \Phi_{k,k+t}\left(\eta'\right)\left(x\right)\right| dx \leq \lambda^{t}$$

- Hence we have

$$\Phi_{k,t}\left(\eta_k\right)\left(x_t\right) \approx \Phi_{k,t}\left(\eta'_k\right)\left(x_t\right)$$

  as $\left(t-k\right) \to \infty$.

Putting Everything Together

- Under such strong mixing assumptions

$$\widehat{\eta}_t - \eta_t = \sum_{k=1}^{t} \underbrace{\left[\Phi_{k,t}\left(\widehat{\eta}_k\right) - \Phi_{k,t}\left(\Phi_{k-1,k}\left(\widehat{\eta}_{k-1}\right)\right)\right]}_{\simeq \frac{1}{\sqrt{N}}\lambda^{t-k+1} \text{ for } 0 \leq \lambda \leq 1}$$

Putting Everything Together

- Under such strong mixing assumptions

$$\widehat{\eta}_t - \eta_t = \sum_{k=1}^{t} \underbrace{\left[\Phi_{k,t}\left(\widehat{\eta}_k\right) - \Phi_{k,t}\left(\Phi_{k-1,k}\left(\widehat{\eta}_{k-1}\right)\right)\right]}_{\simeq \frac{1}{\sqrt{N}}\lambda^{t-k+1} \text{ for } 0 \leq \lambda \leq 1}$$

- We can then obtain results such as there exists $B_1 < \infty$ s.t.

$$\mathbb{E}\left[\left|\widehat{\varphi}_t - \overline{\varphi}_t\right|^p\right]^{1/p} \leq \frac{B_1 \ c\left(p\right) \ \left\|\varphi\right\|_\infty}{\sqrt{N}}$$

Putting Everything Together

- Under such strong mixing assumptions

$$\widehat{\eta}_t - \eta_t = \sum_{k=1}^{t} \underbrace{\left[\Phi_{k,t}\left(\widehat{\eta}_k\right) - \Phi_{k,t}\left(\Phi_{k-1,k}\left(\widehat{\eta}_{k-1}\right)\right)\right]}_{\simeq \frac{1}{\sqrt{N}} \lambda^{t-k+1} \text{ for } 0 \leq \lambda \leq 1}$$

- We can then obtain results such as there exists $B_1 < \infty$ s.t.

$$\mathbb{E}\left[\left|\widehat{\varphi}_t - \overline{\varphi}_t\right|^p\right]^{1/p} \leq \frac{B_1 \; c\left(p\right) \; \left\|\varphi\right\|_\infty}{\sqrt{N}}$$

- Much work has been done recently on removing such strong mixing assumptions; e.g. Whiteley (2012) for much weaker and realistic assumptions.

Summary

- SMC provide consistent estimates under weak assumptions.

Summary

- SMC provide consistent estimates under weak assumptions.
- Under stability assumptions, we have uniform in time stability of the SMC estimates of $\{p(x_t | y_{1:t})\}_{t\geq 1}$.

Summary

- SMC provide consistent estimates under weak assumptions.
- Under stability assumptions, we have uniform in time stability of the SMC estimates of $\{p(x_t | y_{1:t})\}_{t \geq 1}$.
- Under stability assumptions, the relative variance of the SMC estimate of $\{p(y_{1:t})\}_{t \geq 1}$ only increases linearly with t.

Summary

- SMC provide consistent estimates under weak assumptions.
- Under stability assumptions, we have uniform in time stability of the SMC estimates of $\{p(x_t|\, y_{1:t})\}_{t\geq 1}$.
- Under stability assumptions, the relative variance of the SMC estimate of $\{p(y_{1:t})\}_{t\geq 1}$ only increases linearly with t.
- Even under stability assumptions, one cannot expect to obtain uniform in time stability for SMC estimates of $\{p(x_{1:t}|\, y_{1:t})\}_{t\geq 1}$; this is due to the degeneracy problem.

Summary

- SMC provide consistent estimates under weak assumptions.
- Under stability assumptions, we have uniform in time stability of the SMC estimates of $\{p(x_t | y_{1:t})\}_{t \geq 1}$.
- Under stability assumptions, the relative variance of the SMC estimate of $\{p(y_{1:t})\}_{t \geq 1}$ only increases linearly with t.
- Even under stability assumptions, one cannot expect to obtain uniform in time stability for SMC estimates of $\{p(x_{1:t} | y_{1:t})\}_{t \geq 1}$; this is due to the degeneracy problem.
- Is it possible to Q1: eliminate, Q2: mitigate the degeneracy problem?

- SMC provide consistent estimates under weak assumptions.
- Under stability assumptions, we have uniform in time stability of the SMC estimates of $\{p(x_t | y_{1:t})\}_{t \geq 1}$.
- Under stability assumptions, the relative variance of the SMC estimate of $\{p(y_{1:t})\}_{t \geq 1}$ only increases linearly with t.
- Even under stability assumptions, one cannot expect to obtain uniform in time stability for SMC estimates of $\{p(x_{1:t} | y_{1:t})\}_{t \geq 1}$; this is due to the degeneracy problem.
- Is it possible to Q1: eliminate, Q2: mitigate the degeneracy problem?
- Answer: Q1: no, Q2: yes.

Is Resampling Really Necessary?

- Resampling is the source of the degeneracy problem and might appear wasteful.

Is Resampling Really Necessary?

- Resampling is the source of the degeneracy problem and might appear wasteful.
- The resampling step is an unbiased operation

$$\mathbb{E}\left[\widehat{p}\left(x_{1:t}\middle|\,y_{1:t}\right)\middle|\,\widetilde{p}\left(x_{1:t}\middle|\,y_{1:t}\right)\right]=\widetilde{p}\left(x_{1:t}\middle|\,y_{1:t}\right)$$

but clearly it introduces some errors “locally” in time. That is for any test function, we have

$$\mathbb{V}\left[\int\varphi\left(x_{1:t}\right)\widehat{p}\left(x_{1:t}\middle|\,y_{1:t}\right)dx_{1:t}\right]\geq\mathbb{V}\left[\int\varphi\left(x_{1:t}\right)\widetilde{p}\left(x_{1:t}\middle|\,y_{1:t}\right)dx_{1:t}\right]$$

Is Resampling Really Necessary?

- Resampling is the source of the degeneracy problem and might appear wasteful.
- The resampling step is an unbiased operation

$$\mathbb{E}\left[\widehat{p}\left(x_{1:t}\middle|\, y_{1:t}\right)\middle|\,\widetilde{p}\left(x_{1:t}\middle|\, y_{1:t}\right)\right]=\widetilde{p}\left(x_{1:t}\middle|\, y_{1:t}\right)$$

  but clearly it introduces some errors "locally" in time. That is for any test function, we have

$$\mathbb{V}\left[\int\varphi\left(x_{1:t}\right)\widehat{p}\left(x_{1:t}\middle|\, y_{1:t}\right)dx_{1:t}\right]\geq\mathbb{V}\left[\int\varphi\left(x_{1:t}\right)\widetilde{p}\left(x_{1:t}\middle|\, y_{1:t}\right)dx_{1:t}\right]$$

- What about eliminating the resampling step?

Sequential Importance Samping: SMC Without Resampling

- In this case, the estimate of the posterior is

$$\widehat{p}_{\text{SIS}}\left(x_{1:t}\middle|\, y_{1:t}\right)=\sum_{i=1}^{N}W_t^{(i)}\delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right)$$

where $X_{1:t}^{(i)}\sim p\left(x_{1:t}\right)$ and

$$W_t^{(i)}\propto p\left(y_{1:t}\middle|\, X_{1:t}^{(i)}\right)\propto\prod_{k=1}^{t}g\left(y_k\middle|\, X_t^{(i)}\right).$$

Sequential Importance Samping: SMC Without Resampling

- In this case, the estimate of the posterior is

$$\widehat{p}_{\text{SIS}}\left(x_{1:t} \middle| y_{1:t} \right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{1:t}^{(i)}}\left(x_{1:t} \right)$$

where $X_{1:t}^{(i)} \sim p\left(x_{1:t} \right)$ and

$$W_t^{(i)} \propto p\left(y_{1:t} \middle| X_{1:t}^{(i)} \right) \propto \prod_{k=1}^{t} g\left(y_k \middle| X_t^{(i)} \right).$$

- In this case, the marginal likelihood estimate is

$$\widehat{p}_{\text{SIS}}\left(y_{1:t} \right) = \frac{1}{N} \sum_{i=1}^{N} p\left(y_{1:t} \middle| X_{1:t}^{(i)} \right)$$

Sequential Importance Samping: SMC Without Resampling

- In this case, the estimate of the posterior is

$$\widehat{p}_{\text{SIS}}\left(x_{1:t}\middle|\, y_{1:t}\right)=\sum_{i=1}^{N}W_t^{(i)}\delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right)$$

where $X_{1:t}^{(i)}\sim p\left(x_{1:t}\right)$ and

$$W_t^{(i)}\propto p\left(y_{1:t}\middle|\, X_{1:t}^{(i)}\right)\propto\prod_{k=1}^{t}g\left(y_k\middle|\, X_t^{(i)}\right).$$

- In this case, the marginal likelihood estimate is

$$\widehat{p}_{\text{SIS}}\left(y_{1:t}\right)=\frac{1}{N}\sum_{i=1}^{N}p\left(y_{1:t}\middle|\, X_{1:t}^{(i)}\right)$$

- Relative variance of $p\left(y_{1:t}\middle|\, X_{1:t}^{(i)}\right)=\prod_{k=1}^{t}g\left(y_k\middle|\, X_t^{(i)}\right)$ is increasing exponentially fast...

SIS For Stochastic Volatility Model

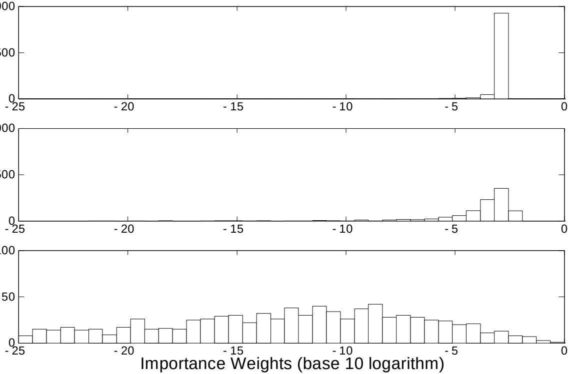


Figure: Histograms of $\log_{10}\left(W_t^{(i)}\right)$ for $t = 1$ (top), $t = 50$ (middle) and $t = 100$ (bottom).

- The algorithm performance collapse as t increases as expected.

Central Limit Theorems

- For both SIS and SMC, we have a CLT for the estimates of the marginal likelihood

$$
\begin{aligned}
\sqrt{N}\left(\frac{\widehat{p}_{\text{SIS}}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)}-1\right) & \Rightarrow \mathcal{N}\left(0,\sigma_{t,\text{SIS}}^{2}\right), \\
\sqrt{N}\left(\frac{\widehat{p}_{\text{SMC}}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)}-1\right) & \Rightarrow \mathcal{N}\left(0,\sigma_{t,\text{SMC}}^{2}\right).
\end{aligned}
$$

Central Limit Theorems

- For both SIS and SMC, we have a CLT for the estimates of the marginal likelihood

$$\sqrt{N}\left(\frac{\widehat{p}_{\text{SIS}}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)}-1\right) \Rightarrow \mathcal{N}\left(0,\sigma^2_{t,\text{SIS}}\right),$$
$$\sqrt{N}\left(\frac{\widehat{p}_{\text{SMC}}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)}-1\right) \Rightarrow \mathcal{N}\left(0,\sigma^2_{t,\text{SMC}}\right).$$

- The variance expressions are

$$\sigma^2_{t,\text{SIS}} = \int \frac{p^2(x_{1:t}|y_{1:t})}{p(x_{1:t})}dx_{1:t} - 1 = \frac{\int p^2(y_{1:t}|x_{1:t})p(x_{1:t})dx_{1:t}}{p^2(y_{1:t})} - 1$$
$$\sigma^2_{t,\text{SMC}} = \int \frac{p^2(x_1|y_{1:t})}{\mu(x_1)}dx_1 + \sum_{k=2}^{t}\int \frac{p^2(x_{1:k}|y_{1:t})}{p(x_{1:k-1}|y_{1:k-1})f(x_k|x_{k-1})}dx_{1:k} - t$$
$$= \frac{\int g^2(y_1|x_1)\mu(x_1)dx_1}{p^2(y_1)} + \sum_{k=2}^{t}\frac{\int p^2(y_{k:t}|x_k)p(x_k|y_{1:k-1})dx_k}{p^2(y_{k:t}|y_{1:k-1})} - t$$

Central Limit Theorems

- For both SIS and SMC, we have a CLT for the estimates of the marginal likelihood

$$\sqrt{N}\left(\frac{\widehat{p}_{\text{SIS}}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)}-1\right) \Rightarrow \mathcal{N}\left(0,\sigma^2_{t,\text{SIS}}\right),$$
$$\sqrt{N}\left(\frac{\widehat{p}_{\text{SMC}}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)}-1\right) \Rightarrow \mathcal{N}\left(0,\sigma^2_{t,\text{SMC}}\right).$$

- The variance expressions are

$$\sigma^2_{t,\text{SIS}} = \int \frac{p^2(x_{1:t}|y_{1:t})}{p(x_{1:t})}dx_{1:t} - 1 = \frac{\int p^2(y_{1:t}|x_{1:t})p(x_{1:t})dx_{1:t}}{p^2(y_{1:t})} - 1$$
$$\sigma^2_{t,\text{SMC}} = \int \frac{p^2(x_1|y_{1:t})}{\mu(x_1)}dx_1 + \sum_{k=2}^{t}\int \frac{p^2(x_{1:k}|y_{1:t})}{p(x_{1:k-1}|y_{1:k-1})f(x_k|x_{k-1})}dx_{1:k} - t$$
$$= \frac{\int g^2(y_1|x_1)\mu(x_1)dx_1}{p^2(y_1)} + \sum_{k=2}^{t}\frac{\int p^2(y_{k:t}|x_k)p(x_k|y_{1:k-1})dx_k}{p^2(y_{k:t}|y_{1:k-1})} - t$$

- SMC "breaks" the integral over $\mathcal{X}^t$ into t integrals over $\mathcal{X}$.

A Toy Example

- Consider the case where $f(x'|x) = \mu(x') = \mathcal{N}(x'; 0, \sigma^2)$ and $g(y|x) = \mathcal{N}\left(y; 0, 1 - \frac{1}{\sigma^2}\right)$ where $\sigma^2 > 1$.

A Toy Example

- Consider the case where $f(x'|x) = \mu(x') = \mathcal{N}(x'; 0, \sigma^2)$ and $g(y|x) = \mathcal{N}\left(y; 0, 1 - \frac{1}{\sigma^2}\right)$ where $\sigma^2 > 1$.
- Assume we observe $y_1 = \cdots = y_t = 0$ then we have

$$\mathbb{V}\left(\frac{\widehat{p}_{\text{SIS}}(y_{1:t})}{p(y_{1:t})}\right) = \frac{\sigma^2_{t,\text{SIS}}}{N} = \frac{1}{N}\left[\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{t/2} - 1\right],$$

$$\mathbb{V}\left(\frac{\widehat{p}_{\text{SMC}}(y_{1:t})}{p(y_{1:t})}\right) \approx \frac{\sigma^2_{t,\text{SMC}}}{N} = \frac{t}{N}\left[\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{1/2} - 1\right].$$

A Toy Example

- Consider the case where $f(x'|x) = \mu(x') = \mathcal{N}(x'; 0, \sigma^2)$ and $g(y|x) = \mathcal{N}\left(y; 0, 1 - \frac{1}{\sigma^2}\right)$ where $\sigma^2 > 1$.
- Assume we observe $y_1 = \cdots = y_t = 0$ then we have

$$\mathbb{V}\left(\frac{\widehat{p}_{\text{SIS}}(y_{1:t})}{p(y_{1:t})}\right) = \frac{\sigma^2_{t,\text{SIS}}}{N} = \frac{1}{N}\left[\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{t/2} - 1\right],$$

$$\mathbb{V}\left(\frac{\widehat{p}_{\text{SMC}}(y_{1:t})}{p(y_{1:t})}\right) \approx \frac{\sigma^2_{t,\text{SMC}}}{N} = \frac{t}{N}\left[\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{1/2} - 1\right].$$

- If select $\sigma^2 = 1.2$ then it is necessary to use $N \approx 2 \times 10^{23}$ particles to obtain $\frac{\sigma^2_{t,\text{SIS}}}{N} = 10^{-2}$ for $t = 1000$.

A Toy Example

- Consider the case where $f\left(x'\middle|\, x\right) = \mu\left(x'\right) = \mathcal{N}\left(x'; 0, \sigma^2\right)$ and $g\left(y\middle|\, x\right) = \mathcal{N}\left(y; 0, 1 - \frac{1}{\sigma^2}\right)$ where $\sigma^2 > 1$.
- Assume we observe $y_1 = \cdots = y_t = 0$ then we have

$$\mathbb{V}\left(\frac{\widehat{p}_{\text{SIS}}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)}\right) = \frac{\sigma^2_{t,\text{SIS}}}{N} = \frac{1}{N}\left[\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{t/2} - 1\right],$$

$$\mathbb{V}\left(\frac{\widehat{p}_{\text{SMC}}\left(y_{1:t}\right)}{p\left(y_{1:t}\right)}\right) \approx \frac{\sigma^2_{t,\text{SMC}}}{N} = \frac{t}{N}\left[\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{1/2} - 1\right].$$

- If select $\sigma^2 = 1.2$ then it is necessary to use $N \approx 2 \times 10^{23}$ particles to obtain $\frac{\sigma^2_{t,\text{SIS}}}{N} = 10^{-2}$ for $t = 1000$.
- To obtain $\frac{\sigma^2_{t,\text{SMC}}}{N} = 10^{-2}$, SMC requires only $N \approx 10^4$ particles: improvement by 19 orders of magnitude!

Better Resampling Schemes

- Better resampling steps can be designed such that $\mathbb{E}\left[N_t^{(i)}\right] = NW_t^{(i)}$ but $\mathbb{V}\left[N_t^{(i)}\right] < NW_t^{(i)}\left(1 - W_t^{(i)}\right)$; residual resampling, minimal entropy resampling etc. (Cappé et al., 2005).

Better Resampling Schemes

- Better resampling steps can be designed such that $\mathbb{E}\left[N_t^{(i)}\right] = NW_t^{(i)}$ but $\mathbb{V}\left[N_t^{(i)}\right] < NW_t^{(i)}\left(1 - W_t^{(i)}\right)$; residual resampling, minimal entropy resampling etc. (Cappé et al., 2005).
- *Residual Resampling*. Set $\widetilde{N}_t^{(i)} = \left\lfloor NW_t^{(i)} \right\rfloor$, sample $\overline{N}_t^{1:N}$ from a multinomial of parameters $\left(N, \overline{W}_t^{(1:N)}\right)$ where $\overline{W}_t^{(i)} \propto W_t^{(i)} - N^{-1}\widetilde{N}_t^{(i)}$ then set $N_t^{(i)} = \widetilde{N}_t^{(i)} + \overline{N}_t^{(i)}$.

Better Resampling Schemes

- Better resampling steps can be designed such that $\mathbb{E}\left[N_t^{(i)}\right] = NW_t^{(i)}$ but $\mathbb{V}\left[N_t^{(i)}\right] < NW_t^{(i)}\left(1 - W_t^{(i)}\right)$; residual resampling, minimal entropy resampling etc. (Cappé et al., 2005).
- *Residual Resampling*. Set $\widetilde{N}_t^{(i)} = \left\lfloor NW_t^{(i)} \right\rfloor$, sample $\overline{N}_t^{1:N}$ from a multinomial of parameters $\left(N, \overline{W}_t^{(1:N)}\right)$ where $\overline{W}_t^{(i)} \propto W_t^{(i)} - N^{-1}\widetilde{N}_t^{(i)}$ then set $N_t^{(i)} = \widetilde{N}_t^{(i)} + \overline{N}_t^{(i)}$.
- *Systematic Resampling*. Sample $U_1 \sim \mathcal{U}\left[0, \frac{1}{N}\right]$ and define $U_i = U_1 + \frac{i-1}{N}$ for $i = 2, ..., N$, then set $N_t^i = \left|\left\{U_j : \sum_{k=1}^{i-1} W_t^{(k)} \leq U_j \leq \sum_{k=1}^{i} W_t^{(k)}\right\}\right|$ with the convention $\sum_{k=1}^{0} := 0$.

Measuring Variability of the Weights

- To measure the variation of the weights, we can use the Effective Sample Size (ESS)

$$ESS = \left(\sum_{i=1}^{N} \left(W_t^{(i)} \right)^2 \right)^{-1}$$

Measuring Variability of the Weights

- To measure the variation of the weights, we can use the Effective Sample Size (ESS)

$$ESS = \left(\sum_{i=1}^{N} \left(W_t^{(i)} \right)^2 \right)^{-1}$$

- We have $ESS = N$ if $W_t^{(i)} = 1/N$ for any i and $ESS = 1$ if $W_t^{(i)} = 1$ and $W_t^{(j)} = 1$ for $j \neq i$.

Measuring Variability of the Weights

- To measure the variation of the weights, we can use the Effective Sample Size (ESS)

$$ESS = \left(\sum_{i=1}^{N} \left(W_t^{(i)} \right)^2 \right)^{-1}$$

- We have $ESS = N$ if $W_t^{(i)} = 1/N$ for any i and $ESS = 1$ if $W_t^{(i)} = 1$ and $W_t^{(j)} = 1$ for $j \neq i$.
- Liu (1996) showed that for simple importance sampling for φ "regular enough"

$$\mathbb{V}\left(\sum_{i=1}^{N} W_t^{(i)} \varphi \left(X_t^{(i)} \right) \right) \approx \mathbb{V}_{p(x_{1:t}|y_{1:t})} \left(\frac{1}{ESS} \sum_{i=1}^{ESS} \varphi \left(X_t^{(i)} \right) \right);$$

i.e. the estimate is roughly as accurate as using an iid sample of size ESS from $p(x_{1:t}|y_{1:t})$.

Dynamic Resampling

- Resampling at each time step can be harmful: only resample when necessary.

Dynamic Resampling

- Resampling at each time step can be harmful: only resample when necessary.
- **Dynamic Resampling**: If the variation of the weights as measured by ESS is too high, e.g. $ESS < N/2$, then resample the particles.

Dynamic Resampling

- Resampling at each time step can be harmful: only resample when necessary.
- **Dynamic Resampling**: If the variation of the weights as measured by ESS is too high, e.g. $ESS < N/2$, then resample the particles.
- We can also use the entropy

$$Ent = -\sum_{i=1}^{N} W_t^{(i)} \log_2 \left(W_t^{(i)} \right)$$

Dynamic Resampling

- Resampling at each time step can be harmful: only resample when necessary.
- **Dynamic Resampling**: If the variation of the weights as measured by *ESS* is too high, e.g. $ESS < N/2$, then resample the particles.
- We can also use the entropy

$$Ent = -\sum_{i=1}^{N} W_t^{(i)} \log_2 \left(W_t^{(i)} \right)$$

- We have $Ent = \log_2 (N)$ if $W_t^{(i)} = 1/N$ for any i. We have $Ent = 0$ if $W_t^{(i)} = 1$ and $W_t^{(j)} = 1$ for $j \neq i$.

Improving the Sampling Step

- **Bootstrap filter**. Sample particles blindly according to the prior without taking into account the observation
  $\rightsquigarrow$ Very inefficient for vague prior/peaky likelihood.

Improving the Sampling Step

- **Bootstrap filter**. Sample particles blindly according to the prior without taking into account the observation
  $\rightsquigarrow$ Very inefficient for vague prior/peaky likelihood.
- **Optimal proposal/Perfect adaptation**. Implement the following alternative update-propagate Bayesian recursion

$$\begin{array}{ll} \text{Update} & p\left(x_{1:t-1} \mid y_{1:t}\right) = \frac{p(y_t|x_{t-1})p(x_{1:t-1}|y_{1:t-1})}{p(y_t|y_{1:t-1})} \\ \text{Propagate} & p\left(x_{1:t} \mid y_{1:t}\right) = p\left(x_{1:t-1} \mid y_{1:t}\right) p\left(x_t \mid y_t, x_{t-1}\right) \end{array}$$

where

$$p\left(x_t \mid y_t, x_{t-1}\right) = \frac{f\left(x_t \mid x_{t-1}\right) g\left(y_t \mid x_{t-1}\right)}{p\left(y_t \mid x_{t-1}\right)}$$

$\rightsquigarrow$ Much more efficient when applicable; e.g.
$f\left(x_t \mid x_{t-1}\right) = \mathcal{N}\left(x_t; \varphi\left(x_{t-1}\right), \Sigma_v\right)$, $g\left(y_t \mid x_t\right) = \mathcal{N}\left(y_t; x_t, \Sigma_w\right)$.

A General Bayesian Recursion

- Introduce an arbitrary proposal distribution $q(x_t | y_t, x_{t-1})$; i.e. an approximation to $p(x_t | y_t, x_{t-1})$.

A General Bayesian Recursion

- Introduce an arbitrary proposal distribution $q(x_t | y_t, x_{t-1})$; i.e. an approximation to $p(x_t | y_t, x_{t-1})$.
- We have seen that

$$p(x_{1:t} | y_{1:t}) = \frac{g(y_t | x_t) \; f(x_t | x_{t-1}) \, p(x_{1:t-1} | y_{1:t-1})}{p(y_t | y_{1:t-1})}$$

so clearly

$$p(x_{1:t} | y_{1:t}) = \frac{w(x_{t-1}, x_t, y_t) \, q(x_t | y_t, x_{t-1}) \, p(x_{1:t-1} | y_{1:t-1})}{p(y_t | y_{1:t-1})}$$

where

$$w(x_{t-1}, x_t, y_t) = \frac{g(y_t | x_t) \; f(x_t | x_{t-1})}{q(x_t | y_t, x_{t-1})}$$

A General Bayesian Recursion

- Introduce an arbitrary proposal distribution $q(x_t|y_t, x_{t-1})$; i.e. an approximation to $p(x_t|y_t, x_{t-1})$.
- We have seen that

$$p(x_{1:t}|y_{1:t}) = \frac{g(y_t|x_t)\; f(x_t|x_{t-1})\, p(x_{1:t-1}|y_{1:t-1})}{p(y_t|y_{1:t-1})}$$

  so clearly

$$p(x_{1:t}|y_{1:t}) = \frac{w(x_{t-1}, x_t, y_t)\, q(x_t|y_t, x_{t-1})\, p(x_{1:t-1}|y_{1:t-1})}{p(y_t|y_{1:t-1})}$$

  where

$$w(x_{t-1}, x_t, y_t) = \frac{g(y_t|x_t)\; f(x_t|x_{t-1})}{q(x_t|y_t, x_{t-1})}$$

- This suggests a more general SMC algorithm.

