A (gentle) introduction to particle filters

nicolas.chopin@ensae.fr
(based on a forthcoming book with Omiros Papaspiliopoulos)

(Mis)conceptions about particle filters

- Something useful only for very specific models (hidden Markov models, state-space models);
- Or alternatively something as versatile as MCMC

- Something useful only for very specific models (hidden Markov models, state-space models);
- Or alternatively something as versatile as MCMC

Which one it is?

Quick look

Let's have a quick look at a particle filter.

Structure

Algorithm 0.1: Basic PF algorithm

Operations involving index n must be performed for all $n \in 1:N$.

At time 0:

(a) Generate $X_0^n \sim M_0(\mathrm{d}x_0)$.

(b) Compute $w_0^n = G_0(X_0^n)$, and $W_0^n = w_0^n / \sum_{m=1}^N w_0^m$.

Recursively, for $t = 1, \ldots, T$:

(a) Generate ancestor variables $A_t^n \in 1:N$ independently from $\mathcal{M}(W_{t-1}^{1:N})$.

(b) Generate $X_t^n \sim M_t(X_{t-1}^{A_t^n}, \mathrm{d}x_t)$.

(c) Compute $w_t^n = G_t(X_{t-1}^{A_t^n}, X_t^n)$, and $W_t^n = w_t^n / \sum_{m=1}^N w_t^m$.

Comments

- All particle filters have (essentially) this structure. (Let's ignore variations based on alternative resampling schemes, etc.)

Comments

- All particle filters have (essentially) this structure. (Let's ignore variations based on alternative resampling schemes, etc.)
- The user must specify:
  - kernel $M_t(x_{t-1}, \mathrm{d}x_t)$: that's how we simulate particle X_t^n, given a certain ancestor $X_{t-1}^{A_t^n}$;
  - Function $G_t(x_{t-1}, x_t)$; that's how we reweight/grade particle X_t^n (and its ancestor).

Comments

- All particle filters have (essentially) this structure. (Let's ignore variations based on alternative resampling schemes, etc.)
- The user must specify:
  - kernel $M_t(x_{t-1}, \mathrm{d}x_t)$: that's how we simulate particle X_t^n, given a certain ancestor $X_{t-1}^{A_t^n}$;
  - Function $G_t(x_{t-1}, x_t)$; that's how we reweight/grade particle X_t^n (and its ancestor).
- Easy part: **How**. Less easy: **Why**

State-space models

nicolas.chopin@ensae.fr
(based on a forthcoming book with Omiros Papaspiliopoulos)

Outline

Objectives

The aim of this chapter is to define state-space models, give examples of such models from various areas of science, and discuss their main properties.

A first definition (with functions)

A time series model that consists of two discrete-time processes $\{X_t\} := (X_t)_{t \geq 0}$, $\{Y_t\} := (Y_t)_{t \geq 0}$, taking values respectively in spaces $\mathcal{X}$ and $\mathcal{Y}$, such that

$$
\begin{aligned}
X_t &= K_t(X_{t-1}, U_t, \theta), \quad t \geq 1 \\
Y_t &= H_t(X_t, V_t, \theta), \quad t \geq 0
\end{aligned}
$$

where K_0, K_t, H_t, are determistic functions, $\{U_t\}$, $\{V_t\}$ are sequences of i.i.d. random variables (*noises*, or *shocks*), and $\theta \in \Theta$ is an unknown parameter.

A first definition (with functions)

A time series model that consists of two discrete-time processes $\{X_t\}:=(X_t)_{t\geq 0}$, $\{Y_t\}:=(Y_t)_{t\geq 0}$, taking values respectively in spaces $\mathcal{X}$ and $\mathcal{Y}$, such that

$$
\begin{aligned}
X_t &= K_t(X_{t-1}, U_t, \theta), \quad t \geq 1 \\
Y_t &= H_t(X_t, V_t, \theta), \quad t \geq 0
\end{aligned}
$$

where K_0, K_t, H_t, are determistic functions, $\{U_t\}$, $\{V_t\}$ are sequences of i.i.d. random variables (*noises*, or *shocks*), and $\theta \in \Theta$ is an unknown parameter.

This is a popular way to define SSMs in Engineering. Rigorous, but not sufficiently general.

A second definition (with densities)

$$
\begin{aligned}
p_\theta(x_0) &= p_0^\theta(x_0) \\
p_\theta(x_t|x_{0:t-1}) &= p_t^\theta(x_t|x_{t-1}) \quad t \geq 1 \\
p_\theta(y_t|x_{0:t}, y_{0:t-1}) &= f_t^\theta(y_t|x_t)
\end{aligned}
\tag{0.1}
$$

A second definition (with densities)

$$
\begin{aligned}
p_\theta(x_0) &= p_0^\theta(x_0) \\
p_\theta(x_t|x_{0:t-1}) &= p_t^\theta(x_t|x_{t-1}) \quad t \geq 1 \\
p_\theta(y_t|x_{0:t}, y_{0:t-1}) &= f_t^\theta(y_t|x_t)
\end{aligned} \tag{0.1}
$$

Not so rigorous (or not general enough): some models are such that $X_t|X_{t-1}$ does not admit a probability density (with respect to a fixed dominating measure).

Outline

Signal processing: tracking, positioning, navigation

X_t is position of a moving object, e.g.

$$X_t = X_{t-1} + U_t, \quad U_t \sim \mathcal{N}_2(0, \sigma^2 I_2),$$

and Y_t is a measurement obtained by e.g. a radar,

$$Y_t = \mathrm{atan}\left(\frac{X_t(2)}{X_t(1)}\right) + V_t, \quad V_t \sim \mathcal{N}_1(0, \sigma_Y^2).$$

and $\theta = (\sigma^2, \sigma_Y^2)$.

Signal processing: tracking, positioning, navigation

X_t is position of a moving object, e.g.

$$X_t = X_{t-1} + U_t, \quad U_t \sim \mathcal{N}_2(0, \sigma^2 I_2),$$

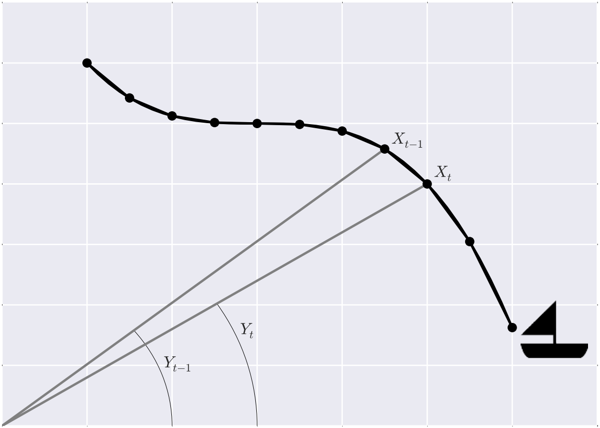
and Y_t is a measurement obtained by e.g. a radar,

$$Y_t = \mathrm{atan}\left(\frac{X_t(2)}{X_t(1)}\right) + V_t, \quad V_t \sim \mathcal{N}_1(0, \sigma_Y^2).$$

and $\theta = (\sigma^2, \sigma_Y^2)$.

(This is called the **bearings-only tracking** model.)

Corresponding plot



GPS

In GPS applications, the velocity v_t of the vehicle is observed, so motion model is (some variation of):

$$X_t = X_{t-1} + v_t + U_t, \qquad U_t \sim \mathcal{N}_2(0, \sigma^2 I_2).$$

Also Y_t usually consists of more than one measurement.

More advanced motion model

A random walk is too erratic for modelling the position of the target; assume instead its velocitity follows a random walk. Then define:

$$X_t = \begin{pmatrix} I_2 & I_2 \\ 0_2 & I_2 \end{pmatrix} X_{t-1} + \begin{pmatrix} 0_2 & 0_2 \\ 0_2 & U_t \end{pmatrix}, \quad U_t \sim \mathcal{N}_2(0, \sigma^2 I_2),$$

with obvious meanings for matrices 0_2 and I_2.

More advanced motion model

A random walk is too erratic for modelling the position of the target; assume instead its velocitity follows a random walk. Then define:

$$X_t = \begin{pmatrix} I_2 & I_2 \\ 0_2 & I_2 \end{pmatrix} X_{t-1} + \begin{pmatrix} 0_2 & 0_2 \\ 0_2 & U_t \end{pmatrix}, \quad U_t \sim \mathcal{N}_2(0, \sigma^2 I_2),$$

with obvious meanings for matrices 0_2 and I_2.

Note: $X_t(1)$ and $X_t(2)$ (position) are deterministic functions of X_{t-1}: no probability density for $X_t|X_{t-1}$.

multi-target tracking

Same ideas except $\{X_t\}$ now represent the position (and velocity if needed) of a set of targets (of random size); i.e. $\{X_t\}$ is a point process.

Time series of counts (neuro-decoding, astrostatistics, genetics)

- Neuro-decoding: Y_t is a vector of d_y counts (spikes from neuron k),

$$Y_t(k)|X_t \sim \mathcal{P}(\lambda_k(X_t)), \quad \log \lambda_k(X_t) = \alpha_k + \beta_k X_t,$$

  and X_t is position+velocity of the subject's hand (in 3D).
- astro-statistics: Y_t is number of photon emissions; intensity varies over time (according to an auto-regressive process)
- Y_t is the number of 'reads', which is a noisy measurement of the transcription level X_t at position t in the genome;

Time series of counts (neuro-decoding, astrostatistics, genetics)

- Neuro-decoding: Y_t is a vector of d_y counts (spikes from neuron k),

$$Y_t(k)|X_t \sim \mathcal{P}(\lambda_k(X_t)), \quad \log \lambda_k(X_t) = \alpha_k + \beta_k X_t,$$

  and X_t is position+velocity of the subject's hand (in 3D).
- astro-statistics: Y_t is number of photon emissions; intensity varies over time (according to an auto-regressive process)
- Y_t is the number of 'reads', which is a noisy measurement of the transcription level X_t at position t in the genome;

Note: 'functional' definition of state-space models is less convenient in this case.

Stochastic volatility (basic model)

Y_t is log-return of asset price, $Y_t = \log(p_t/p_{t-1})$,

$$Y_t | X_t = x_t \sim \mathcal{N}(0, \exp(x_t))$$

where $\{X_t\}$ is an auto-regressive process:

$$X_t - \mu = \phi(X_{t-1} - \mu) + U_t, \quad U_t \sim \mathcal{N}(0, \sigma^2)$$

and $\theta = (\mu, \phi, \sigma^2)$.

Stochastic volatility (basic model)

Y_t is log-return of asset price, $Y_t = \log(p_t/p_{t-1})$,

$$Y_t | X_t = x_t \sim \mathcal{N}(0, \exp(x_t))$$

where $\{X_t\}$ is an auto-regressive process:

$$X_t - \mu = \phi(X_{t-1} - \mu) + U_t, \quad U_t \sim \mathcal{N}(0, \sigma^2)$$

and $\theta = (\mu, \phi, \sigma^2)$.

Take $|\phi| < 1$ and $X_0 \sim N(\mu, \sigma^2/(1-\rho^2))$ to impose stationarity.

Stochastic volatility (variations)

- Student dist' for noises
- skewness: $Y_t = \alpha X_t + \exp(X_t/2) V_t$
- leverage effect: correlation between U_t and V_t
- multivariate extensions

Nonlinear dynamic systems in Ecology, Epidemiology, and other fields

$Y_t = X_t + V_t$, where $\{X_t\}$ is some complex nonlinear dynamic system. In Ecology for instance,

$$X_t = X_{t-1} + \theta_1 - \theta_2 \exp(\theta_3 X_{t-1}) + U_t$$

where X_t is log of population size. For some values of θ, process is nearly chaotic.

Nonlinear dynamic systems: Lokta-Volterra

Predator-prey model, where $\mathcal{X} = (\mathbb{Z}^+)^2$, $X_t(1)$ is the number of preys, $X_t(2)$ is the number of predators, and, working in continuous-time:

$$\begin{aligned} X_t(1) &\overset{\theta_1}{\rightarrow} 2X_t(1) \\ X_t(1) + X_t(2) &\overset{\theta_2}{\rightarrow} 2X_t(2), \quad t \in \mathbb{R}^+ \\ X_t(2) &\overset{\theta_3}{\rightarrow} 0 \end{aligned}$$

(but Y_t still observed in discrete time.)

Nonlinear dynamic systems: Lokta-Volterra

Predator-prey model, where $\mathcal{X} = (\mathbb{Z}^+)^2$, $X_t(1)$ is the number of preys, $X_t(2)$ is the number of predators, and, working in continuous-time:

$$\begin{aligned} X_t(1) &\xrightarrow{\theta_1} 2X_t(1) \\ X_t(1) + X_t(2) &\xrightarrow{\theta_2} 2X_t(2), \quad t \in \mathbb{R}^+ \\ X_t(2) &\xrightarrow{\theta_3} 0 \end{aligned}$$

(but Y_t still observed in discrete time.)

Intractable dynamics: We can simulate from $X_t|X_{t-1}$, but we can't compute $p_t(x_t|x_{t-1})$.

Nonlinear dynamic systems: Lokta-Volterra

Predator-prey model, where $\mathcal{X} = (\mathbb{Z}^+)^2$, $X_t(1)$ is the number of preys, $X_t(2)$ is the number of predators, and, working in continuous-time:

$$\begin{aligned} X_t(1) &\xrightarrow{\theta_1} 2X_t(1) \\ X_t(1) + X_t(2) &\xrightarrow{\theta_2} 2X_t(2), \quad t \in \mathbb{R}^+ \\ X_t(2) &\xrightarrow{\theta_3} 0 \end{aligned}$$

(but Y_t still observed in discrete time.)

Intractable dynamics: We can simulate from $X_t | X_{t-1}$, but we can't compute $p_t(x_t | x_{t-1})$.

