Perfect simulation algorithm of a trajectory under a Feynman-Kac law

Data Assimilation, 24th-28th September 2012, Oxford-Man Institute

C. Andrieu, N. Chopin, A. Doucet, S. Rubenthaler

University of Bristol, CREST-ENSAE, University of Oxford, Université de Nice-Sophia Antipolis
Perfect simulation algorithm of a trajectory under a Feynman-Kac law

Introduction

Notations

- $X_1, X_2, \ldots$ is a Markov chain in $E$ with initial law $M_1$ and transition $M$ (say $E = \mathbb{R}^d$ or $\mathbb{Z}^d$)
- $G_1, G_2, \cdots : E \to \mathbb{R}_+$ are potentials
- total time : $P$

One is interested in the law :

$$
\pi(f) = \frac{\mathbb{E}(f(X_1, \ldots, X_P) \prod_{i=1}^{P-1} G_i(X_i))}{\mathbb{E}(\prod_{i=1}^{P-1} G_i(X_i))}.
$$
Approximation

We can take a SMC approximating $\eta_P(f) = \frac{\mathbb{E}(f(X_P) \prod_{i=1}^{P-1} G_i(X_i))}{\mathbb{E}(\prod_{i=1}^{P-1} G_i(X_i))}$ and then take an ancestral line.
Perfect simulation

Example

If $G_i : E \rightarrow [0, 1]$, draw $X_1, \ldots, X_P$, $U \sim U([0, 1])$ until

$$G_1(X_1)G_2(X_1)\ldots G_{P-1}(X_{P-1}) \geq U.$$ 

Cost is exponential in $P$.

Here we will use

- a Metropolis-like algorithm on an extended space (ideas from Andrieu, Doucet, Holenstein, Particle Markov chain Monte Carlo methods, Journal of the Royal Statistical Society: Series B (Statistical Methodology)),
- coupling from the past.
Branching system

- Start with $N_1$ particles.
- The particle $X^i_n$ ($i$-th particle at time $n$) has $A^i_{n+1}$ offsprings with law $\mathbb{P}(A^i_{n+1} = j) = f_{n+1}(G_n(X^i_n), j)$ (independant of other particles).
- Total number of particles : $N_{n+1} = \sum_{i=1}^{N_n} A^i_{n+1}$.

Density :

$$q_0(N_2, \ldots, N_P, (A^i_n), (X^i_n)) = \prod_{i=1}^{N_1} M_1(X^i_1) \prod_{n=2}^{P} \left( \prod_{i=1}^{N_{n-1}} f_n(G_{n-1}(X^i_{n-1}), A^i_n) \prod_{j \in \ldots} M(X^i_{n-1}, X^j_n) \right)$$
Perfect simulation algorithm of a trajectory under a Feynman-Kac law

Metropolis-like algorithm on extended space

Branching system
How to make it work

We do not want $N_n \to 0$ or $N_n \to +\infty$.

- First run a SMC system $(\xi^i_n)$ ($1 \leq n \leq P$, $1 \leq i \leq N_1$).
- At time $n$, the particles of the branching system are such that $\frac{1}{N_n} \sum_{i=1}^{N_n} \delta x^i_n \approx \eta_n$.
- Take $f_n$ such that $\frac{1}{N_n} \sum_{i=1}^{N_n} \sum_{j=1}^{+\infty} j f_n(G_n(\xi^i_n), j) = 1$, we get

$$
\frac{1}{N_n} \sum_{i=1}^{N_n} \sum_{j=1}^{+\infty} j f_n(G_n(x^i_n), j) = 1. \quad (1)
$$
Extension

Take a trajectory and draw a branching system conditionned to contain this trajectory.
Perfect simulation algorithm of a trajectory under a Feynman-Kac law
Metropolis-like algorithm on extended space

**Extension**

Take a trajectory and draw a branching system conditioned to contain this trajectory.
Extension

When a particle is blue at position \( x \) at time \( n - 1 \), the number of children is chosen with law:

\[
\mathbb{P}(\text{j offsprings}) = \hat{f}_n(G_{n-1}(x), j) = \frac{f_n(G_{n-1}(x), j)}{1 - f_n(G_{n-1}(x), 0)}.
\]
Extension

When a particle is blue at position $x$ at time $n - 1$, the number of children is chosen with law:

$\mathbb{P}(j \text{ offsprings}) = \hat{f}_n(G_{n-1}(x), j) = \frac{f_n(G_{n-1}(x), j)}{1 - f_n(G_{n-1}(x), 0)}.$

We choose $f_n$ such that $f_n(g, 0) = 1 - \frac{g}{\|G_n\|_\infty}$, $f_n(g, j) = \frac{g}{k_n \|G_n\|_\infty}$, $1 \leq j \leq k_n$).

