Population-based simulation for static inference

François Caron

UBC

November 20, 2007 / MLRG

Overview

1. Introduction

2. Population-based MCMC

3. Sequential Monte Carlo Samplers

4. Comparison of both approaches

Introduction

- Common problem in Bayesian statistics: approximation of

$$\mathbb{E}_{\pi}\left[h\right] = \int_{E} h(x)\pi(x)dx$$

  where $\pi(x) = \frac{\gamma(x)}{Z}$, $\gamma(x)$ can be evaluated pointwise and Z is unknown

- MCMC draw T samples $x^{(i)}$, $i = 1, \ldots, T$ approximately distributed from π with an ergodic Markov kernel K with invariant distribution π and approximate $\mathbb{E}_{\pi}\left[h\right]$ with

$$S_T(h) = \frac{1}{T}\sum_{i=1}^{T} h(x^{(i)})$$

Introduction

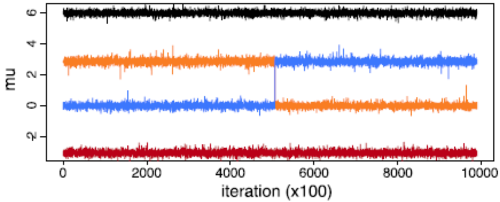
- Distribution π is often highly dimensional, multimodal

Introduction

- Distribution π is often highly dimensional, multimodal
- Lead to poor mixing properties of MCMC algorithms, based on local moves

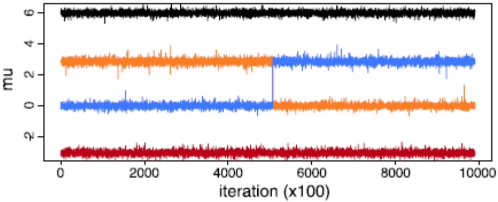
Introduction

- Distribution π is often highly dimensional, multimodal
- Lead to poor mixing properties of MCMC algorithms, based on local moves
- Example: Mixture models with K elements



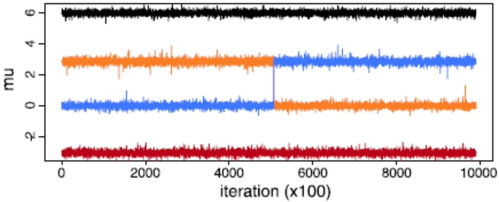
Introduction

- Distribution π is often highly dimensional, multimodal
- Lead to poor mixing properties of MCMC algorithms, based on local moves
- Example: Mixture models with K elements
- $K!$ symmetric modes - non convergence of most of the MCMC algorithms



Introduction

- Distribution π is often highly dimensional, multimodal
- Lead to poor mixing properties of MCMC algorithms, based on local moves
- Example: Mixture models with K elements
- $K!$ symmetric modes - non convergence of most of the MCMC algorithms
- Note that it is only taken as an example, as the exploration of the $K!$ is redundant from a statistical point of view



Population-based simulation

- Sequence of distributions π_n, $n = 1, \ldots N$, defined on E

Population-based simulation

- Sequence of distributions π_n, $n = 1, \dots N$, defined on E
- At least one of the distributions is the target distribution π

Population-based simulation

- Sequence of distributions π_n, $n = 1, \dots N$, defined on E
- At least one of the distributions is the target distribution π
- Idea: generate a collection of samples in parallel

Population-based simulation

- Sequence of distributions π_n, $n = 1, \ldots N$, defined on E
- At least one of the distributions is the target distribution π
- Idea: generate a collection of samples in parallel
- Population-based MCMC

Population-based simulation

- Sequence of distributions π_n, $n = 1, \dots N$, defined on E
- At least one of the distributions is the target distribution π
- Idea: generate a collection of samples in parallel
- Population-based MCMC
- Sequential Monte Carlo Sampler