State-space models with an intractable or degenerate observation process

We have seen models such that $X_t|X_{t-1}$ is intractable; $Y_t|X_t$ may be intractable as well. Let

$$X'_t = (X_t, Y_t), \quad Y'_t = Y_t + V_t, \quad V_t \sim \mathcal{N}(0, \sigma^2)$$

and use $\{(X'_t, Y'_t)\}$ as an approximation of $\{(X_t, Y_t)\}$.

State-space models with an intractable or degenerate observation process

We have seen models such that $X_t|X_{t-1}$ is intractable; $Y_t|X_t$ may be intractable as well. Let

$$X_t' = (X_t, Y_t), \quad Y_t' = Y_t + V_t, \quad V_t \sim \mathcal{N}(0, \sigma^2)$$

and use $\{(X_t', Y_t')\}$ as an approximation of $\{(X_t, Y_t)\}$.

$\Rightarrow$ Connection with ABC (likelihood-free inference).

Finite state-space models (aka hidden Markov models)

$\mathcal{X} = \{1, \ldots, K\}$, uses in e.g.

- speech processing; X_t is a word, Y_t is an acoustic measurement (possibly the earliest application of HMMs). Note K is quite large.
- time-series modelling to deal with heterogenity (e.g. in medecine, X_t is state of patient)
- rediscovered in Economics as Markov-switching models; there X_t is the state of the Economy (recession, growth), and Y_t is some economic indicator (e.g. GDP) which follows an ARMA process (with parameters that depend on X_t)
- also related: change-point models

Finite state-space models (aka hidden Markov models)

$\mathcal{X} = \{1, \ldots, K\}$, uses in e.g.

- speech processing; X_t is a word, Y_t is an acoustic measurement (possibly the earliest application of HMMs). Note K is quite large.
- time-series modelling to deal with heterogenity (e.g. in medecine, X_t is state of patient)
- rediscovered in Economics as Markov-switching models; there X_t is the state of the Economy (recession, growth), and Y_t is some economic indicator (e.g. GDP) which follows an ARMA process (with parameters that depend on X_t)
- also related: change-point models

Note: Not of direct interest to us, as sequential analysis may be performed *exactly* using Baum-Petrie filter.

Outline

Definition

The phrase *state-space models* refers not only to its definition (in terms of $\{X_t\}$ and $\{Y_t\}$) but also to a particular **inferential scenario**: $\{Y_t\}$ is observed (data denoted $y_0, \ldots$), $\{X_t\}$ is not, and one wishes to recover the X_t's given the Y_t's, often sequentially (over time).

Filtering, prediction, smoothing

Conditional distributions of interest (at every time t)

- Filtering: $X_t | Y_{0:t}$
- Prediction: $X_t | Y_{0:t-1}$
- data prediction: $Y_t | Y_{0:t-1}$
- fixed-lag smoothing: $X_{t-h:t} | Y_{0:t}$ for $h \geq 1$
- complete smoothing: $X_{0:t} | Y_{0:t}$
- likelihood factor: density of $Y_t | Y_{0:t-1}$ (so as to compute the full likelihood)

Parameter estimation

All these tasks are usually performed for a fixed θ (assuming the model depends on some parameter θ). To deal additionally with parameter uncertainty, we could adopt a Bayesian approach, and consider e.g. the law of (θ, X_t) given $Y_{0:t}$ (for filtering). But this is often more involved.

Formal notations

- $\{X_t\}$ is a Markov process with initial law $P_0(\mathrm{d}x_0)$, and Markov kernel $P_t(x_{t-1}, \mathrm{d}x_t)$.
- $\{Y_t\}$ has conditional distribution $F_t(x_t, \mathrm{d}y_t)$, which admits probability density $f_t(y_t|x_t)$ (with respect to common dominating measure $\nu(\mathrm{d}y_t)$).
- when needed, dependence on θ will be made explicit as follows: $P_t^\theta(x_{t-1}, \mathrm{d}x_t)$, $f_t^\theta(y_t|x_t)$, etc.

Formal notations

- $\{X_t\}$ is a Markov process with initial law $P_0(\mathrm{d}x_0)$, and Markov kernel $P_t(x_{t-1}, \mathrm{d}x_t)$.
- $\{Y_t\}$ has conditional distribution $F_t(x_t, \mathrm{d}y_t)$, which admits probability density $f_t(y_t|x_t)$ (with respect to common dominating measure $\nu(\mathrm{d}y_t)$).
- when needed, dependence on θ will be made explicit as follows: $P_t^\theta(x_{t-1}, \mathrm{d}x_t)$, $f_t^\theta(y_t|x_t)$, etc.

Algorithms, calculations, etc may be extended straightforwardly to non-standard situations such that $\mathcal{X}$, $\mathcal{Y}$ vary over time, or such that $Y_t|X_t$ also depends on $Y_{0:t-1}$, but for simplicity, we will stick to these notations.

Applications of SMC beyond state-space models

nicolas.chopin@ensae.fr
(based on a forthcoming book with Omiros Papaspiliopoulos)

Consider the simulation of Markov process $\{X_t\}$, conditional on $X_t \in A_t$ for each t.

Consider the simulation of Markov process $\{X_t\}$, conditional on $X_t \in A_t$ for each t.

Take $Y_t = \mathbb{1}(X_t \in A_t)$, $y_t = 1$, then this tasks amounts to smoothing the corresponding state-space model.

A particular example: self-avoiding random walk

Consider a random walk in $\mathbb{Z}^2$, (i.e. at each time we may move north, south, east or west, wit probability $1/4$). We would to simulate $\{X_t\}$ conditional on the trajectory $X_{0:T}$ never visiting the same point more than once.

A particular example: self-avoiding random walk

Consider a random walk in $\mathbb{Z}^2$, (i.e. at each time we may move north, south, east or west, wit probability $1/4$). We would to simulate $\{X_t\}$ conditional on the trajectory $X_{0:T}$ never visiting the same point more than once.

How to define $\{X_t\}$ in this case?

For a Bayesian model, with parameter θ, data $Y_0, \ldots, Y_t, \ldots$ (no latent variables), we would like to approximate recursively the posterior $p_t(\theta|y_{0:t})$. Could we treat this as a **filtering** problem, where

$$X_t = \Theta$$

is a constant process?

We wish to simulate from (or compute the normalising constant of):

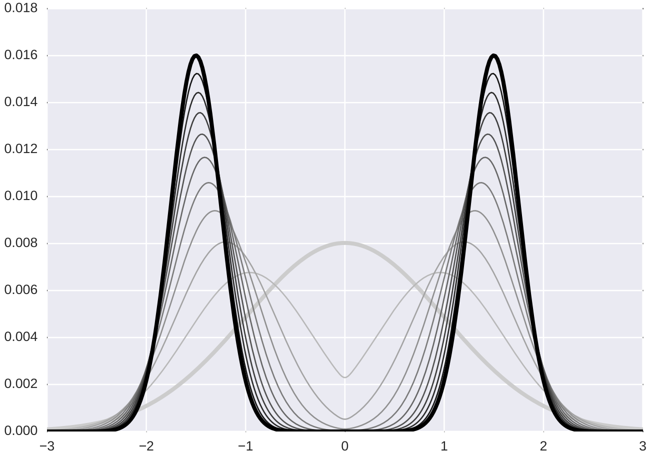
$$\pi(\mathrm{d}\theta) \propto \exp\{-V(\theta)\}\mathrm{d}\theta$$

To do so, we introduce a **tempering** sequence:

$$\mathbb{P}_t(\mathrm{d}\theta) \propto \exp\{-\lambda_t V(\theta)\}$$

where $0 = \lambda_0 < \ldots < \lambda_T = 1$, and use SMC to target recursively $\mathbb{P}_0$, $\mathbb{P}_1$, ...

Plot of tempering sequence



Fundamental question

In all these applications, how are we going to set the Markov kernels M_t to simulate the particles?

Fundamental question

In all these applications, how are we going to set the Markov kernels M_t to simulate the particles?

Hint: use MCMC.

Laying out the foundations: importance sampling, resampling, Feynman-Kac

nicolas.chopin@ensae.fr
(based on a forthcoming book with Omiros Papaspiliopoulos)

Outline

Basic identity

$$\int_{\mathcal{X}} \varphi(x) q(x) \mathrm{d}x = \int_{\mathcal{X}} \varphi(x) \frac{q(x)}{m(x)} m(x) \mathrm{d}x$$

Basic identity

$$\int_{\mathcal{X}} \varphi(x) q(x) \mathrm{d}x = \int_{\mathcal{X}} \varphi(x) \frac{q(x)}{m(x)} m(x) \mathrm{d}x$$

Warning: valid only if $m(x) = 0 \Rightarrow q(x) = 0$.

Basic identity

$$\int_{\mathcal{X}} \varphi(x)q(x)\mathrm{d}x = \int_{\mathcal{X}} \varphi(x)\frac{q(x)}{m(x)}m(x)\mathrm{d}x$$

Warning: valid only if $m(x) = 0 \Rightarrow q(x) = 0$.

Normalised IS estimator:

$$\frac{1}{N}\sum_{n=1}^{N} w(X^n)\varphi(X^n)$$

where $X^n \sim m$, $w(x) = q(x)/m(x)$.

Auto-normalised IS

Sometimes, we can compute densities m or q only **up to a constant**. However:

$$\int_{\mathcal{X}} \varphi(x) q(x) \, \mathrm{d}x = \frac{\int_{\mathcal{X}} \varphi(x) \frac{q(x)}{m(x)} m(x) \, \mathrm{d}x}{\int_{\mathcal{X}} \frac{q(x)}{m(x)} m(x) \, \mathrm{d}x}$$

$$= \frac{\int_{\mathcal{X}} \varphi(x) \frac{q_u(x)}{m_u(x)} m(x) \, \mathrm{d}x}{\int_{\mathcal{X}} \frac{q_u(x)}{m_u(x)} m(x) \, \mathrm{d}x}$$

Auto-normalised IS

Sometimes, we can compute densities m or q only **up to a constant**. However:

$$\begin{aligned}
\int_{\mathcal{X}} \varphi(x) q(x)\,\mathrm{d}x &= \frac{\int_{\mathcal{X}} \varphi(x) \frac{q(x)}{m(x)} m(x)\,\mathrm{d}x}{\int_{\mathcal{X}} \frac{q(x)}{m(x)} m(x)\,\mathrm{d}x} \\
&= \frac{\int_{\mathcal{X}} \varphi(x) \frac{q_u(x)}{m_u(x)} m(x)\,\mathrm{d}x}{\int_{\mathcal{X}} \frac{q_u(x)}{m_u(x)} m(x)\,\mathrm{d}x}
\end{aligned}$$

This suggests the autonormalised IS estimator:

$$\sum_{n=1}^{N} W^n \varphi(X^n), \qquad W^n = \frac{w(X^n)}{\sum_{m=1}^{N} w(X^m)}.$$

Change of measure

In a more general setting, the proposal and the target may be probability measures $\mathbb{M}(\mathrm{d}x)$, $\mathbb{Q}(\mathrm{d}x)$, and provided that $\mathbb{M}$ dominates $\mathbb{Q}$, we way reweight according to a function proportional to the **Radon-Nykodim derivative**.

Change of measure

In a more general setting, the proposal and the target may be probability measures $\mathbb{M}(\mathrm{d}x)$, $\mathbb{Q}(\mathrm{d}x)$, and provided that $\mathbb{M}$ dominates $\mathbb{Q}$, we way reweight according to a function proportional to the **Radon-Nykodim derivative**.

This is equivalent to applying a change of measure:

$$\mathbb{Q}(\mathrm{d}x) = \frac{1}{L}\mathbb{M}(\mathrm{d}x)w(x)$$

where $L = \mathbb{M}(w) \in (0, \infty)$.

Approximating moments, or approximating a distribution?

Since, for any function φ, we have

$$\sum_{n=1}^{N} W^n \varphi(X^n) \approx \mathbb{Q}(\varphi)$$

we could say that:

$$\mathbb{Q}^n(\mathrm{d}x) \approx \mathbb{Q}(\mathrm{d}x)$$

where $\mathbb{Q}^n$ is the following **random distribution**:

$$\mathbb{Q}^N(\mathrm{d}x) = \sum_{n=1}^{N} W^n \delta_{X^n}(\mathrm{d}x)$$

(In particular, $\mathbb{Q}^N(\varphi) = \sum_{n=1}^{N} \varphi(X^n)$.)

ESS (Effective sample size)

A popular criterion:

$$\mathrm{ESS} = \frac{1}{\sum_{n=1}^{N}(W^n)^2} = \frac{\left(\sum_{n=1}^{N} w(X^n)\right)^2}{\sum_{n=1}^{N} w(X^n)^2}$$

which has several justifications:

- $\mathrm{ESS} \in [1, N]$.
- If $w(x) = \mathbb{1}_A(x)$, ESS is number of non-zero weights.
- N/ESS converges to the **chi-square (pseudo-)distance** of q relative to m: $\int_{\mathcal{X}} m(q/m-1)^2$.

Curse of dimensionality in importance sampling

Now assume that both m and q are densities of IID variables $X_0, \ldots, X_T$; then

$$\frac{q(x)}{m(x)} = \prod_{t=0}^{T} \frac{q_1(x_t)}{m_1(x_t)}$$

and the variance of the weights is of the form $r^{T+1} - 1$, with $r \geq 1$.

Curse of dimensionality in importance sampling

Now assume that both m and q are densities of IID variables $X_0, \ldots, X_T$; then

$$\frac{q(x)}{m(x)} = \prod_{t=0}^{T} \frac{q_1(x_t)}{m_1(x_t)}$$

and the variance of the weights is of the form $r^{T+1} - 1$, with $r \geq 1$.

IID scenario not completely fictitious.