Then

$$\frac{\hat{f}_n(g, j)}{f_n(g, j)} = \frac{\|G_n\|_\infty}{g}$$

$(\forall n, j, g)$ and (1) is easy to fulfill.
Target density

Start with trajectory of law $\pi$ and extend it into a forest. Density is

$$\hat{\pi}(\zeta_1, \ldots, \zeta_P, N_2, \ldots, N_P, (A^i_n), (X^i_n))$$

$$= \pi(\zeta_1, \ldots, \zeta_P) \times \frac{q_0(N_2, \ldots, N_P, (A^i_n), (X^i_n))}{M_1(\zeta_1) \prod_{n=2}^{P} M(\zeta_{n-1}, \zeta_n)} \times \prod_{n=2}^{P} \frac{\hat{f}_n(G_{n-1}(\zeta_{n-1}), A^i_n)}{f_n(G_{n-1}(\zeta_{n-1}), A^i_n)} .$$
Proposal

Take a branching system like above, select a particle at time $P$ and its ancestral line. We get some density $q$ on the space of (size of each generation) $\times$ (numbers of offsprings) $\times$ (positions) $\times$ (special trajectory).
Proposal

Take a branching system like above, select a particle at time $P$ and its ancestral line. We get some density $q$ on the space of (size of each generation) $\times$ (numbers of offsprings) $\times$ (positions) $\times$ (special trajectory).
Proposal

Take a branching system like above, select a particle at time $P$ and its ancestral line. We get some density $q$ on the space of (size of each generation)×(numbers of offsprings)×(positions)×(special trajectory).
Proposal

Take a branching system like above, select a particle at time $P$ and its ancestral line. We get some density $q$ on the space of (size of each generation) $\times$ (numbers of offsprings) $\times$ (positions) $\times$ (special trajectory).
Proposal

We get the density

$$q(\zeta_1, \ldots, \zeta_P, N_2, \ldots, N_P, (A^i_n), (X^i_n))$$

$$= q_0(N_2, \ldots, N_P, (A^i_n), (X^i_n)) \times \frac{1}{N_P}.$$
Acception/rejection

Target law: trajectory with the law $\pi$, to which we add a (conditionned) branching system. We have a law $\hat{\pi}$ on “forests”.

$$
\frac{\hat{\pi}(\ldots)}{q(\ldots)} = \frac{N_P \prod_{i=1}^{P-1} \| G_i \|_\infty}{N_1 Z},
$$

with $Z := \mathbb{E}(\prod_{n=1}^{P-1} G_n(X_n))$ (partition function).
Markov chain

- start with trajectory, extend it into a “forest”
Perfect simulation algorithm of a trajectory under a Feynman-Kac law

Metropolis-like algorithm on extended space

Markov chain

- start with trajectory, extend it into a “forest”
- when at point $\mathcal{X}$ (in “forests”): propose $\overline{\mathcal{X}}$, accept with probability $\hat{\pi}(\mathcal{X}) q(\mathcal{X})/\hat{\pi}(\mathcal{X}) q(\mathcal{X}) \wedge 1 \leq N_{\mathcal{P}} N_{\mathcal{P}} \wedge 1$, then prune the forest to have only the colored trajectory.
Markov chain

- start with trajectory, extend it into a “forest”
- when at point $X$ (in “forests”): propose $\overline{X}$,
- accept with probability $\frac{\hat{\pi}(\overline{X})q(\overline{X})}{\hat{\pi}(X)q(X)} \land 1 = \frac{N_P}{N_P} \land 1$,
Markov chain