Example

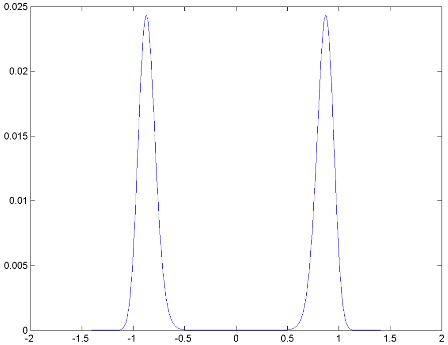
- For $t = 1, \ldots, T$

$$z_t \sim \mathcal{N}(x^2, 1)$$

where

$$x \sim \mathcal{N}(0, 3)$$

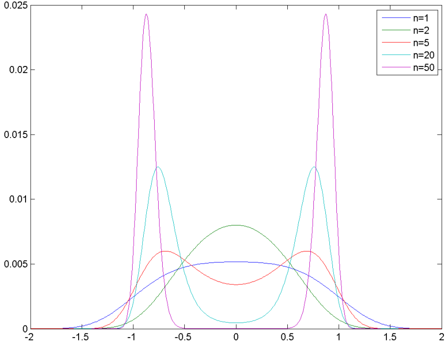
- Bimodal posterior distribution $p(x|z_{1:T})$



Example

- Sequence of distributions for $n = 1, \dots, T$

$$\pi_n(x) \propto p(z_{1:n}|x)p(x)$$

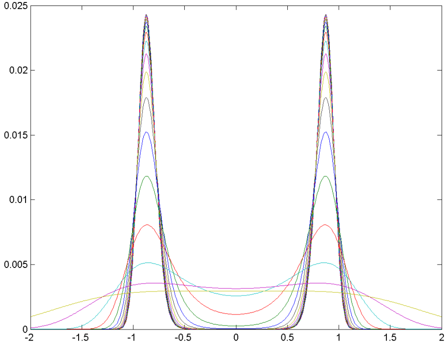


Example

- Sequence of distributions for $n = 1, \ldots, N$

$$\pi_n(x) \propto p(z_{1:T}|x)^{\xi_n} p(x)$$

where ξ_n is a non decreasing sequence in $[0, 1]$ and $\xi_N = 1$.



Population-based MCMC

- Sequence of distributions π_n, $n = 1, \dots N$

Population-based MCMC

- Sequence of distributions π_n, $n = 1, \dots N$
- New target

$$\pi^*(x_{1:N}) = \prod_{n=1}^{N} \pi_n(x_n)$$

where $\pi \equiv \pi_n$ for at least one $n \in \{1, \dots, N\}$.

Population-based MCMC

- Sequence of distributions π_n, $n = 1, \dots N$
- New target

$$\pi^*(x_{1:N}) = \prod_{n=1}^{N} \pi_n(x_n)$$

where $\pi \equiv \pi_n$ for at least one $n \in \{1, \dots, N\}$.

- Markov kernel which admits π^* as its invariant distribution

Population-based MCMC

- Sequences of densities chosen so that they are easy to sample compared to the target density π
- Population of samples allows more global moves than a single chain

Population moves

- Mutation: update a single component of the population with a Markov kernel $K(x_n, \cdot)$ which is π_n stationary

Population moves

- Mutation: update a single component of the population with a Markov kernel $K(x_n, \cdot)$ which is π_n stationary
- Exchange: Swap the value of two chains n and q, accepted with probability $\min(1, \frac{\pi_n(x_q)\pi_q(x_n)}{\pi_n(x_n)\pi_q(x_q)})$