A General SMC Algorithm

Assume we have N weighted particles $\left\{ W_{t-1}^{(i)}, X_{1:t-1}^{(i)} \right\}$ approximating $p\left(x_{1:t-1} \middle| y_{1:t-1} \right)$ then at time t,

- Sample $\widetilde{X}_t^{(i)} \sim q\left(x_t \middle| y_t, X_{t-1}^{(i)} \right)$, set $\widetilde{X}_{1:t}^{(i)} = \left(X_{1:t-1}^{(i)}, \widetilde{X}_t^{(i)} \right)$ and

$$
\begin{aligned}
\widetilde{p}\left(x_{1:t} \middle| y_{1:t} \right) &= \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t} \right), \\
W_t^{(i)} &\propto W_{t-1}^{(i)} \frac{f\left(\widetilde{X}_t^{(i)} \middle| X_{t-1}^{(i)} \right) g\left(y_t \middle| \widetilde{X}_t^{(i)} \right)}{q\left(\widetilde{X}_t^{(i)} \middle| y_t, X_{t-1}^{(i)} \right)}.
\end{aligned}
$$

A General SMC Algorithm

Assume we have N weighted particles $\left\{W_{t-1}^{(i)}, X_{1:t-1}^{(i)}\right\}$ approximating $p\left(x_{1:t-1} \mid y_{1:t-1}\right)$ then at time t,

- Sample $\widetilde{X}_t^{(i)} \sim q\left(x_t \mid y_t, X_{t-1}^{(i)}\right)$, set $\widetilde{X}_{1:t}^{(i)} = \left(X_{1:t-1}^{(i)}, \widetilde{X}_t^{(i)}\right)$ and

$$\widetilde{p}\left(x_{1:t} \mid y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\widetilde{X}_{1:t}^{(i)}}\left(x_{1:t}\right),$$

$$W_t^{(i)} \propto W_{t-1}^{(i)} \frac{f\left(\widetilde{X}_t^{(i)} \middle| X_{t-1}^{(i)}\right) g\left(y_t \middle| \widetilde{X}_t^{(i)}\right)}{q\left(\widetilde{X}_t^{(i)} \middle| y_t, X_{t-1}^{(i)}\right)}.$$

- If ESS$< N/2$ resample $X_{1:t}^{(i)} \sim \widetilde{p}\left(x_{1:t} \mid y_{1:t}\right)$ and set $W_t^{(i)} \leftarrow \frac{1}{N}$ to obtain $\widehat{p}\left(x_{1:t} \mid y_{1:t}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}}\left(x_{1:t}\right)$.

Building Proposals

- Our aim is to select $q(x_t | y_t, x_{t-1})$ as "close" as possible to $p(x_t | y_t, x_{t-1})$ as this minimizes the variance of

$$w(x_{t-1}, x_t, y_t) = \frac{g(y_t | x_t)\; f(x_t | x_{t-1})}{q(x_t | y_t, x_{t-1})}.$$

Building Proposals

- Our aim is to select $q\left(x_t \mid y_t, x_{t-1}\right)$ as "close" as possible to $p\left(x_t \mid y_t, x_{t-1}\right)$ as this minimizes the variance of
$$w\left(x_{t-1}, x_t, y_t\right) = \frac{g\left(y_t \mid x_t\right)\ f\left(x_t \mid x_{t-1}\right)}{q\left(x_t \mid y_t, x_{t-1}\right)}.$$
- **Example - EKF proposal**: Let
$$X_t = \varphi\left(X_{t-1}\right) + V_t,\ \ Y_t = \Psi\left(X_t\right) + W_t,$$
with $V_t \sim \mathcal{N}(0, \Sigma_v)$, $W_t \sim \mathcal{N}(0, \Sigma_w)$. We perform local linearization
$$Y_t \approx \Psi\left(\varphi\left(X_{t-1}\right)\right) + \left.\frac{\partial \Psi\left(x\right)}{\partial x}\right|_{\varphi(X_{t-1})} \left(X_t - \varphi\left(X_{t-1}\right)\right) + W_t$$
and use as a proposal.
$$q\left(x_t \mid y_t, x_{t-1}\right) \propto \widehat{g}\left(y_t \mid x_t\right)\ f\left(x_t \mid x_{t-1}\right).$$

Building Proposals

- Our aim is to select $q\left(x_t | y_t, x_{t-1}\right)$ as "close" as possible to $p\left(x_t | y_t, x_{t-1}\right)$ as this minimizes the variance of

$$w\left(x_{t-1}, x_t, y_t\right) = \frac{g\left(y_t | x_t\right) \; f\left(x_t | x_{t-1}\right)}{q\left(x_t | y_t, x_{t-1}\right)}.$$

- **Example - EKF proposal**: Let

$$X_t = \varphi\left(X_{t-1}\right) + V_t, \;\; Y_t = \Psi\left(X_t\right) + W_t,$$

with $V_t \sim \mathcal{N}(0, \Sigma_v)$, $W_t \sim \mathcal{N}(0, \Sigma_w)$. We perform local linearization

$$Y_t \approx \Psi\left(\varphi\left(X_{t-1}\right)\right) + \left.\frac{\partial \Psi\left(x\right)}{\partial x}\right|_{\varphi\left(X_{t-1}\right)} \left(X_t - \varphi\left(X_{t-1}\right)\right) + W_t$$

and use as a proposal.

$$q\left(x_t | y_t, x_{t-1}\right) \propto \widehat{g}\left(y_t | x_t\right) \; f\left(x_t | x_{t-1}\right).$$

- Any standard suboptimal filtering methods can be used: Unscented Particle filter, Gaussan Quadrature particle filter etc.

Implicit Proposals

- Proposed recently by Chorin (2012). Let

$$F\left(x_{t-1}, x_t\right) = \log g\left(\left. y_t \right| x_t\right) + \log\ f\left(\left. x_t \right| x_{t-1}\right)$$

and

$$x_t^* = \arg\max F\left(x_{t-1}, x_t\right) = \arg\max\ p\left(\left. x_t \right| y_t, x_{t-1}\right)$$

Implicit Proposals

- Proposed recently by Chorin (2012). Let

$$F(x_{t-1}, x_t) = \log g(y_t | x_t) + \log f(x_t | x_{t-1})$$

and

$$x_t^* = \arg\max F(x_{t-1}, x_t) = \arg\max p(x_t | y_t, x_{t-1})$$

- We sample $Z \sim \mathcal{N}(0, I_{n_x})$, then we solve in X_t

$$F(x_{t-1}, x_t^*) - F(x_{t-1}, X_t) = \frac{1}{2} Z^{\mathsf{T}} Z, \quad Z \sim \mathcal{N}(0, I_{n_x})$$

so if there is a unique solution

$$\begin{aligned} q(x_t | y_t, x_{t-1}) &= p_Z(z) \left|\det \partial z / \partial x_t\right| \\ &\propto \frac{\exp(-F(x_{t-1}, x_t^*))}{\left|\det \partial x_t / \partial z\right|} g(y_t | x_t) \; f(x_t | x_{t-1}) \end{aligned}$$

Implicit Proposals

- Proposed recently by Chorin (2012). Let

$$F(x_{t-1}, x_t) = \log g(y_t | x_t) + \log f(x_t | x_{t-1})$$

and

$$x_t^* = \arg\max F(x_{t-1}, x_t) = \arg\max p(x_t | y_t, x_{t-1})$$

- We sample $Z \sim \mathcal{N}(0, I_{n_x})$, then we solve in X_t

$$F(x_{t-1}, x_t^*) - F(x_{t-1}, X_t) = \frac{1}{2} Z^{\mathsf{T}} Z, \quad Z \sim \mathcal{N}(0, I_{n_x})$$

so if there is a unique solution

$$\begin{aligned} q(x_t | y_t, x_{t-1}) &= p_Z(z) \left|\det \partial z / \partial x_t\right| \\ &\propto \frac{\exp(-F(x_{t-1}, x_t^*))}{\left|\det \partial x_t / \partial z\right|} g(y_t | x_t) \; f(x_t | x_{t-1}) \end{aligned}$$

- The incremental weight is

$$\frac{g(y_t | x_t) \; f(x_t | x_{t-1})}{q(x_t | y_t, x_{t-1})} \propto \left|\det \partial x_t / \partial z\right| \exp(F(x_{t-1}, x_t^*))$$

Auxiliary Particle Filters

- Popular variation introduced by (Pitt & Shephard, 1999).

Auxiliary Particle Filters

- Popular variation introduced by (Pitt & Shephard, 1999).
- This corresponds to a standard SMC algorithm (Johansen & D., 2008) where we target

$$\widehat{p}\left(x_{1:t} \middle| y_{1:t+1}\right) \propto p\left(x_{1:t} \middle| y_{1:t}\right) \widehat{p}\left(y_{t+1} \middle| x_t\right)$$

  where $\widehat{p}\left(y_{t+1} \middle| x_t\right) \approx p\left(y_{t+1} \middle| x_t\right)$ using a proposal $\widehat{p}\left(x_t \middle| y_t, x_{t-1}\right)$.

Auxiliary Particle Filters

- Popular variation introduced by (Pitt & Shephard, 1999).
- This corresponds to a standard SMC algorithm (Johansen & D., 2008) where we target

$$\widehat{p}\left(x_{1:t} \middle| y_{1:t+1}\right) \propto p\left(x_{1:t} \middle| y_{1:t}\right) \widehat{p}\left(y_{t+1} \middle| x_t\right)$$

  where $\widehat{p}\left(y_{t+1} \middle| x_t\right) \approx p\left(y_{t+1} \middle| x_t\right)$ using a proposal $\widehat{p}\left(x_t \middle| y_t, x_{t-1}\right)$.
- When $\widehat{p}\left(y_{t+1} \middle| x_t\right) = p\left(y_{t+1} \middle| x_t\right)$ and $\widehat{p}\left(x_{t+1} \middle| y_{t+1}, x_t\right) = p\left(x_{t+1} \middle| y_{t+1}, x_t\right)$ then we are back to "perfect adaptation".

Block Sampling Proposals

- **Problem**: we only sample X_t at time t so, even if you use $p\left(x_t \mid y_t, x_{t-1} \right)$, the SMC estimates could have high variance if $\mathbb{V}_{p\left(x_{t-1} \mid y_{1:t-1} \right)} \left[p\left(y_t \mid x_{t-1} \right) \right]$ is high.

Block Sampling Proposals

- **Problem**: we only sample X_t at time t so, even if you use $p\left(x_t \mid y_t, x_{t-1}\right)$, the SMC estimates could have high variance if $\mathbb{V}_{p(x_{t-1}|y_{1:t-1})}\left[p\left(y_t \mid x_{t-1}\right)\right]$ is high.
- **Block sampling idea**: allows yourself to sample again $X_{t-L+1:t-1}$ as well as X_t in light of y_t. Optimally we would like at time t to sample

$$X_{t-L+1:t}^{(i)} \sim p\left(x_{t-L+1:t} \middle| y_{t-L+1:t}, X_{t-L}^{(i)}\right)$$

and

$$\begin{aligned} W_t^{(i)} &\propto W_{t-1}^{(i)} \frac{p\left(X_{1:t}^{(i)} \middle| y_{1:t}\right)}{p\left(X_{1:t-L}^{(i)} \middle| y_{1:t-1}\right) p\left(X_{t-L+1:t}^{(i)} \middle| y_{t-L+1:t}, X_{t-L}^{(i)}\right)} \\ &\propto W_{t-1}^{(i)} p\left(y_t \middle| y_{t-L+1:t-1}, X_{t-L}^{(i)}\right) \end{aligned}$$

Block Sampling Proposals

- **Problem**: we only sample X_t at time t so, even if you use $p\left(x_t \middle| y_t, x_{t-1}\right)$, the SMC estimates could have high variance if $\mathbb{V}_{p(x_{t-1}|y_{1:t-1})}\left[p\left(y_t \middle| x_{t-1}\right)\right]$ is high.
- **Block sampling idea**: allows yourself to sample again $X_{t-L+1:t-1}$ as well as X_t in light of y_t. Optimally we would like at time t to sample

$$X_{t-L+1:t}^{(i)} \sim p\left(x_{t-L+1:t} \middle| y_{t-L+1:t}, X_{t-L}^{(i)}\right)$$

and

$$\begin{aligned} W_t^{(i)} &\propto W_{t-1}^{(i)} \frac{p\left(X_{1:t}^{(i)} \middle| y_{1:t}\right)}{p\left(X_{1:t-L}^{(i)} \middle| y_{1:t-1}\right) p\left(X_{t-L+1:t}^{(i)} \middle| y_{t-L+1:t}, X_{t-L}^{(i)}\right)} \\ &\propto W_{t-1}^{(i)} p\left(y_t \middle| y_{t-L+1:t-1}, X_{t-L}^{(i)}\right) \end{aligned}$$

- When $p\left(x_{t-L+1:t} \middle| y_{t-L+1:t}, x_{t-L}\right)$ and $p\left(y_t \middle| y_{t-L+1:t-1}, x_{t-L}\right)$ are not available, we can use analytical approximations of them and still have consistent estimates (D., Briers & Senecal, 2006).

Block Sampling Proposals

- Computational cost is increased from $\mathcal{O}(N)$ to $\mathcal{O}(LN)$ so is it worth it?

Block Sampling Proposals

- Computational cost is increased from $\mathcal{O}(N)$ to $\mathcal{O}(LN)$ so is it worth it?
- Consider the ideal scenario where

$$\begin{aligned} X_t &= X_{t-1} + V_t \\ Y_t &= X_t + W_t \end{aligned}$$

where $X_1 \sim \mathcal{N}(0,1)$ and $V_t, W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$.

Block Sampling Proposals

- Computational cost is increased from $\mathcal{O}(N)$ to $\mathcal{O}(LN)$ so is it worth it?
- Consider the ideal scenario where

$$\begin{aligned} X_t &= X_{t-1} + V_t \\ Y_t &= X_t + W_t \end{aligned}$$

where $X_1 \sim \mathcal{N}(0,1)$ and $V_t, W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$.

- In this case, we have

$$|p(y_t|y_{t-L+1:t-1}, x_{t-L}) - p(y_t|y_{t-L+1:t-1}, x'_{t-L})| < c|x_{t-L} - x'_{t-L}|/2^L$$

where the rate of exponential convergence depends upon the signal-to-noise ratio if more general Gaussian AR are considered.

Block Sampling Proposals

- Computational cost is increased from $\mathcal{O}(N)$ to $\mathcal{O}(LN)$ so is it worth it?
- Consider the ideal scenario where

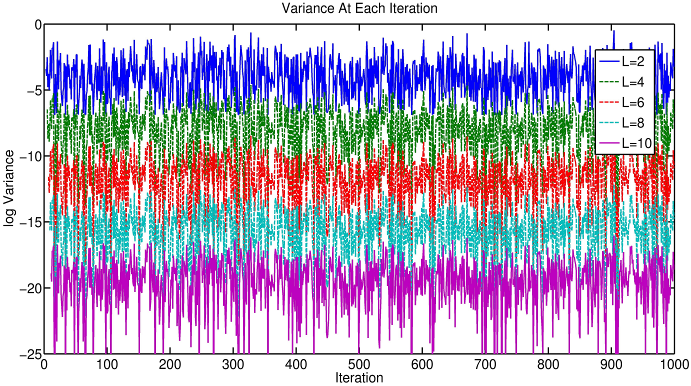
$$\begin{aligned} X_t &= X_{t-1} + V_t \\ Y_t &= X_t + W_t \end{aligned}$$

  where $X_1 \sim \mathcal{N}(0,1)$ and $V_t, W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$.
- In this case, we have

$$|p(y_t|y_{t-L+1:t-1}, x_{t-L}) - p(y_t|y_{t-L+1:t-1}, x'_{t-L})| < c|x_{t-L} - x'_{t-L}|/2^L$$

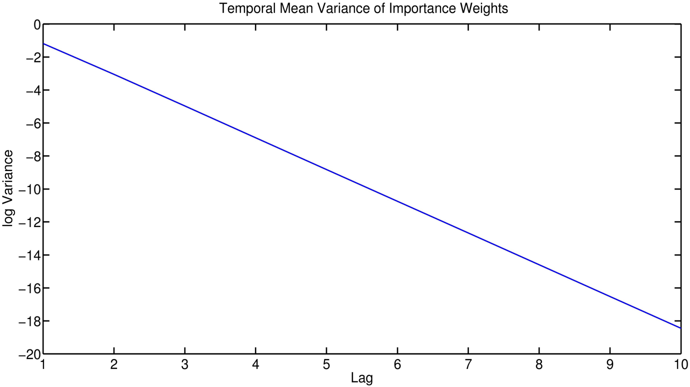
  where the rate of exponential convergence depends upon the signal-to-noise ratio if more general Gaussian AR are considered.
- We can obtain an analytic expression of the variance of the (normalized) weight.

Block Sampling Proposals



Variance of incremental weight w.r.t. $p\left(x_{1:t-L} \middle| y_{1:t-1}\right)$.

Block Sampling Proposals



Time averaged variance of of incremental weight w.r.t. $p\left(x_{1:t-L}\middle| y_{1:t-1}\right)$.

Fighting Degeneracy Using MCMC Steps

- The design of "good" proposals can be complicated and/or time consuming so, after the resampling step, a few particles might inherit many offspring.

Fighting Degeneracy Using MCMC Steps

- The design of "good" proposals can be complicated and/or time consuming so, after the resampling step, a few particles might inherit many offspring.
- A standard way to limit degeneracy is known as the Resample-Move algorithm (Gilks & Berzuini, 2001); i.e. using MCMC kernels as a principled way to "jitter" the particle locations.

Fighting Degeneracy Using MCMC Steps

- The design of "good" proposals can be complicated and/or time consuming so, after the resampling step, a few particles might inherit many offspring.
- A standard way to limit degeneracy is known as the Resample-Move algorithm (Gilks & Berzuini, 2001); i.e. using MCMC kernels as a principled way to "jitter" the particle locations.
- A MCMC kernel $K_t\left(x'_{1:t} \middle| x_{1:t}\right)$ of invariant distribution $p\left(x_{1:t} \middle| y_{1:t}\right)$ is a Markov transition kernel with the property that

$$p\left(x'_{1:t} \middle| y_{1:t}\right) = \int p\left(x_{1:t} \middle| y_{1:t}\right) K_t\left(x'_{1:t} \middle| x_{1:t}\right) dx_{1:t},$$

  i.e. if $X_{1:t} \sim p\left(x_{1:t} \middle| y_{1:t}\right)$ and $X'_{1:t} \middle| X_{1:t} \sim K_t\left(x'_{1:t} \middle| X_{1:t}\right)$ then marginally $X'_{1:t} \sim p\left(x_{1:t} \middle| y_{1:t}\right)$.

Fighting Degeneracy Using MCMC Steps

- *Example 1: Gibbs moves.* Set $X'_{1:t-L} = X_{1:t-L}$ then sample X'_{t-L+1} from $p\left(x_{t-L+1}\middle|\, y_{t-L+1}, x'_{t-L}, x_{t-L+2}\right)$, sample X'_{t-L+2} from $p\left(x_{t-L+2}\middle|\, y_{t-L+2}, x'_{t-L+1}, x_{t-L+3}\right)$ and so on until we sample X'_t from $p\left(x_t\middle|\, y_t, x'_{t-1}\right)$; that is

$$K_t\left(x'_{1:t}\middle|\, x_{1:t}\right) = \delta_{x_{1:t-L}}\left(x'_{1:t-L}\right)\prod_{k=t-L+1}^{t-1} p\left(x'_k\middle|\, y_k, x'_{k-1}, x_{k+1}\right) \times p\left(x'_t\middle|\, y_t, x'_{t-1}\right)$$

Fighting Degeneracy Using MCMC Steps

- *Example 1: Gibbs moves.* Set $X'_{1:t-L} = X_{1:t-L}$ then sample X'_{t-L+1} from $p\left(x_{t-L+1} \middle| y_{t-L+1}, x'_{t-L}, x_{t-L+2}\right)$, sample X'_{t-L+2} from $p\left(x_{t-L+2} \middle| y_{t-L+2}, x'_{t-L+1}, x_{t-L+3}\right)$ and so on until we sample X'_t from $p\left(x_t \middle| y_t, x'_{t-1}\right)$; that is

$$\begin{aligned} K_t\left(x'_{1:t} \middle| x_{1:t}\right) &= \delta_{x_{1:t-L}}\left(x'_{1:t-L}\right) \prod_{k=t-L+1}^{t-1} p\left(x'_k \middle| y_k, x'_{k-1}, x_{k+1}\right) \\ &\quad \times p\left(x'_t \middle| y_t, x'_{t-1}\right) \end{aligned}$$

- *Example 2: Metropolis-Hastings moves.* Set $X'_{1:t-L} = X_{1:t-L}$ then sample X^*_{t-L+1} from $q\left(x'_{t-L+1:t} \middle| x_{t-L}, x_{t-L+1:t}\right)$ and set $X'_{t-L+1} = X^*_{t-L+1}$ with proba.

$$1 \wedge \frac{p\left(x^*_{t-L+1:t} \middle| y_{t-L+1}, x_{t-L}\right)}{p\left(x_{t-L+1:t} \middle| y_{t-L+1}, x_{t-L}\right)} \frac{q\left(x_{t-L+1:t} \middle| x_{t-L}, x^*_{t-L+1:t}\right)}{q\left(x^*_{t-L+1:t} \middle| x_{t-L}, x_{t-L+1:t}\right)},$$

otherwise set $X'_{t-L+1} = X_{t-L+1}$.

Fighting Degeneracy Using MCMC Steps

- *Example 1: Gibbs moves.* Set $X'_{1:t-L} = X_{1:t-L}$ then sample X'_{t-L+1} from $p\left(x_{t-L+1} \middle| y_{t-L+1}, x'_{t-L}, x_{t-L+2}\right)$, sample X'_{t-L+2} from $p\left(x_{t-L+2} \middle| y_{t-L+2}, x'_{t-L+1}, x_{t-L+3}\right)$ and so on until we sample X'_t from $p\left(x_t \middle| y_t, x'_{t-1}\right)$; that is

$$K_t\left(x'_{1:t} \middle| x_{1:t}\right) = \delta_{x_{1:t-L}}\left(x'_{1:t-L}\right) \prod_{k=t-L+1}^{t-1} p\left(x'_k \middle| y_k, x'_{k-1}, x_{k+1}\right) \times p\left(x'_t \middle| y_t, x'_{t-1}\right)$$

- *Example 2: Metropolis-Hastings moves.* Set $X'_{1:t-L} = X_{1:t-L}$ then sample X^*_{t-L+1} from $q\left(x'_{t-L+1:t} \middle| x_{t-L}, x_{t-L+1:t}\right)$ and set $X'_{t-L+1} = X^*_{t-L+1}$ with proba.

$$1 \wedge \frac{p\left(x^*_{t-L+1:t} \middle| y_{t-L+1}, x_{t-L}\right)}{p\left(x_{t-L+1:t} \middle| y_{t-L+1}, x_{t-L}\right)} \frac{q\left(x_{t-L+1:t} \middle| x_{t-L}, x^*_{t-L+1:t}\right)}{q\left(x^*_{t-L+1:t} \middle| x_{t-L}, x_{t-L+1:t}\right)},$$

otherwise set $X'_{t-L+1} = X_{t-L+1}$.
- Contrary to MCMC, we typically do not use ergodic kernels in SMC.

Example: Bearings-only-tracking

- Target modelled using a standard constant velocity model

$$X_t = AX_{t-1} + V_t$$

where $V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$. The state vector $X_t = \begin{pmatrix} X_t^1 & X_t^2 & X_t^3 & X_t^4 \end{pmatrix}^\mathsf{T}$ contains location and velocity components.

Example: Bearings-only-tracking

- Target modelled using a standard constant velocity model

$$X_t = AX_{t-1} + V_t$$

where $V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$. The state vector $X_t = \begin{pmatrix} X_t^1 & X_t^2 & X_t^3 & X_t^4 \end{pmatrix}^{\mathsf{T}}$ contains location and velocity components.

- One only receives observations of the bearings of the target

$$Y_t = \tan^{-1}\left(\frac{X_t^3}{X_t^1}\right) + W_t$$

where $W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, 10^{-4}\right)$; i.e. the observations are almost noiseless.

Example: Bearings-only-tracking

- Target modelled using a standard constant velocity model

$$X_t = AX_{t-1} + V_t$$

where $V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$. The state vector $X_t = \begin{pmatrix} X_t^1 & X_t^2 & X_t^3 & X_t^4 \end{pmatrix}^{\mathsf{T}}$ contains location and velocity components.

- One only receives observations of the bearings of the target

$$Y_t = \tan^{-1}\left(\frac{X_t^3}{X_t^1}\right) + W_t$$

where $W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, 10^{-4}\right)$; i.e. the observations are almost noiseless.

- We compare Bootstrap filter, SMC-EKF with $L = 5, 10$, MCMC moves $L = 5, 10$ using dynamic resampling.