Outline

Feynman-Kac structure

Consider the following **generic** class of distributions: for each $t \geq 0$:

- $\mathbb{M}_t(\mathrm{d}x_{0:t})$ is the distribution of a *Markov* process $\{X_t\}$; with density:

$$= m_0(x_0)m_1(x_1|x_0)\dots m_t(x_t|x_{t-1})$$

- $\mathbb{Q}_t(\mathrm{d}x_{0:t})$ is the distribution that corresponds to the following **change of measure**, that is the distribution with density

$$= \frac{1}{L_t} m_0(x_0)m_1(x_1|x_0)\dots m_t(x_t|x_{t-1}) \prod_{s=0}^{t} G_s(x_s)$$

How to approximate the $\mathbb{Q}_t$'s?

Importance sampling? Curse of dimensionality.

How to approximate the $\mathbb{Q}_t$'s?

Importance sampling? Curse of dimensionality.

However, if we are only interested in certain **marginal distributions** of the $\mathbb{Q}_t$, we might be able to express our calculations in a much smaller dimension. This is the key observation.

Forward recursion

Suppose we have computed the marginal density $q_{t-1}(x_{t-1})$ (of variable X_{t-1} with respect to $\mathbb{Q}_{t-1}$. Then:

1. Extend:

$$q_{t-1}(x_{t-1}, x_t) = q_{t-1}(x_{t-1}) m_t(x_t | x_{t-1}).$$

2. Embrace (the next potential function):

$$q_t(x_{t-1}, x_t) \propto q_{t-1}(x_{t-1}, x_t) G_t(x_t)$$

3. Extinguish (marginalize out X_{t-1})

$$q_t(x_t) = \int_{\mathcal{X}} q_t(x_{t-1}, x_t) \mathrm{d}x_{t-1}$$

Why do we care?

Let's go back to state-space models. The smooting distribution at time t is the distribution of $X_{0:t}$ given $Y_{0:t} = y_{0:t}$, and has the expression:

$$\propto p_0(x_0) \prod_{s=1}^{t} p_t(x_t|x_{t-1}) \prod_{s=0}^{t} f_t(y_t|x_t)$$

hence, the same structure as $\mathbb{Q}_t(\mathrm{d}x_{0:t})$ provided we take:

- $m_t(x_t|x_{t-1}) = p_t(x_t|x_{t-1})$
- $G_t(x_{t-1}, x_t) = f_t(y_t|x_t)$

Why do we care?

Let's go back to state-space models. The smooting distribution at time t is the distribution of $X_{0:t}$ given $Y_{0:t} = y_{0:t}$, and has the expression:

$$\propto p_0(x_0) \prod_{s=1}^{t} p_t(x_t|x_{t-1}) \prod_{s=0}^{t} f_t(y_t|x_t)$$

hence, the same structure as $\mathbb{Q}_t(\mathrm{d}x_{0:t})$ provided we take:

- $m_t(x_t|x_{t-1}) = p_t(x_t|x_{t-1})$
- $G_t(x_{t-1}, x_t) = f_t(y_t|x_t)$

In particular, the forward recursion may be used to compute recursively the filtering distributions.

Practical implementations of the forward recursions

- finite state-space: replace integrals by sums, exact calculations, complexity $\mathcal{O}(K^2)$ per time step (Baum-Petrie)
- linear-Gaussian state-space models: propagatinng mean/variance through the **Kalman filter**
- other state-space models: importance sampling and resampling $\Rightarrow$ particle filters.

Exercise

Rewrite the forward recursion when:

- function G_t depends on both X_t and X_{t-1} (for $t \geq 1$);
- The Markov process $\{X_t\}$ is defined through Markov kernels $M_t(x_{t-1}, \mathrm{d}x_t)$ (which does not necessarily admit a density $m_t(x_t|x_{t-1})$ w.r.t. a fixed measure).

Outline

1. Importance sampling
2. Feynman-Kac
3. Resampling
   - Motivating examples

Motivation

$$\mathbb{Q}_0^N(\mathrm{d}x_0) = \sum_{n=1}^N W_0^n \delta_{X_0^n}, \qquad X^n \sim \mathbb{M}_0, \qquad W_0^n = \frac{w_0(X_0^n)}{\sum_{m=1}^N w_0(X_0^m)},$$

and now interested in

$$(\mathbb{Q}_0 M_1)(\mathrm{d}x_{0:1}) = \mathbb{Q}_0(\mathrm{d}x_0) M_1(x_0, \mathrm{d}x_1).$$

Two solutions:

First solution

IS from $\mathbb{M}_1 = \mathbb{M}_0 M_1$ to $\mathbb{Q}_0 M_1$:

(a) sample (X_0^n, X_1^n) from $\mathbb{M}_0 M_1$;
(b) compute weights.

This ignores the intermediate approximation of $\mathbb{Q}$ by $\mathbb{Q}_0^N$.

Second solution: resampling

$$\mathbb{Q}_0^N(\mathrm{d}x_0)M_1(x_0,\mathrm{d}x_1) = \sum_{n=1}^N W_0^n M_1(X_0^n,\mathrm{d}x_1)$$

and now we *sample* from this approximation:

$$\frac{1}{N}\sum_{n=1}^N \delta_{\widetilde{X}_{0:1}^n}, \qquad \text{where} \widetilde{X}_{0:1}^n \sim \mathbb{Q}_0^N(\mathrm{d}x_0)M_1(x_0,\mathrm{d}x_1).$$

One way to obtain such samples is to do:

$$\widetilde{X}_{0:1}^n = (X_0^{A_1^n}, X_1^n), \qquad A_1^{1:N} \sim \mathcal{M}(W_0^{1:N}), \qquad X_1^n \sim M_1(X_0^{A_1^n},\mathrm{d}x_1)$$

Why resample??

Toy example

- $\mathcal{X} = \mathbb{R}$, $\mathbb{M}_0$ is $\mathcal{N}(0,1)$, $w_0(x) = \mathbb{1}(|x| > \tau)$; thus $\mathbb{Q}_0$ is a truncated Gaussian distribution
- $M_1(x_0, \mathrm{d}x_1)$ so that $X_1 = \rho X_0 + \sqrt{1-\rho^2}U$, with $U \sim N(0,1)$
- $\varphi(x_1) = x_1$; note that $(\mathbb{Q}_0 M_1)(\varphi) = 0$

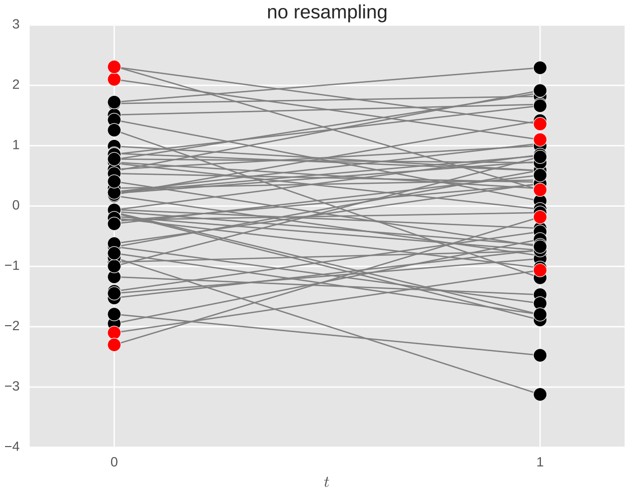
Toy example

- $\mathcal{X} = \mathbb{R}$, $\mathbb{M}_0$ is $\mathcal{N}(0,1)$, $w_0(x) = \mathbb{1}(|x| > \tau)$; thus $\mathbb{Q}_0$ is a truncated Gaussian distribution
- $M_1(x_0, \mathrm{d}x_1)$ so that $X_1 = \rho X_0 + \sqrt{1-\rho^2}U$, with $U \sim N(0,1)$
- $\varphi(x_1) = x_1$; note that $(\mathbb{Q}_0 M_1)(\varphi) = 0$

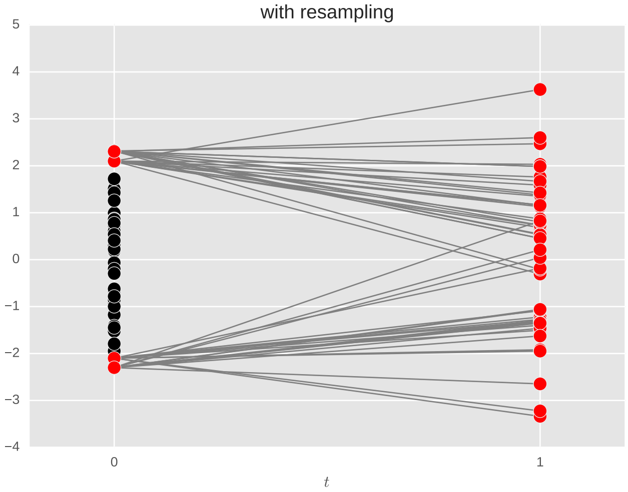
$$\widehat{\varphi}_{\mathrm{IS}} = \sum_{n=1}^{N} W_0^n X_1^n, \qquad (X_0^n, X_1^n) \sim \mathbb{M}_0 M_1$$

$$\widehat{\varphi}_{\mathrm{IR}} = N^{-1} \sum_{n=1}^{N} X_1^n, \qquad X_1^n \sim \mathbb{Q}_0^N M_1$$

No resampling



With resampling



Assume that, among the N particles X_0^n, k have a non-zero weight, then

$$\mathrm{var}[\widehat{\varphi}_{\mathrm{IS}}] \approx \frac{\rho^2 C(\tau)}{k} + \frac{1-\rho^2}{k}$$
$$\mathrm{var}[\widehat{\varphi}_{\mathrm{IR}}] \approx \frac{\rho^2 C'(\tau)}{k} + \frac{1-\rho^2}{N}$$

In words:

- IS: only k particles are "alive".
- IR: all N particles are alive, but they are correlated.
- if ρ not too large, IR beats IS.
- If τ gets larger and larger, relative performance of IS vs IR deteriorates quickly. $\Rightarrow$ Resampling is the safe option.

Bottom line

Resampling sacrifices the past to save the future.

Particle filtering

nicolas.chopin@ensae.fr
(based on a forthcoming book with Omiros Papaspiliopoulos)

Outline

Objectives

- introduce a generic PF algorithm for a given Feynman-Kac model $\{(M_t, G_t)\}_{t=0}^T$
- discuss the different algorithms one may obtain for a given state-space model, by using different Feynman-Kac formalisms.
- give more details on the implementation, complexity, and so on of the algorithm.

Outline

Input

- A Feynman-Kac model $\{(M_t, G_t)\}_{t=0}^{T}$ such that:
  - the weight function G_t may be evaluated pointwise (for all t);
  - it is possible to simulate from $M_0(\mathrm{d}x_0)$ and from $M_t(x_{t-1}, \mathrm{d}x_t)$ (for any x_{t-1} and t)
- The number of particles N

Structure

Algorithm 0.1: Basic PF algorithm

All operations to be performed for all $n \in 1:N$.

At time 0:

(a) Generate $X_0^n \sim M_0(\mathrm{d}x_0)$.

(b) Compute $w_0^n = G_0(X_0^n)$, $W_0^n = w_0^n / \sum_{m=1}^N w_0^m$, and $L_0^N = N^{-1} \sum_{n=1}^N w_0^n$.

Recursively, for $t = 1, \ldots, T$:

(a) Generate ancestor variables $A_t^n \in 1:N$ independently from $\mathcal{M}(W_{t-1}^{1:N})$.

(b) Generate $X_t^n \sim M_t(X_{t-1}^{A_t^n}, \mathrm{d}x_t)$.

Output

the algorithm delivers the following approximations at each time t:

$$\frac{1}{N}\sum_{n=1}^{N}\delta_{X_t^n} \quad \text{approximates } \mathbb{Q}_{t-1}(\mathrm{d}x_t)$$

$$\mathbb{Q}_t^N(\mathrm{d}x_t) = \sum_{n=1}^{N} W_t^n \delta_{X_t^n} \quad \text{approximates } \mathbb{Q}_t(\mathrm{d}x_t)$$

$$L_t^N \quad \text{approximates } L_t$$

some comments

- by *approximates*, we mean: for any test function φ, the quantity

$$\mathbb{Q}_t^N(\varphi) = \sum_{n=1}^{N} W_t^n \varphi(X_t^n)$$

converges to $\mathbb{Q}_t(\varphi)$ as $N \to +\infty$ (at the standard Monte Carlo rate $\mathcal{O}_P(N^{-1/2})$).

some comments

- by *approximates*, we mean: for any test function φ, the quantity

$$\mathbb{Q}_t^N(\varphi) = \sum_{n=1}^{N} W_t^n \varphi(X_t^n)$$

converges to $\mathbb{Q}_t(\varphi)$ as $N \to +\infty$ (at the standard Monte Carlo rate $\mathcal{O}_P(N^{-1/2})$).

- complexity is $\mathcal{O}(N)$ per time step.

Outline

Principle

We now consider a given state-space model:

- with initial law $P_0(\mathrm{d}x_0)$ and Markov kernel $P_t(x_{t-1}, \mathrm{d}x_t)$ for $\{X_t\}$;
- with conditional probability density $f_t(y_t|x_t)$ for $Y_t|X_t$

and discuss how the choice of a particular Feynman-Kac formalism leads to more or less efficient particle algorithms.

The bootstrap filter

Bootstrap Feynman-Kac formalism:

$$M_t(x_{t-1}, \mathrm{d}x_t) = P_t(x_{t-1}, \mathrm{d}x_t), \quad G_t(x_{t-1}, x_t) = f_t(y_t|x_t)$$

then $\mathbb{Q}_t$ is the filtering distribution, L_t is the likelihood of $y_{0:t}$, and so on.

The resulting algorithm is called the **boostrap filter**, and is particularly simple to interpret: we sample particles from Markov transition $P_t(x_{t-1}, \mathrm{d}x_t)$, and we reweight particles according to how compatible they are with the data.

The boostrap filter: algorithm

All operations to be performed for all $n \in 1:N$.