Population moves

- Mutation: update a single component of the population with a Markov kernel $K(x_n, \cdot)$ which is π_n stationary
- Exchange: Swap the value of two chains n and q, accepted with probability $\min(1, \frac{\pi_n(x_q)\pi_q(x_n)}{\pi_n(x_n)\pi_q(x_q)})$
- Crossover: Crossover the l^{th} position in the vector for chains n and q such that

$$
\begin{aligned}
x'_n &= (x_{1n}, \ldots, x_{(l-1)n}, x_{lq}, \ldots, x_{dq}) \\
x'_q &= (x_{1q}, \ldots, x_{(l-1)q}, x_{ln}, \ldots, x_{dn})
\end{aligned}
$$

Population moves

- Mutation: update a single component of the population with a Markov kernel $K(x_n, \cdot)$ which is π_n stationary
- Exchange: Swap the value of two chains n and q, accepted with probability $\min(1, \frac{\pi_n(x_q)\pi_q(x_n)}{\pi_n(x_n)\pi_q(x_q)})$
- Crossover: Crossover the l^{th} position in the vector for chains n and q such that

$$
\begin{aligned}
x_n' &= (x_{1n}, \ldots, x_{(l-1)n}, x_{lq}, \ldots, x_{dq}) \\
x_q' &= (x_{1q}, \ldots, x_{(l-1)q}, x_{ln}, \ldots, x_{dn})
\end{aligned}
$$

- Snooker moves: Propose moves close to other members of the population

Population-based MCMC algorithm

- Initialize the chain $x_{1:N}$
- For $t = 1, \ldots, T$,
  1. (Mutation)
     - ★ Select a chain n with a fixed probability and update x_n with a Markov kernel which admits π_n as its invariant distribution
  2. Make a random choice between step 3 or 4
  3. (Crossover)
     - ★ Select two chains n and q randomly and perform a crossover move between x_n and x_q
  4. (Exchange)
     - ★ Select two chains n and q randomly and perform an exchange move between x_n and x_q

Issues

- Tempering or data point tempering?
- How to set the schedule of the tempering approaches?
- Number of distributions?
- Exchange and/or crossover moves?

Sequential Monte Carlo Samplers

- Sequence of distributions π_n, $n = 1, \ldots, p$, defined on E

Sequential Monte Carlo Samplers

- Sequence of distributions π_n, $n = 1, \dots, p$, defined on E
- $\pi_p = \pi$ the target of interest

Sequential Monte Carlo Samplers

- Sequence of distributions π_n, $n = 1, \ldots, p$, defined on E
- $\pi_p = \pi$ the target of interest
- Objective: use sampling/importance sampling/resampling methods to sequentially draw weighted samples from $\pi_1, \pi_2, \ldots, \pi_p$

Sequential Monte Carlo Samplers

- Define new target densities $\widetilde{\pi}_n(x_{1:n}) = \pi_n(x_n) \prod_{j=2}^{n} L_{j-1}(x_j, x_{j-1})$ defined on E^n

Sequential Monte Carlo Samplers

- Define new target densities $\widetilde{\pi}_n(x_{1:n}) = \pi_n(x_n) \prod_{j=2}^{n} L_{j-1}(x_j, x_{j-1})$ defined on E^n
- Arbitrary backward kernels L_n such that $E_1 \subset E_2 \subset \ldots \subset E_p$

Sequential Monte Carlo Samplers

- Define new target densities $\widetilde{\pi}_n(x_{1:n}) = \pi_n(x_n) \prod_{j=2}^{n} L_{j-1}(x_j, x_{j-1})$ defined on E^n
- Arbitrary backward kernels L_n such that $E_1 \subset E_2 \subset \ldots \subset E_p$
- Approximate $\widetilde{\pi}_n(x_{1:n})$, for $n = 1, \ldots, p$ by the empirical distribution

$$P_n(dx_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{x_{1:n}^{(i)}}(dx_{1:n})$$

Sequential Monte Carlo Samplers

0 Initialization

- Set $n = 1$
- For $i = 1, \ldots, N$, draw $X_1^{(i)} \sim \nu$
- Set $W_1^{(i)} \propto \frac{\pi_1(x_1^{(i)})}{\nu(x_1^{(i)})}$ with $\sum_i W_1^{(i)} = 1$