Degeneracy for Various Proposals

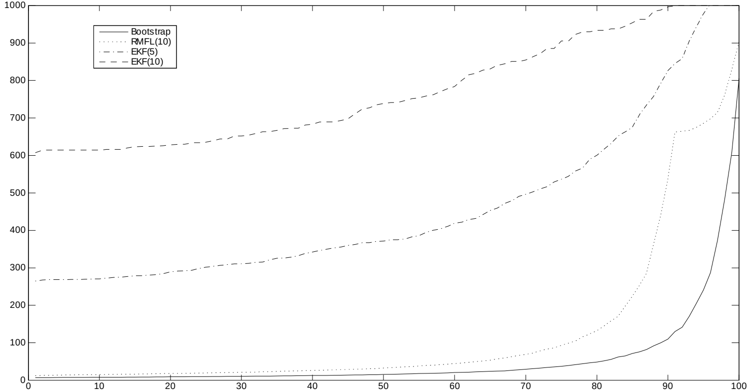


Figure: Average number of unique particles $X_t^{(i)}$ approximating $p\left(x_t \middle| y_{1:100} \right)$; time on x-axis, average number of unique particles on y-axis.

Summary

- SMC provide consistent estimates under weak assumptions.

Summary

- SMC provide consistent estimates under weak assumptions.
- We can estimate $\{p(x_t | y_{1:t})\}_{t \geq 1}$ satisfactorily but our approximations of $\{p(x_{1:t} | y_{1:t})\}_{t \geq 1}$ degenerates as t increases because of resampling steps.

Summary

- SMC provide consistent estimates under weak assumptions.
- We can estimate $\{p(x_t | y_{1:t})\}_{t \geq 1}$ satisfactorily but our approximations of $\{p(x_{1:t} | y_{1:t})\}_{t \geq 1}$ degenerates as t increases because of resampling steps.
- Resampling is crucial.

Summary

- SMC provide consistent estimates under weak assumptions.
- We can estimate $\{p(x_t | y_{1:t})\}_{t\geq 1}$ satisfactorily but our approximations of $\{p(x_{1:t} | y_{1:t})\}_{t\geq 1}$ degenerates as t increases because of resampling steps.
- Resampling is crucial.
- We can mitigate but not eliminate the degeneracy problem by the design of "clever" proposals.

Summary

- SMC provide consistent estimates under weak assumptions.
- We can estimate $\{p(x_t | y_{1:t})\}_{t\geq 1}$ satisfactorily but our approximations of $\{p(x_{1:t} | y_{1:t})\}_{t\geq 1}$ degenerates as t increases because of resampling steps.
- Resampling is crucial.
- We can mitigate but not eliminate the degeneracy problem by the design of "clever" proposals.
- Smoothing methods to estimate $p(x_{1:T} | y_{1:T})$ can come to the rescue.

Smoothing in State-Space Models

- **Smoothing problem**: given a fixed time T, we are interested in $p\left(x_{1:T} \middle| y_{1:T}\right)$ or some of its marginals, e.g. $\left\{p\left(x_t \middle| y_{1:T}\right)\right\}_{t=1}^{T}$.

Smoothing in State-Space Models

- **Smoothing problem**: given a fixed time T, we are interested in $p\left(x_{1:T} \middle| y_{1:T} \right)$ or some of its marginals, e.g. $\left\{ p\left(x_t \middle| y_{1:T} \right) \right\}_{t=1}^{T}$.
- Smoothing is crucial to parameter estimation.

Smoothing in State-Space Models

- **Smoothing problem**: given a fixed time T, we are interested in $p\left(x_{1:T} \mid y_{1:T}\right)$ or some of its marginals, e.g. $\left\{p\left(x_t \mid y_{1:T}\right)\right\}_{t=1}^{T}$.
- Smoothing is crucial to parameter estimation.
- Direct SMC approximations of $p\left(x_{1:T} \mid y_{1:T}\right)$ and its marginals $p\left(x_k \mid y_{1:T}\right)$ are poor if T is large.

Smoothing in State-Space Models

- **Smoothing problem**: given a fixed time T, we are interested in $p\left(x_{1:T} \middle| y_{1:T} \right)$ or some of its marginals, e.g. $\left\{ p\left(x_t \middle| y_{1:T} \right) \right\}_{t=1}^{T}$.
- Smoothing is crucial to parameter estimation.
- Direct SMC approximations of $p\left(x_{1:T} \middle| y_{1:T} \right)$ and its marginals $p\left(x_k \middle| y_{1:T} \right)$ are poor if T is large.
- SMC provide "good" approximations of marginals $\left\{ p\left(x_t \middle| y_{1:t} \right) \right\}_{t \geq 1}$. This can be used to develop efficient smoothing estimates.
  - $\rightsquigarrow$ Fixed-lag smoothing
  - $\rightsquigarrow$ Forward-backward smoothing
  - $\rightsquigarrow$ (Generalized) two-filter smoothing

Fixed-Lag Smoothing

- The fixed-lag smoothing approximation relies on

$$p\left(x_t \mid y_{1:T}\right) \approx p\left(x_t \mid y_{1:t+\Delta}\right) \text{ for } \Delta \text{ large enough.}$$

  and quantitative bounds can be established under stability assumptions.

Fixed-Lag Smoothing

- The fixed-lag smoothing approximation relies on

$$p\left(x_t \mid y_{1:T}\right) \approx p\left(x_t \mid y_{1:t+\Delta}\right) \text{ for } \Delta \text{ large enough.}$$

  and quantitative bounds can be established under stability assumptions.
- This can be exploited by SMC methods (Kitagawa & Sato, 2001)

Fixed-Lag Smoothing

- The fixed-lag smoothing approximation relies on

$$p\left(x_t \mid y_{1:T}\right) \approx p\left(x_t \mid y_{1:t+\Delta}\right) \text{ for } \Delta \text{ large enough.}$$

  and quantitative bounds can be established under stability assumptions.
- This can be exploited by SMC methods (Kitagawa & Sato, 2001)
- **Algorithmically**: stop resampling $\left\{X_t^{(i)}\right\}$ beyond time $t+\Delta$ (Kitagawa & Sato, 2001).

Fixed-Lag Smoothing

- The fixed-lag smoothing approximation relies on

$$p(x_t | y_{1:T}) \approx p(x_t | y_{1:t+\Delta}) \text{ for } \Delta \text{ large enough.}$$

  and quantitative bounds can be established under stability assumptions.
- This can be exploited by SMC methods (Kitagawa & Sato, 2001)
- **Algorithmically**: stop resampling $\left\{X_t^{(i)}\right\}$ beyond time $t+\Delta$ (Kitagawa & Sato, 2001).
- Computational cost is $\mathcal{O}(N)$ but non-vanishing bias as $N \to \infty$ (Olsson & al., 2008).

Fixed-Lag Smoothing

- The fixed-lag smoothing approximation relies on

$$p(x_t | y_{1:T}) \approx p(x_t | y_{1:t+\Delta}) \text{ for } \Delta \text{ large enough.}$$

  and quantitative bounds can be established under stability assumptions.
- This can be exploited by SMC methods (Kitagawa & Sato, 2001)
- **Algorithmically**: stop resampling $\left\{ X_t^{(i)} \right\}$ beyond time $t + \Delta$ (Kitagawa & Sato, 2001).
- Computational cost is $\mathcal{O}(N)$ but non-vanishing bias as $N \rightarrow \infty$ (Olsson & al., 2008).
- Picking Δ is difficult: Δ too small results in $p(x_t | y_{1:t+\Delta})$ being a poor approximation of $p(x_t | y_{1:T})$. Δ too large improves the approximation but degeneracy creeps in.

Forward Backward Smoothing

- Forward Backward (FB) decomposition states

$$\begin{aligned} p(x_{1:T} | y_{1:T}) &= p(x_T | y_{1:T}) \prod_{t=1}^{T-1} p(x_t | y_{1:T}, x_{t+1:T}) \\ &= p(x_T | y_{1:T}) \prod_{t=1}^{T-1} p(x_t | y_{1:t}, x_{t+1}) \end{aligned}$$

where

$$p(x_t | y_{1:t}, x_{t+1}) = \frac{f(x_{t+1} | x_t) \, p(x_t | y_{1:t})}{p(x_{t+1} | y_{1:t})}.$$

Forward Backward Smoothing

- Forward Backward (FB) decomposition states

$$\begin{aligned} p(x_{1:T}|\, y_{1:T}) &= p(x_T|\, y_{1:T}) \prod_{t=1}^{T-1} p(x_t|\, y_{1:T}, x_{t+1:T}) \\ &= p(x_T|\, y_{1:T}) \prod_{t=1}^{T-1} p(x_t|\, y_{1:t}, x_{t+1}) \end{aligned}$$

where

$$p(x_t|\, y_{1:t}, x_{t+1}) = \frac{f(x_{t+1}|\, x_t)\, p(x_t|\, y_{1:t})}{p(x_{t+1}|\, y_{1:t})}.$$

- Conditioned upon $y_{1:T}$, $\{X_t\}_{t=1}^{T}$ is a backward Markov chain of initial distribution $p(x_T|\, y_{1:T})$ and inhomogeneous Markov transitions $\{p(x_t|\, y_{1:t}, x_{t+1})\}_{t=1}^{T-1}$.

Forward Filtering Backward Sampling

- To obtain a sample from $p(x_{1:T} | y_{1:T})$,

Forward Filtering Backward Sampling

- To obtain a sample from $p\left(x_{1:T}\middle|\,y_{1:T}\right)$,
  - **Forward filtering**: compute and store $\left\{p\left(x_t\middle|\,y_{1:t}\right)\right\}_{t=1}^{T}$

Forward Filtering Backward Sampling

- To obtain a sample from $p(x_{1:T} | y_{1:T})$,
  - **Forward filtering**: compute and store $\{p(x_t | y_{1:t})\}_{t=1}^{T}$
  - **Backward sampling**: sample $X_T \sim p(x_T | y_{1:T})$ then for $t = T-1, ..., 1$ sample $X_t \sim p(x_t | y_{1:t}, X_{t+1})$.

Forward Filtering Backward Sampling

- To obtain a sample from $p(x_{1:T} | y_{1:T})$,
  - **Forward filtering**: compute and store $\{p(x_t | y_{1:t})\}_{t=1}^{T}$
  - **Backward sampling**: sample $X_T \sim p(x_T | y_{1:T})$ then for $t = T-1, ..., 1$ sample $X_t \sim p(x_t | y_{1:t}, X_{t+1})$.
- SMC to obtain an approximate sample from $p(x_{1:T} | y_{1:T})$

Forward Filtering Backward Sampling

- To obtain a sample from $p\left(\left.x_{1:T}\right| y_{1:T}\right)$,
  - **Forward filtering**: compute and store $\left\{p\left(\left.x_t\right| y_{1:t}\right)\right\}_{t=1}^{T}$
  - **Backward sampling**: sample $X_T \sim p\left(\left.x_T\right| y_{1:T}\right)$ then for $t = T-1, ..., 1$ sample $X_t \sim p\left(\left.x_t\right| y_{1:t}, X_{t+1}\right)$.
- SMC to obtain an approximate sample from $p\left(\left.x_{1:T}\right| y_{1:T}\right)$
  - **Forward filtering**: compute and store $\left\{\widehat{p}\left(\left.x_t\right| y_{1:t}\right)\right\}_{t=1}^{T}$.

Forward Filtering Backward Sampling

- To obtain a sample from $p\left(x_{1:T} \mid y_{1:T}\right)$,
  - **Forward filtering**: compute and store $\left\{p\left(x_t \mid y_{1:t}\right)\right\}_{t=1}^{T}$
  - **Backward sampling**: sample $X_T \sim p\left(x_T \mid y_{1:T}\right)$ then for $t = T-1, ..., 1$ sample $X_t \sim p\left(x_t \mid y_{1:t}, X_{t+1}\right)$.
- SMC to obtain an approximate sample from $p\left(x_{1:T} \mid y_{1:T}\right)$
  - **Forward filtering**: compute and store $\left\{\widehat{p}\left(x_t \mid y_{1:t}\right)\right\}_{t=1}^{T}$.
  - **Backward sampling**: sample $X_T \sim \widehat{p}\left(x_T \mid y_{1:T}\right)$ then for $t = T-1, ..., 1$ sample $X_t \sim \widehat{p}\left(x_t \mid y_{1:t}, X_{t+1}\right)$ where

$$\begin{aligned}\widehat{p}\left(x_t \mid y_{1:t}, X_{t+1}\right) &\propto f\left(X_{t+1} \mid x_t\right) \widehat{p}\left(x_t \mid y_{1:t}\right) \\ &\propto \sum_{i=1}^{N} f\left(X_{t+1} \mid X_t^{(i)}\right) \delta_{X_t^{(i)}}\left(x_t\right)\end{aligned}$$

Forward Filtering Backward Sampling

- To obtain a sample from $p\left(x_{1:T} \mid y_{1:T}\right)$,
  - **Forward filtering**: compute and store $\left\{p\left(x_t \mid y_{1:t}\right)\right\}_{t=1}^{T}$
  - **Backward sampling**: sample $X_T \sim p\left(x_T \mid y_{1:T}\right)$ then for $t = T-1, ..., 1$ sample $X_t \sim p\left(x_t \mid y_{1:t}, X_{t+1}\right)$.
- SMC to obtain an approximate sample from $p\left(x_{1:T} \mid y_{1:T}\right)$
  - **Forward filtering**: compute and store $\left\{\widehat{p}\left(x_t \mid y_{1:t}\right)\right\}_{t=1}^{T}$.
  - **Backward sampling**: sample $X_T \sim \widehat{p}\left(x_T \mid y_{1:T}\right)$ then for $t = T-1, ..., 1$ sample $X_t \sim \widehat{p}\left(x_t \mid y_{1:t}, X_{t+1}\right)$ where

$$
\begin{aligned}
\widehat{p}\left(x_t \mid y_{1:t}, X_{t+1}\right) &\propto f\left(X_{t+1} \mid x_t\right) \widehat{p}\left(x_t \mid y_{1:t}\right) \\
&\propto \sum_{i=1}^{N} f\left(X_{t+1} \mid X_t^{(i)}\right) \delta_{X_t^{(i)}}\left(x_t\right)
\end{aligned}
$$

- Direct implementation $\mathcal{O}(NT)$ (Godsill, D. & West, 2004). Rejection sampling possible if $f\left(x_{t+1} \mid x_t\right) \leq C\left(x_{t+1}\right)$ (Douc et al., 2011) and cost $\mathcal{O}(NT)$.

Forward Filtering Backward Smoothing

- Assume you want to compute the marginal smoothing distributions $\{p(x_t | y_{1:T})\}_{t=1}^{T}$ instead of sampling from them.

Forward Filtering Backward Smoothing

- Assume you want to compute the marginal smoothing distributions $\{p(x_t | y_{1:T})\}_{t=1}^{T}$ instead of sampling from them.
- **Forward filtering Backward smoothing** (FFBS).

$$
\begin{aligned}
\overbrace{p(x_t | y_{1:T})}^{\text{smoother at } t} &= \int p(x_t, x_{t+1} | y_{1:T}) \, dx_{t+1} \\
&= \int p(x_{t+1} | y_{1:T}) \, p(x_t | y_{1:t}, x_{t+1}) \, dx_{t+1} \\
&= \int \overbrace{p(x_{t+1} | y_{1:T})}^{\text{smoother at } t+1} \underbrace{\frac{f(x_{t+1} | x_t) \overbrace{p(x_t | y_{1:t})}^{\text{filter at } t}}{p(x_{t+1} | y_{1:t})}}_{\text{backward transition } p(x_t | y_{1:t}, x_{t+1})} \, dx_{t+1}.
\end{aligned}
$$

Forward Filtering Backward Smoothing

- Assume you want to compute the marginal smoothing distributions $\{p(x_t | y_{1:T})\}_{t=1}^{T}$ instead of sampling from them.
- **Forward filtering Backward smoothing** (FFBS).

$$
\begin{aligned}
\overbrace{p(x_t | y_{1:T})}^{\text{smoother at } t} &= \int p(x_t, x_{t+1} | y_{1:T})\, dx_{t+1} \\
&= \int p(x_{t+1} | y_{1:T})\, p(x_t | y_{1:t}, x_{t+1})\, dx_{t+1} \\
&= \int \overbrace{p(x_{t+1} | y_{1:T})}^{\text{smoother at } t+1} \underbrace{\frac{f(x_{t+1} | x_t)\, \overbrace{p(x_t | y_{1:t})}^{\text{filter at } t}}{p(x_{t+1} | y_{1:t})}}_{\text{backward transition } p(x_t | y_{1:t}, x_{t+1})} dx_{t+1}.
\end{aligned}
$$

- For finite state-space HMM, it is surprisingly and unfortunately not the recursion usually implemented (Rabiner et al., 1989).

SMC Forward Filtering Backward Smoothing

- **Forward filtering**: compute and store $\left\{ \widehat{p}\left(x_t \mid y_{1:t} \right) \right\}_{t=1}^{T}$ using your favourite SMC.

SMC Forward Filtering Backward Smoothing

- **Forward filtering**: compute and store $\{\widehat{p}(x_t | y_{1:t})\}_{t=1}^{T}$ using your favourite SMC.
- **Backward smoothing**: For $t = T-1, ..., 1$, we have $\widehat{p}(x_t | y_{1:T}) = \sum_{i=1}^{N} W_{t|T}^{(i)} \delta_{X_t^{(i)}}(x_t)$ with $W_{T|T}^{(i)} = 1/N$ and

$$\widehat{p}(x_t | y_{1:T}) = \underbrace{\widehat{p}(x_t | y_{1:t})}_{\frac{1}{N}\sum_{i=1}^{N}\delta_{X_t^{(i)}}(x_t)} \int \underbrace{\widehat{p}(x_{t+1} | y_{1:T})}_{\sum_{j=1}^{N} W_{t+1|T}^{(j)} \delta_{X_{t+1}^{(j)}}(x_{t+1})} \frac{f(x_{t+1}|x_t)}{\int f(x_{t+1}|x_t)\widehat{p}(x_t|y_{1:t})dx_t} dx_{t+1}$$

$$= \sum_{i=1}^{N} W_{t|T}^{(i)} \delta_{X_t^{(i)}}(x_t)$$

where

$$W_{t|T}^{(i)} = \sum_{j=1}^{N} W_{t+1|T}^{(j)} \frac{f\left(X_{t+1}^{(j)}|X_t^{(i)}\right)}{\sum_{l=1}^{N} f\left(X_{t+1}^{(j)}|X_t^{(l)}\right)}.$$

SMC Forward Filtering Backward Smoothing

- **Forward filtering**: compute and store $\{\widehat{p}(x_t | y_{1:t})\}_{t=1}^{T}$ using your favourite SMC.
- **Backward smoothing**: For $t = T-1, ..., 1$, we have $\widehat{p}(x_t | y_{1:T}) = \sum_{i=1}^{N} W_{t|T}^{(i)} \delta_{X_t^{(i)}}(x_t)$ with $W_{T|T}^{(i)} = 1/N$ and

$$\widehat{p}(x_t | y_{1:T}) = \underbrace{\widehat{p}(x_t | y_{1:t})}_{\frac{1}{N}\sum_{i=1}^{N}\delta_{X_t^{(i)}}(x_t)} \int \underbrace{\widehat{p}(x_{t+1} | y_{1:T})}_{\sum_{j=1}^{N} W_{t+1|T}^{(j)} \delta_{X_{t+1}^{(j)}}(x_{t+1})} \frac{f(x_{t+1}|x_t)}{\int f(x_{t+1}|x_t)\widehat{p}(x_t|y_{1:t})dx_t} dx_{t+1}$$

$$= \sum_{i=1}^{N} W_{t|T}^{(i)} \delta_{X_t^{(i)}}(x_t)$$

where

$$W_{t|T}^{(i)} = \sum_{j=1}^{N} W_{t+1|T}^{(j)} \frac{f\left(X_{t+1}^{(j)} | X_t^{(i)}\right)}{\sum_{l=1}^{N} f\left(X_{t+1}^{(j)} | X_t^{(l)}\right)}.$$

- Computational complexity is $\mathcal{O}(TN^2)$.

Two-Filter Smoothing

- An alternative to FB smoothing is the Two-Filter (TF) formula

$$p\left(x_t, x_{t+1} \middle| y_{1:T}\right) \propto \overbrace{p\left(x_t \middle| y_{1:t}\right)}^{\text{forward filter}} f\left(x_{t+1} \middle| x_t\right) \overbrace{p\left(y_{t+1:T} \middle| x_{t+1}\right)}^{\text{backward filter}}$$

Two-Filter Smoothing

- An alternative to FB smoothing is the Two-Filter (TF) formula

$$p(x_t, x_{t+1} | y_{1:T}) \propto \overbrace{p(x_t | y_{1:t})}^{\text{forward filter}} f(x_{t+1} | x_t) \overbrace{p(y_{t+1:T} | x_{t+1})}^{\text{backward filter}}$$

- The backward information filter satisfies $p(y_T | x_T) = g(y_T | x_T)$ and

$$\begin{aligned} p(y_{t:T} | x_t) &= \int p(y_t, y_{t+1:T}, x_{t+1} | x_t)\, dx_{t+1} \\ &= g(y_t | x_t) \int p(y_{t+1:T} | x_{t+1})\, f(x_{t+1} | x_t)\, dx_{t+1} \end{aligned}$$

Two-Filter Smoothing

- An alternative to FB smoothing is the Two-Filter (TF) formula

$$p\left(x_t, x_{t+1} \middle| y_{1:T}\right) \propto \overbrace{p\left(x_t \middle| y_{1:t}\right)}^{\text{forward filter}} f\left(x_{t+1} \middle| x_t\right) \overbrace{p\left(y_{t+1:T} \middle| x_{t+1}\right)}^{\text{backward filter}}$$

- The backward information filter satisfies $p\left(y_T \middle| x_T\right) = g\left(y_T \middle| x_T\right)$ and

$$\begin{aligned} p\left(y_{t:T} \middle| x_t\right) &= \int p\left(y_t, y_{t+1:T}, x_{t+1} \middle| x_t\right) dx_{t+1} \\ &= g\left(y_t \middle| x_t\right) \int p\left(y_{t+1:T} \middle| x_{t+1}\right) f\left(x_{t+1} \middle| x_t\right) dx_{t+1} \end{aligned}$$

- Various particle methods have been proposed to approximate $\left\{p\left(y_{t:T} \middle| x_t\right)\right\}_{t=1}^{T}$ but rely implicitly on $\int p\left(y_{t:T} \middle| x_t\right) dx_t < \infty$ and try to come up with a backward dynamics; e.g. solve

$$X_{t+1} = \varphi\left(X_t, V_{t+1}\right) \Leftrightarrow X_t = \varphi^{-1}\left(X_t, V_{t+1}\right).$$

This is incorrect.

Generalized Two-Filter Smoothing

- **Generalized Two-Filter smoothing** (Briers, D. & Maskell, 2004-2010)

$$p\left(x_t, x_{t+1} \middle| y_{1:T}\right) \propto \frac{\overbrace{p\left(x_t \middle| y_{1:t}\right)}^{\text{forward filter}} f\left(x_{t+1} \middle| x_t\right) \overbrace{\overline{p}\left(x_{t+1} \middle| y_{t+1:T}\right)}^{\text{backward filter}}}{\underbrace{\overline{p}\left(x_{t+1}\right)}_{\text{artificial prior}}}$$

where

$$\overline{p}\left(x_{t+1} \middle| y_{t+1:T}\right) \propto p\left(y_{t+1:T} \middle| x_{t+1}\right) \overline{p}\left(x_{t+1}\right).$$

- **Generalized Two-Filter smoothing** (Briers, D. & Maskell, 2004-2010)

$$p(x_t, x_{t+1} | y_{1:T}) \propto \frac{\overbrace{p(x_t | y_{1:t})}^{\text{forward filter}} f(x_{t+1} | x_t) \overbrace{\overline{p}(x_{t+1} | y_{t+1:T})}^{\text{backward filter}}}{\underbrace{\overline{p}(x_{t+1})}_{\text{artificial prior}}}$$

  where

$$\overline{p}(x_{t+1} | y_{t+1:T}) \propto p(y_{t+1:T} | x_{t+1}) \overline{p}(x_{t+1}).$$

- By construction, we now have integrable $\overline{p}(x_{t+1} | y_{t+1:T})$ which we can approximate using a backward SMC algorithm targeting $\{\overline{p}(x_{t+1:T} | y_{t+1:T})\}_{t=T}^{1}$ where

$$\overline{p}(x_t | y_{t:T}) \propto \overline{p}(x_t) \prod_{k=t+1}^{T} f(x_k | x_{k-1}) \prod_{k=t}^{T} g(y_k | x_k).$$

SMC Generalized Two-Filter Smoothing

- **Forward filter**: compute and store $\{\widehat{p}(x_t | y_{1:t})\}_{t=1}^{T}$ using your favourite SMC.

SMC Generalized Two-Filter Smoothing

- **Forward filter**: compute and store $\{\widehat{p}(x_t | y_{1:t})\}_{t=1}^{T}$ using your favourite SMC.
- **Backward filter**: compute and store $\{\widehat{\widetilde{p}}(x_t | y_{t:T})\}_{t=1}^{T}$ using your favourite SMC.

SMC Generalized Two-Filter Smoothing

- **Forward filter**: compute and store $\left\{\widehat{p}\left(\left.x_t\right| y_{1:t}\right)\right\}_{t=1}^{T}$ using your favourite SMC.
- **Backward filter**: compute and store $\left\{\widehat{\overline{p}}\left(\left.x_t\right| y_{t:T}\right)\right\}_{t=1}^{T}$ using your favourite SMC.
- **Combination step**: for any $t \in \{1, ..., T\}$ we have

$$\begin{aligned}
\widehat{p}\left(\left.x_t, x_{t+1}\right| y_{1:T}\right) &\propto \widehat{p}\left(\left.x_t\right| y_{1:T}\right) \frac{f\left(\left.x_{t+1}\right| x_t\right)}{\overline{p}\left(x_{t+1}\right)} \widehat{\overline{p}}\left(\left.x_{t+1}\right| y_{t+1:t}\right) \\
&\propto \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{f\left(\left.\overline{X}_{t+1}^{(j)}\right| X_t^{(i)}\right)}{\overline{p}\left(\overline{X}_{t+1}^{(j)}\right)} \delta_{X_t^{(i)}, \overline{X}_{t+1}^{(j)}}\left(x_t, x_{t+1}\right).
\end{aligned}$$

SMC Generalized Two-Filter Smoothing

- **Forward filter**: compute and store $\{\widehat{p}(x_t | y_{1:t})\}_{t=1}^{T}$ using your favourite SMC.
- **Backward filter**: compute and store $\{\widehat{\overline{p}}(x_t | y_{t:T})\}_{t=1}^{T}$ using your favourite SMC.
- **Combination step**: for any $t \in \{1, ..., T\}$ we have

$$
\begin{aligned}
\widehat{p}(x_t, x_{t+1} | y_{1:T}) &\propto \widehat{p}(x_t | y_{1:T}) \frac{f(x_{t+1} | x_t)}{\overline{p}(x_{t+1})} \widehat{\overline{p}}(x_{t+1} | y_{t+1:t}) \\
&\propto \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{f\left(\overline{X}_{t+1}^{(j)} \middle| X_t^{(i)}\right)}{\overline{p}\left(\overline{X}_{t+1}^{(j)}\right)} \delta_{X_t^{(i)}, \overline{X}_{t+1}^{(j)}}(x_t, x_{t+1}).
\end{aligned}
$$

- Cost $\mathcal{O}(N^2 T)$ but $\mathcal{O}(NT)$ through importance sampling (Briers, D. & Singh, 2005; Fearnhead, Wyncoll & Tawn, 2010) and fast computational methods (Klaas et al., 2005).

Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_t \middle| y_{2:t}, X_1 = x_1 \right) - p\left(x_t \middle| y_{2:t}, X_1 = x_1' \right) \right| dx_t \leq \alpha^t \text{ for } |\alpha| < 1.$$

Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_t \middle| y_{2:t}, X_1 = x_1 \right) - p\left(x_t \middle| y_{2:t}, X_1 = x_1' \right) \right| dx_t \leq \alpha^t \text{ for } |\alpha| < 1.$$

- Here $\widehat{\varphi}_T$ denotes SMC estimates obtained using direct, fixed-lag FB or TF method.

Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_t \middle| y_{2:t}, X_1 = x_1 \right) - p\left(x_t \middle| y_{2:t}, X_1 = x_1' \right) \right| dx_t \leq \alpha^t \text{ for } |\alpha| < 1.$$

- Here $\widehat{\varphi}_T$ denotes SMC estimates obtained using direct, fixed-lag FB or TF method.
- **Marginal distribution**. If $\varphi_T(x_{1:T}) = \varphi(x_t)$, we have for the standard path-based SMC estimate

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_T - \overline{\varphi}_T\right) \Rightarrow \mathcal{N}\left(0, \sigma_T^2\right), \underline{A}\left(T - t + 1\right) \leq \sigma_T^2 \leq \overline{A}\left(T - t + 1\right)$$

whereas for FB and TF estimates there exists B independent of T s.t.

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_T - \overline{\varphi}_T\right) \Rightarrow \mathcal{N}\left(0, \sigma_T^2\right) \text{ where } \sigma_T^2 \leq B$$

Comparison Direct Method vs FB and TF

- Assume the model is stable and we are interested in approximating $\overline{\varphi}_T = \int \varphi(x_t)\, p(x_t | y_{1:T})\, dx_t$ using SMC.

Method	Fixed-lag	Direct SMC	FB/TF
# particles	N	N	N
cost	$\mathcal{O}(TN)$	$\mathcal{O}(TN)$	$\mathcal{O}(TN^2)$,$\mathcal{O}(TN)$
Variance	$\mathcal{O}(1/N)$	$\mathcal{O}((T-t+1)/N)$	$\mathcal{O}(1/N)$
Bias	δ	$\mathcal{O}(1/N)$	$\mathcal{O}(1/N)$
MSE=Bias^2+Var	$\delta^2 + \mathcal{O}(1/N)$	$\mathcal{O}((T-t+1)/N)$	$\mathcal{O}(1/N)$

Comparison Direct Method vs FB and TF

- Assume the model is stable and we are interested in approximating $\overline{\varphi}_T = \int \varphi(x_t) \, p(x_t | y_{1:T}) \, dx_t$ using SMC.

Method	Fixed-lag	Direct SMC	FB/TF
# particles	N	N	N
cost	$\mathcal{O}(TN)$	$\mathcal{O}(TN)$	$\mathcal{O}(TN^2)$,$\mathcal{O}(TN)$
Variance	$\mathcal{O}(1/N)$	$\mathcal{O}((T-t+1)/N)$	$\mathcal{O}(1/N)$
Bias	δ	$\mathcal{O}(1/N)$	$\mathcal{O}(1/N)$
MSE=Bias^2+Var	$\delta^2 + \mathcal{O}(1/N)$	$\mathcal{O}((T-t+1)/N)$	$\mathcal{O}(1/N)$

- FB/TF provide uniformly "good" approximations of $\{p(x_t | y_{1:T})\}_{t=1}^{T}$ whereas direct method provide only "good" approximation for $|T-t|$ "small".

Comparison Direct Method vs FB and TF

- Assume the model is stable and we are interested in approximating $\overline{\varphi}_T = \int \varphi(x_t)\, p(x_t | y_{1:T})\, dx_t$ using SMC.

Method	Fixed-lag	Direct SMC	FB/TF
# particles	N	N	N
cost	$\mathcal{O}(TN)$	$\mathcal{O}(TN)$	$\mathcal{O}(TN^2)$,$\mathcal{O}(TN)$
Variance	$\mathcal{O}(1/N)$	$\mathcal{O}((T-t+1)/N)$	$\mathcal{O}(1/N)$
Bias	δ	$\mathcal{O}(1/N)$	$\mathcal{O}(1/N)$
MSE=Bias^2+Var	$\delta^2 + \mathcal{O}(1/N)$	$\mathcal{O}((T-t+1)/N)$	$\mathcal{O}(1/N)$

- FB/TF provide uniformly "good" approximations of $\{p(x_t | y_{1:T})\}_{t=1}^T$ whereas direct method provide only "good" approximation for $|T-t|$ "small".
- "Fast" implementations FB and TF of computational complexity $\mathcal{O}(NT)$ outperform other approaches as MSE is $\mathcal{O}(1/N)$ whereas it is $\mathcal{O}((T-t+1)/N)$ for direct SMC.

Convergence Results for Smoothed Additive Functionals

- Consider now the case where $\varphi_T(x_{1:T}) = \sum_{t=1}^{T} \varphi(x_t)$, so that

$$\begin{aligned}\overline{\varphi}_T &= \int \varphi_T(x_{1:T})\, p(x_{1:T}|\, y_{1:T})\, dx_{1:T} \\ &= \sum_{t=1}^{T} \int \varphi(x_t)\, p(x_t|\, y_{1:T})\, dx_t\end{aligned}$$

Convergence Results for Smoothed Additive Functionals

- Consider now the case where $\varphi_T(x_{1:T}) = \sum_{t=1}^{T} \varphi(x_t)$, so that

$$\begin{aligned}\overline{\varphi}_T &= \int \varphi_T(x_{1:T})\, p(x_{1:T} | y_{1:T})\, dx_{1:T} \\ &= \sum_{t=1}^{T} \int \varphi(x_t)\, p(x_t | y_{1:T})\, dx_t\end{aligned}$$

- This type of functionals is crucial when performing ML parameter estimation.

Convergence Results for Smoothed Additive Functionals

- Consider now the case where $\varphi_T(x_{1:T}) = \sum_{t=1}^{T} \varphi(x_t)$, so that

$$\begin{aligned}\overline{\varphi}_T &= \int \varphi_T(x_{1:T})\, p(x_{1:T}|\, y_{1:T})\, dx_{1:T} \\ &= \sum_{t=1}^{T} \int \varphi(x_t)\, p(x_t|\, y_{1:T})\, dx_t\end{aligned}$$

- This type of functionals is crucial when performing ML parameter estimation.
- We have for the standard path-based SMC estimate (Poyiadjis, D. & Singh, 2010)

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_T - \overline{\varphi}_T\right) \Rightarrow \mathcal{N}\left(0, \sigma_T^2\right) \text{ where } \underline{A}T^2 \leq \sigma_T^2 \leq \overline{A}T^2.$$

For the FB and TF estimates (Douc et al., 2009; Del Moral, D. & Singh, 2009), we have

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_T - \overline{\varphi}_T\right) \Rightarrow \mathcal{N}\left(0, \sigma_T^2\right) \text{ where } \sigma_T^2 \leq CT$$

Comparison Direct Method vs FB and TF

- Assume we are interested in approximating $\overline{\varphi}_T = \sum_{t=1}^{T} \int \varphi(x_t)\, p(x_t | y_{1:T})\, dx_t$ using SMC.

Method	Fixed-lag	Direct SMC	FB/TF
# particles	N	N	N
cost	$\mathcal{O}(TN)$	$\mathcal{O}(TN)$	$\mathcal{O}(TN^2)$, $\mathcal{O}(TN)$
Var.	$\mathcal{O}(T/N)$	$\mathcal{O}(T^2/N)$	$\mathcal{O}(T/N)$
Bias	$T\delta$	$\mathcal{O}(T/N)$	$\mathcal{O}(T/N)$
MSE=Bias^2+Var	$T^2\delta^2+\mathcal{O}(T/N)$	$\mathcal{O}(T^2/N)$	$\mathcal{O}(T^2/N^2)$

Comparison Direct Method vs FB and TF

- Assume we are interested in approximating $\overline{\varphi}_T = \sum_{t=1}^{T} \int \varphi(x_t)\, p(x_t | y_{1:T})\, dx_t$ using SMC.

Method	Fixed-lag	Direct SMC	FB/TF
# particles	N	N	N
cost	$\mathcal{O}(TN)$	$\mathcal{O}(TN)$	$\mathcal{O}(TN^2), \mathcal{O}(TN)$
Var.	$\mathcal{O}(T/N)$	$\mathcal{O}(T^2/N)$	$\mathcal{O}(T/N)$
Bias	$T\delta$	$\mathcal{O}(T/N)$	$\mathcal{O}(T/N)$
MSE=Bias^2+Var	$T^2\delta^2+\mathcal{O}(T/N)$	$\mathcal{O}(T^2/N)$	$\mathcal{O}(T^2/N^2)$

- "Naive" implementations FB and TF have MSE of same order as direct method for fixed computational complexity but MSE is bias dominated for FB/TF whereas it is variance dominated for Direct SMC.

Comparison Direct Method vs FB and TF

- Assume we are interested in approximating $\overline{\varphi}_T = \sum_{t=1}^{T} \int \varphi(x_t)\, p(x_t | y_{1:T})\, dx_t$ using SMC.

Method	Fixed-lag	Direct SMC	FB/TF
# particles	N	N	N
cost	$\mathcal{O}(TN)$	$\mathcal{O}(TN)$	$\mathcal{O}(TN^2)$, $\mathcal{O}(TN)$
Var.	$\mathcal{O}(T/N)$	$\mathcal{O}(T^2/N)$	$\mathcal{O}(T/N)$
Bias	$T\delta$	$\mathcal{O}(T/N)$	$\mathcal{O}(T/N)$
MSE=Bias^2+Var	$T^2\delta^2 + \mathcal{O}(T/N)$	$\mathcal{O}(T^2/N)$	$\mathcal{O}(T^2/N^2)$

- "Naive" implementations FB and TF have MSE of same order as direct method for fixed computational complexity but MSE is bias dominated for FB/TF whereas it is variance dominated for Direct SMC.
- "Fast" implementations FB and TF of computational complexity $\mathcal{O}(NT)$ outperform other approaches as MSE is $\mathcal{O}(T^2/N^2)$ whereas it is $\mathcal{O}(T^2/N)$ for direct SMC.

Experimental Results

- Consider a linear Gaussian model

$$X_t = 0.8X_{t-1} + 0.5V_t, \;\; V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$
$$Y_t = X_t + W_t, \;\; W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

Experimental Results

- Consider a linear Gaussian model

$$X_t = 0.8X_{t-1} + 0.5V_t,\ \ V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$
$$Y_t = X_t + W_t,\ \ W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

- We simulate 10,000 observations and compute SMC estimates of

$$\int \varphi_T(x_{1:T})\ \ p(x_{1:T}|\,y_{1:T})\, dx_{1:T}$$

  for 4 different additive functionals $\varphi_t(x_{1:t}) = \varphi_{t-1}(x_{1:t-1}) + \varphi(x_{t-1}, x_t, y_t)$ including $\varphi^1(x_{t-1}, x_t, y_t) = x_{t-1}x_t$, $\varphi^2(x_{t-1}, x_t, y_t) = x_t^2$. [Ground truth can be computed using Kalman smoother.]

Experimental Results

- Consider a linear Gaussian model

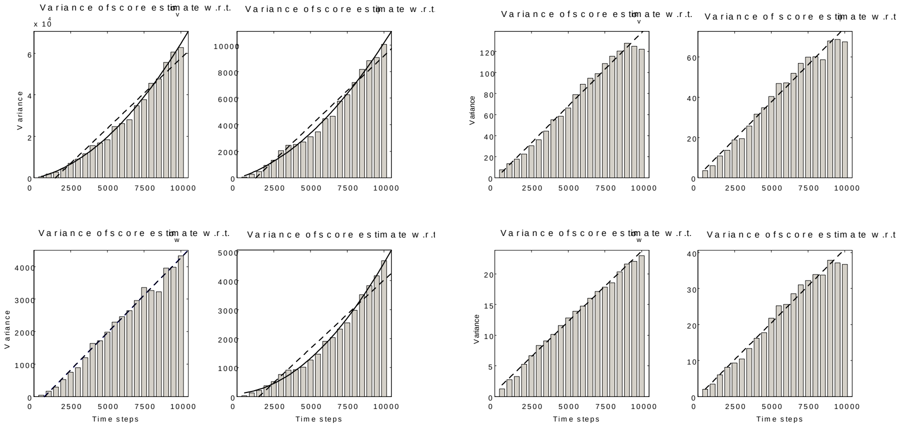
$$X_t = 0.8X_{t-1} + 0.5V_t, \;\; V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$
$$Y_t = X_t + W_t, \;\; W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

- We simulate 10,000 observations and compute SMC estimates of

$$\int \varphi_T(x_{1:T}) \; p(x_{1:T} | y_{1:T}) \, dx_{1:T}$$

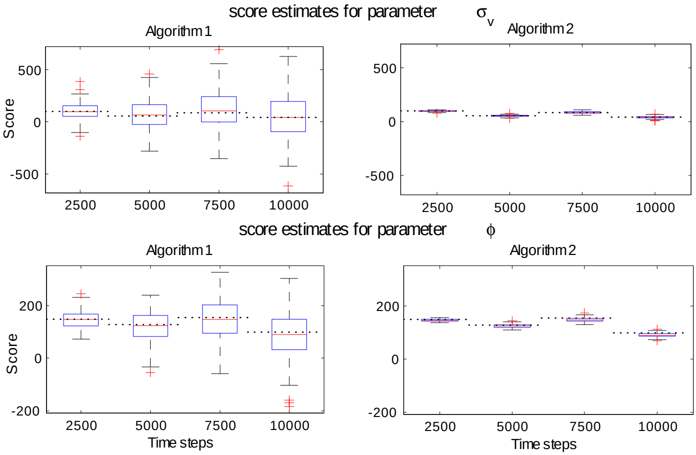
  for 4 different additive functionals $\varphi_t(x_{1:t}) = \varphi_{t-1}(x_{1:t-1}) + \varphi(x_{t-1}, x_t, y_t)$ including $\varphi^1(x_{t-1}, x_t, y_t) = x_{t-1}x_t$, $\varphi^2(x_{t-1}, x_t, y_t) = x_t^2$. [Ground truth can be computed using Kalman smoother.]
- We use SMC over 100 replications on the same dataset to estimate the empirical variance.

Empirical Variance for Direct vs FB



Direct (left) vs FB (right); the vertical scale is different

Boxplots of SMC Estimates for Direct vs FB



Direct (left) vs FB (right)

Summary

- SMC smoothing techniques allow us to “solve” the degeneracy problem.

Summary

- SMC smoothing techniques allow us to "solve" the degeneracy problem.
- SMC fixed-lag smoothing is the simplest one but has non-vanishing bias difficult to quantify.

Summary

- SMC smoothing techniques allow us to "solve" the degeneracy problem.
- SMC fixed-lag smoothing is the simplest one but has non-vanishing bias difficult to quantify.
- SMC FB and SMC TF algorithms provide uniformly "good" approximations of marginal smoothing distributions contrary to direct method.

Summary

- SMC smoothing techniques allow us to "solve" the degeneracy problem.
- SMC fixed-lag smoothing is the simplest one but has non-vanishing bias difficult to quantify.
- SMC FB and SMC TF algorithms provide uniformly "good" approximations of marginal smoothing distributions contrary to direct method.
- In terms of MSE, only "fast" implementations of SMC FB/TF provide a gain in terms of MSE.

Summary

- SMC smoothing techniques allow us to "solve" the degeneracy problem.
- SMC fixed-lag smoothing is the simplest one but has non-vanishing bias difficult to quantify.
- SMC FB and SMC TF algorithms provide uniformly "good" approximations of marginal smoothing distributions contrary to direct method.
- In terms of MSE, only "fast" implementations of SMC FB/TF provide a gain in terms of MSE.
- For direct implementation SMC FB/TF, MSE is of the same order but SMC FB/TF is bias dominated and direct SMC is variance dominated.

ML Parameter Estimation in State-Space Models

- In most scenarios of interest, the state-space model contains an unknown static parameter $\theta \in \Theta$ so that

$$X_1 \sim \mu_\theta(x_1) \text{ and } X_t | (X_{t-1} = x_{t-1}) \sim f_\theta(x_t | x_{t-1}).$$

ML Parameter Estimation in State-Space Models

- In most scenarios of interest, the state-space model contains an unknown static parameter $\theta \in \Theta$ so that

$$X_1 \sim \mu_\theta(x_1) \text{ and } X_t|(X_{t-1} = x_{t-1}) \sim f_\theta(x_t|x_{t-1}).$$

- The observations $\{Y_t\}_{t\geq 1}$ are conditionally independent given $\{X_t\}_{t\geq 1}$ and

$$Y_t|(X_t = x_t) \sim g_\theta(y_t|x_t).$$

ML Parameter Estimation in State-Space Models

- In most scenarios of interest, the state-space model contains an unknown static parameter $\theta \in \Theta$ so that

$$X_1 \sim \mu_\theta(x_1) \text{ and } X_t | (X_{t-1} = x_{t-1}) \sim f_\theta(x_t | x_{t-1}).$$

- The observations $\{Y_t\}_{t\geq 1}$ are conditionally independent given $\{X_t\}_{t\geq 1}$ and

$$Y_t | (X_t = x_t) \sim g_\theta(y_t | x_t).$$

- In many applications, we actually only care about θ and would like to estimate it off-line or on-line.

Examples

- **Stochastic Volatility model**

$$\begin{aligned} X_t &= \phi X_{t-1} + \sigma V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \\ Y_t &= \beta \exp(X_t/2)\, W_t, \; W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \end{aligned}$$

where $\theta = \left(\phi, \sigma^2, \beta\right)$.

Examples

- **Stochastic Volatility model**

$$\begin{aligned} X_t &= \phi X_{t-1} + \sigma V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \\ Y_t &= \beta \exp(X_t/2) W_t, \ W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \end{aligned}$$

where $\theta = (\phi, \sigma^2, \beta)$.

- **Biochemical Network model**

$$\begin{aligned} &\Pr\left(X^1_{t+dt}{=}x^1_t{+}1, X^2_{t+dt}{=}x^2_t \,\middle|\, x^1_t, x^2_t\right) = \alpha\, x^1_t\, dt + o(dt), \\ &\Pr\left(X^1_{t+dt}{=}x^1_t{-}1, X^2_{t+dt}{=}x^2_t{+}1 \,\middle|\, x^1_t, x^2_t\right) = \beta\, x^1_t\, x^2_t\, dt + o(dt), \\ &\Pr\left(X^1_{t+dt}{=}x^1_t, X^2_{t+dt}{=}x^2_t{-}1 \,\middle|\, x^1_t, x^2_t\right) = \gamma\, x^2_t\, dt + o(dt), \end{aligned}$$

with

$$Y_k = X^1_{k\Delta T} + W_k \text{ with } W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$$

where $\theta = (\alpha, \beta, \gamma)$.

Likelihood Function Estimation

- Let $y_{1:T}$ being given, the log-(marginal) likelihood is given by

$$\ell(\theta) = \log p_\theta \left(y_{1:T} \right).$$

Likelihood Function Estimation

- Let $y_{1:T}$ being given, the log-(marginal) likelihood is given by

$$\ell(\theta) = \log p_\theta(y_{1:T}).$$

- For any $\theta \in \Theta$, one can estimate $\ell(\theta)$ using standard SMC. methods, variance $\mathcal{O}(T/N)$.

Likelihood Function Estimation

- Let $y_{1:T}$ being given, the log-(marginal) likelihood is given by

$$\ell(\theta) = \log p_{\theta}\left(y_{1:T}\right).$$

- For any $\theta \in \Theta$, one can estimate $\ell(\theta)$ using standard SMC. methods, variance $\mathcal{O}\left(T/N\right)$.
- Direct maximization of $\ell(\theta)$ difficult as SMC estimate $\widehat{\ell}(\theta)$ is not a smooth function of θ even for fixed random seed.

Likelihood Function Estimation

- Let $y_{1:T}$ being given, the log-(marginal) likelihood is given by

$$\ell(\theta) = \log p_\theta\left(y_{1:T}\right).$$

- For any $\theta \in \Theta$, one can estimate $\ell(\theta)$ using standard SMC. methods, variance $\mathcal{O}\left(T/N\right)$.
- Direct maximization of $\ell(\theta)$ difficult as SMC estimate $\widehat{\ell}(\theta)$ is not a smooth function of θ even for fixed random seed.
- For $\dim\left(X_t\right) = 1$, we can obtain smooth estimate of log-likelihood function by using a smoothed resampling step (e.g. Pitt, 2002-2011); i.e. piecewise linear approximation of $\Pr\left(X_t < x \middle| y_{1:t}\right)$.

Likelihood Function Estimation

- Let $y_{1:T}$ being given, the log-(marginal) likelihood is given by

$$\ell(\theta) = \log p_\theta(y_{1:T}).$$

- For any $\theta \in \Theta$, one can estimate $\ell(\theta)$ using standard SMC. methods, variance $\mathcal{O}(T/N)$.
- Direct maximization of $\ell(\theta)$ difficult as SMC estimate $\widehat{\ell}(\theta)$ is not a smooth function of θ even for fixed random seed.
- For $\dim(X_t) = 1$, we can obtain smooth estimate of log-likelihood function by using a smoothed resampling step (e.g. Pitt, 2002-2011); i.e. piecewise linear approximation of $\Pr(X_t < x | y_{1:t})$.
- For $\dim(X_t) > 1$, we can obtain estimates of $\ell(\theta)$ highly positively correlated for neigbouring values in Θ (e.g. Lee, 2008).

Gradient Ascent

- To maximise $\ell(\theta)$ w.r.t θ, use at iteration $k+1$

$$\theta_{k+1} = \theta_k + \gamma_k \nabla \ell(\theta)|_{\theta=\theta_k}$$

where $\nabla \ell(\theta)|_{\theta=\theta_k}$ is the so-called score vector.

Gradient Ascent

- To maximise $\ell(\theta)$ w.r.t θ, use at iteration $k+1$

$$\theta_{k+1} = \theta_k + \gamma_k \left. \nabla \ell(\theta) \right|_{\theta=\theta_k}$$

where $\left. \nabla \ell(\theta) \right|_{\theta=\theta_k}$ is the so-called score vector.

- $\left. \nabla \ell(\theta) \right|_{\theta=\theta_k}$ can be estimated using finite differences but more efficiently using Fisher's identity (e.g. Cappé et al., 2005)

$$\nabla \ell(\theta) = \int \nabla \log p_\theta \left(x_{1:T}, y_{1:T} \right) \, p_\theta \left(x_{1:T} \middle| y_{1:T} \right) dx_{1:T}$$

where

$$\nabla \log p_\theta \left(x_{1:T}, y_{1:T} \right) = \nabla \log \mu_\theta \left(x_1 \right) \\ + \sum_{t=2}^{T} \nabla \log f_\theta \left(x_t \middle| x_{t-1} \right) + \sum_{t=1}^{T} \nabla \log g_\theta \left(y_t \middle| x_t \right).$$

Gradient Ascent

- To maximise $\ell(\theta)$ w.r.t θ, use at iteration $k+1$

$$\theta_{k+1} = \theta_k + \gamma_k \left. \nabla \ell(\theta) \right|_{\theta=\theta_k}$$

  where $\left. \nabla \ell(\theta) \right|_{\theta=\theta_k}$ is the so-called score vector.

- $\left. \nabla \ell(\theta) \right|_{\theta=\theta_k}$ can be estimated using finite differences but more efficiently using Fisher's identity (e.g. Cappé et al., 2005)

$$\nabla \ell(\theta) = \int \nabla \log p_\theta \left(x_{1:T}, y_{1:T} \right) \; p_\theta \left(x_{1:T} \middle| y_{1:T} \right) dx_{1:T}$$

  where

$$\begin{aligned} &\nabla \log p_\theta \left(x_{1:T}, y_{1:T} \right) = \nabla \log \mu_\theta \left(x_1 \right) \\ &+ \textstyle\sum_{t=2}^{T} \nabla \log f_\theta \left(x_t \middle| x_{t-1} \right) + \textstyle\sum_{t=1}^{T} \nabla \log g_\theta \left(y_t \middle| x_t \right). \end{aligned}$$

- An alternative is to use IPA (Coquelin, Deguest & Munos, 2009).

Example: SV Model

- Remember that

$$\begin{aligned} X_t &= \theta X_{t-1} + \sigma V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \\ Y_t &= \beta \exp(X_t/2)\, W_t, \ \ W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \end{aligned}$$

where we assume here that (σ^2, β) are known so that $\theta = \phi$.

Example: SV Model

- Remember that

$$\begin{aligned} X_t &= \theta X_{t-1} + \sigma V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \\ Y_t &= \beta \exp(X_t/2)\, W_t, \quad W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \end{aligned}$$

where we assume here that (σ^2, β) are known so that $\theta = \phi$.