At time 0:

(a) Generate $X_0^n \sim P_0(\mathrm{d}x_0)$.

(b) Compute $w_0^n = f_0(y_0|X_0^n)$, $W_0^n = w_0^n / \sum_{m=1}^N w_0^m$, and $L_0^N = N^{-1} \sum_{n=1}^N w_0^n$.

Recursively, for $t = 1, \ldots, T$:

(a) Generate ancestor variables $A_t^n \in 1:N$ independently from $\mathcal{M}(W_{t-1}^{1:N})$.

(b) Generate $X_t^n \sim P_t(X_{t-1}^{A_t^n}, \mathrm{d}x_t)$.

(c) Compute $w_t^n = f_t(y_t|X_t^n)$, $W_t^n = w_t^n / \sum_{m=1}^N w_t^m$, and $L_t^N = L_{t-1}^N \{N^{-1} \sum_{n=1}^N w_t^n\}$.

The bootstrap filter: output

$$\frac{1}{N}\sum_{n=1}^{N}\varphi(X_t^n) \quad \text{approximates } \mathbb{E}[\varphi(X_t)|Y_{0:t-1}=y_{0:t-1}]$$

$$\sum_{n=1}^{N}W_t^n\varphi(X_t^n) \quad \text{approximates } \mathbb{E}[\varphi(X_t)|Y_{0:t}=y_{0:t}]$$

$$L_t^N \quad \text{approximates } p(y_{0:t})$$

The bootstrap filter: pros and cons

Pros:

- particularly simple
- does not require to compute the density $X_t|X_{t-1}$: we can apply it to models with **intractable dynamics**

Cons:

- We simulate particles *blindly*: if $Y_t|X_t$ is very informative, few particles will get a non-negligible weight.

The guided PF

Guided Feynman-Kac formalism: M_t is a user-chosen **proposal** kernel such that $M_t(x_{t-1}, \mathrm{d}x_t)$ dominates $P_t(x_{t-1}, \mathrm{d}x_t)$, and

$$G_t(x_{t-1}, x_t) = \frac{f_t(y_t|x_t) P_t(x_{t-1}, \mathrm{d}x_t)}{M_t(x_{t-1}, \mathrm{d}x_t)}$$
$$= \frac{f_t(y_t|x_t) p_t(x_t|x_{t-1})}{m_t(x_t|x_{t-1})}$$

(assuming in the second line that both kernels admit a density wrt a common measure). We still have that $\mathbb{Q}_t(\mathrm{d}x_t)$ is the filtering distribution, and L_t is the likelihood.
We call the resulting algorithm the **guided particle filter**, as in practice we would like to choose M_t so as to **guide** particles to regions of high likelihood.

The guided PF: choice of M_t (local optimality)

Suppose that (G_s, M_s) have been chosen to satisfy (**??**) for $s \leq t-1$. Among all pairs (M_t, G_t) that satisfy (**??**), the Markov kernel

$$M_t^{\mathrm{opt}}(x_{t-1}, \mathrm{d}x_t) = \frac{f_t(y_t|x_t)}{\int_{\mathcal{X}} f(y_t|x')\, P_t(x_{t-1}, \mathrm{d}x')} P_t(x_{t-1}, \mathrm{d}x_t)$$

minimises the variance of the weights, $\mathrm{Var}\left[G_t(X_{t-1}^{A_t^n}, X_t^n)\right]$.

Interpretation and discussion of this result

- M_t^{opt} is simply the law of X_t given X_{t-1} and Y_t. In a sense it is the perfect compromise between the information brought by $P_t(x_{t-1}, \mathrm{d}x_t)$ and by $f_t(y_t|x_t)$.
- In most practical cases, M_t^{opt} is not tractable, hence this result is mostly indicative (on how to choose M_t).
- Note also that the local optimality criterion is debatable. For instance, we do not consider the effect of *future* datapoints.

A first example: stochastic volatility

There, the log-density of $X_t | X_{t-1}, Y_t$ is (up to a constant):

$$-\frac{1}{2\sigma^2}\left\{x_t - \mu - \phi(x_{t-1} - \mu)\right\}^2 - \frac{x_t}{2} - \frac{e^{-x_t}}{2} y_t^2$$

We can use $e^{x - x_0} \approx 1 + (x - x_0) + (x - x_0)^2/2$ to get a Gaussian approximation.

A second example: bearings-only tracking

In that case, $P_t(x_{t-1}, \mathrm{d}x_t)$ imposes deterministic constraints:

$$X_t(k) = X_{t-1}(k) + X_{t-1}(k+2), \quad k = 1, 2$$

We can choose a M_t that imposes the same constraints. However, in this case, we find that

$$M_t^{\mathrm{opt}}(x_{t-1}, \mathrm{d}x_t) = P_t(x_{t-1}, \mathrm{d}x_t).$$

Discuss.

Guided particle filter pros and cons

Pro:

- may work much better that bootstrap filter when $Y_t|X_t$ is informative (provided we are able to derive a good proposal).

Cons:

- requires to be able to compute density $p_t(x_t|x_{t-1})$.
- sometimes local optimality criterion is not so sound.

The auxiliary particle filter

In the auxiliary Feynman-Kac formalism, an extra degree of freedom is gained by introducing **auxiliary** function η_t, and set:

$$G_0(x_0) = f_0(y_0|x_0)\frac{P_0(\mathrm{d}x_0)}{M_0(\mathrm{d}x_0)}\eta_0(x_0),$$

$$G_t(x_{t-1}, x_t) = f_t(y_t|x_t)\frac{P_t(x_{t-1}, \mathrm{d}x_t)}{M_t(x_{t-1}, \mathrm{d}x_t)}\frac{\eta_t(x_t)}{\eta_{t-1}(x_{t-1})}\,.$$

so that

$$\mathbb{Q}_t(\mathrm{d}x_{0:t}) \propto \mathbb{P}(\mathrm{d}x_{0:t}|Y_{0:t} = y_{0:t})\eta_t(x_t)$$

and we recover the filtering distribution by reweighting by $1/\eta_t$.
Idea: choose η_t so that G_t is as constant as possible.

Output of APF

Let $\tilde{w}_t^n := w_t^n / \eta_t(X_t^n)$, $\tilde{W}_t^n := \tilde{w}_t^n / \sum_{m=1}^N \tilde{w}_t^m$, then

$$\frac{1}{\sum_{m=1}^N \frac{\tilde{W}_t^m}{f(y_t|X_t^m)}} \sum_{n=1}^N \frac{\tilde{W}_t^n}{f_t(y_t|X_t^n)} \varphi(X_t^n) \quad \text{approx.} \quad \mathbb{E}[\varphi(X_t)|Y_{0:t-1} = y_{0:t-1}]$$

$$\sum_{n=1}^N \tilde{W}_t^n \varphi(X_t^n) \quad \text{approx.} \quad \mathbb{E}[\varphi(X_t)|Y_{0:t} = y_{0:t}]$$

$$L_t^N \times N^{-1} \sum_{n=1}^N \tilde{w}_t^n \quad \text{approx.} \quad p(y_{0:t})$$

Local optimality for M_t and η_t

For a given state-space model, suppose that (G_s, M_s) have been chosen to satisfy (**??**) for $s \leq t-2$, and M_{t-1} has also been chosen. Among all pairs (M_t, G_t) that satisfy (**??**) and functions η_{t-1}, the Markov kernel

$$M_t^{\mathrm{opt}}(x_{t-1}, \mathrm{d}x_t) = \frac{f_t(y_t|x_t)}{\int_{\mathcal{X}} f(y_t|x')\, P_t(x_{t-1}, \mathrm{d}x')} P_t(x_{t-1}, \mathrm{d}x_t)$$

and the function

$$\eta_{t-1}^{\mathrm{opt}}(x_{t-1}) = \int_{\mathcal{X}} f(y_t|x')\, P_t(x_{t-1}, \mathrm{d}x')$$

minimise $\mathrm{Var}\left[G_t(X_{t-1}^{A_t^n}, X_t^n)/\eta_t(X_t^n)\right]$.

Interpretation and discussion

- We find again that the optimal proposal is the law of X_t given X_{t-1} and Y_t. In addition, the optimal auxiliary function is the probability density of Y_t given X_{t-1}.
- For this ideal algorithm, we would have

$$G_t(x_{t-1}, x_t) = \eta_t^{\mathrm{opt}}(x_t);$$

the density of Y_{t+1} given $X_t = x_t$; not constant, but intuitively less variable than $f_t(y_t|x_t)$ (as in the bootstrap filter).

Example: stochastic volatility

We use the same ideas as for the guided PF: Taylor expansion of log-density, then we integrate wrt x_t.

APF pros and cons

Pros:

- usually gives some extra performance.

Cons:

- a bit difficult to interpret and use;
- they are some (contrived) examples where the auxiliary particle filter actually performs worse than the bootstrap filter.

Note on the generality of APF

From the previous descriptions, we see that:

- the guided PF is a particular instance of the auxiliary particle filter (take $\eta_t = 1$);
- the bootstrap filter is a particular instance of the guided PF(take $M_t = P_t$).

This is why some recent papers focus on the APF.

Which resampling to use in practice?

- Systematic resampling is fast, easy to implement, and seems to work best; but no supporting theory.
- We **do** have some theoretical results regarding the fact that multinomial resampling is dominated by most other resampling schemes. (So don't use it!)
- On the other hand, multinomial resampling is easier to study formally (because again it is based on IID sampling).

Outline

Resampling or not resampling, that is the question

For the moment, we resample every time. When we introduced resampling, we explained that the decision to resample was based on a trade-off: adding noise at time $t-1$, while hopefully reducing noise at time t (assuming that $\{X_t\}$ forgets its past).

Resampling or not resampling, that is the question

For the moment, we resample every time. When we introduced resampling, we explained that the decision to resample was based on a trade-off: adding noise at time $t-1$, while hopefully reducing noise at time t (assuming that $\{X_t\}$ forgets its past).
We do know that never resample would be a bad idea: consider $M_t(x_{t-1}, \mathrm{d}x_t)$ defined such that the X_t are IID $\mathcal{N}(0,1)$, $G_t(x_t) = \mathbb{1}(x_t > 0)$. (More generally, recall the curse of dimensionality of importance sampling.)

The ESS recipe

Trigger the resampling step whenever the variability of the weights is too large, as measured by e.g. the ESS (effective sample size):

$$\mathrm{ESS}(W_t^{1:N}) := \frac{1}{\sum_{n=1}^N (W_t^n)^2} = \frac{\{\sum_{n=1}^N w_t(X^n)\}^2}{\sum_{n=1}^N w_t(X^n)^2}.$$

Recall that $\mathrm{ESS}(W_t^{1:N}) \in [1, N]$, and that if k weights equal one, and $N - k$ weights equal zero, then $\mathrm{ESS}(W_t^{1:N}) = k$.

PF with adaptive resampling

(Same operations at $t = 0$.)
Recursively, for $t = 1, \dots, T$:

(a) **If** $\mathrm{ESS}(W_{t-1}^{1:N}) < \gamma N$
generate ancestor variables $A_{t-1}^{1:N}$ from resampling distribution $\mathcal{RS}(W_{t-1}^{1:N})$, and set $\hat{W}_{t-1}^{n} = W_{t-1}^{A_t^n}$;
Else (no resampling)
set $A_{t-1}^{n} = n$ and $\hat{W}_{t-1}^{n} = 1/N$

(b) Generate $X_t^n \sim M_t(X_{t-1}^{A_t^n}, \mathrm{d}x_t)$.

(c) Compute $w_t^n = (N\hat{W}_{t-1}^n) \times G_t(X_{t-1}^{A_t^n}, X_t^n)$,
$L_t^N = L_{t-1}^N \{N^{-1} \sum_{n=1}^N w_t^n\}$, $W_t^n = w_t^n / \sum_{m=1}^N w_t^m$.

Outline

Linear Gaussian example

$$X_t = \rho X_{t-1} + \sigma_X U_t$$
$$Y_t = X_t + \sigma_Y V_t$$

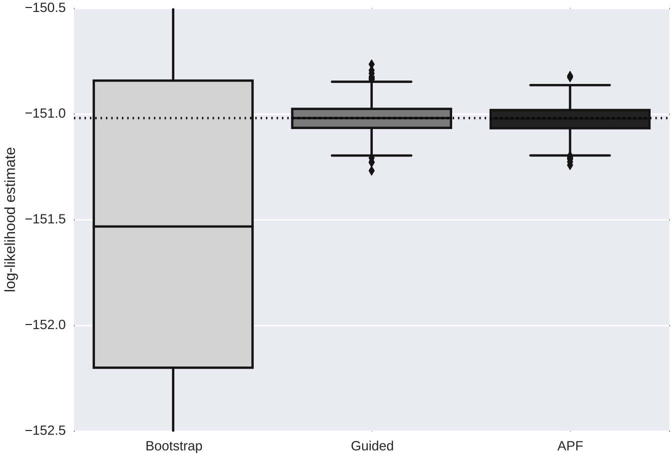
with $\rho = 0.9$, $\sigma_X = 1$, $\sigma_Y = 0.2$.

Linear Gaussian example

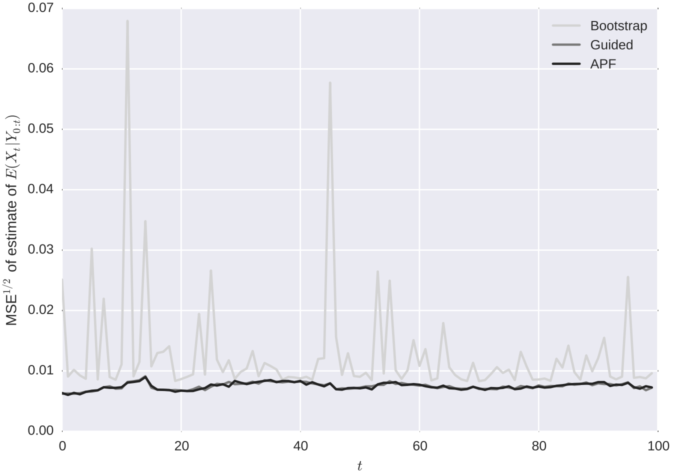
$$X_t = \rho X_{t-1} + \sigma_X U_t$$
$$Y_t = X_t + \sigma_Y V_t$$

with $\rho = 0.9$, $\sigma_X = 1$, $\sigma_Y = 0.2$.
We can implement the perfect guided filter and the perfect APF.