Iterate steps 1 and 2

1. Selection
   - If ESS<Threshold, resample the particles and set the weights to $1/N$
2. Mutation
   - Set $n = n + 1$
   - For $i = 1, \ldots, N$, draw $X_n^{(i)} \sim K_n(x_{n-1}, \cdot)$
   - Compute

$$
\begin{aligned}
W_n^{(i)} &\propto W_{n-1}^{(i)} \frac{\widetilde{\pi}_n(x_{1:n}^{(i)})}{\widetilde{\pi}_{n-1}(x_{1:n-1}^{(i)}) K_n(x_{n-1}^{(i)}, x_n^{(i)})} \\
&\propto W_{n-1}^{(i)} \frac{\pi_n(x_n^{(i)}) L_{n-1}(x_n^{(i)}, x_{n-1}^{(i)})}{\pi_{n-1}(x_{n-1}^{(i)}) K_n(x_{n-1}^{(i)}, x_n^{(i)})}
\end{aligned}
$$

Sequential Monte Carlo Samplers

- Kernel K_n
  - MCMC kernel with invariant distribution π_n
  - Mixture of Markov kernels
  - Approximate Gibbs move (similar to the approximate optimal importance distribution in standard SMC)
  - non-MCMC kernels, etc.

Sequential Monte Carlo Samplers

- Backward Kernel L_{n-1}
  - Introduced to have nested intervals and to avoid computing the marginal importance distribution

Sequential Monte Carlo Samplers

- Backward Kernel L_{n-1}
  - Introduced to have nested intervals and to avoid computing the marginal importance distribution
  - Arbitrary, but need to be optimized with respect to K_n to minimize the variance of the weights

Sequential Monte Carlo Samplers

- Backward Kernel L_{n-1}
  - Introduced to have nested intervals and to avoid computing the marginal importance distribution
  - Arbitrary, but need to be optimized with respect to K_n to minimize the variance of the weights
  - Optimal backward kernel

$$L_{n-1}^{opt} = \frac{\nu_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}{\nu_n(x_n)}$$

where $\nu_n = \nu K_{2:n}$

Sequential Monte Carlo Samplers

- Backward Kernel L_{n-1}
  - Introduced to have nested intervals and to avoid computing the marginal importance distribution
  - Arbitrary, but need to be optimized with respect to K_n to minimize the variance of the weights
  - Optimal backward kernel

    $$L_{n-1}^{opt} = \frac{\nu_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}{\nu_n(x_n)}$$

    where $\nu_n = \nu K_{2:n}$
  - Cannot be computed

Sequential Monte Carlo Samplers

- Backward Kernel L_{n-1}
  - Introduced to have nested intervals and to avoid computing the marginal importance distribution
  - Arbitrary, but need to be optimized with respect to K_n to minimize the variance of the weights
  - Optimal backward kernel

$$L_{n-1}^{opt} = \frac{\nu_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}{\nu_n(x_n)}$$

    where $\nu_n = \nu K_{2:n}$
  - Cannot be computed
  - Selecting L_{n-1} as close as possible to L_{n-1}^{opt} is crucial for the method to be efficient

Sequential Monte Carlo Samplers

- If K_n is an MCMC kernel with invariant distribution π_n, a good approximation is

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1})K_n(x_{n-1}, x_n)}{\pi_n(x_n)}$$

Sequential Monte Carlo Samplers

- If K_n is an MCMC kernel with invariant distribution π_n, a good approximation is

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1}) K_n(x_{n-1}, x_n)}{\pi_n(x_n)}$$

- Good approximation of L_n may be difficult to construct for non-MCMC moves and may lead to fast impoverishment

Comparison of both approaches

- MCMC is often easier to calibrate
- SMC requires no burn-in
- SMC is a richer method, allowing more freedom to design good samplers