- In this scenario

$$\begin{aligned} \log f_\theta(x_t | x_{t-1}) &= -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(x_t - \theta x_{t-1})^2, \\ \nabla \log f_\theta(x_t | x_{t-1}) &= \frac{x_{t-1}(x_t - \theta x_{t-1})}{\sigma^2} = \frac{x_{t-1}x_t}{\sigma^2} - \frac{\theta x_{t-1}^2}{\sigma^2}, \end{aligned}$$

hence

$$\nabla \ell(\theta) = \frac{\mathbb{E}_\theta\left(\sum_{t=2}^{T} X_{t-1}X_t \middle| y_{1:T}\right)}{\sigma^2} - \frac{\theta \mathbb{E}_\theta\left(\sum_{t=2}^{T} X_{t-1}^2 \middle| y_{1:T}\right)}{\sigma^2}.$$

Gradient Ascent using SMC

- An obvious SMC approximation is given by

$$\theta_{k+1} = \theta_k + \gamma_k \left. \widehat{\nabla \ell(\theta)} \right|_{\theta=\theta_k}$$

where $\left. \widehat{\nabla \ell(\theta)} \right|_{\theta=\theta_k}$ is estimated by your favourite SMC smoothing technique.

Gradient Ascent using SMC

- An obvious SMC approximation is given by

$$\theta_{k+1} = \theta_k + \gamma_k \left. \widehat{\nabla \ell(\theta)} \right|_{\theta=\theta_k}$$

where $\left. \widehat{\nabla \ell(\theta)} \right|_{\theta=\theta_k}$ is estimated by your favourite SMC smoothing technique.

- As $\nabla \ell(\theta)$ is a smoothed additive functional, all previously presented SMC methods and results do apply; see previous numerical results.

Gradient Ascent using SMC

- An obvious SMC approximation is given by

$$\theta_{k+1} = \theta_k + \gamma_k \left. \widehat{\nabla \ell(\theta)} \right|_{\theta=\theta_k}$$

  where $\left. \widehat{\nabla \ell(\theta)} \right|_{\theta=\theta_k}$ is estimated by your favourite SMC smoothing technique.
- As $\nabla \ell(\theta)$ is a smoothed additive functional, all previously presented SMC methods and results do apply; see previous numerical results.
- Similarly, it is possible to estimate the observed information matrix $-\nabla^2 \ell(\theta)$ using SMC based on Louis identity (e.g. Cappé et al., 2005) to implement a Newton-Raphson algorithm (Poyadjis, D. & Singh, 2010).

ML Parameter Estimation using EM

- The Expectation-Maximization (EM) algorithm is a celebrated alternative to gradient ascent technique.

ML Parameter Estimation using EM

- The Expectation-Maximization (EM) algorithm is a celebrated alternative to gradient ascent technique.
- To maximise $\ell(\theta)$ w.r.t θ, the EM uses

$$\theta_{k+1} = \arg\max \; Q(\theta_k, \theta).$$

where

$$Q(\theta_k, \theta) = \int \log p_\theta(x_{1:T}, y_{1:T}) \; p_{\theta_k}(x_{1:T} | y_{1:T}) dx_{1:T}$$

and we know that

$$\ell(\theta_{k+1}) \geq \ell(\theta_k).$$

ML Parameter Estimation using EM

- The Expectation-Maximization (EM) algorithm is a celebrated alternative to gradient ascent technique.
- To maximise $\ell(\theta)$ w.r.t θ, the EM uses

$$\theta_{k+1} = \arg\max \ Q(\theta_k, \theta).$$

where

$$Q(\theta_k, \theta) = \int \log p_\theta(x_{1:T}, y_{1:T}) \ p_{\theta_k}(x_{1:T} | y_{1:T}) dx_{1:T}$$

and we know that

$$\ell(\theta_{k+1}) \geq \ell(\theta_k).$$

- If $p_\theta(x_{1:T}, y_{1:T})$ is in the exponential family then we have

$$\theta_{k+1} = \Lambda \left(T^{-1} \varphi_T^{\theta_k} \right)$$

where

$$\varphi_T^\theta = \int \left(\sum_{t=2}^{T} \varphi\left(x_{t-1}, x_t, y_t\right) \right) p_\theta(x_{1:T} | y_{1:T}) dx_{1:T}$$

Example: SV Model

- Remember that

$$
\begin{aligned}
X_t &= \theta X_{t-1} + \sigma V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \\
Y_t &= \beta \exp(X_t/2)\, W_t, \; W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)
\end{aligned}
$$

where we assume here that (σ^2, β) are known so that $\theta = \phi$.

Example: SV Model

- Remember that

$$\begin{aligned} X_t &= \theta X_{t-1} + \sigma V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \\ Y_t &= \beta \exp(X_t/2)\, W_t, \quad W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \end{aligned}$$

where we assume here that (σ^2, β) are known so that $\theta = \phi$.

- In this scenario

$$\begin{aligned} \log f_\theta(x_t | x_{t-1}) &= -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(x_t - \theta x_{t-1})^2 \\ &= -\frac{1}{2}\log(2\pi\sigma^2) - \frac{x_t^2}{2\sigma^2} - \frac{\theta^2 x_{t-1}^2}{2\sigma^2} + \frac{\theta x_{t-1} x_t}{\sigma^2} \end{aligned}$$

so that

$$\theta_{k+1} = \frac{\mathbb{E}_{\theta_k}\left(\sum_{t=2}^{T} X_{t-1} X_t \middle| y_{1:T}\right)}{\mathbb{E}_{\theta_k}\left(\sum_{t=2}^{T} X_{t-1}^2 \middle| y_{1:T}\right)}.$$

EM using SMC

- SMC approximation of the EM is direct.

EM using SMC

- SMC approximation of the EM is direct.
- As EM requires computing smoothed additive functionals $\varphi_T^\theta = \int \left(\sum_{t=2}^T \varphi\left(x_{t-1}, x_t, y_t\right)\right) p_\theta(x_{1:T} | y_{1:T}) dx_{1:T}$, all previously presented SMC smoothing methods and results do apply.

EM using SMC

- SMC approximation of the EM is direct.
- As EM requires computing smoothed additive functionals $\varphi_T^\theta = \int \left(\sum_{t=2}^{T} \varphi\left(x_{t-1}, x_t, y_t\right) \right) p_\theta(x_{1:T} | y_{1:T}) dx_{1:T}$, all previously presented SMC smoothing methods and results do apply.
- There is obviously no more guarantee that $\ell(\theta_{k+1}) \geq \ell(\theta_k)$ for finite N but many positive experimental results; e.g. (Schon, Wills & Ninness, 2011).

ML Parameter Estimation using Online Gradient

- In many applications, we would like to estimate the parameter on-line.

ML Parameter Estimation using Online Gradient

- In many applications, we would like to estimate the parameter on-line.
- *Recursive maximum likelihood* (Titterington, 1984; LeGland & Mevel, 1997) proceeds as follows

$$\theta_{t+1} = \theta_t + \gamma_t \nabla \log \; p_{\theta_{1:t}} \left(y_t \middle| y_{1:t-1} \right)$$

where $p_{\theta_{1:t}} \left(y_t \middle| y_{1:t-1} \right)$ is computed using θ_k at time k and $\sum_t \gamma_t = \infty$, $\sum_t \gamma_t^2 < \infty$. Under regularity conditions, this converges towards a local maximum of the (average) log-likelihood.

ML Parameter Estimation using Online Gradient

- In many applications, we would like to estimate the parameter on-line.
- *Recursive maximum likelihood* (Titterington, 1984; LeGland & Mevel, 1997) proceeds as follows

$$\theta_{t+1} = \theta_t + \gamma_t \nabla \log p_{\theta_{1:t}} (y_t | y_{1:t-1})$$

where $p_{\theta_{1:t}} (y_t | y_{1:t-1})$ is computed using θ_k at time k and $\sum_t \gamma_t = \infty$, $\sum_t \gamma_t^2 < \infty$. Under regularity conditions, this converges towards a local maximum of the (average) log-likelihood.

- Note that

$$\nabla \log p_{\theta_{1:t}} (y_t | y_{1:t-1}) = \nabla \log p_{\theta_{1:t}} (y_{1:t}) - \nabla \log p_{\theta_{1:t-1}} (y_{1:t-1})$$

is given by the difference of two pseudo-score vectors where

$$\nabla \log p_{\theta_{1:t}} (y_{1:t}) := \int \left(\sum_{k=2}^{t} \nabla \log f_\theta (x_k | x_{k-1}) \big|_{\theta_k} + \nabla \log g_\theta (y_k | x_k) \big|_{\theta_k} \right) p_{\theta_{1:t}} (x_{1:t} | y_{1:t}) \, dx_{1:t}.$$

ML Parameter Estimation using SMC Online Gradient

- SMC approximation follows

$$\theta_{t+1} = \theta_t + \gamma_t \widehat{\nabla \log p_{\theta_{1:t}}} (y_t | y_{1:t-1})$$

where

$$\widehat{\nabla \log p_{\theta_{1:t}}} (y_t | y_{1:t-1}) = \widehat{\nabla \log p_{\theta_{1:t}}} (y_{1:t}) - \widehat{\nabla \log p_{\theta_{1:t-1}}} (y_{1:t-1})$$

is given by the difference of SMC estimates of pseudo-score vectors (Poyadjis, D. & Singh, 2011).

ML Parameter Estimation using SMC Online Gradient

- SMC approximation follows

$$\theta_{t+1} = \theta_t + \gamma_t \; \widehat{\nabla \log \; p}_{\theta_{1:t}} \left(y_t \middle| \, y_{1:t-1} \right)$$

where

$$\widehat{\nabla \log \; p}_{\theta_{1:t}} \left(y_t \middle| \, y_{1:t-1} \right) = \widehat{\nabla \log \; p}_{\theta_{1:t}} \left(y_{1:t} \right) - \widehat{\nabla \log \; p}_{\theta_{1:t-1}} \left(y_{1:t-1} \right)$$

is given by the difference of SMC estimates of pseudo-score vectors (Poyadjis, D. & Singh, 2011).

- Asymptotic variance of $\widehat{\nabla \log \; p}_{\theta_{1:t}} \left(y_t \middle| \, y_{1:t-1} \right)$ is uniformly bounded for FB estimate (Del Moral, D. & Singh, 2011) whereas it increases linearly with t for direct SMC method.

ML Parameter Estimation using SMC Online Gradient

- SMC approximation follows

$$\theta_{t+1} = \theta_t + \gamma_t \widehat{\nabla \log} p_{\theta_{1:t}}(y_t | y_{1:t-1})$$

where

$$\widehat{\nabla \log} p_{\theta_{1:t}}(y_t | y_{1:t-1}) = \widehat{\nabla \log} p_{\theta_{1:t}}(y_{1:t}) - \widehat{\nabla \log} p_{\theta_{1:t-1}}(y_{1:t-1})$$

is given by the difference of SMC estimates of pseudo-score vectors (Poyadjis, D. & Singh, 2011).

- Asymptotic variance of $\widehat{\nabla \log} p_{\theta_{1:t}}(y_t | y_{1:t-1})$ is uniformly bounded for FB estimate (Del Moral, D. & Singh, 2011) whereas it increases linearly with t for direct SMC method.
- **Major Problem**: If we use FB, this is not an online algorithm anymore as it requires a backward pass of order $\mathcal{O}(t)$ to approximate $\nabla \log p_{\theta_{1:t}}(y_{1:t})$ ...

Variance of the Gradient Estimate for Direct vs FB

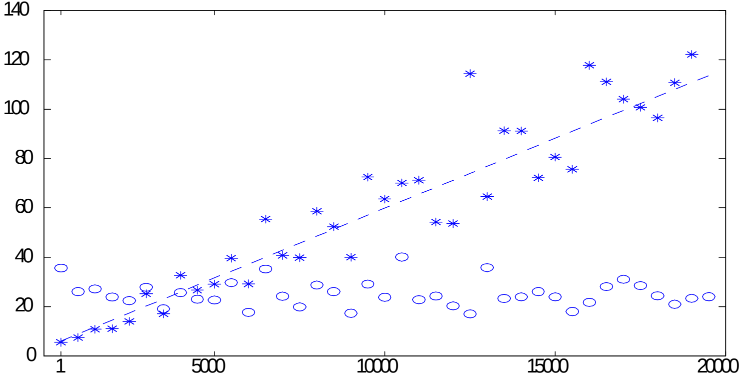


Figure: Empirical variance of the gradient estimate for standard versus FB approximations (SV model)

Online SMC ML Estimation using Direct Approximation

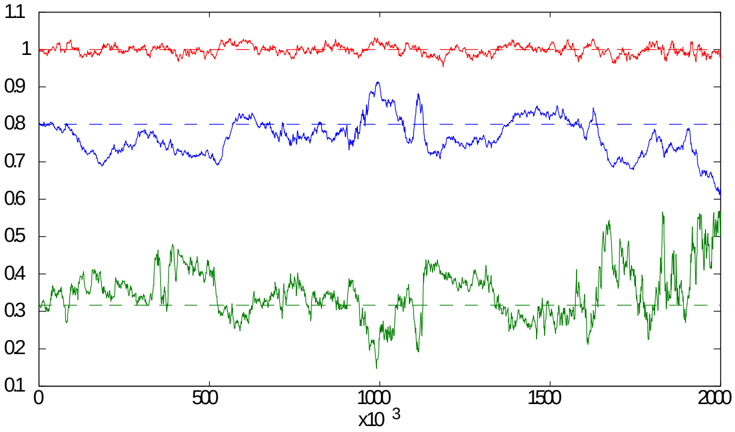


Figure: $N = 10,000$ particles, online parameter estimates for SV model.

SMC ML Estimation for SV Model using FB

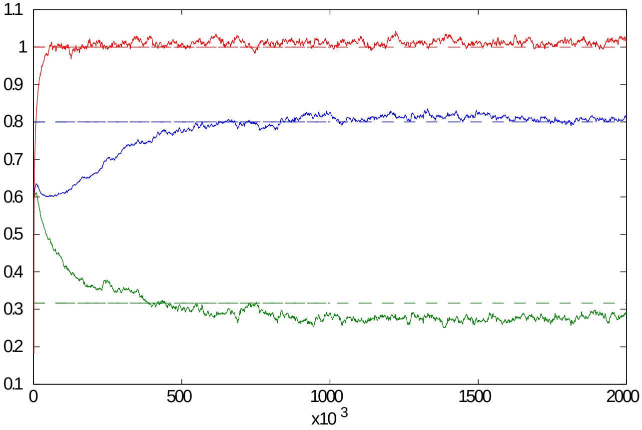


Figure: $N = 50$ particles, online parameter estimates for SV model.

Forward only Smoothing

- For the time being, we do not have an online implementation as a backward pass of length t is required at time t.

Forward only Smoothing

- For the time being, we do not have an online implementation as a backward pass of length t is required at time t.
- It is possible to completely bypass the backward pass to compute using FB

$$\varphi_t^{\theta} = \int_t \varphi_t\left(x_{1:t}\right)\ p_\theta\left(\left. x_{1:t} \right| y_{1:t}\right) dx_{1:t}$$

where

$$\varphi_t\left(x_{1:t}\right) = \sum_{k=1}^{t} \varphi\left(x_{k-1:k}, y_k\right)$$

using a dynamic programming trick for the "backward" Markov chain of transition densities $\left\{p_\theta\left(\left. x_k \right| y_{1:k}, x_{k+1}\right)\right\}$.

Forward only Smoothing

- For the time being, we do not have an online implementation as a backward pass of length t is required at time t.
- It is possible to completely bypass the backward pass to compute using FB

$$\varphi_t^{\theta} = \int_t \varphi_t\left(x_{1:t}\right)\ p_{\theta}\left(x_{1:t} \middle| y_{1:t}\right) dx_{1:t}$$

where

$$\varphi_t\left(x_{1:t}\right) = \sum_{k=1}^{t} \varphi\left(x_{k-1:k}, y_k\right)$$

using a dynamic programming trick for the "backward" Markov chain of transition densities $\left\{p_{\theta}\left(x_k \middle| y_{1:k}, x_{k+1}\right)\right\}$.
- Let us introduce the "value" function

$$V_t^{\theta}\left(x_t\right) := \int \varphi_t\left(x_{1:t}\right)\ p_{\theta}\left(x_{1:t-1} \middle| y_{1:t-1}, x_t\right) dx_{1:t-1}$$

then

$$\varphi_t^{\theta} = \int V_t^{\theta}\left(x_t\right) p_{\theta}\left(x_t \middle| y_{1:t}\right) dx_t.$$

Forward only Smoothing

- *Forward smoothing recursion*

$$V_t^{\theta}(x_t) = \int \left[V_{t-1}^{\theta}(x_{t-1}) + \varphi(x_{t-1:t}, y_t) \right] p_{\theta}(x_{t-1} | y_{1:t-1}, x_t) \, dx_{t-1}$$

Forward only Smoothing

- *Forward smoothing recursion*

$$V_t^\theta(x_t) = \int \left[V_{t-1}^\theta(x_{t-1}) + \varphi(x_{t-1:t}, y_t) \right] p_\theta(x_{t-1} | y_{1:t-1}, x_t)\, dx_{t-1}$$

- Proof is trivial

$$\begin{aligned}
&V_t^\theta(x_t) = \int \varphi_t(x_{1:t})\ p_\theta(x_{1:t-1} | y_{1:t-1}, x_t)\, dx_{1:t-1} \\
&= \int \left[\varphi_{t-1}(x_{1:t-1}) + \varphi(x_{t-1:t}, y_t) \right]\ p_\theta(x_{1:t-2} | y_{1:t-2}, x_{t-1}) \\
&\qquad\qquad\qquad \times p_\theta(x_{t-1} | y_{1:t-1}, x_t)\, dx_{1:t-1} \\
&= \int (\underbrace{\int \varphi_{t-1}(x_{1:t-1})\, p_\theta(x_{1:t-2} | y_{1:t-2}, x_{t-1})\, dx_{1:t-2}}_{V_{t-1}^\theta(x_{t-1})} \\
&\qquad\qquad + \varphi(x_{t-1:t}, y_t))\ p_\theta(x_{t-1} | y_{1:t-1}, x_t)\, dx_{t-1}
\end{aligned}$$

Forward only Smoothing

- *Forward smoothing recursion*

$$V_t^{\theta}(x_t) = \int \left[V_{t-1}^{\theta}(x_{t-1}) + \varphi(x_{t-1:t}, y_t) \right] p_{\theta}(x_{t-1} | y_{1:t-1}, x_t)\, dx_{t-1}$$

- Proof is trivial

$$\begin{aligned} V_t^{\theta}(x_t) &= \int \varphi_t(x_{1:t})\ p_{\theta}(x_{1:t-1} | y_{1:t-1}, x_t)\, dx_{1:t-1} \\ &= \int \left[\varphi_{t-1}(x_{1:t-1}) + \varphi(x_{t-1:t}, y_t) \right]\ p_{\theta}(x_{1:t-2} | y_{1:t-2}, x_{t-1}) \\ &\qquad \times p_{\theta}(x_{t-1} | y_{1:t-1}, x_t)\, dx_{1:t-1} \\ &= \int \Big(\underbrace{\int \varphi_{t-1}(x_{1:t-1})\, p_{\theta}(x_{1:t-2} | y_{1:t-2}, x_{t-1})\, dx_{1:t-2}}_{V_{t-1}^{\theta}(x_{t-1})} \\ &\qquad + \varphi(x_{t-1:t}, y_t) \Big)\ p_{\theta}(x_{t-1} | y_{1:t-1}, x_t)\, dx_{t-1} \end{aligned}$$

- Appears implicitly in Elliott, Aggoun & Moore (1996), Ford (1998) and rediscovered a few times... Presentation follows here (Del Moral, D. & Singh, 2009).

SMC Forward only Smoothing

- At time $t-1$, we have $\widehat{p}_\theta\left(x_{t-1}\middle|\, y_{1:t-1}\right)=\frac{1}{N}\sum_{i=1}^{N}\delta_{X_{t-1}^{(i)}}\left(x_{t-1}\right)$ and $\left\{\widehat{V}_{t-1}^{\theta}\left(X_{t-1}^{(i)}\right)\right\}_{1\leq i\leq N}$.

SMC Forward only Smoothing

- At time $t-1$, we have $\widehat{p}_\theta\left(x_{t-1} \mid y_{1:t-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{t-1}^{(i)}}\left(x_{t-1}\right)$ and $\left\{\widehat{V}_{t-1}^{\theta}\left(X_{t-1}^{(i)}\right)\right\}_{1\leq i\leq N}$.
- At time t, compute $\widehat{p}_\theta\left(x_t \mid y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_t^{(i)}}\left(x_t\right)$ and set

$$
\begin{aligned}
\widehat{V}_t^{\theta}\left(X_t^{(i)}\right) &= \int \left[\widehat{V}_{t-1}^{\theta}\left(x_{t-1}\right) + \varphi\left(x_{t-1:t}, y_t\right)\right] \widehat{p}_\theta\left(x_{t-1} \mid y_{1:t-1}, X_t^{(i)}\right) dx_{t-1} \\
&= \frac{\sum_{j=1}^{N} f_\theta\left(X_t^{(i)} | X_{t-1}^{(j)}\right)\left[\widehat{V}_{t-1}^{\theta}\left(X_{t-1}^{(j)}\right) + \varphi\left(X_{t-1}^{(j)}, X_t^{(i)}, y_t\right)\right]}{\sum_{j=1}^{N} f_\theta\left(X_t^{(i)} | X_{t-1}^{(j)}\right)},
\end{aligned}
$$

$$
\widehat{\varphi}_t^{\theta} = \frac{1}{N}\sum_{i=1}^{N} \widehat{V}_t^{\theta}\left(X_t^{(i)}\right).
$$

SMC Forward only Smoothing

- At time $t-1$, we have $\widehat{p}_\theta\left(x_{t-1} \mid y_{1:t-1}\right)=\frac{1}{N}\sum_{i=1}^{N}\delta_{X_{t-1}^{(i)}}\left(x_{t-1}\right)$ and $\left\{\widehat{V}_{t-1}^{\theta}\left(X_{t-1}^{(i)}\right)\right\}_{1\leq i\leq N}$.
- At time t, compute $\widehat{p}_\theta\left(x_t \mid y_{1:t}\right)=\sum_{i=1}^{N}W_t^{(i)}\delta_{X_t^{(i)}}\left(x_t\right)$ and set

$$\begin{aligned}\widehat{V}_t^{\theta}\left(X_t^{(i)}\right) &= \int\left[\widehat{V}_{t-1}^{\theta}\left(x_{t-1}\right)+\varphi\left(x_{t-1:t},y_t\right)\right]\widehat{p}_\theta\left(x_{t-1} \mid y_{1:t-1},X_t^{(i)}\right)dx_{t-1}\\ &= \frac{\sum_{j=1}^{N}f_\theta\left(X_t^{(i)}|X_{t-1}^{(j)}\right)\left[\widehat{V}_{t-1}^{\theta}\left(X_{t-1}^{(j)}\right)+\varphi\left(X_{t-1}^{(j)},X_t^{(i)},y_t\right)\right]}{\sum_{j=1}^{N}f_\theta\left(X_t^{(i)}|X_{t-1}^{(j)}\right)},\end{aligned}$$

$$\widehat{\varphi}_t^{\theta}=\frac{1}{N}\sum_{i=1}^{N}\widehat{V}_t^{\theta}\left(X_t^{(i)}\right).$$

- This estimate is exactly the same as the SMC FB estimate, computational complexity $\mathcal{O}\left(N^2\right)$.

ML Parameter Estimation using SMC Online Gradient

- At time $t-1$, we have $\widehat{p}_{\theta_{1:t-1}}\left(x_{t-1}\middle|\,y_{1:t-1}\right)$, $\left\{\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(X_{t-1}^{(i)}\right)\right\}$and $\widehat{\nabla \log\ } p_{\theta_{1:t-1}}\left(y_{1:t-1}\right)=\int \widehat{V}_{t-1}^{\theta_{1:t-1}}\left(x_{t-1}\right)\widehat{p}_{\theta_{1:t-1}}\left(x_{t-1}\middle|\,y_{1:t-1}\right)dx_{t-1}$ and obtained θ_t.