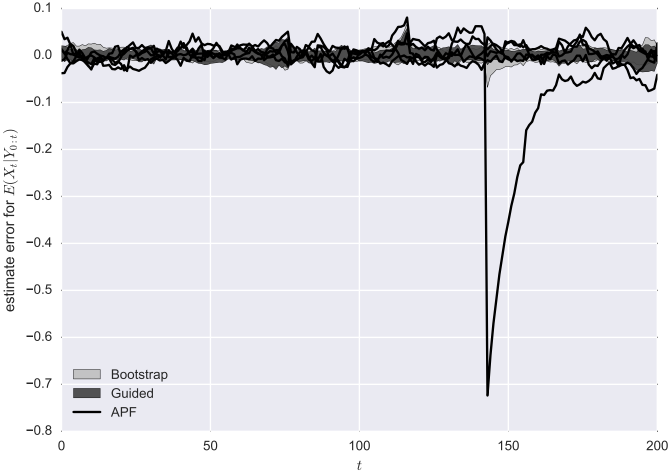
Likelihood


−150.5
−151.0
−151.5
−152.0
−152.5
log-likelihood estimate
Bootstrap
Guided
APF

Filtering


0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
20
40
60
80
100
t
MSE$^{1/2}$ of estimate of $E(X_t|Y_{0:t})$
Bootstrap
Guided
APF

Stochastic volatility



SMC samplers

nicolas.chopin@ensae.fr
(based on a previous PG course with O. Papaspiliopoulos)

Summary

- Motivating problems: sequential (or non-sequential) inference and simulation outside SSMs (including normalising constant calculation)
- Feynman-Kac formalisation of such problems
- Specific algorithms: IBIS, tempering SMC, SMC-ABC
- An overarching framework: SMC samplers

Outline

1. Motivating problems
   - Sequential Bayesian learning
   - Tempering
   - Rare event simulation

2. Notation and statement of problem

Sequential Bayesian learning

$\mathbb{P}_t(\mathrm{d}\theta)$ posterior distribution of parameters θ, given observations $y_{0:t}$, where $\theta \in \Theta$; typically:

$$\mathbb{P}_t(\mathrm{d}\theta) = \frac{1}{p_t(y_{0:t})} p_t^\theta(y_{0:t})\nu(\mathrm{d}\theta)$$

with $\nu(\mathrm{d}\theta)$ the prior distribution, $p_t^\theta(y_{0:t})$ likelihood and $p_t(y_{0:t})$ marginal likelihood.

Note that

$$\frac{\mathbb{P}_t(\mathrm{d}\theta)}{\mathbb{P}_{t-1}(\mathrm{d}\theta)} \propto p_t^\theta(y_t|y_{0:t-1}).$$

Practical motivations

- sequential learning
- Detection of outliers and structural changes
- Sequential model choice/composition
- ‘Big’ data
- Data tempering effect

Tempering

Suppose we wish to either sample from, or compute the normalising constant of

$$\mathbb{P}(\mathrm{d}\theta) = \frac{1}{L} \exp\{-V(\theta)\}\mu(\mathrm{d}\theta).$$

Idea: introduce for any $a \in [0, 1]$,

$$\mathbb{P}^a(\mathrm{d}\theta) = \frac{1}{L_a} \exp\{-aV(\theta)\}\mu(\mathrm{d}\theta).$$

Note that

$$\frac{\mathbb{P}^b(\mathrm{d}\theta)}{\mathbb{P}^a(\mathrm{d}\theta)} \propto \exp\{(a-b)V(\theta)\}$$

Rare events

Suppose we wish to either sample from, or compute the normalising constant of

$$\mathbb{P}(\mathrm{d}\theta) = \frac{1}{L}\mathbb{1}_E(\theta)\mu(\mathrm{d}\theta).$$

for some set E.

As for tempering, we could introduce a sequence of sets $\Theta = E_0 \supset \ldots \supset E_n = E$, and the corresponding sequence of distributions.

Outline

1. Motivating problems
   - Sequential Bayesian learning
   - Tempering
   - Rare event simulation
2. Notation and statement of problem

Statement

Sequence of probability distributions on a common space $(\Theta, \mathcal{B}(\Theta))$, $\mathbb{P}_0(\mathrm{d}\theta), \ldots, \mathbb{P}_T(\mathrm{d}\theta)$. In certain applications interest only in $\mathbb{P}_T$, in others for all $\mathbb{P}_t$, in others mainly interested in normalising constants.

For simplicity, assume that $\mathbb{P}_t(\mathrm{d}\theta)$ has density $\gamma_t(\theta)/L_t$ (wrt to some common dominating measure).

Forward recursion

- Let $G_t(\theta)$ such that $\frac{\mathbb{P}_t(\mathrm{d}\theta)}{\mathbb{P}_{t-1}(\mathrm{d}\theta)} \propto G_t(\theta)$.
- Suppose we can construct a **MCMC** kernel M_t that leaves invariant $\mathbb{P}_{t-1}(\mathrm{d}\theta)$.

Then

$$\begin{aligned} \mathbb{P}_t(\mathrm{d}\theta') &= \frac{\mathbb{P}_t(\mathrm{d}\theta')}{\mathbb{P}_{t-1}(\mathrm{d}\theta')}\mathbb{P}_{t-1}(\mathrm{d}\theta') \\ &= G_t(\theta') \int_\Theta M_t(\theta, \mathrm{d}\theta')\mathbb{P}_{t-1}(\mathrm{d}\theta) \end{aligned}$$

$\Rightarrow$ We recognise the forward recursion of a Feynman-Kac model.

In practice

This means that, provided:

- We can compute $\gamma_t(\theta)/\gamma_{t-1}(\theta)$ pointwise;
- We can sample from $M_t(\theta_{t-1}, \mathrm{d}\theta)$, a MCMC kernel that leaves invariant $\mathbb{P}_{t-1}(\mathrm{d}\theta)$;

We are able to implement a SMC sampler that targets $\mathbb{P}_t(\mathrm{d}\theta)$ at every iteration t. (Same algorithm as usual!)

How to choose the MCMC kernels?

A standard choice for MCMC kernel M_t is a Gaussian random walk Metropolis. Then we can calibrate the random walk variance on the empirical variance of the resampled particles.

It is also possible to automatically choose when to do resampling+MCMC:

- for sequential inference, trigger resampling+MCMC when ESS is below (say) $N/2$.
- for tempering SMC, one may choose recursively $\delta_i = a_i - a_{i-1}$ by solving numerically $\mathrm{ESS} = N/2$ (say).

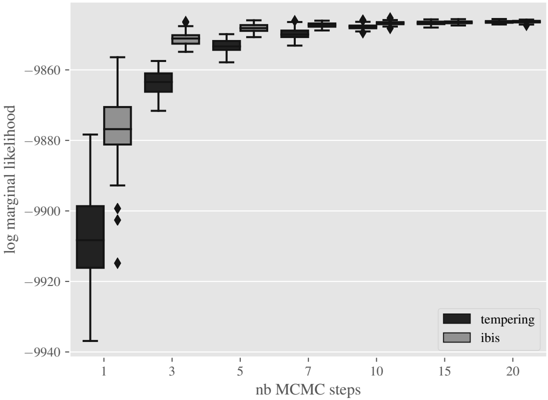
Numerical experiment

Logistic regression, two datasets:

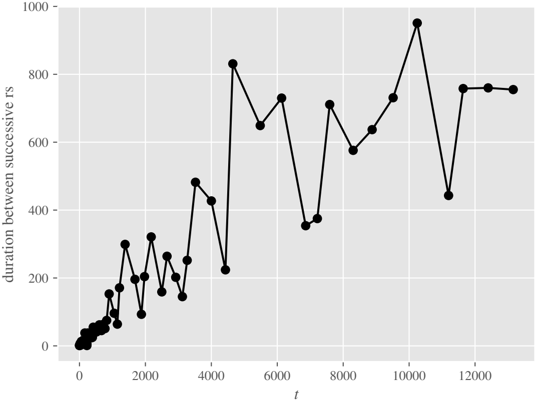
- EEG (tall): $d = 15$, $T \approx 15000$
- Sonar (big): $d = 60$, $T = 200$

We compare IBIS vs tempering, for estimating the marginal likelihood and the posterior expectations.

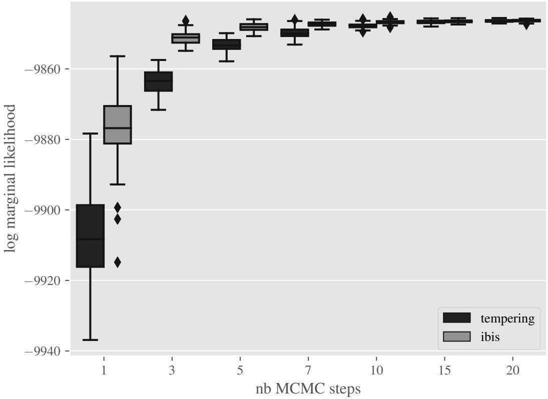
EEG (tall): IBIS behaviour



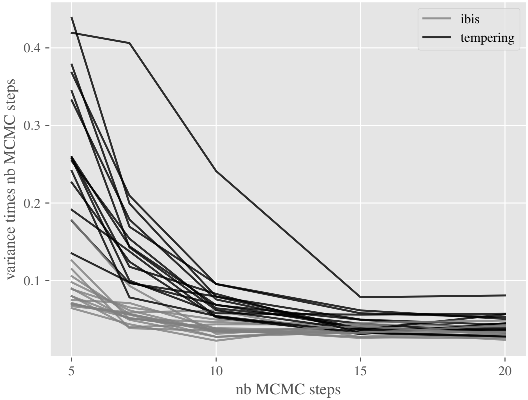
EEG (tall): IBIS behaviour



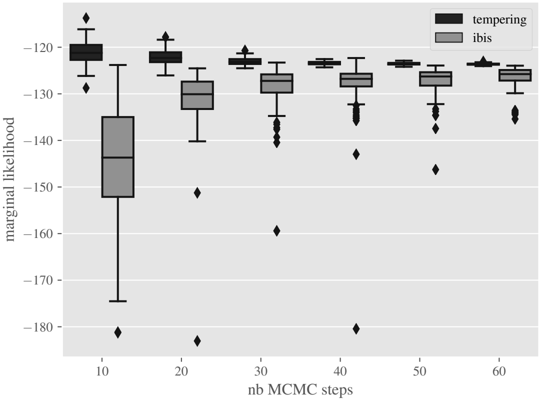
EEG (tall): marginal likelihood



EEG (tall): posterior expectations



Sonar (big): marginal likelihood



Conclusion

- Even more general SMC samplers may be obtained by considering kernels that are not invariant; see Del Moral et al (2006).
- However, even these general algorithms are special instances of the generic SMC algorithm.
- In practice, the main appeals of SMC samplers are:
  1. parallelisation;
  2. easy to make them adaptive;
  3. estimate of the marginal likelihood for free.

Particles as auxiliary variables: PMCMC and related algorithms

nicolas.chopin@ensae.fr
(based on a previous PG course with O. Papaspiliopoulos)

Particles as auxiliary variables: PMCMC and related algorithms

nicolas.chopin@ensae.fr
(based on a previous PG course with O. Papaspiliopoulos)

Outline

Tractable models

For a standard Bayesian model, defined by (a) prior $p(\theta)$, and (b) likelihood $p(y|\theta)$, a standard approach is to use the Metropolis-Hastings algorithm to sample from the posterior

$$p(\theta|y) \propto p(\theta)p(y|\theta).$$

Metropolis-Hastings

From current point θ_m

1. Sample $\theta_\star \sim H(\theta_m, \mathrm{d}\theta_\star)$
2. With probability $1 \wedge r$, take $\theta_{m+1} = \theta_\star$, otherwise $\theta_{m+1} = \theta_m$, where

$$r = \frac{p(\theta_\star)p(y|\theta_\star)h(\theta_m|\theta_\star)}{p(\theta_m)p(y|\theta_m)h(\theta_\star|\theta_m)}$$

This generates a Markov chain which leaves $p(\theta|y)$ invariant.

Metropolis Proposal

Note that proposal kernel $H(\theta_m, \mathrm{d}\theta_\star)$ (to simulate proposed value $\theta^\star$, conditional on current value θ_m). Popular choices are:

- random walk proposal: $h(\theta^\star|\theta_m) = N(\theta^\star; \theta_m, \Sigma)$; usual recommendation is to take $\Sigma \approx c_d \Sigma_{\mathrm{post}}$, with $c_d = 2.38^2/d$.
- independent proposal: $h(\theta^\star|\theta_m) = h(\theta^\star)$.
- Langevin proposals.

Intractable models

This generic approach cannot be applied in the following situations:

1. The likelihood is $p(y|\theta) = h_\theta(y)/Z(\theta)$, where $Z(\theta)$ is an intractable normalising constant; e.g. log-linear models, network models, Ising models.
2. The likelihood $p(y|\theta)$ is an intractable integral

$$p(y|\theta) = \int_{\mathcal{X}} p(y, x|\theta)\, dx.$$

3. The likelihood is even more complicated, because it corresponds to some scientific model involving some complicate *generative* process (scientific models, "likelihood-free inference", ABC).

Example of likelihoods as intractable integrals

When $p(y|\theta) = \int p(y, x|\theta)\, dx$.