- start with trajectory, extend it into a “forest”
- when at point $\mathcal{X}$ (in “forests”): propose $\overline{\mathcal{X}}$,
- accept with probability $\frac{\hat{\pi}(\mathcal{X})q(\mathcal{X})}{\hat{\pi}(\mathcal{X})q(\mathcal{X})} \wedge 1 = \frac{\overline{N}_P}{N_P} \wedge 1$,
- then prune the forest to have only the colored trajectory.
Markov chain

- start with trajectory, extend it into a “forest”
- when at point $X$ (in “forests”): propose $\overline{X}$,
- accept with probability $\frac{\hat{\pi}(\overline{X})q(\overline{X})}{\hat{\pi}(X)q(X)} \wedge 1 = \frac{N_P}{N_P} \wedge 1$,
- then prune the forest to have only the colored trajectory.

We have here a Markov process on the trajectory space whose invariant law is $\pi$. 

Perfect simulation algorithm of a trajectory under a Feynman-Kac law

Metropolis-like algorithm on extended space
Perfect simulation algorithm of a trajectory under a Feynman-Kac law

Metropolis-like algorithm on extended space

Markov chain

extension

or

proposal

acception/rejection

pruning
Coupling from the past in a nutshell

Transition $Q$ of a Markov chain expressed with uniform variables:

$$Z = h(z, U) \sim Q(z, .).$$

Suppose $Q$ has invariant law $\pi$. Take $(U_k)$ i.i.d. For a starting point $z$ and $n \geq 0$, set

$$Z_{-n}^z = z, \ Z_{-n+1}^z = h(Z_{-n}^z, U_n), \ldots, Z_0^z = h(Z_{-1}^z, U_1).$$

If $T$ such that: $\forall z, z': \ Z_{-T}^z = z, \ Z_{-T}^z' = z'$, we have $Z_0^z = Z_0^{z'}$, then $Z_0^z \sim \pi$. 
Coupling from the past algorithm

The easy case

- state space is totally ordered with a max and a min element
- $z \leq z' \Rightarrow h(z, U) \leq h(z', U)$
Coupling from the past algorithm

Start with law $\pi$

$\max$

$\min$

U_4 U_3 U_2 U_1

Time

Space
Perfect simulation algorithm of a trajectory under a Feynman-Kac law

Coupling from the past

**Coupling from the past algorithm**

![Diagram showing space-time evolution with marked time points and arrows indicating transitions between states.](image)
Detection of a coupling time

Look for a time such that the red proposal is accepted for all possible blue trajectories.

Bound the number of squares at the bottom and you bound the acceptation ratio.
Directed polymers in $\mathbb{Z}$

Draw $U(i,j)$ i.i.d. of Bernoulli law ($i \in \mathbb{N}, j \in \mathbb{Z}$). Take $(X_n)$ the simple random walk in $\mathbb{Z}$ with $X_0 = 0$. Set $G_i(j) = \exp(-\beta U(i,j))$. To draw a trajectory of length $n$, the cost is $O(n^3)$.

Figure: Green path has higher potential than blue path
Directed polymers in $\mathbb{Z}$

Figure: 500 trajectories
Directed polymers in $\mathbb{Z}$

Can recover that \( \frac{1}{\log P} \mathbb{E}_\pi(\max_{1 \leq n \leq P} |X_n|) \xrightarrow{P \to +\infty} \frac{2}{3}. \)

Figure: 100 trajectories
Perfect simulation algorithm of a trajectory under a Feynman-Kac law

Examples

Radar detection

- Transition (in $\mathbb{R}$): $M(x, dy) = \frac{1}{\sqrt{2\pi b^2}} \exp\left(-\frac{(y-ax)^2}{2b^2}\right)$
  \[ X_{n+1} = aX_n + W_{n+1}. \]
- Observations: $Y_n = X_n + c \epsilon_n$ ($\epsilon_n \sim \mathcal{N}(0, 1)$). Potential
  \[ G_n(x) = \frac{1}{\sqrt{2\pi c^2}} \exp\left(-\frac{(x-Y_n)^2}{2c^2}\right) \]

You can bound the number of offspring of any trajectory by discretizing the space (works for $a \in [-1, 1]$).

- : $X$
- - : $Y$
- - : perfect simulation