ML Parameter Estimation using SMC Online Gradient

- At time $t-1$, we have $\widehat{p}_{\theta_{1:t-1}}\left(x_{t-1}\middle|\,y_{1:t-1}\right)$, $\left\{\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(X_{t-1}^{(i)}\right)\right\}$and $\widehat{\nabla \log\ p}_{\theta_{1:t-1}}\left(y_{1:t-1}\right)=\int \widehat{V}_{t-1}^{\theta_{1:t-1}}\left(x_{t-1}\right)\widehat{p}_{\theta_{1:t-1}}\left(x_{t-1}\middle|\,y_{1:t-1}\right)dx_{t-1}$ and obtained θ_t.
- At time t, use SMC to compute $\widehat{p}_{\theta_{1:t}}\left(x_t\middle|\,y_{1:t}\right)$ and

$$\widehat{V}_t^{\theta_{1:t}}\left(X_t^{(i)}\right)=\int\left[\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(x_{t-1}\right)+\varphi\left(x_{t-1:t},y_t\right)\right]\widehat{p}_{\theta_{1:t}}\left(x_{t-1}\middle|\,y_{1:t-1},X_t^{(i)}\right)dx_{t-1}$$

$$\varphi\left(x_{t-1:t},y_t\right)=\nabla\log f_\theta\left(x_t\middle|\,x_{t-1}\right)\big|_{\theta_t}+\nabla\log g_\theta\left(y_t\middle|\,x_t\right)\big|_{\theta_t}$$

and

$$\widehat{\nabla \log\ p}_{\theta_{1:t}}\left(y_{1:t}\right)=\int\widehat{V}_t^{\theta_{1:t}}\left(x_t\right)\widehat{p}_{\theta_{1:t}}\left(x_t\middle|\,y_{1:t}\right)dx_t$$

ML Parameter Estimation using SMC Online Gradient

- At time $t-1$, we have $\widehat{p}_{\theta_{1:t-1}}\left(x_{t-1}\right| y_{1:t-1})$, $\left\{\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(X_{t-1}^{(i)}\right)\right\}$and $\widehat{\nabla \log\ p}_{\theta_{1:t-1}}\left(y_{1:t-1}\right)=\int\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(x_{t-1}\right)\widehat{p}_{\theta_{1:t-1}}\left(x_{t-1}\right| y_{1:t-1})\,dx_{t-1}$ and obtained θ_t.
- At time t, use SMC to compute $\widehat{p}_{\theta_{1:t}}\left(x_t\right| y_{1:t})$ and

$$\widehat{V}_t^{\theta_{1:t}}\left(X_t^{(i)}\right)=\int\left[\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(x_{t-1}\right)+\varphi\left(x_{t-1:t},y_t\right)\right]\widehat{p}_{\theta_{1:t}}\left(x_{t-1}\middle| y_{1:t-1},X_t^{(i)}\right)dx_{t-1}$$

$$\varphi\left(x_{t-1:t},y_t\right)=\left.\nabla\log f_\theta\left(x_t\right| x_{t-1})\right|_{\theta_t}+\left.\nabla\log g_\theta\left(y_t\right| x_t)\right|_{\theta_t}$$

and

$$\widehat{\nabla \log\ p}_{\theta_{1:t}}\left(y_{1:t}\right)=\int\widehat{V}_t^{\theta_{1:t}}\left(x_t\right)\widehat{p}_{\theta_{1:t}}\left(x_t\right| y_{1:t})\,dx_t$$

- Parameter update

$$\theta_{t+1}=\theta_t+\gamma_t\left(\widehat{\nabla \log\ p}_{\theta_{1:t}}\left(y_{1:t}\right)-\widehat{\nabla \log\ p}_{\theta_{1:t-1}}\left(y_{1:t-1}\right)\right)$$

Online ML Parameter Estimation through EM

- Batch EM uses

$$\begin{aligned} \varphi_T^{\theta_k} &= \int \left(\sum_{t=2}^{T} \varphi\left(x_{t-1:t}, y_t\right) \right) p_{\theta_k}\left(x_{1:T} \middle| y_{1:T}\right) dx_{1:T}, \\ \theta_{k+1} &= \Lambda\left(T^{-1} \varphi_T^{\theta_k}\right) \end{aligned}$$

Online ML Parameter Estimation through EM

- Batch EM uses

$$\varphi_T^{\theta_k} = \int \left(\sum_{t=2}^{T} \varphi\left(x_{t-1:t}, y_t\right) \right) p_{\theta_k}\left(x_{1:T} \middle| y_{1:T}\right) dx_{1:T},$$

$$\theta_{k+1} = \Lambda\left(T^{-1}\varphi_T^{\theta_k}\right)$$

- Online EM uses

$$\varphi_{t+1}^{\theta_{1:t}} = \gamma_{t+1} \int \varphi\left(x_{t:t+1}, y_{t+1}\right) p_{\theta_{1:t}}\left(x_t, x_{t+1} \middle| y_{1:t+1}\right) dx_{t:t+1}$$
$$+ \left(1-\gamma_{t+1}\right) \textstyle\sum_{k=1}^{t} \left(\displaystyle\prod_{l=k+2}^{t} \left(1-\gamma_l\right) \right) \gamma_{k+1}$$
$$\times \int \varphi\left(x_{k-1:k}, y_k\right) p_{\theta_{1:t}}\left(x_{k-1}, x_k \middle| y_{1:t+1}\right) dx_{k-1:k}$$

then set $\theta_{t+1} = \Lambda\left(\varphi_{t+1}^{\theta_{1:t}}\right)$ for $\{\gamma_t\}_{t\geq 1}$ satisfying $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$; e.g. $\gamma_t = t^{-\alpha}$ with $0.5 < \alpha \leq 1$.

Online ML Parameter Estimation through EM

- Batch EM uses

$$\varphi_T^{\theta_k} = \int \left(\sum_{t=2}^{T} \varphi\left(x_{t-1:t}, y_t\right) \right) p_{\theta_k}(x_{1:T} | y_{1:T}) dx_{1:T},$$

$$\theta_{k+1} = \Lambda\left(T^{-1} \varphi_T^{\theta_k} \right)$$

- Online EM uses

$$\varphi_{t+1}^{\theta_{1:t}} = \gamma_{t+1} \int \varphi\left(x_{t:t+1}, y_{t+1}\right) p_{\theta_{1:t}}(x_t, x_{t+1} | y_{1:t+1}) dx_{t:t+1}$$
$$+ (1 - \gamma_{t+1}) \sum_{k=1}^{t} \left(\prod_{l=k+2}^{t} (1 - \gamma_l) \right) \gamma_{k+1}$$
$$\times \int \varphi\left(x_{k-1:k}, y_k\right) p_{\theta_{1:t}}(x_{k-1}, x_k | y_{1:t+1}) dx_{k-1:k}$$

then set $\theta_{t+1} = \Lambda\left(\varphi_{t+1}^{\theta_{1:t}}\right)$ for $\{\gamma_t\}_{t \geq 1}$ satisfying $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$; e.g. $\gamma_t = t^{-\alpha}$ with $0.5 < \alpha \leq 1$.

- Under regularity conditions, this converges towards a local maximum of the (average) log-likelihood (well not yet proven for HMM)

Online ML Parameter Estimation through SMC EM

- At time $t-1$, we have $\widehat{p}_{\theta_{1:t-1}}\left(x_{t-1} \middle| y_{1:t-1}\right)$, $\left\{\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(X_{t-1}^{(i)}\right)\right\}$ and obtained θ_t.

Online ML Parameter Estimation through SMC EM

- At time $t-1$, we have $\widehat{p}_{\theta_{1:t-1}}\left(x_{t-1}\middle|\, y_{1:t-1}\right)$, $\left\{\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(X_{t-1}^{(i)}\right)\right\}$ and obtained θ_t.
- At time t, use SMC to compute $\widehat{p}_{\theta_{1:t}}\left(x_{t-1}\middle|\, y_{1:t-1}\right)$ and

$$\widehat{V}_t^{\theta_{1:t}}\left(X_t^{(i)}\right) = \int \left[\left(1-\gamma_t\right)\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(x_{t-1}\right) + \gamma_t \varphi\left(x_{t-1:t}, y_t\right)\right] \times \widehat{p}_{\theta_{1:t}}\left(x_{t-1}\middle|\, y_{1:t-1}, X_t^{(i)}\right) dx_{t-1},$$

$$\varphi_t^{\theta_{1:t}} = \int \widehat{V}_t^{\theta_{1:t}}\left(x_t\right)\widehat{p}_{\theta_{1:t}}\left(x_t\middle|\, y_{1:t}\right) dx_t$$

Online ML Parameter Estimation through SMC EM

- At time $t-1$, we have $\widehat{p}_{\theta_{1:t-1}}\left(x_{t-1}\middle|\, y_{1:t-1}\right)$, $\left\{\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(X_{t-1}^{(i)}\right)\right\}$ and obtained θ_t.
- At time t, use SMC to compute $\widehat{p}_{\theta_{1:t}}\left(x_{t-1}\middle|\, y_{1:t-1}\right)$ and

$$\widehat{V}_t^{\theta_{1:t}}\left(X_t^{(i)}\right) = \int \left[\left(1-\gamma_t\right)\widehat{V}_{t-1}^{\theta_{1:t-1}}\left(x_{t-1}\right) + \gamma_t \varphi\left(x_{t-1:t}, y_t\right)\right] \times \widehat{p}_{\theta_{1:t}}\left(x_{t-1}\middle|\, y_{1:t-1}, X_t^{(i)}\right) dx_{t-1},$$

$$\varphi_t^{\theta_{1:t}} = \int \widehat{V}_t^{\theta_{1:t}}\left(x_t\right)\widehat{p}_{\theta_{1:t}}\left(x_t\middle|\, y_{1:t}\right) dx_t$$

- Parameter update

$$\theta_{t+1} = \Lambda\left(\varphi_t^{\theta_{1:t}}\right)$$

Application to SV Model

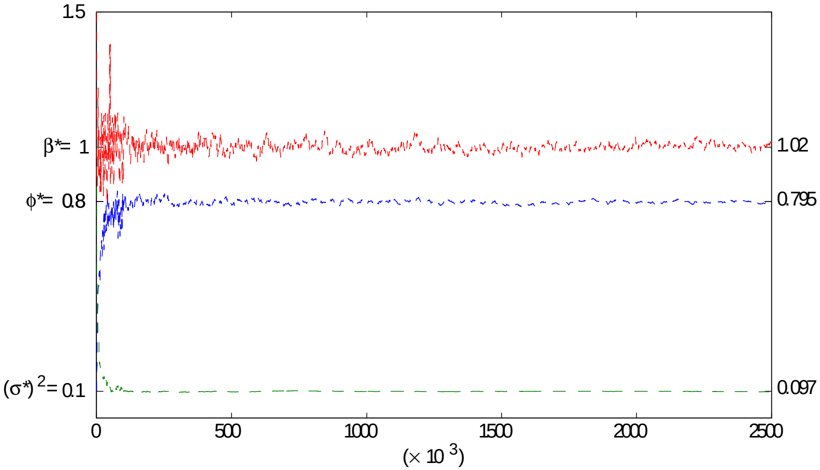


Figure: Online EM algorithm with $N = 200$ initialized at $\left(\phi, \sigma^2, \beta^2\right) = (0.1, 1, 2)$; the true values are $\left(\phi, \sigma^2, \beta^2\right) = (0.8, 0.1, 1)$.

Direct SMC vs Forward Smoothing for Online EM

- For online gradient techniques, forward smoothing is stable contrary to the direct method.

Direct SMC vs Forward Smoothing for Online EM

- For online gradient techniques, forward smoothing is stable contrary to the direct method.
- Structure of online EM is significantly different.

Direct SMC vs Forward Smoothing for Online EM

- For online gradient techniques, forward smoothing is stable contrary to the direct method.
- Structure of online EM is significantly different.
- We have seen previously that the MSE for smoothed additive functionals is of the same order for direct and FB estimates.

Direct SMC vs Forward Smoothing for Online EM

- For online gradient techniques, forward smoothing is stable contrary to the direct method.
- Structure of online EM is significantly different.
- We have seen previously that the MSE for smoothed additive functionals is of the same order for direct and FB estimates.
- Direct method is variance dominated, FB is bias dominated.

Direct SMC vs Forward Smoothing for Online EM

- For online gradient techniques, forward smoothing is stable contrary to the direct method.
- Structure of online EM is significantly different.
- We have seen previously that the MSE for smoothed additive functionals is of the same order for direct and FB estimates.
- Direct method is variance dominated, FB is bias dominated.
- We compare experimentally both methods on a simple linear Gaussian model over 100 runs.

Experimental Comparisons of Direct vs Forward Smoothing for online EM

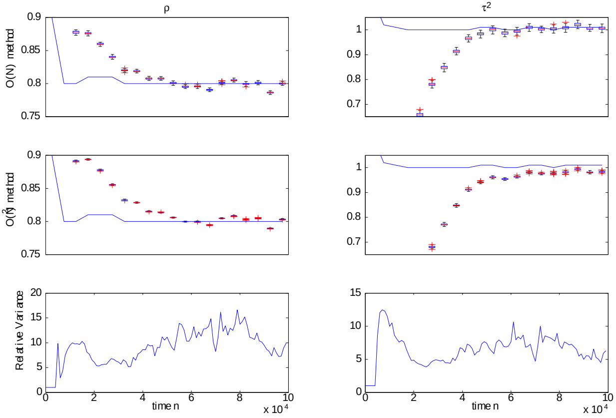


Figure: Parameter estimates for online EM obtained over 50 runs compared to ground truth: direct (left) vs forward smoothing (right).

Summary

- SMC smoothing techniques can be used to perform off-line and on-line ML parameter estimation.

Summary

- SMC smoothing techniques can be used to perform off-line and on-line ML parameter estimation.
- FB estimates for smoothed additive functionals can be computed using a forward only procedure.

Summary

- SMC smoothing techniques can be used to perform off-line and on-line ML parameter estimation.
- FB estimates for smoothed additive functionals can be computed using a forward only procedure.
- Forward smoothing allows us to implement a degeneracy free on-line gradient ascent algorithm.

Summary

- SMC smoothing techniques can be used to perform off-line and on-line ML parameter estimation.
- FB estimates for smoothed additive functionals can be computed using a forward only procedure.
- Forward smoothing allows us to implement a degeneracy free on-line gradient ascent algorithm.
- For on-line EM, forward smoothing and direct methods have both pros and cons with no clear winner.

Summary

- SMC smoothing techniques can be used to perform off-line and on-line ML parameter estimation.
- FB estimates for smoothed additive functionals can be computed using a forward only procedure.
- Forward smoothing allows us to implement a degeneracy free on-line gradient ascent algorithm.
- For on-line EM, forward smoothing and direct methods have both pros and cons with no clear winner.
- Bias reduction approaches are currently under study.

Bayesian Parameter Inference in State-Space Models

- Assume we have

$$X_t | (X_{t-1} = x_{t-1}) \sim f_\theta(x_t | x_{t-1}),$$
$$Y_t | (X_t = x_t) \sim g_\theta(y_t | x_t),$$

where θ is an *unknown* static parameter with prior $p(\theta)$.

Bayesian Parameter Inference in State-Space Models

- Assume we have

$$X_t | (X_{t-1} = x_{t-1}) \sim f_\theta(x_t | x_{t-1}),$$
$$Y_t | (X_t = x_t) \sim g_\theta(y_t | x_t),$$

where θ is an *unknown* static parameter with prior $p(\theta)$.

- Given data $y_{1:t}$, inference relies on

$$p(\theta, x_{1:t} | y_{1:t}) = p(\theta | y_{1:t})\, p_\theta(x_{1:t} | y_{1:t})$$

where

$$p(\theta | y_{1:t}) \propto p_\theta(y_{1:t})\, p(\theta).$$

Bayesian Parameter Inference in State-Space Models

- Assume we have

$$X_t|\left(X_{t-1}=x_{t-1}\right)\sim f_\theta\left(x_t|\,x_{t-1}\right),$$
$$Y_t|\left(X_t=x_t\right)\sim g_\theta\left(y_t|\,x_t\right),$$

  where θ is an *unknown* static parameter with prior $p(\theta)$.

- Given data $y_{1:t}$, inference relies on

$$p\left(\theta,x_{1:t}|\,y_{1:t}\right)=p\left(\theta|\,y_{1:t}\right)p_\theta\left(x_{1:t}|\,y_{1:t}\right)$$

  where

$$p\left(\theta|\,y_{1:t}\right)\propto p_\theta\left(y_{1:t}\right)p\left(\theta\right).$$

- SMC methods apply as it is a standard model with extended state $Z_t=(X_t,\theta_t)$ where

$$f\left(z_t|\,z_{t-1}\right)=\underbrace{\delta_{\theta_{t-1}}\left(\theta_t\right)}_{\text{practical problems}}f_{\theta_t}\left(x_t|\,x_{t-1}\right),\;g\left(y_t|\,z_t\right)=g_{\theta_t}\left(y_t|\,x_t\right).$$

Cautionary Warning

- For fixed θ, $\mathbb{V}\left[\widehat{p}_{\theta}\left(y_{1:t}\right)\right] / p_{\theta}^{2}\left(y_{1:t}\right)$ is in $\mathcal{O}\left(t/N\right)$.

Cautionary Warning

- For fixed θ, $\mathbb{V}\left[\widehat{p}_{\theta}\left(y_{1:t}\right)\right] / p_{\theta}^{2}\left(y_{1:t}\right)$ is in $\mathcal{O}\left(t/N\right)$.
- In a Bayesian context, the problem is even more complex as $p\left(\theta \mid y_{1:t}\right) \propto p_{\theta}\left(y_{1:t}\right) p\left(\theta\right)$ and we have $\theta_t = \theta$ for all t so the latent process does not enjoy mixing properties.

Cautionary Warning

- For fixed θ, $\mathbb{V}\left[\widehat{p}_{\theta}\left(y_{1:t}\right)\right] / p_{\theta}^{2}\left(y_{1:t}\right)$ is in $\mathcal{O}\left(t/N\right)$.
- In a Bayesian context, the problem is even more complex as $p\left(\theta \mid y_{1:t}\right) \propto p_{\theta}\left(y_{1:t}\right) p\left(\theta\right)$ and we have $\theta_t = \theta$ for all t so the latent process does not enjoy mixing properties.
- A seemingly attractive idea consists of using MCMC steps on θ; e.g. (Andrieu, De Freitas & D.,1999; Fearnhead, 2002; Gilks & Berzuini 2001; Storvik, 2002; Carvalho et al., 2010) so as to introduce some "noise" on the θ component of the state.

Cautionary Warning

- For fixed θ, $\mathbb{V}\left[\widehat{p}_{\theta}\left(y_{1:t}\right)\right] / p_{\theta}^{2}\left(y_{1:t}\right)$ is in $\mathcal{O}\left(t/N\right)$.
- In a Bayesian context, the problem is even more complex as $p\left(\theta \mid y_{1:t}\right) \propto p_{\theta}\left(y_{1:t}\right) p\left(\theta\right)$ and we have $\theta_t = \theta$ for all t so the latent process does not enjoy mixing properties.
- A seemingly attractive idea consists of using MCMC steps on θ; e.g. (Andrieu, De Freitas & D.,1999; Fearnhead, 2002; Gilks & Berzuini 2001; Storvik, 2002; Carvalho et al., 2010) so as to introduce some "noise" on the θ component of the state.
- When $p\left(\theta \mid y_{1:t}, x_{1:t}\right) = p\left(\theta \mid s_t\left(x_{1:t}, y_{1:t}\right)\right)$ where $s_t\left(x_{1:t}, y_{1:t}\right)$ is a fixed-dimensional of sufficient statistics, the algorithm is particularly elegant but still implicitly relies on SMC approximation of $p\left(x_{1:t} \mid y_{1:t}\right)$ so degeneracy will creep in.

Cautionary Warning

- For fixed θ, $\mathbb{V}\left[\widehat{p}_{\theta}\left(y_{1:t}\right)\right] / p_{\theta}^{2}\left(y_{1:t}\right)$ is in $\mathcal{O}\left(t/N\right)$.
- In a Bayesian context, the problem is even more complex as $p\left(\theta \mid y_{1:t}\right) \propto p_{\theta}\left(y_{1:t}\right) p\left(\theta\right)$ and we have $\theta_t = \theta$ for all t so the latent process does not enjoy mixing properties.
- A seemingly attractive idea consists of using MCMC steps on θ; e.g. (Andrieu, De Freitas & D.,1999; Fearnhead, 2002; Gilks & Berzuini 2001; Storvik, 2002; Carvalho et al., 2010) so as to introduce some "noise" on the θ component of the state.
- When $p\left(\theta \mid y_{1:t}, x_{1:t}\right) = p\left(\theta \mid s_t\left(x_{1:t}, y_{1:t}\right)\right)$ where $s_t\left(x_{1:t}, y_{1:t}\right)$ is a fixed-dimensional of sufficient statistics, the algorithm is particularly elegant but still implicitly relies on SMC approximation of $p\left(x_{1:t} \mid y_{1:t}\right)$ so degeneracy will creep in.
- As $\dim\left(Z_t\right) = \dim\left(X_t\right) + \dim\left(\theta\right)$, such methods are not recommended for high-dimensional θ, especially with vague priors.

SMC with MCMC Step for Parameter Estimation

- Given at time $t-1$, the approximation

$$\widehat{p}\left(\theta, x_{1:t-1} \middle| y_{1:t-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_{t-1}^{(i)}, X_{1:t-1}^{(i)}\right)}\left(\theta, x_{1:t-1}\right),$$

we update the approximation as follows at time t.

SMC with MCMC Step for Parameter Estimation

- Given at time $t-1$, the approximation

$$\widehat{p}\left(\theta, x_{1:t-1} \middle| y_{1:t-1}\right) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\left(\theta_{t-1}^{(i)}, X_{1:t-1}^{(i)}\right)}\left(\theta, x_{1:t-1}\right),$$

we update the approximation as follows at time t.

- Sample $\widetilde{X}_t^{(i)} \sim f_{\theta_{t-1}^{(i)}}\left(\cdot \middle| X_{t-1}^{(i)}\right)$, set $\widetilde{X}_{1:t}^{(i)} = \left(X_{1:t-1}^{(i)}, \widetilde{X}_t^{(i)}\right)$ and

$$\widetilde{p}\left(\theta, x_{1:t} \middle| y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)}\delta_{\left(\theta_{t-1}^{(i)}, \widetilde{X}_{1:t}^{(i)}\right)}\left(\theta, x_{1:t}\right),$$

$$W_t^{(i)} \propto g_{\theta_{t-1}^{(i)}}\left(y_t \middle| \widetilde{X}_t^{(i)}\right).$$

SMC with MCMC Step for Parameter Estimation

- Given at time $t-1$, the approximation

$$\widehat{p}\left(\theta, x_{1:t-1} \middle| y_{1:t-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_{t-1}^{(i)}, X_{1:t-1}^{(i)}\right)}\left(\theta, x_{1:t-1}\right),$$

  we update the approximation as follows at time t.

- Sample $\widetilde{X}_t^{(i)} \sim f_{\theta_{t-1}^{(i)}}\left(\cdot \middle| X_{t-1}^{(i)}\right)$, set $\widetilde{X}_{1:t}^{(i)} = \left(X_{1:t-1}^{(i)}, \widetilde{X}_t^{(i)}\right)$ and

$$\widetilde{p}\left(\theta, x_{1:t} \middle| y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\left(\theta_{t-1}^{(i)}, \widetilde{X}_{1:t}^{(i)}\right)}\left(\theta, x_{1:t}\right),$$

$$W_t^{(i)} \propto g_{\theta_{t-1}^{(i)}}\left(y_t \middle| \widetilde{X}_t^{(i)}\right).$$

- Resample $X_{1:t}^{(i)} \sim \widetilde{p}\left(x_{1:t} \middle| y_{1:t}\right)$ then sample $\theta_t^{(i)} \sim p\left(\theta \middle| y_{1:t}, X_{1:t}^{(i)}\right)$ to obtain $\widehat{p}\left(\theta, x_{1:t} \middle| y_{1:t}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_t^{(i)}, X_{1:t}^{(i)}\right)}\left(\theta, x_{1:t}\right)$.

A Toy Example

- Linear Gaussian state-space model

$$X_t = \theta X_{t-1} + \sigma_V V_t, \ \ V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$
$$Y_t = X_t + \sigma_W W_t, \ \ W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

A Toy Example

- Linear Gaussian state-space model

$$X_t = \theta X_{t-1} + \sigma_V V_t, \;\; V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$
$$Y_t = X_t + \sigma_W W_t, \;\; W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

- We set $p(\theta) \propto 1_{(-1,1)}(\theta)$ so

$$p(\theta | y_{1:t}, x_{1:t}) \propto \mathcal{N}\left(\theta; m_t, \sigma_t^2\right) 1_{(-1,1)}(\theta)$$

where

$$\sigma_t^2 = S_{2,t}^{-1}, \;\; m_t = S_{2,t}^{-1} S_{1,t}$$

with

$$S_{1,t} = \sum_{k=2}^{t} x_{k-1} x_k, \;\; S_{2,t} = \sum_{k=2}^{t} x_{k-1}^2$$

SMC with MCMC Step for Parameter Estimation

- At time $t-1$, $\left(\theta_{t-1}^{(i)}, X_{t-1}^{(i)}, S_{t-1}^{(i)}\right)$ we have

$$\widehat{p}\left(\theta, x_{t-1}, s_{t-1} \middle| y_{1:t-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_{t-1}^{(i)}, X_{t-1}^{(i)}, S_{t-1}^{(i)}\right)}\left(\theta, x_{t-1}, s_{t-1}\right).$$

SMC with MCMC Step for Parameter Estimation

- At time $t-1$, $\left(\theta_{t-1}^{(i)}, X_{t-1}^{(i)}, S_{t-1}^{(i)}\right)$ we have

$$\widehat{p}\left(\theta, x_{t-1}, s_{t-1} \middle| y_{1:t-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_{t-1}^{(i)}, X_{t-1}^{(i)}, S_{t-1}^{(i)}\right)}\left(\theta, x_{t-1}, s_{t-1}\right).$$

- Sample $\widetilde{X}_t^{(i)} \sim f_{\theta_{t-1}^{(i)}}\left(\cdot \middle| X_{t-1}^{(i)}\right)$, set $\widetilde{S}_{1,t}^{(i)} = S_{1,t-1}^{(i)} + X_{t-1}^{(i)}\widetilde{X}_t^{(i)}$, $\widetilde{S}_{2,t}^{(i)} = S_{2,t-1}^{(i)} + \left(X_{t-1}^{(i)}\right)^2$, $W_t^{(i)} \propto g_{\theta_{t-1}^{(i)}}\left(y_t \middle| \widetilde{X}_t^{(i)}\right)$ and

$$\widetilde{p}\left(\theta, x_t, s_t \middle| y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\left(\theta_{t-1}^{(i)}, \widetilde{X}_t^{(i)}, \widetilde{S}_t^{(i)}\right)}\left(\theta, x_t, s_t\right).$$

SMC with MCMC Step for Parameter Estimation

- At time $t-1$, $\left(\theta_{t-1}^{(i)}, X_{t-1}^{(i)}, S_{t-1}^{(i)}\right)$ we have

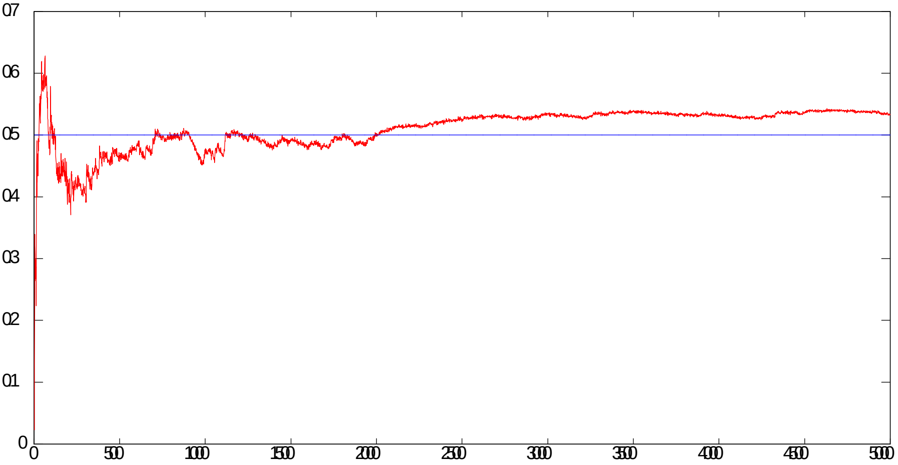
$$\widehat{p}\left(\theta, x_{t-1}, s_{t-1} \mid y_{1:t-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_{t-1}^{(i)}, X_{t-1}^{(i)}, S_{t-1}^{(i)}\right)}\left(\theta, x_{t-1}, s_{t-1}\right).$$

- Sample $\widetilde{X}_t^{(i)} \sim f_{\theta_{t-1}^{(i)}}\left(\cdot \mid X_{t-1}^{(i)}\right)$, set $\widetilde{S}_{1,t}^{(i)} = S_{1,t-1}^{(i)} + X_{t-1}^{(i)}\widetilde{X}_t^{(i)}$, $\widetilde{S}_{2,t}^{(i)} = S_{2,t-1}^{(i)} + \left(X_{t-1}^{(i)}\right)^2$, $W_t^{(i)} \propto g_{\theta_{t-1}^{(i)}}\left(y_t \mid \widetilde{X}_t^{(i)}\right)$ and

$$\widetilde{p}\left(\theta, x_t, s_t \mid y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{\left(\theta_{t-1}^{(i)}, \widetilde{X}_t^{(i)}, \widetilde{S}_t^{(i)}\right)}\left(\theta, x_t, s_t\right).$$

- Resample $\left(X_t^{(i)}, S_t^{(i)}\right) \sim \widetilde{p}\left(x_t, s_t \mid y_{1:t}\right)$ then sample $\theta_t^{(i)} \sim \mathcal{N}\left(\theta; \left(S_{2,t}^{(i)}\right)^{-1} S_{1,t}^{(i)}, \left(S_{2,t}^{(i)}\right)^{-1}\right) 1_{(-1,1)}(\theta)$ to obtain $\widehat{p}\left(\theta, x_t, s_t \mid y_{1:t}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_t^{(i)}, X_t^{(i)}, S_t^{(i)}\right)}\left(\theta, x_t, s_t\right)$.