- phylogenetic trees (Beaumont, 2003);
- state-space models (see later);
- other models with latent variables.

We will focus on this case, but certain ideas may also be applied to the two other cases.

Outline

General framework

Consider posterior

$$\pi(\theta, x) \propto p(\theta)p(x|\theta)p(y|x,\theta)$$

where typically x is of much larger dimension than θ.
One potential approach to sample from the posterior is *Gibbs sampling*: iteratively sample $\theta|x,y$, then $x|\theta,y$. However, there are many cases where Gibbs is either difficult to implement, or quite inefficient.
Instead, we would like to sample *marginally* from

$$\pi(\theta) \propto p(\theta)p(y|\theta), \quad p(y|\theta) = \int_{\mathcal{X}} p(x,y|\theta)\,dx$$

but again $p(y|\theta)$ is intractable...

Importance sampling

I cannot compute $p(y|\theta)$, but I can compute an *unbiased* estimator of this quantity:

$$\hat{p}(y|\theta) = \frac{1}{N}\sum_{n=1}^{N}\frac{p(y, x^n|\theta)}{q(x^n)}, \quad x^{1:N} \overset{iid}{\sim} q(x)$$

using *importance sampling*.

The pseudo-marginal approach

GIMH (Beaumont, 2003)

From current point θ_m

1. Sample $\theta_\star \sim H(\theta_m, \mathrm{d}\theta_\star)$
2. With prob. $1 \wedge r$, take $\theta_{m+1} = \theta_\star$, otherwise $\theta_{m+1} = \theta_m$, with

$$r = \frac{p(\theta_\star)\hat{p}(y|\theta_\star)h(\theta_m|\theta_\star)}{p(\theta_m)\hat{p}(y|\theta_m)h(\theta_\star|\theta_m)}$$

Note that $\hat{p}(y|\theta_\star)$ is based on independent samples generated at iteration m.

Question: Is GIMH a *non-standard* HM sampler w.r.t. *standard* target $\pi(\theta)$?

Validity of GIMH

Property 1

The following function

$$\bar{\pi}(\theta, x^{1:N}) = \prod_{n=1}^{N} q(x^n) \frac{p(\theta)\hat{p}(y|\theta)}{p(y)}$$

is a joint PDF, whose θ-marginal is $\pi(\theta) \propto p(\theta)p(y|\theta)$.

Proof: Direct consequence of unbiasedness; fix θ then

$$\int \prod_{n=1}^{N} q(x^n) p(\theta)\hat{p}(y|\theta)\, dx^{1:N} = p(\theta)\mathbb{E}\left[\hat{p}(y|\theta)\right] = p(\theta)p(y|\theta)$$

GIMH as a Metropolis sampler

Property 2

GIMH is a Metropolis sampler with respect to joint distribution $\bar{\pi}(\theta, x^{1:N})$. The proposal density is $h(\theta_\star|\theta_m)\prod_{n=1}^N q(x_\star^n)$.

Proof: current point is $(\theta_m, x_m^{1:N})$, proposed point is $(\theta_\star, x_\star^{1:N})$ and HM ratio is

$$r = \frac{\cancel{\prod_{n=1}^N q(x_\star^n)} p(\theta_\star)\hat{p}(y|\theta_\star)h(\theta_m|\theta_\star)\cancel{\prod_{n=1}^N q(x_m^n)}}{\cancel{\prod_{n=1}^N q(x_m^n)} p(\theta_m)\hat{p}(y|\theta_m)h(\theta_\star|\theta_m)\cancel{\prod_{n=1}^N q(x_\star^n)}}$$

Thus, GIMH is a *standard* Metropolis sampler w.r.t. *non-standard* (extended) target $\bar{\pi}(\theta, x^{1:N})$.

There is more to life than this

Property 3

Extend $\bar{\pi}(\theta, x^{1:N})$ with $k|\theta, x^{1:N} \propto \pi(\theta, x^k)/q(x^k)$, then,

- the marginal dist. of (θ, x^k) is $\pi(\theta, x)$.
- Conditional on (θ, x^k), $x_n \sim q$ for $n \neq k$, independently.

Proof: let

$$\bar{\pi}(\theta, x^{1:N}, k) = \left\{ \prod_{n=1}^{N} q(x^n) \right\} \frac{\pi(\theta, x^k)}{q(x^k)} = \left\{ \prod_{n \neq k} q(x^n) \right\} \pi(\theta, x^k)$$

then clearly the sum w.r.t. k gives $\bar{\pi}(\theta, x^{1:N})$, while the above properties hold.

We can do Gibbs!

One consequence of Property 3 is that we gain the ability to perform *Gibbs*, in order to regenerate the $N-1$ non-selected points x^n, $n \neq k$. More precisely:

1. Sample $k \sim \pi(k|\theta, x^{1:N}) \propto \pi(\theta, x^k)/q(x^k)$
2. regenerate $x^n \sim q$, for all $n \neq k$.

Could be useful for instance to avoid "getting stuck", because say the current value $\hat{\pi}(\theta)$ is too high.

Main lessons

- We can replace an intractable quantity by an unbiased estimate, *without introducing any approximation*.
- In fact, we can do more: with Proposition 3, we have obtained that
  1. it is possible to sample from $\pi(\theta, x)$ jointly;
  2. it is possible to do a Gibbs step where the $N-1$ x^n, $n \neq k$ are regenerated (useful when GIMH "get stucks"?)
- but careful, it is possible to get it wrong...

Unbiasedness without an auxiliary variable representation

This time, consider instead a target $\pi(\theta)$ (no x), involving an intractable *denominator*; an important application is Bayesian inference on likelihoods with intractable normalising constants:

$$\pi(\theta) \propto p(\theta)p(y|\theta) = p(\theta)\frac{h_\theta(y)}{Z(\theta)}$$

Liang & Lin (2010)'s sampler

From current point θ_m

1. Sample $\theta_\star \sim H(\theta^m, \mathrm{d}\theta_\star)$
2. With prob. $1 \wedge r$, take $\theta_{m+1} = \theta_\star$, otherwise $\theta_{m+1} = \theta_m$, with

$$r = \left(\widehat{\frac{Z(\theta_m)}{Z(\theta_\star)}}\right)\frac{p(\theta_\star)h_{\theta_\star}(y)h(\theta^m|\theta_\star)}{p(\theta_m)h_{\theta_m}(y)h(\theta_\star|\theta^m)}.$$

Russian roulette

See the Russian roulette paper of Girolami et al (2013, arxiv) for a valid algorithm for this type of problem. Basically they compute an unbiased estimator of $Z(\theta)^{-1}$ at every iteration.

Note the connection with Bernoulli factories: from unbiased estimates $\hat{Z}_i(\theta)$ of $Z(\theta)$, how do you obtain an unbiased estimate of $\varphi(Z(\theta))$? here $\varphi(z) = 1/z$.

Outline

PMCMC: introduction

PMCMC (Andrieu et al., 2010) is akin to GIMH, except a more complex proposal mechanism is used: a PF (particle filter).
The same remarks will apply:

- Unbiasedness (of the likelihood estimated provided by the PF) is only an intermediate result for establishing the validity of the whole approach.
- Unbiasedness is not enough to give you intuition on the validity of e.g. Particle Gibbs.

Objective

Objectives

Sample from

$$p(\mathrm{d}\theta, \mathrm{d}x_{0:T} | y_{0:T})$$

for a given state-space model.

Why are these models difficult?

Because the likelihood is intractable

$$p_T^\theta(y_{0:T}) = \int \prod_{t=0}^{T} f_t^\theta(y_t|x_t) \prod_{t=1}^{T} p_t^\theta(x_t|x_{t-1}) p_0^\theta(x_0)$$

Feynman-Kac formalism

Taking $\{M_t^\theta, G_t^\theta\}_{t\geq 0}$ so that

- $M_t^\theta(x_{t-1}, \mathrm{d}x_t)$ is a Markov kernel (for fixed θ), with density $m_t^\theta(x_t|x_{t-1})$
- and

$$G_t^\theta(x_{t-1}, x_t) = \frac{f_t^\theta(y_t|x_t)p_t^\theta(x_t|x_{t-1})}{m_t^\theta(x_t|x_{t-1})}$$

we obtain the Feynman-Kac representation associated to a guided PF that approximates the filtering distribution at every time t.

If we take $m_t^\theta(x_t|x_{t-1}) = p_t^\theta(x_t|x_{t-1})$, we recover the bootstrap filter (which does not require to be able to evaluate $p_t^\theta(x_t|x_{t-1})$ pointwise).

Particle filters: pseudo-code

All operations to be performed for all $n \in 1:N$.

At time 0:

(a) Generate $X_0^n \sim M_0^\theta(\mathrm{d}x_0)$.

(b) Compute $w_0^n = G_0^\theta(X_0^n)$, $W_0^n = w_0^n / \sum_{m=1}^N w_0^m$, and $L_0^N = N^{-1} \sum_{n=1}^N w_0^n$.

Recursively, for $t = 1, \ldots, T$:

(a) Generate ancestor variables $A_t^n \in 1:N$ independently from $\mathcal{M}(W_{t-1}^{1:N})$.

(b) Generate $X_t^n \sim M_t^\theta(X_{t-1}^{A_t^n}, \mathrm{d}x_t)$.

(c) Compute $w_t^n = G_t^\theta(x_{t-1}, x_t)$, $W_t^n = w_t^n / \sum_{m=1}^N w_t^m$, and $L_t^N(\theta) = L_{t-1}^N(\theta) \times \{N^{-1} \sum_{n=1}^N w_t^n\}$.

Unbiased likelihood estimator

A by-product of PF output is that

$$L_T^N(\theta) = \left(\frac{1}{N} \sum_{n=1}^{N} G_0^\theta(X_0^n) \right) \prod_{t=1}^{T} \left(\frac{1}{N} \sum_{n=1}^{N} G_t^\theta(x_{t-1}, x_t) \right)$$

is an *unbiased* estimator of the likelihood $L_T(\theta) = p(y_{0:T}|\theta)$.

(Not trivial, see e.g Proposition 7.4.1 in Pierre Del Moral's book.)

PMCMC

Breakthrough paper of Andrieu et al. (2011), based on the unbiasedness of the PF estimate of the likelihood.

Marginal PMCMC

From current point θ_m (and current PF estimate $L_T^N(\theta_m)$):

1. Sample $\theta_\star \sim H(\theta_m, \mathrm{d}\theta_\star)$
2. Run a PF so as to obtain $L_T^N(\theta_\star)$, an unbiased estimate of $L_T(\theta_\star) = p(y_{0:T}|\theta_\star)$.
3. With probability $1 \wedge r$, set $\theta_{m+1} = \theta_\star$, otherwise $\theta_{m+1} = \theta_m$ with
$$r = \frac{p(\theta_\star)L_T^N(\theta_\star)h(\theta_m|\theta_\star)}{p(\theta_m)L_T^N(\theta_m)h(\theta_\star|\theta_m)}$$

Validity

Property 1

Let $\psi_{T,\theta}(\mathrm{d}x_{0:T}^{1:N}, \mathrm{d}a_{1:T}^{1:N})$ be the joint dist' of all the the rv's generated by a PF (for fixed θ), then

$$\pi_T(\mathrm{d}\theta, \mathrm{d}x_{0:T}^{1:N}, \mathrm{d}a_{1:T}^{1:N}) = \frac{p(\mathrm{d}\theta)}{p(y_{0:T})}\psi_{T,\theta}(\mathrm{d}x_{0:T}^{1:N}, \mathrm{d}a_{1:T}^{1:N})L_T^N(\theta)$$

is a joint pdf, such that the θ-marginal is $p(\theta|y_{0:T})\mathrm{d}\theta$.

Proof: fix θ, and integrate wrt the other variables:

$$\begin{aligned}\int \pi_T(\cdot) &= \frac{p(\theta)}{p(y_{0:T})}\mathbb{E}\left[L_T^N(\theta)\right]\mathrm{d}\theta \\ &= \frac{p(\theta)p(y_{0:T}|\theta)}{p(y_{0:T})}\mathrm{d}\theta = p(\theta|y_{0:T})\mathrm{d}\theta\end{aligned}$$

More direct proof for $T = 1$

$$\psi_{1,\theta}(\mathrm{d}x_{0:1}^{1:N}, \mathrm{d}a_1^{1:N}) = \prod_{n=1}^{N} M_0^\theta(\mathrm{d}x_0^n) \left\{ \prod_{n=1}^{N} M_1^\theta(x_0^{a_1^n}, \mathrm{d}x_1^n) W_{0,\theta}^{a_1^n} \mathrm{d}a_1^n \right\}$$

with $W_{0,\theta}^n = G_0^\theta(x_0^n) / \sum_{m=1}^N G_0^\theta(x_0^m)$. So

$$\pi_1(\cdot) = \frac{p(\theta)}{p(y_{0:t})} \psi_{1,\theta}(\cdot) \left\{ \frac{1}{N} \sum_{n=1}^{N} G_0^\theta(x_0^n) \right\} \left\{ \frac{1}{N} \sum_{n=1}^{N} G_1^\theta(x_0^{a_1^n}, x_1^n) \right\}$$

$$= \frac{p(\theta)}{N^2 p(y_{0:t})} \sum_{n=1}^{N} G_1^\theta(x_0^{a_1^n}, x_1^n) M_1^\theta(x_0^{a_1^n}, x_1^n) \frac{G_0^\theta(x_0^{a_1^n})}{\sum_{m=1}^N G_0^\theta(x_0^m)} \left\{ \sum_{m=1}^{N} G_0^\theta(x_0^m) \right\}$$

$$\times M_0^\theta(\mathrm{d}x_0^{a_1^n}) \left\{ \prod_{i \neq a_1^n} M_0^\theta(\mathrm{d}x_0^i) \right\} \left\{ \prod_{i \neq n} M_1^\theta(x_0^{a_1^i}, \mathrm{d}x_1^i) W_1^{a_1^i} \mathrm{d}a_1^i \right\}$$

Interpretation

$$\pi_1(\mathrm{d}\theta, \mathrm{d}x_{0:1}^{1:N}, \mathrm{d}a_1^{1:N}) = \frac{1}{N} \times \left[\frac{1}{N} \sum_{n=1}^{N} p(\mathrm{d}\theta, \mathrm{d}x_0^{a_1^n}, \mathrm{d}x_1^n | y_{0:1}) \right.$$
$$\left. \prod_{i \neq a_1^n} M_0^\theta(\mathrm{d}x_0^i) \left\{ \prod_{i \neq n} M_1^\theta(x_0^{a_1^i}, \mathrm{d}x_1^i) W_0^{a_1^i} \right\} \right]$$

which is a mixture distribution, with probability $1/N$ that path n follows $p(\mathrm{d}\theta, \mathrm{d}x_{0:1}|y_{0:1})$, A_1^n is Uniform in $1:N$, and other paths follows a conditional SMC distribution (the distribution of a particle filter conditional on one trajectory being fixed). From this calculation, one easily deduce the unbiasedness property (directly!) but also properties similar to those of the GIMH.

Additional properties (similar to GIMH)

Property 2

Marginal PMCMC is a Metropolis sampler with invariant distribution π_T, and proposal distribution $h(\theta_\star|\theta)\mathrm{d}\theta_\star\psi_{T,\theta_\star}(\cdot)$. (In particular, it leaves invariant the posterior $p(\mathrm{d}\theta|y_{0:T})$.)

Proof: write the MH ratio, same type of cancellations as for GIMH.

Additional properties (similar to GIMH)

Property 3

If we extend π_T by adding component $k \in 1:N$ with conditional probability $\propto W_T^k$, then the joint pdf $\pi_T(\mathrm{d}\theta, \mathrm{d}x_{0:T}^{1:N}, \mathrm{d}a_{1:T-1}^{1:N}, \mathrm{d}k)$ is such that

(a) $(\theta, X_{0:T}^\star) \sim p(\mathrm{d}\theta, \mathrm{d}x_{0:T} | y_{0:T})$ marginally; and

(b) Given $(\theta, X_{0:T}^\star)$, the $N-1$ remaining trajectories follow the conditional SMC distribution.

where $X_{0:T}^\star$ is the $k-$th *complete* trajectory: $X_t^\star = X_t^{B_t}$ for all t, with $B_T = k$, $B_{T-1} = A_T^k$, ... $B_0 = A_1^{B_1}$.

Outline

Don't listen to Jeff!

Proposal: Gaussian random walk, variance Σ.
Naive approach:

- Fix N
- target acceptance rate 0.234

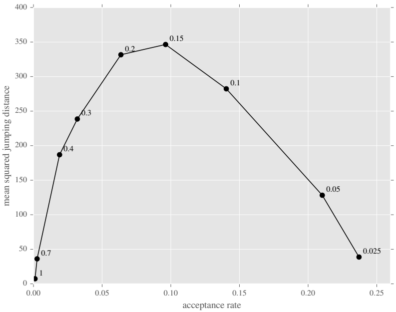


Figure: Acceptance rate vs N, when $\Sigma = \tau I_3$, and τ varies, PMMH for a toy linear Gaussian model

Recommended approach

- Through pilot runs, try to find N such that variance of log-likelihood estimate is $<< 1$;
- Then calibrate in order to minimise the SJD (squared jumping distance) or some other criterion;
- "Best" acceptance rate will be $<< 0.234$.
- Adaptative MCMC is kind of dangerous in this context; consider SMC^2 instead.

Also: state-space model likelihoods are nasty

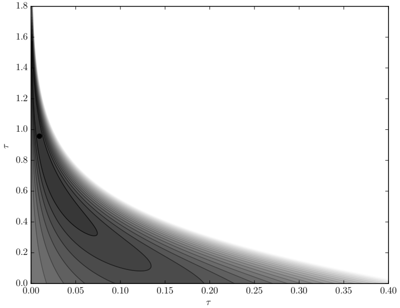


Figure: Log-likelihood contour for nutria data and Ricker state-space model (third parameter is fixed).

Outline

CSMC

- The formalisation of PMCMC offers the possibility to regenerate the $N-1$ trajectories that have not been selected; this is essentially a Gibbs step, conditional on θ, and the selected trajectory $X^{\star}_{0:T}$.
- This CSMC step cannot be analysed with the same tools as marginal PMCMC, as in Andrieu and Vihola (2012).

From now on, we drop θ from the notations.

Algorithmic description ($T = 1$)

Assume selected trajectory is $X^\star_{0:1} = (X_0^1, X_1^1)$; i.e. $k = 1$, $A_1^k = 1$.
At time $t = 0$:

(a) sample $X_0^n \sim M_0(\mathrm{d}x_0)$ for $n \in 2:N$.

(b) Compute weights $w_0^n = G_0(X_0^n)$ and normalise, $W_0^n = w_0^n / \sum_{m=1}^N w_0^m$.

At time $t = 1$:

(a) Sample $A_1^{2:N}$ $\mathcal{M}(W_0^{1:N})$.

(b) Sample $X_1^n \sim M_1(X_1^{A_0^n}, \mathrm{d}x_1)$ for $n \in 2:N$.

(c) Compute weights $w_1^n = G_1(X_0^{A_1^n}, X_1^n)$ and normalise, $W_1^n = w_1^n / \sum_{m=1}^N w_1^m$.

(d) select new trajectory k with probability W_1^k.

then return $\tilde{X}^\star_{0:1} = (X_0^{A_1^k}, X_1^k)$.

Some remarks

- One may show that the CSMC update does not depend on the labels of the frozen trajectory. This is why we set these arbitrarily to $(1, \ldots, 1)$. Formally, this means that the CSMC kernel is such that $K^N_{\mathrm{CSMC}} : \mathcal{X}^T \to \mathcal{P}(\mathcal{X}^T)$.
- This remains true for other resampling schemes (than multinomial); see next two* slides for an example

Properties of the CSMC kernel

Theorem

Under appropriate conditions, one has, for any $\varepsilon > 0$,

$$\left| K^N_{\mathrm{CSMC}}(\varphi)(x_{0:T}) - K^N_{\mathrm{CSMC}}(\varphi)(x'_{0:T}) \right| \leq \varepsilon$$

for N large enough, and $\varphi : \mathcal{X}^T \to [-1, 1]$.

This implies uniform ergodicity. Proof based on a coupling construction.

Assumptions

- G_t is upper bounded, $G_t(x_t) \leq g_t$.
- We have

$$\int M_0(dx_0) G_0(x_0) \geq \frac{1}{g_0}, \quad \int M_t(x_{t-1}, dx_t) G_t(x_t) \geq \frac{1}{g_t}$$

But no assumptions on the kernels M_t.

Backward sampling

Nick Whiteley (in his RSS discussion of PMCMC) suggested to add an extra *backward* step to CSMC, where one tries to modify (recursively, backward in time) the ancestry of the selected trajectory.
In our $T = 1$ example, and for multinomial resampling, this amounts to draw A_1^k from

$$\mathbb{P}(A_1^k = a | k, x_{0:1}^{1:N}) \propto W_0^a m_1(x_1^k | x_0^a)$$

where $m_1(x_1^k | x_0^a)$ is the PDF at point x_1^k of $M_1(x_0^a, dx_1)$, then return $x_{0:1}^\star = (x_0^a, x_1^k)$.

BS for other resampling schemes

More generally, BS amounts to draw a_1^k from

$$P(a_1^k = a | k, x_{1:2}^{1:N}) \propto \rho_1(W_1^{1:N}; a_1^k = a | a_1^{-k}) m_2(x_1^a, x_2^k)$$

where a_1^{-k} is $a_1^{1:N}$ minus a_1^k.

So we need to be able the conditional probability $\rho_1(W_1^{1:N}; a_1^k = a | a_1^{-k})$ for alternative resampling schemes.

Why BS would bring an improvement?

C. and Singh (2014) prove that CSMC+BS dominates CSMC in efficiency ordering (i.e. asymptotic variance). To do so, they prove that these two kernels are reversible; see Tierney (1998), Mira & Geyer (1999).

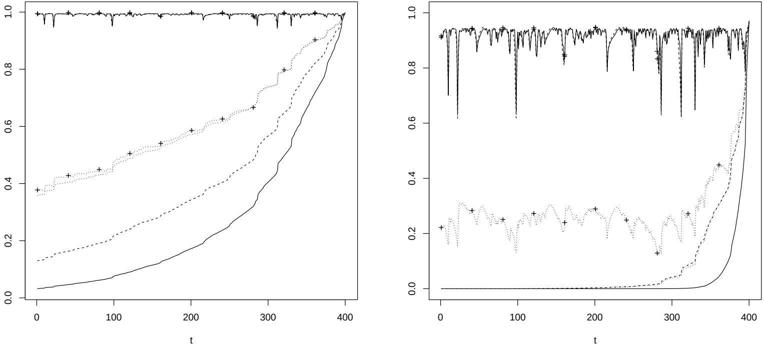
Simulations

See the plots in next slide, based on the following simple state-space model, with $\theta = (\mu, \phi, \sigma)$:

$$x_t - \mu = \phi(x_{t-1} - \mu) + \sigma\epsilon_t, \quad \epsilon_t \sim N(0,1)$$

$$y_t | x_t \sim \mathrm{Poisson}(e^{x_t})$$

Update rate of X_t



Left: $N = 200$, right: $N = 20$. Solid line: multinomial, Dashed line: residual; Dotted line: Systematic. Crosses mean BS has been used.

Conclusion

- When the backward step is possible, it should be implemented, because it improves mixing dramatically. In that case, multinomial resampling is good enough.
- When the backward step cannot be implemented, switching to systematic resampling helps.

But what's the point of PG?

It's a bit the same discussion as marginal Metropolis (in θ-space) versus Gibbs:

- Gibbs does not work so well when they are strong correlations (here between θ and $X^{\star}_{0:T}$);
- Metropolis requires a good proposal to work well.

In some cases, combining the two is helpful: in this way, the CSMC update will refresh the particle system, which may help to get "unstuck".

SMC^2

Outline

1. SMC^2
2. Conclusion

Preliminary

So far, we have played with replacing intractable quantities with unbiased estimates within Metropolis samplers. Note however we could do the same within an importance sampler. For instance, the following approach has been used in Chopin and Robert (2007).

To compute the evidence $p(y)$ of some state-space model

- Sample points θ^n from the prior $p(\theta)$.
- For each θ^n, run a PF (for fixed $\theta = \theta^n$) to obtain an estimate $\hat{p}(y|\theta^n)$ of the likelihood.
- Compute

$$\hat{p}(y) = \frac{1}{N}\sum_{n=1}^{N} \hat{p}(y|\theta^n)$$

Objectives

1. to derive sequentially

$$p(\mathrm{d}\theta, \mathrm{d}x_{0:t} | Y_{0:t} = y_{0:t}), \quad p(y_{0:t}), \quad \text{for all } t \in \{0, \ldots, T\}$$

2. to obtain a black box algorithm (automatic calibration).

Main tools of our approach

- Particle filter algorithms for state-space models (this will be to estimate the likelihood, for a fixed θ).
- Iterated Batch Importance Sampling for sequential Bayesian inference for parameters (this will be the theoretical algorithm we will try to approximate).

Both are sequential Monte Carlo (SMC) methods.

IBIS (C., 2001)

SMC method for particle approximation of the sequence $p(\theta|y_{0:t})$, $t = 0 : T$. Based on the sequence of importance sampling steps:

$$\frac{p(\theta|y_{0:t})}{p(\theta|y_{0:t-1})} \propto p(y_t|y_{0:t-1}, \theta)$$

but doing only IS steps would not well. Resampling alone will not help, because θ is not an ergodic process.

$\Rightarrow$ introduces an artificial dynamics by moving the θ particles through a MCMC step (that leaves $p(\theta|y_{0:t})$ invariant).
In next slide, operations with superscript m must be understood as operations performed for all $m \in 1 : N_\theta$, where N_θ is the total number of θ-particles.

Sample θ^m from $p(\theta)$ and set $\omega^m \leftarrow 1$. Then, at time $t = 0, \ldots, T$

(a) Compute incremental weights

$$u_t(\theta^m) = p(y_t | y_{0:t-1}, \theta^m), \quad L_t = \frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \times \sum_{m=1}^{N_\theta} \omega^m u_t(\theta^m),$$

(b) Update the importance weights,

$$\omega^m \leftarrow \omega^m u_t(\theta^m). \tag{1.1}$$

(c) If some degeneracy criterion is fulfilled, sample $\tilde{\theta}^m$ independently from the mixture distribution

$$\frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \sum_{m=1}^{N_\theta} \omega^m K_t\left(\theta^m, \cdot\right).$$

Finally, replace the current weighted particle system:

$$(\theta^m, \omega^m) \leftarrow (\tilde{\theta}^m, 1).$$

Observations

- Cost of lack of ergodicity in θ: the occasional MCMC move
- Still, in regular problems resampling happens at diminishing frequency (logarithmically)
- K_t is an MCMC kernel invariant wrt $\pi(\theta \mid y_{1:t})$. Its parameters can be chosen using information from current population of θ-particles
- L_t is a MC estimator of the model evidence
- Infeasible to implement for state-space models: intractable incremental weights, and MCMC kernel

Our algorithm: SMC^2

We provide a generic (black box) algorithm for recovering the sequence of parameter posterior distributions, but as well filtering, smoothing and predictive.

We give next a pseudo-code; the code seems to only track the parameter posteriors, but actually it does all other jobs. Superficially, it looks an approximation of IBIS, but in fact it does not produce any systematic errors (unbiased MC).