Illustration of the Degeneracy Problem



SMC estimate of $\mathbb{E}\left[\theta \middle| y_{1:t}\right]$, as t increases the degeneracy creeps in.

Another Toy Example

- Linear Gaussian state-space model

$$X_t = \rho X_{t-1} + V_t, \;\; V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$
$$Y_t = X_t + \sigma W_t, \;\; W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

Another Toy Example

- Linear Gaussian state-space model

$$X_t = \rho X_{t-1} + V_t, \ V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$
$$Y_t = X_t + \sigma W_t, \ W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

- We set $\rho \sim \mathcal{U}_{(-1,1)}$ and $\sigma^2 \sim \mathcal{IG}(1,1)$.

Another Toy Example

- Linear Gaussian state-space model

$$X_t = \rho X_{t-1} + V_t, \; V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$
$$Y_t = X_t + \sigma W_t, \; W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

- We set $\rho \sim \mathcal{U}_{(-1,1)}$ and $\sigma^2 \sim \mathcal{IG}(1,1)$.
- We use particle filter with perfect adaptation and Gibbs moves with $N = 10000$; particle learning (Andrieu, D. & De Freitas, 1999; Carvalho et al., 2010)

Another Toy Example

- Linear Gaussian state-space model

$$X_t = \rho X_{t-1} + V_t, \;\; V_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$
$$Y_t = X_t + \sigma W_t, \;\; W_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

- We set $\rho \sim \mathcal{U}_{(-1,1)}$ and $\sigma^2 \sim \mathcal{IG}(1,1)$.
- We use particle filter with perfect adaptation and Gibbs moves with $N = 10000$; particle learning (Andrieu, D. & De Freitas, 1999; Carvalho et al., 2010)
- We compare to the ground truth obtained using Kalman filter on states and grid on parameters.

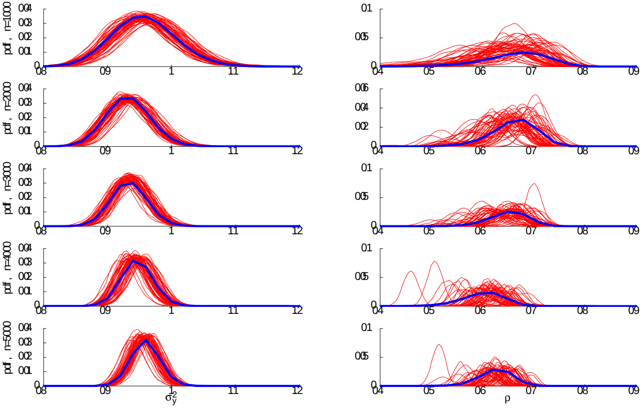


Figure: Estimates of $p(\rho | y_{1:t})$ and $p(\sigma^2 | y_{1:t})$ over 50 runs (red) vs ground truth (blue) for $t = 10^3, 2.10^3, ..., 5.10^3$ for $N = 10^4$.

Online Bayesian Parameter Estimation

- All proposed procedures for online Bayesian parameter estimation are deficient.
- Some artificial dynamics can be introduced but then we do not approximate $\{p(\theta, x_{1:t} | y_{1:t})\}_{t \geq 1}$; e.g. (Liu & West, 2001; Flury & Shephard, 2010).
- Methods based on MCMC steps are elegant but do suffer from the degeneracy problem and provide unreliable approximations.

Offline Bayesian Parameter Estimation

- Given a collection of observations $y_{1:T} := (y_1, ..., y_T)$, T being fixed, inference relies on the posterior density

$$\begin{aligned} p(\theta, x_{1:T} | y_{1:T}) &= p(\theta | y_{1:T}) \, p_\theta(x_{1:T} | y_{1:T}) \\ &\propto p(\theta, x_{1:T}, y_{1:T}) \end{aligned}$$

where

$$p(\theta, x_{1:T}, y_{1:T}) \propto p(\theta) \, \mu_\theta(x_1) \prod_{t=2}^{T} f_\theta(x_t | x_{t-1}) \prod_{t=1}^{T} g_\theta(y_t | x_t) \ .$$

Offline Bayesian Parameter Estimation

- Given a collection of observations $y_{1:T} := (y_1, ..., y_T)$, T being fixed, inference relies on the posterior density

$$\begin{aligned} p(\theta, x_{1:T} | y_{1:T}) &= p(\theta | y_{1:T}) \, p_\theta(x_{1:T} | y_{1:T}) \\ &\propto p(\theta, x_{1:T}, y_{1:T}) \end{aligned}$$

where

$$p(\theta, x_{1:T}, y_{1:T}) \propto p(\theta) \, \mu_\theta(x_1) \prod_{t=2}^{T} f_\theta(x_t | x_{t-1}) \prod_{t=1}^{T} g_\theta(y_t | x_t) \; .$$

- We show how to address this problem using particle MCMC (Andrieu, D. & Holenstein, *JRSS* B, 2010).

Common MCMC Approaches and Limitations

- **MCMC Idea**: Simulate an ergodic Markov chain $\{\theta(i), X_{1:T}(i)\}_{i\geq 0}$ of invariant distribution $p(\theta, x_{1:T} | y_{1:T})$... infinite number of possibilities.

Common MCMC Approaches and Limitations

- **MCMC Idea**: Simulate an ergodic Markov chain $\{\theta(i), X_{1:T}(i)\}_{i\geq 0}$ of invariant distribution $p(\theta, x_{1:T} | y_{1:T})$... infinite number of possibilities.
- Typical strategies consists of updating iteratively $X_{1:T}$ conditional upon θ then θ conditional upon $X_{1:T}$.

Common MCMC Approaches and Limitations

- **MCMC Idea**: Simulate an ergodic Markov chain $\{\theta(i), X_{1:T}(i)\}_{i\geq 0}$ of invariant distribution $p(\theta, x_{1:T} | y_{1:T})$... infinite number of possibilities.
- Typical strategies consists of updating iteratively $X_{1:T}$ conditional upon θ then θ conditional upon $X_{1:T}$.
- To update $X_{1:T}$ conditional upon θ, use MCMC kernels updating subblocks according to $p_\theta(x_{t:t+K-1} | y_{t:t+K-1}, x_{t-1}, x_{t+K})$.

Common MCMC Approaches and Limitations

- **MCMC Idea**: Simulate an ergodic Markov chain $\{\theta(i), X_{1:T}(i)\}_{i\geq 0}$ of invariant distribution $p(\theta, x_{1:T} | y_{1:T})$... infinite number of possibilities.
- Typical strategies consists of updating iteratively $X_{1:T}$ conditional upon θ then θ conditional upon $X_{1:T}$.
- To update $X_{1:T}$ conditional upon θ, use MCMC kernels updating subblocks according to $p_\theta(x_{t:t+K-1} | y_{t:t+K-1}, x_{t-1}, x_{t+K})$.
- Standard MCMC algorithms are inefficient if θ and $X_{1:T}$ are strongly correlated.

Common MCMC Approaches and Limitations

- **MCMC Idea**: Simulate an ergodic Markov chain $\{\theta(i), X_{1:T}(i)\}_{i\geq 0}$ of invariant distribution $p(\theta, x_{1:T} | y_{1:T})$... infinite number of possibilities.
- Typical strategies consists of updating iteratively $X_{1:T}$ conditional upon θ then θ conditional upon $X_{1:T}$.
- To update $X_{1:T}$ conditional upon θ, use MCMC kernels updating subblocks according to $p_\theta(x_{t:t+K-1} | y_{t:t+K-1}, x_{t-1}, x_{t+K})$.
- Standard MCMC algorithms are inefficient if θ and $X_{1:T}$ are strongly correlated.
- Strategy impossible to implement when it is only possible to sample from the prior but impossible to evaluate it pointwise.

Metropolis-Hastings (MH) Sampling

- To bypass these problems, we want to update jointly θ and $X_{1:T}$.

Metropolis-Hastings (MH) Sampling

- To bypass these problems, we want to update jointly θ and $X_{1:T}$.
- Assume that the current state of our Markov chain is $(\theta, x_{1:T})$, we propose to update simultaneously the parameter and the states using a proposal

$$q\left(\left(\theta^{*}, x_{1:T}^{*}\right) \middle| \left(\theta, x_{1:T}\right)\right) = q\left(\theta^{*} \middle| \theta\right) \; q_{\theta^{*}}\left(x_{1:T}^{*} \middle| y_{1:T}\right).$$

Metropolis-Hastings (MH) Sampling

- To bypass these problems, we want to update jointly θ and $X_{1:T}$.
- Assume that the current state of our Markov chain is $(\theta, x_{1:T})$, we propose to update simultaneously the parameter and the states using a proposal

$$q\left(\left(\theta^{*}, x_{1:T}^{*}\right)\middle|\left(\theta, x_{1:T}\right)\right) = q\left(\theta^{*}\middle|\theta\right)\ q_{\theta^{*}}\left(x_{1:T}^{*}\middle|y_{1:T}\right).$$

- The proposal $(\theta^{*}, x_{1:T}^{*})$ is accepted with MH acceptance probability

$$1 \wedge \frac{p\left(\theta^{*}, x_{1:T}^{*}\middle|y_{1:T}\right)}{p\left(\theta, x_{1:T}\middle|y_{1:T}\right)} \frac{q\left(\left(x_{1:T}, \theta\right)\middle|\left(x_{1:T}^{*}, \theta^{*}\right)\right)}{q\left(\left(x_{1:T}^{*}, \theta^{*}\right)\middle|\left(x_{1:T}, \theta\right)\right)}$$

Metropolis-Hastings (MH) Sampling

- To bypass these problems, we want to update jointly θ and $X_{1:T}$.
- Assume that the current state of our Markov chain is $(\theta, x_{1:T})$, we propose to update simultaneously the parameter and the states using a proposal

$$q\left(\left(\theta^*, x^*_{1:T}\right) \middle| \left(\theta, x_{1:T}\right)\right) = q\left(\theta^* \middle| \theta\right)\; q_{\theta^*}\left(x^*_{1:T} \middle| y_{1:T}\right).$$

- The proposal $(\theta^*, x^*_{1:T})$ is accepted with MH acceptance probability

$$1 \wedge \frac{p\left(\theta^*, x^*_{1:T} \middle| y_{1:T}\right)}{p\left(\theta, x_{1:T} \middle| y_{1:T}\right)} \frac{q\left(\left(x_{1:T}, \theta\right) \middle| \left(x^*_{1:T}, \theta^*\right)\right)}{q\left(\left(x^*_{1:T}, \theta^*\right) \middle| \left(x_{1:T}, \theta\right)\right)}$$

- **Problem**: Designing a proposal $q_{\theta^*}\left(x^*_{1:T} \middle| y_{1:T}\right)$ such that the acceptance probability is not extremely small is very difficult.

"Idealized" Marginal MH Sampler

- Consider the following so-called marginal Metropolis-Hastings (MH) algorithm which uses as a proposal

$$q\left(\left(x_{1:T}^{*},\theta^{*}\right)\middle|\left(x_{1:T},\theta\right)\right)=q\left(\theta^{*}\middle|\theta\right)p_{\theta^{*}}\left(x_{1:T}^{*}\middle|y_{1:T}\right).$$

"Idealized" Marginal MH Sampler

- Consider the following so-called marginal Metropolis-Hastings (MH) algorithm which uses as a proposal

$$q\left(\left(x_{1:T}^{*},\theta^{*}\right)\middle|\left(x_{1:T},\theta\right)\right)=q\left(\theta^{*}\middle|\theta\right)p_{\theta^{*}}\left(x_{1:T}^{*}\middle|y_{1:T}\right).$$

- The MH acceptance probability is

$$1\wedge\frac{p\left(\theta^{*},x_{1:T}^{*}\middle|y_{1:T}\right)}{p\left(\theta,x_{1:T}\middle|y_{1:T}\right)}\frac{q\left(\left(x_{1:T},\theta\right)\middle|\left(x_{1:T}^{*},\theta^{*}\right)\right)}{q\left(\left(x_{1:T}^{*},\theta^{*}\right)\middle|\left(x_{1:T},\theta\right)\right)}$$

$$=1\wedge\frac{p_{\theta^{*}}\left(y_{1:T}\right)p\left(\theta^{*}\right)}{p_{\theta}\left(y_{1:T}\right)p\left(\theta\right)}\frac{q\left(\theta\middle|\theta^{*}\right)}{q\left(\theta^{*}\middle|\theta\right)}$$

"Idealized" Marginal MH Sampler

- Consider the following so-called marginal Metropolis-Hastings (MH) algorithm which uses as a proposal

$$q\left(\left(x^*_{1:T},\theta^*\right)\middle|\left(x_{1:T},\theta\right)\right)=q\left(\theta^*\middle|\theta\right)p_{\theta^*}\left(x^*_{1:T}\middle|y_{1:T}\right).$$

- The MH acceptance probability is

$$1\wedge\frac{p\left(\theta^*,x^*_{1:T}\middle|y_{1:T}\right)}{p\left(\theta,x_{1:T}\middle|y_{1:T}\right)}\frac{q\left(\left(x_{1:T},\theta\right)\middle|\left(x^*_{1:T},\theta^*\right)\right)}{q\left(\left(x^*_{1:T},\theta^*\right)\middle|\left(x_{1:T},\theta\right)\right)}$$
$$=1\wedge\frac{p_{\theta^*}\left(y_{1:T}\right)p\left(\theta^*\right)}{p_{\theta}\left(y_{1:T}\right)p\left(\theta\right)}\frac{q\left(\theta\middle|\theta^*\right)}{q\left(\theta^*\middle|\theta\right)}$$

- In this MH algorithm, $X_{1:T}$ has been essentially integrated out.

Implementation Issues

- **Problem 1**: We do not know $p_\theta\left(y_{1:T}\right) = \int p_\theta\left(x_{1:T}, y_{1:T}\right) dx_{1:T}$ analytically.

Implementation Issues

- **Problem 1**: We do not know $p_\theta(y_{1:T}) = \int p_\theta(x_{1:T}, y_{1:T})\, dx_{1:T}$ analytically.
- **Problem 2:** We do not know how to sample from $p_\theta(x_{1:T} | y_{1:T})$.

Implementation Issues

- **Problem 1**: We do not know $p_\theta(y_{1:T}) = \int p_\theta(x_{1:T}, y_{1:T})\, dx_{1:T}$ analytically.
- **Problem 2:** We do not know how to sample from $p_\theta(x_{1:T} | y_{1:T})$.
- **"Idea"**: Use SMC approximations of $p_\theta(x_{1:T} | y_{1:T})$ and $p_\theta(y_{1:T})$.

- Given θ, SMC methods provide approximations of $p_\theta(x_{1:T} | y_{1:T})$ and $p_\theta(y_{1:T})$.

- Given θ, SMC methods provide approximations of $p_\theta\left(x_{1:T} \mid y_{1:T}\right)$ and $p_\theta\left(y_{1:T}\right)$.
- At time T, we obtain the following approximation of the posterior of interest

$$\widehat{p}_\theta\left(x_{1:T} \mid y_{1:T}\right) = \frac{1}{N} \sum_{k=1}^{N} \delta_{X_{1:T}^{(k)}}\left(x_{1:T}\right)$$

and an approximation of $p_\theta\left(y_{1:T}\right)$ is given by

$$\widehat{p}_\theta\left(y_{1:T}\right) = \widehat{p}_\theta\left(y_1\right) \prod_{t=2}^{T} \widehat{p}_\theta\left(y_t \mid y_{1:t-1}\right) = \prod_{t=1}^{T} \left(\frac{1}{N} \sum_{k=1}^{N} g_\theta\left(y_t \mid X_t^{(k)}\right) \right)$$

if we use $f_\theta\left(x_t \mid x_{t-1}\right)$ as a proposal.

Reminder...

- Under *mixing assumptions*, we have

$$\frac{\mathbb{V}\left[\widehat{p}_{\theta}\left(y_{1:T}\right)\right]}{p_{\theta}^{2}\left(y_{1:T}\right)} \leq D_{\theta}\frac{T}{N}.$$

Reminder...

- Under *mixing assumptions*, we have

$$\frac{\mathbb{V}\left[\widehat{p}_{\theta}\left(y_{1:T}\right)\right]}{p_{\theta}^{2}\left(y_{1:T}\right)} \leq D_{\theta}\frac{T}{N}.$$

- Under *mixing assumptions*, we also have

$$\int \left|\mathbb{E}\left[\widehat{p}_{\theta}\left(x_{1:T}\right| y_{1:T}\right)\right] - p_{\theta}\left(x_{1:T}\right| y_{1:T}\left.\right)\right| dx_{1:T} \leq C_{\theta}\frac{T}{N}$$

so if I run an SMC method to obtain $\widehat{p}_{\theta}\left(x_{1:T}\middle| y_{1:T}\right)$ then $X_{1:T} \sim \widehat{p}_{\theta}\left(x_{1:T}\middle| y_{1:T}\right)$, unconditionally $X_{1:T} \sim \mathbb{E}\left[\widehat{p}_{\theta}\left(\cdot\middle| y_{1:T}\right)\right]$.

Reminder...

- Under *mixing assumptions*, we have

$$\frac{\mathbb{V}\left[\widehat{p}_\theta\left(y_{1:T}\right)\right]}{p_\theta^2\left(y_{1:T}\right)} \leq D_\theta \frac{T}{N}.$$

- Under *mixing assumptions*, we also have

$$\int \left|\mathbb{E}\left[\widehat{p}_\theta\left(x_{1:T} \middle| y_{1:T}\right)\right] - p_\theta\left(x_{1:T} \middle| y_{1:T}\right)\right| dx_{1:T} \leq C_\theta \frac{T}{N}$$

  so if I run an SMC method to obtain $\widehat{p}_\theta\left(x_{1:T} \middle| y_{1:T}\right)$ then $X_{1:T} \sim \widehat{p}_\theta\left(x_{1:T} \middle| y_{1:T}\right)$, unconditionally $X_{1:T} \sim \mathbb{E}\left[\widehat{p}_\theta\left(\cdot \middle| y_{1:T}\right)\right]$.
- **Problem**: We cannot compute analytically the particle filter proposal $q_\theta\left(x_{1:T} \middle| y_{1:T}\right) = \mathbb{E}\left[\widehat{p}_\theta\left(x_{1:T} \middle| y_{1:T}\right)\right]$ as it involves an expectation w.r.t all the variables appearing in the particle algorithm...

"Idealized" Marginal MH Sampler

At iteration i

- Sample $\theta^* \sim q\left(\theta \middle| \theta\left(i-1\right)\right)$.

“Idealized” Marginal MH Sampler

At iteration i

- Sample $\theta^* \sim q(\theta | \theta(i-1))$.
- Sample $X^*_{1:T} \sim p_{\theta^*}(x_{1:T} | y_{1:T})$.

"Idealized" Marginal MH Sampler

At iteration i

- Sample $\theta^* \sim q\left(\theta | \theta(i-1)\right)$.
- Sample $X_{1:T}^* \sim p_{\theta^*}\left(x_{1:T} | y_{1:T}\right)$.
- With probability

$$1 \wedge \frac{p_{\theta^*}\left(y_{1:T}\right) p\left(\theta^*\right)}{p_{\theta(i-1)}\left(y_{1:T}\right) p\left(\theta(i-1)\right)} \frac{q\left(\theta(i-1) | \theta^*\right)}{q\left(\theta^* | \theta(i-1)\right)}$$

set $\theta(i) = \theta^*$, $X_{1:T}(i) = X_{1:T}^*$ otherwise set $\theta(i) = \theta(i-1)$, $X_{1:T}(i) = X_{1:T}(i-1)$.

Particle Marginal MH Sampler

At iteration i

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

Particle Marginal MH Sampler

At iteration i

- Sample $\theta^{*} \sim q\left(\theta \mid \theta\left(i-1\right)\right)$ and run an SMC algorithm to obtain $\widehat{p}_{\theta^{*}}\left(x_{1:T} \mid y_{1:T}\right)$ and $\widehat{p}_{\theta^{*}}\left(y_{1:T}\right)$.
- Sample $X_{1:T}^{*} \sim \widehat{p}_{\theta^{*}}\left(x_{1:T} \mid y_{1:T}\right)$.

Particle Marginal MH Sampler

At iteration i

- Sample $\theta^* \sim q\left(\theta | \theta(i-1)\right)$ and run an SMC algorithm to obtain $\widehat{p}_{\theta^*}\left(x_{1:T} | y_{1:T}\right)$ and $\widehat{p}_{\theta^*}\left(y_{1:T}\right)$.
- Sample $X^*_{1:T} \sim \widehat{p}_{\theta^*}\left(x_{1:T} | y_{1:T}\right)$.
- With probability

$$1 \wedge \frac{\widehat{p}_{\theta^*}\left(y_{1:T}\right) p\left(\theta^*\right)}{\widehat{p}_{\theta(i-1)}\left(y_{1:T}\right) p\left(\theta(i-1)\right)} \frac{q\left(\theta(i-1) | \theta^*\right)}{q\left(\theta^* | \theta(i-1)\right)}$$

set $\theta(i) = \theta^*$, $X_{1:T}(i) = X^*_{1:T}$ otherwise set $\theta(i) = \theta(i-1)$, $X_{1:T}(i) = X_{1:T}(i-1)$.

Validity of the Particle Marginal MH Sampler

- **Proposition**. Assume that the 'idealized' marginal MH sampler chain is ergodic then, under very weak assumptions, the PMMH sampler chain is ergodic and admits $p(\theta, x_{1:T} \mid y_{1:T})$ **whatever being** $N \geq 1$.

Validity of the Particle Marginal MH Sampler

- **Proposition**. Assume that the 'idealized' marginal MH sampler chain is ergodic then, under very weak assumptions, the PMMH sampler chain is ergodic and admits $p(\theta, x_{1:T} | y_{1:T})$ **whatever being** $N \geq 1$.
- It is easy to show the simpler result that the PMMH admits $p(\theta | y_{1:T})$ as invariant distribution **whatever being** $N \geq 1$.