Sample θ^m from $p(\theta)$ and set $\omega^m \leftarrow 1$. Then, at time $t = 0, \ldots, T$,

(a) For each particle θ^m, perform iteration t of the PF: If $t = 0$, sample independently $X_0^{1:N_x,m}$ from ψ_{0,θ^m}, and compute

$$\hat{p}(y_0|\theta^m) = \frac{1}{N_x}\sum_{n=1}^{N_x} w_0^{\theta}(x_0^{n,m});$$

If $t > 0$, sample $\left(X_t^{1:N_x,m}, A_t^{1:N_x,m}\right)$ from ψ_{t,θ^m} conditional on $\left(X_{0:t-1}^{1:N_x,m}, A_{1:t-1}^{1:N_x,m}\right)$, and compute

$$\hat{p}(y_t|y_{1:t-1},\theta^m) = \frac{1}{N_x}\sum_{n=1}^{N_x} w_t^{\theta}(X_{t-1}^{A_t^{n,m},m}, X_t^{n,m}).$$

(b) Update the importance weights,

$$\omega^m \leftarrow \omega^m \hat{p}(y_t|y_{0:t-1}, \theta^m)$$

(c) If some degeneracy criterion is fulfilled, sample $\left(\tilde{\theta}^m, \tilde{X}_{0:t}^{1:N_x,m}, \tilde{A}_{1:t}^{1:N_x}\right)$ independently from

$$\frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \sum_{m=1}^{N_\theta} \omega^m K_t \left\{ \left(\theta^m, x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m}\right), \cdot \right\}$$

Finally, replace current weighted particle system:

$$(\theta^m, X_{0:t}^{1:N_x,m}, A_{1:t}^{1:N_x,m}, \omega^m) \leftarrow (\tilde{\theta}^m, \tilde{X}_{0:t}^{1:N_x,m}, \tilde{A}_{1:t-1}^{1:N_x,m}, 1)$$

Observations

- It appears as approximation to IBIS. For $N_x = \infty$ it is IBIS.
- However, no approximation is done whatsoever. This algorithm really samples from $p(\theta|y_{0:t})$ and all other distributions of interest.
- The validity of algorithm is essentially based on two results: i) the particles are weighted due to unbiasedness of PF estimator of likelihood; ii) the MCMC kernel is appropriately constructed to maintain invariance wrt to an expanded distribution which admits those of interest as marginals; it is a Particle MCMC kernel.
- The algorithm does not suffer from the path degeneracy problem due to the MCMC updates.

The MCMC step

(a) Sample $\tilde{\theta}$ from proposal kernel, $\tilde{\theta} \sim h(\theta, d\tilde{\theta})$.

(b) Run a new PF for $\tilde{\theta}$: sample independently $(\tilde{X}_{0:t}^{1:N_x}, \tilde{A}_{1:t}^{1:N_x})$ from $\psi_{t,\tilde{\theta}}$, and compute $\hat{L}_t(\tilde{\theta}, \tilde{X}_{0:t}^{1:N_x}, \tilde{A}_{1:t-1}^{1:N_x})$.

(c) Accept the move with probability

$$1 \wedge \frac{p(\tilde{\theta})\hat{L}_t(\tilde{\theta}, \tilde{X}_{0:t}^{1:N_x}, \tilde{A}_{1:t}^{1:N_x})h(\tilde{\theta}, \theta)}{p(\theta)\hat{L}_t(\theta, X_{0:t}^{1:N_x}, A_{1:t}^{1:N_x})h(\theta, \tilde{\theta})}.$$

It can be shown that this is a standard Hastings-Metropolis kernel with proposal

$$q_\theta(\tilde{\theta}, \tilde{x}_{0:t}^{1:N_x}, \tilde{a}_{1:t}^{1:N_x}) = h(\theta, \tilde{\theta})\psi_{t,\tilde{\theta}}(\tilde{x}_{0:t}^{1:N_x}, \tilde{a}_{1:t}^{1:N_x})$$

invariant w.r.t. to an extended distribution $\pi_t(\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x})$.

Some advantages of the algorithm

- Immediate estimates of filtering and predictive distributions
- Immediate and sequential estimator of model evidence.
- Easy recovery of smoothing distributions.
- Principled framework for automatic calibration of N_x.
- Population Monte Carlo advantages.

Validity

SMC^2 is simply a SMC sampler with respect to the sequence:

$$\pi_t(\mathrm{d}\theta, \mathrm{d}x_{0:t}^{1:N_x}, \mathrm{d}a_{1:t}^{1:N_x})$$

- the reweigthing step $t-1 \to t$ (a) extends the dimension, by sampling $X_t^{1:N}, a_t^{1:N}$; and (b) computes $\pi_t(\cdot)/\pi_{t-1}(\cdot)$.
- The move step is a PMCMC step that leaves π_t invariant.

Technical point

As in PMCMC, one may extend π_t by adding index k that picks some trajectory, which, jointly with θ, is sampled from the current posterior $p(\theta, x_{0:t}|y_{0:t})$. However, it is more difficult to define an importance sampling step with respect to the extended space (that includes k), so, we must discard k before progressing to time $t+1$.

How to choose N_x?

PMCMC: valid whatever N_x, but one needs to take $N_x = O(T)$ in order to obtain a non-negligible acceptance rate. This is related to the following type of results (Cérou et al, 2011; Whiteley, 2011):

$$\mathrm{Var}[\hat{p}(y_{0:T}|\theta)] \leq \frac{CT}{N_x}.$$

For SMC^2, this suggests that one should start with a small value, then increases N_x progressively. But:

1. how to increase N_x at a given time?
2. when should we increase N_x?

How to increase N_x

Two possible strategies to replace our PF's of size N_x with PF's of size N'_x at iteration t:

1. exchange step: generate a new PF of size N'_x, then do an importance sampling step in order to swap the old PF and the new PF.
2. a CSMC (Particle Gibbs step), when we select one trajectory, throw away the $N_x - 1$ remaining ones, and regenerate $N'_x - 1$ new trajectories using CSMC.

The latter should suffer less from weigh degeneracy, but it suffers from a higher memory cost, i.e. $O(TN_xN_\theta)$ at time t.

When to increase N_x?

Currently, we monitor the acceptance rate of the PMCMC rejuvenation step; when it's too small, we trigger an exchange step (from N_x to $2N_x$).

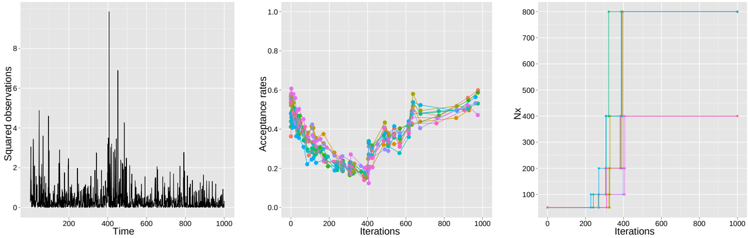
We're working on more refined versions based on PG steps, and better criteria to determine when and by how much we should increase N_x (on-going work).

Complexity

The overall complexity of SMC^2 is $O(N_\theta T^2)$ if run until time T:

1. The cost of iteration t without a rejuvenation step is $O(N_\theta N_x)$;
2. as explained before, we need to increase N_x progressively, $N_x = O(t)$;
3. The cost of the PMCMC rejuvenation step is $O(tN_\theta N_x)$, but we obtained the following result: if it is triggered whenever ESS$< \gamma$, and $N_x = O(t)$, then the occurence times are geometric $(\tau^k,\ k = 1, 2, \ldots)$.

Numerical illustrations: SV



(a) (b) (c)

Figure: Squared observations (synthetic data set), acceptance rates, and illustration of the automatic increase of N_x.

See the model

Numerical illustrations: SV

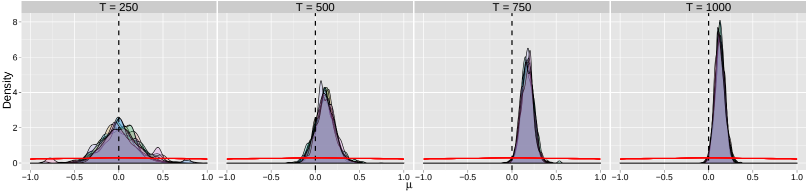


Figure: Concentration of the posterior distribution for parameter μ.

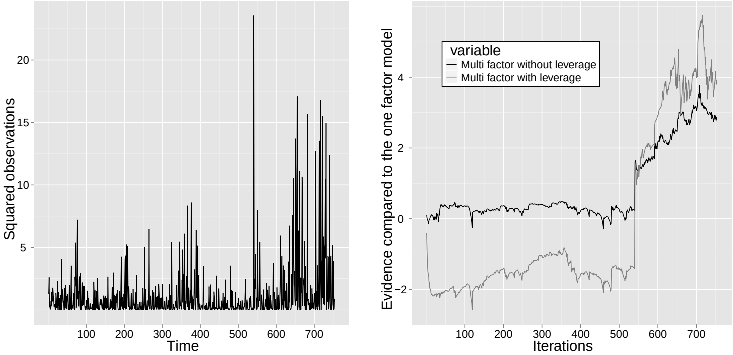
Numerical illustrations: SV

Multifactor model

$$y_t = \mu + \beta v_t + v_t^{1/2}\epsilon_t + \rho_1 \sum_{j=1}^{k_1} e_{1,j} + \rho_2 \sum_{j=1}^{k_2} e_{2,j} - \xi(w\rho_1\lambda_1 + (1-w)\rho_2\lambda_2)$$

where $v_t = v_{1,t} + v_{2,t}$, and $(v_i, z_i)_{n=1,2}$ are following the same dynamics with parameters $(w_i\xi, w_i\omega^2, \lambda_i)$ and $w_1 = w$, $w_2 = 1 - w$.

Numerical illustrations: SV



(a) (b)

Figure: S&P500 squared observations, and log-evidence comparison between models (relative to the one-factor model).

Numerical illustrations

Athletics records model

$$g(y_{1:2,t}|\mu_t,\xi,\sigma) = \{1 - G(y_{2,t}|\mu_t,\xi,\sigma)\} \prod_{n=1}^{2} \frac{g(y_{i,t}|\mu_t,\xi,\sigma)}{1 - G(y_{i,t}|\mu_t,\xi,\sigma)}$$

$$x_t = (\mu_t, \dot{\mu}_t)', \quad x_{t+1} \mid x_t, \nu \sim \mathcal{N}(Fx_t, Q) ,$$

with

$$F = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \text{ and } Q = \nu^2 \begin{pmatrix} 1/3 & 1/2 \\ 1/2 & 1 \end{pmatrix}$$

$$G(y|\mu,\xi,\sigma) = 1 - \exp\left[- \left\{ 1 - \xi \left(\frac{y-\mu}{\sigma} \right) \right\}_{+}^{-1/\xi} \right]$$

Numerical illustrations

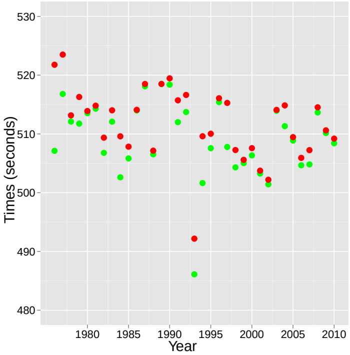


Figure: Best two times of each year, in women's 3000 metres events between 1976 and 2010.

Numerical illustrations: Athletics records

Motivating question

How unlikely is Wang Junxia's record in 1993?

A smoothing problem

We want to estimate the likelihood of Wang Junxia's record in 1993, given that we observe a better time than the previous world record. We want to use all the observations from 1976 to 2010 to answer the question.

Note

We exclude observations from the year 1993.

See the model

Numerical illustrations

Some probabilities of interest

$$p_t^y = \mathbb{P}(y_t \leq y | y_{1976:2010})$$
$$= \int_\Theta \int_\mathcal{X} G(y|\mu_t, \theta) p(\mu_t | y_{1976:2010}, \theta) p(\theta | y_{1976:2010}) \, d\mu_t d\theta$$

The interest lies in $p_{1993}^{486.11}$, $p_{1993}^{502.62}$ and $p_t^{cond} := p_t^{486.11} / p_t^{502.62}$.

Numerical illustrations

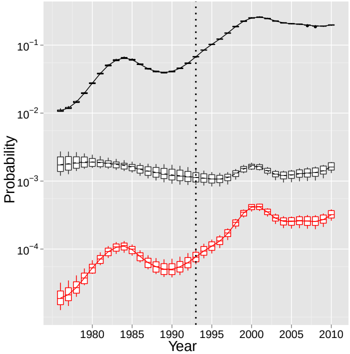


Figure: Estimates of the probability of interest (top) $p_t^{502.62}$, (middle) p_t^{cond} and (bottom) $p_t^{486.11}$, obtained with the SMC^2 algorithm. The y-axis is in log scale, and the dotted line indicates the year 1993 which motivated the study.

Final Remarks on SMC^2

A powerful framework

- A generic algorithm for sequential estimation and state inference in state space models: only requirements are to be able (a) to simulate the Markov transition $p_t^\theta(x_t|x_{t-1})$, and (b) to evaluate the likelihood term $f_t^\theta(y_t|x_t)$.
- The article is available on arXiv and our web pages
- A package is available at:

http://code.google.com/p/py-smc2/.

Outline

General conclusions

- Auxiliary variables algorithms are not so complicated, when they are understood as standard samplers on extended spaces.
- offers excellent performance, at little cost (in the user's time dimension); almost magic.
- Many applications not yet fully explored; e.g. variable selection, see C. Schäfer's PhD. thesis.
- Many avenues for future research, e.g. the active particle framework of Anthony Lee (work with Arnaud Doucet and Christophe Andrieu).

References

- C, P. Jacob, and O. Papaspiliopoulos. "SMC2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates." JR Stat. Soc. B (2013)
- C and S.S. Singh. "On the particle Gibbs sampler." Bernoulli (2014, in press).