Validity of the Particle Marginal MH Sampler

- **Proposition**. Assume that the 'idealized' marginal MH sampler chain is ergodic then, under very weak assumptions, the PMMH sampler chain is ergodic and admits $p(\theta, x_{1:T} | y_{1:T})$ **whatever being** $N \geq 1$.
- It is easy to show the simpler result that the PMMH admits $p(\theta | y_{1:T})$ as invariant distribution **whatever being** $N \geq 1$.
- Let U denote all the r.v. introduce to build the SMC estimate then write $\widehat{p}_\theta(y_{1:T}) = \widehat{p}_\theta(y_{1:T}, U)$ and from unbiasedness

$$\int \widehat{p}_\theta(y_{1:T}, u)\, q_\theta(u)\, du = p_\theta(y_{1:T}).$$

An Incomplete But Trivial Proof

- The PMMH targets the distribution

$$\widetilde{\pi}\left(\theta, u\right) \propto p\left(\theta\right) \widehat{p}_{\theta}\left(y_{1:T}, u\right) q_{\theta}\left(u\right)$$

which satisfies

$$\widetilde{\pi}\left(\theta\right) = p(\theta | y_{1:T}).$$

An Incomplete But Trivial Proof

- The PMMH targets the distribution

$$\widetilde{\pi}\left(\theta, u\right) \propto p\left(\theta\right) \widehat{p}_{\theta}\left(y_{1:T}, u\right) q_{\theta}\left(u\right)$$

which satisfies

$$\widetilde{\pi}\left(\theta\right) = p(\theta | y_{1:T}).$$

- The PMMH sampler uses as a proposal

$$q\left(\left(\theta^{*}, u^{*}\right) \middle| \left(\theta, u\right)\right) = q\left(\theta^{*} \middle| \theta\right) q_{\theta^{*}}\left(u^{*}\right)$$

and

$$\begin{aligned}\frac{\widetilde{\pi}(\theta^{*},u^{*})}{\widetilde{\pi}(\theta,u)} \frac{q(\,(\theta,u)|(\theta^{*},u^{*}))}{q(\,(\theta^{*},u^{*})|(\theta,u))} &= \frac{p(\theta^{*})\widehat{p}_{\theta^{*}}(y_{1:T},u^{*})q_{\theta^{*}}(u^{*})}{p(\theta)\widehat{p}_{\theta}(y_{1:T},u)q_{\theta}(u)} \frac{q(\,\theta|\theta^{*})q_{\theta}(u)}{q(\,\theta^{*}|\theta)q_{\theta^{*}}(u^{*})} \\ &= \frac{p(\theta^{*})\widehat{p}_{\theta^{*}}(y_{1:T},u^{*})}{p(\theta)\widehat{p}_{\theta}(y_{1:T},u)} \frac{q(\,\theta|\theta^{*})}{q(\,\theta^{*}|\theta)}\end{aligned}$$

An Incomplete But Trivial Proof

- The PMMH targets the distribution

$$\widetilde{\pi}\left(\theta, u\right) \propto p\left(\theta\right) \widehat{p}_{\theta}\left(y_{1:T}, u\right) q_{\theta}\left(u\right)$$

which satisfies

$$\widetilde{\pi}\left(\theta\right) = p(\theta | \, y_{1:T}).$$

- The PMMH sampler uses as a proposal

$$q\left(\left(\theta^{*}, u^{*}\right) \middle| \left(\theta, u\right)\right) = q\left(\theta^{*} \middle| \theta\right) q_{\theta^{*}}\left(u^{*}\right)$$

and

$$\begin{aligned} \frac{\widetilde{\pi}(\theta^{*},u^{*})}{\widetilde{\pi}(\theta,u)} \frac{q(\,(\theta,u)|(\theta^{*},u^{*}))}{q(\,(\theta^{*},u^{*})|(\theta,u))} &= \frac{p(\theta^{*})\widehat{p}_{\theta^{*}}(y_{1:T},u^{*})q_{\theta^{*}}(u^{*})}{p(\theta)\widehat{p}_{\theta}(y_{1:T},u)q_{\theta}(u)} \frac{q(\,\theta|\theta^{*})q_{\theta}(u)}{q(\,\theta^{*}|\theta)q_{\theta^{*}}(u^{*})} \\ &= \frac{p(\theta^{*})\widehat{p}_{\theta^{*}}(y_{1:T},u^{*})}{p(\theta)\widehat{p}_{\theta}(y_{1:T},u)} \frac{q(\,\theta|\theta^{*})}{q(\,\theta^{*}|\theta)} \end{aligned}$$

- **Trivial but deep result:** if you plug any unbiased likelihood estimate within a MCMC scheme, you do not perturb the invariant distribution.

Explicit Structure of the Target Distribution

- Let first consider the case where $T = 1$.

Explicit Structure of the Target Distribution

- Let first consider the case where $T = 1$.
- *Proposal distribution*

$$\widetilde{q}\left(\left(\theta^{*}, k, x_1^{(1:N)}\right) \middle| \theta\right) = q\left(\theta^{*} \middle| \theta\right) \underbrace{\prod_{m=1}^{N} \mu_{\theta^{*}}\left(x_1^{(m)}\right) \;\; W_1^{(k)}}_{q_{\theta^{*}}(u)}$$

Explicit Structure of the Target Distribution

- Let first consider the case where $T = 1$.
- *Proposal distribution*

$$\widetilde{q}\left(\left(\theta^{*}, k, x_{1}^{(1:N)}\right)\middle|\,\theta\right) = q\left(\theta^{*}\middle|\,\theta\right) \underbrace{\prod_{m=1}^{N} \mu_{\theta^{*}}\left(x_{1}^{(m)}\right) \; W_{1}^{(k)}}_{q_{\theta^{*}}(u)}$$

- *Target distribution*

$$\tilde{\pi}\left(\theta, k, x_{1}^{(1:N)}\right) \propto \; p(\theta) \underbrace{\frac{1}{N}\sum_{m=1}^{N} g_{\theta}\left(y_{1}\middle|\, x_{1}^{(m)}\right)}_{\widehat{p}_{\theta}(y_{1})} \prod_{m=1}^{N} \mu_{\theta}\left(x_{1}^{(m)}\right) \; W_{1}^{(k)}$$

Explicit Structure of the Target Distribution

- Let first consider the case where $T = 1$.
- *Proposal distribution*

$$\widetilde{q}\left(\left(\theta^{*}, k, x_{1}^{(1:N)}\right) \middle| \theta\right) = q\left(\theta^{*} \middle| \theta\right) \underbrace{\prod_{m=1}^{N} \mu_{\theta^{*}}\left(x_{1}^{(m)}\right) \; W_{1}^{(k)}}_{q_{\theta^{*}}(u)}$$

- *Target distribution*

$$\tilde{\pi}\left(\theta, k, x_{1}^{(1:N)}\right) \propto \; p\left(\theta\right) \underbrace{\frac{1}{N} \sum_{m=1}^{N} g_{\theta}\left(y_{1} \middle| x_{1}^{(m)}\right)}_{\widehat{p}_{\theta}(y_{1})} \prod_{m=1}^{N} \mu_{\theta}\left(x_{1}^{(m)}\right) \; W_{1}^{(k)}$$

- We have already shown

$$\frac{\tilde{\pi}\left(\theta^{*}, k, x_{1}^{(1:N)}\right)}{\widetilde{q}^{N}\left(\left(\theta^{*}, k, x_{1}^{(1:N)}\right) \middle| \theta\right)} = \frac{p\left(\theta^{*}\right)}{q\left(\theta^{*} \middle| \theta\right)} \frac{\widehat{p}_{\theta^{*}}\left(y_{1}\right)}{p_{\theta^{*}}\left(y_{1}\right)}$$

Explicit Structure of the Target Distribution

- The target is given by

$$\tilde{\pi}\left(\theta, k, x_1^{(1:N)}\right) \propto p\left(\theta\right) \left(\sum_{m=1}^{N} g_\theta\left(y_1 \middle| x_1^{(m)}\right)\right) \prod_{m=1}^{N} \mu_\theta\left(x_1^{(m)}\right) W_1^{(k)}$$

but $W_1^{(k)} = g_\theta\left(y_1 \middle| x_1^{(k)}\right) / \left(\sum_{m=1}^{N} g_\theta\left(y_1 \middle| x_1^{(m)}\right)\right)$.

Explicit Structure of the Target Distribution

- The target is given by

$$\tilde{\pi}\left(\theta, k, x_1^{(1:N)}\right) \propto p(\theta) \left(\sum_{m=1}^{N} g_\theta\left(y_1 \middle| x_1^{(m)}\right)\right) \prod_{m=1}^{N} \mu_\theta\left(x_1^{(m)}\right) W_1^{(k)}$$

but $W_1^{(k)} = g_\theta\left(y_1 \middle| x_1^{(k)}\right) / \left(\sum_{m=1}^{N} g_\theta\left(y_1 \middle| x_1^{(m)}\right)\right)$.

- Hence, we can actually rewrite the target as

$$\tilde{\pi}^N\left(\theta, k, x_1^{(1:N)}\right) = \frac{p\left(\theta, x_1^{(k)} \middle| y_1\right)}{N} \prod_{m=1; m \neq k}^{N} \mu_\theta\left(x_1^{(m)}\right).$$

Explicit Structure of the Target Distribution

- The target is given by

$$\tilde{\pi}\left(\theta, k, x_1^{(1:N)}\right) \propto p\left(\theta\right) \left(\sum_{m=1}^{N} g_\theta\left(y_1 \middle| x_1^{(m)}\right)\right) \prod_{m=1}^{N} \mu_\theta\left(x_1^{(m)}\right) \; W_1^{(k)}$$

but $W_1^{(k)} = g_\theta\left(y_1 \middle| x_1^{(k)}\right) / \left(\sum_{m=1}^{N} g_\theta\left(y_1 \middle| x_1^{(m)}\right)\right)$.

- Hence, we can actually rewrite the target as

$$\tilde{\pi}^N\left(\theta, k, x_1^{(1:N)}\right) = \frac{p\left(\theta, x_1^{(k)} \middle| y_1\right)}{N} \prod_{m=1; m \neq k}^{N} \mu_\theta\left(x_1^{(m)}\right).$$

- This shows that we are able to sample from $p\left(\theta, x_1 \middle| y_1\right)$ and not only its marginal $p\left(\theta \middle| y_1\right)$.

Sampling from the Target Distribution

- To sample from this target distribution

Sampling from the Target Distribution

- To sample from this target distribution
  - Sample K from a uniform distribution on $\{1, ..., N\}$.

Sampling from the Target Distribution

- To sample from this target distribution
  - Sample K from a uniform distribution on $\{1, ..., N\}$.
  - Sample $\left(\theta, X_1^{(K)}\right)$ from $p\left(\theta, x_1 \mid y_1\right)$. (We do not know how to do this, this is why we use MCMC).

Sampling from the Target Distribution

- To sample from this target distribution
  - Sample K from a uniform distribution on $\{1, ..., N\}$.
  - Sample $\left(\theta, X_1^{(K)}\right)$ from $p\left(\theta, x_1 \mid y_1\right)$. (We do not know how to do this, this is why we use MCMC).
- Sample $X_1^{(m)} \sim \mu_\theta\left(x_1\right)$ for $m \neq K$.

Explicit Structure of the Target Distribution

- This construction can be extended to the case $T > 1$.

Explicit Structure of the Target Distribution

- This construction can be extended to the case $T > 1$.
- To sample from this target distribution

Explicit Structure of the Target Distribution

- This construction can be extended to the case $T > 1$.
- To sample from this target distribution
  - Sample indexes from a uniform distribution on $\{1, ..., N\}^T$ corresponding to an ancestral line.

Explicit Structure of the Target Distribution

- This construction can be extended to the case $T > 1$.
- To sample from this target distribution
  - Sample indexes from a uniform distribution on $\{1, ..., N\}^T$ corresponding to an ancestral line.
  - Sample θ and $X_{1:T}$ for this ancestral line from $p(\theta, x_{1:T} | y_{1:T})$. (We do not know how to do this, this is why we use MCMC).

Explicit Structure of the Target Distribution

- This construction can be extended to the case $T > 1$.
- To sample from this target distribution
  - Sample indexes from a uniform distribution on $\{1, ..., N\}^T$ corresponding to an ancestral line.
  - Sample θ and $X_{1:T}$ for this ancestral line from $p(\theta, x_{1:T} | y_{1:T})$. (We do not know how to do this, this is why we use MCMC).
- Run a conditional SMC algorithm compatible with $X_{1:T}$ and its ancestral lineage; see (Andrieu, D. & Holenstein, 2010).

Conditional SMC

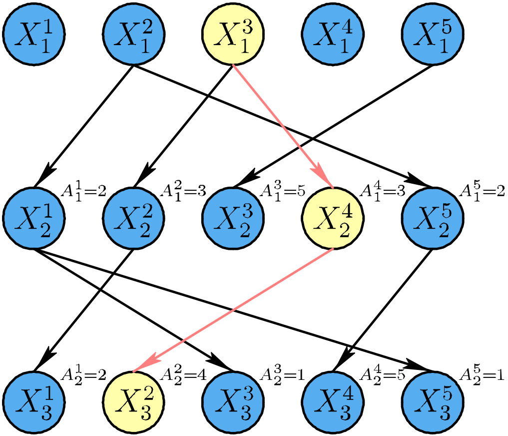


Figure: Example of $N-1=4$ ancestral lineages generated by a conditional SMC algorithm for $N=5$, $T=3$ conditional upon $X_{1:3}^2$ and $B_{1:3}^2$

"Idealized" Gibbs Sampler

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

At iteration i

"Idealized" Gibbs Sampler

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

At iteration i

- Sample $X_{1:T}\left(i\right) \sim p_{\theta(i-1)}\left(x_{1:T} \mid y_{1:T}\right)$.

"Idealized" Gibbs Sampler

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

At iteration i

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

“Idealized” Gibbs Sampler

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

At iteration i

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

- **Problem**: We do not know how to sample from $p_{\theta}\left(x_{1:T} \middle| y_{1:T}\right)$.

"Idealized" Gibbs Sampler

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

<u>*At iteration i*</u>

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

- **Problem**: We do not know how to sample from $p_{\theta}(x_{1:T} | y_{1:T})$.
- Naive particle approximation where $X_{1:T}(i) \sim \widehat{p}(x_{1:T} | y_{1:T}, \theta(i))$ is substituted to $X_{1:T}(i) \sim p(x_{1:T} | y_{1:T}, \theta(i))$ is obviously incorrect.

Particle Gibbs Sampler

At iteration i

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

Particle Gibbs Sampler

At iteration i

- Sample $\theta(i) \sim p(\theta | y_{1:T}, X_{1:T}(i-1))$.
- Run a conditional SMC algorithm for $\theta(i)$ consistent with $X_{1:T}(i-1)$ and its ancestral lineage.

Particle Gibbs Sampler

At iteration i

- Sample $\theta(i) \sim p(\theta | y_{1:T}, X_{1:T}(i-1))$.
- Run a conditional SMC algorithm for $\theta(i)$ consistent with $X_{1:T}(i-1)$ and its ancestral lineage.
- Sample $X_{1:T}(i) \sim \widehat{p}(x_{1:T} | y_{1:T}, \theta(i))$ from the resulting approximation (hence its ancestral lineage too).

Particle Gibbs Sampler

At iteration i

- Sample $\theta(i) \sim p(\theta | y_{1:T}, X_{1:T}(i-1))$.
- Run a conditional SMC algorithm for $\theta(i)$ consistent with $X_{1:T}(i-1)$ and its ancestral lineage.
- Sample $X_{1:T}(i) \sim \widehat{p}(x_{1:T} | y_{1:T}, \theta(i))$ from the resulting approximation (hence its ancestral lineage too).

- **Proposition**. Assume that the 'ideal' Gibbs sampler chain is ergodic then under very weak assumptions the particle Gibbs sampler chain is ergodic and admits $p(\theta, x_{1:T} | y_{1:T})$ as an invariant distribution **for any** $N \geq 2$.

Nonlinear State-Space Model

- Consider the following model

$$
\begin{aligned}
X_t &= \frac{1}{2}X_{t-1} + 25\frac{X_{t-1}}{1+X_{t-1}^2} + 8\cos 1.2t + V_t, \\
Y_t &= \frac{X_t^2}{20} + W_t
\end{aligned}
$$

where $V_t \sim \mathcal{N}\left(0, \sigma_v^2\right)$, $W_t \sim \mathcal{N}\left(0, \sigma_w^2\right)$ and $X_1 \sim \mathcal{N}\left(0, 5^2\right)$.

Nonlinear State-Space Model

- Consider the following model

$$X_t = \frac{1}{2}X_{t-1} + 25\frac{X_{t-1}}{1+X_{t-1}^2} + 8\cos 1.2t + V_t,$$
$$Y_t = \frac{X_t^2}{20} + W_t$$

where $V_t \sim \mathcal{N}\left(0, \sigma_v^2\right)$, $W_t \sim \mathcal{N}\left(0, \sigma_w^2\right)$ and $X_1 \sim \mathcal{N}\left(0, 5^2\right)$.

- Use the prior for $\{X_t\}$ as proposal distribution.

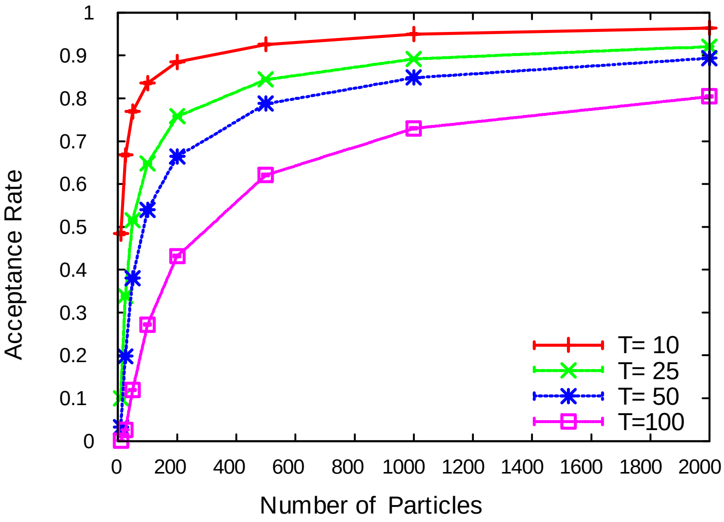
Nonlinear State-Space Model

- Consider the following model

$$
\begin{aligned}
X_t &= \frac{1}{2}X_{t-1} + 25\frac{X_{t-1}}{1+X_{t-1}^2} + 8\cos 1.2t + V_t, \\
Y_t &= \frac{X_t^2}{20} + W_t
\end{aligned}
$$

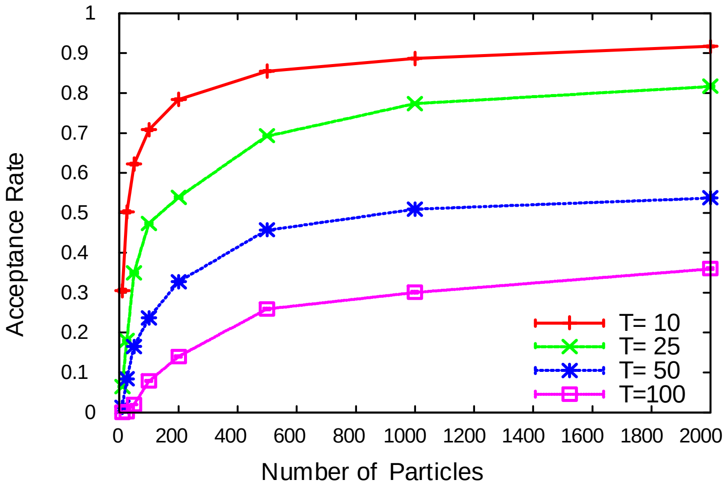
  where $V_t \sim \mathcal{N}\left(0, \sigma_v^2\right)$, $W_t \sim \mathcal{N}\left(0, \sigma_w^2\right)$ and $X_1 \sim \mathcal{N}\left(0, 5^2\right)$.
- Use the prior for $\{X_t\}$ as proposal distribution.
- For a fixed θ, we evaluate the expected acceptance probability as a function of N.

Average Acceptance Probability



Average acceptance probability when $\sigma_v^2 = \sigma_w^2 = 10$

Average Acceptance Probability



Average acceptance probability when $\sigma_v^2 = 10$, $\sigma_w^2 = 1$

- Two species X_t^1 (prey) and X_t^2 (predator)

$$\begin{aligned}
\Pr\left(X_{t+dt}^1=x_t^1+1, X_{t+dt}^2=x_t^2 \,\middle|\, x_t^1, x_t^2\right) &= \alpha\, x_t^1\, dt + o\left(dt\right), \\
\Pr\left(X_{t+dt}^1=x_t^1-1, X_{t+dt}^2=x_t^2+1 \,\middle|\, x_t^1, x_t^2\right) &= \beta\, x_t^1\, x_t^2\, dt + o\left(dt\right), \\
\Pr\left(X_{t+dt}^1=x_t^1, X_{t+dt}^2=x_t^2-1 \,\middle|\, x_t^1, x_t^2\right) &= \gamma\, x_t^2\, dt + o\left(dt\right),
\end{aligned}$$

with

$$Y_k = X_{k\Delta T}^1 + W_k \text{ with } W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right).$$

- Two species X_t^1 (prey) and X_t^2 (predator)

$$\begin{aligned}
&\Pr\left(X_{t+dt}^1{=}x_t^1{+}1, X_{t+dt}^2{=}x_t^2 \,\middle|\, x_t^1, x_t^2\right) = \alpha\, x_t^1\, dt + o\,(dt)\,,\\
&\Pr\left(X_{t+dt}^1{=}x_t^1{-}1, X_{t+dt}^2{=}x_t^2{+}1 \,\middle|\, x_t^1, x_t^2\right) = \beta\, x_t^1\, x_t^2\, dt + o\,(dt)\,,\\
&\Pr\left(X_{t+dt}^1{=}x_t^1, X_{t+dt}^2{=}x_t^2{-}1 \,\middle|\, x_t^1, x_t^2\right) = \gamma\, x_t^2\, dt + o\,(dt)\,,
\end{aligned}$$

with

$$Y_k = X_{k\Delta T}^1 + W_k \text{ with } W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right).$$

- We are interested in the kinetic rate constants $\theta = (\alpha, \beta, \gamma)$ a priori distributed as (Boys et al., 2008; Kunsch, 2011)

$$\alpha \sim \mathcal{G}(1, 10), \quad \beta \sim \mathcal{G}(1, 0.25), \quad \gamma \sim \mathcal{G}(1, 7.5).$$

- Two species X_t^1 (prey) and X_t^2 (predator)

$$\Pr\left(X_{t+dt}^1{=}x_t^1{+}1, X_{t+dt}^2{=}x_t^2 \,\middle|\, x_t^1, x_t^2\right) = \alpha\, x_t^1\, dt + o\left(dt\right),$$
$$\Pr\left(X_{t+dt}^1{=}x_t^1{-}1, X_{t+dt}^2{=}x_t^2{+}1 \,\middle|\, x_t^1, x_t^2\right) = \beta\, x_t^1\, x_t^2\, dt + o\left(dt\right),$$
$$\Pr\left(X_{t+dt}^1{=}x_t^1, X_{t+dt}^2{=}x_t^2{-}1 \,\middle|\, x_t^1, x_t^2\right) = \gamma\, x_t^2\, dt + o\left(dt\right),$$

with

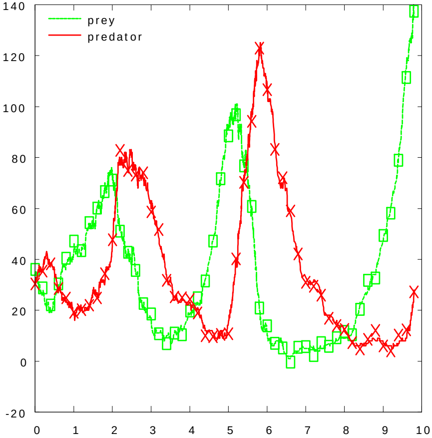
$$Y_k = X_{k\Delta T}^1 + W_k \text{ with } W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right).$$

- We are interested in the kinetic rate constants $\theta = (\alpha, \beta, \gamma)$ a priori distributed as (Boys et al., 2008; Kunsch, 2011)

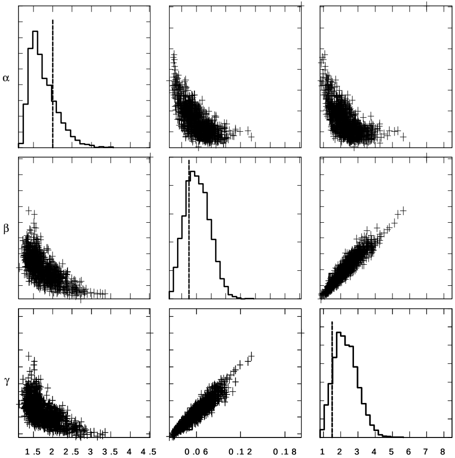
$$\alpha \sim \mathcal{G}(1, 10), \quad \beta \sim \mathcal{G}(1, 0.25), \quad \gamma \sim \mathcal{G}(1, 7.5).$$

- MCMC methods require reversible jumps, Particle MCMC requires only forward simulation.

Experimental Results

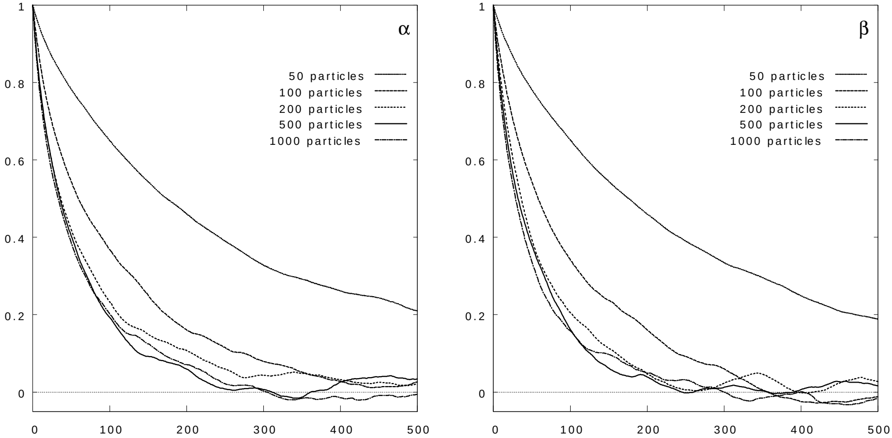


Simulated data



Posterior distributions

Autocorrelation Functions



Autocorrelation of α (left) and β (right) for the PMMH sampler for various N.

Summary

- Offline Bayesian parameter inference is feasible by using SMC proposals within MCMC.

Summary

- Offline Bayesian parameter inference is feasible by using SMC proposals within MCMC.
- This approach does not suffer from degeneracy problem and N scales roughly linearly with T.

Summary

- Offline Bayesian parameter inference is feasible by using SMC proposals within MCMC.
- This approach does not suffer from degeneracy problem and N scales roughly linearly with T.
- Particle MCMC allow us to perform Bayesian inference for dynamic models for which only forward simulation is possible.

Summary

- Offline Bayesian parameter inference is feasible by using SMC proposals within MCMC.
- This approach does not suffer from degeneracy problem and N scales roughly linearly with T.
- Particle MCMC allow us to perform Bayesian inference for dynamic models for which only forward simulation is possible.
- Computationally intensive but several implementations on GPU already available and applications in control, ecology, econometrics, biochemical systems, epidemiology, water resources research etc.

Summary

- Offline Bayesian parameter inference is feasible by using SMC proposals within MCMC.
- This approach does not suffer from degeneracy problem and N scales roughly linearly with T.
- Particle MCMC allow us to perform Bayesian inference for dynamic models for which only forward simulation is possible.
- Computationally intensive but several implementations on GPU already available and applications in control, ecology, econometrics, biochemical systems, epidemiology, water resources research etc.
- Selection of N is a key issue and some guidelines are available (Lee, Andrieu & D., 2012), (D., Pitt & Kohn, 2012).