Monte Carlo Methods and Bayesian Estimation

Niclas Bergman
Data Fusion Group
SaabTech Systems

November 20, 2001

Outline

- Classical Monte Carlo Methods
- Markov Chain Monte Carlo Methods
- Sequential Monte Carlo Methods

Problem definition

Problem 1 Draw samples from a probability distribution $\pi(x)$ (known at least up to a normalizing constant)

Problem 2 Estimate expectations of functions under $\pi(x)$

$$I_\pi(f) \triangleq \int f(x)\pi(x)\,dx$$

where $x \in E \subset \mathbf{R}^{n_x}$, $n_x \gg 1$.

Application: Bayesian Estimation

Consider an experiment with domain Ω.

Sought parameters: $\boldsymbol{x} = \boldsymbol{x}(\omega) : \Omega \to \mathbf{R}^{n_x}$

Observations: $\boldsymbol{y} = \boldsymbol{y}(\omega) : \Omega \to \mathbf{R}^{n_y}$

Statistical model of the experiment: $\boxed{p(x, y) = p(y \,|\, x) p(x)}$

Likelihood of the observed data $p(y \,|\, x)$. Density of $\boldsymbol{y}$ given that we knew that the value of the s.v. $\boldsymbol{x} = x$.

Prior distribution $p(x)$. Information regarding $\boldsymbol{x}$ prior to performing the experiment.

Bayes theorem yields

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

Once the experiment has been carried out, $\boldsymbol{y}$ collapses to a vector of real numbers y, and $p(y)$ is a scalar positive real value.

Thus, posterior to observing the value y

$$\boxed{p(x \,|\, y) \propto p(y \,|\, x) p(x)}$$

says *everything* about the parameters $\boldsymbol{x}$.

Bayesian Estimate

An estimate $\widehat{x}$ is an educated guess regarding the value of $\boldsymbol{x}$ defined to minimize the Bayesian risk with respect to a given penalty function $L(x^\star, x)$

$$\widehat{x} = \arg\min_{x^\star} \int L(x^\star, x) p(x \,|\, y)\, dx \tag{1}$$

A quadratic cost function

$$L(x^\star, x) = (x - x^\star)^T Q (x - x^\star) \quad \text{for any } Q > 0$$

yields the MMSE estimate

$$\widehat{x}^{\text{MMSE}} = \int x\, p(x \,|\, y)\, dx$$

An *estimator* $\widehat{x}(y)$ solves (1) for each value of y.

Bayesian Estimation and Monte Carlo methods

Monte Carlo methods are suitable for drawing samples from the posterior $p(x \mid y)$ to assess features regarding the parameters after observing the outcome of the experiment.

Monte Carlo mehtods are also used directly to determine estimates like

$$\widehat{x}^{\mathsf{MMSE}} = \int x \, p(x \mid y) \, dx$$

using Monte Carlo integration.

Deterministic Numerical Integration

- Common methods exist (quadrature, splines, wavelets,...)

- Efficient in low dimensional cases (say $n_x < 10$)

But...

- Difficult to implement

- Convergence rate depends on n_x and approximation is of the order $O(N^{-1/n_x})$.

Monte Carlo integration

Perfect simulation: Assume N i.i.d. samples $x^{(i)} \sim \pi(x), i = 1, \ldots, N$ are available, then an approximation of $\pi(x)$ is

$$\widehat{\pi}_N(dx) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}}(dx)$$

thus

$$\widehat{I}_\pi^N(f) = \int f(x) \widehat{\pi}_N(dx)\, dx = \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)})$$

Properties of Monte Carlo estimates

- $\boldsymbol{E}\left(\widehat{I}_{\pi}^{N}(f)\right) = I_{\pi}(f)$, i.e., $\widehat{I}_{\pi}^{N}(f)$ is unbiased

- The variance of $\widehat{I}_{\pi}^{N}(f)$ satisfies
$$\boldsymbol{V}\left(\widehat{I}_{\pi}^{N}(f)\right) = \frac{\boldsymbol{V}_{\pi}(f)}{N}$$
Contrary to all deterministic methods, *rate of convergence independent of* n_x.

Properties of Monte Carlo estimates cont'd

- Law of large numbers: If $|I_\pi(f)| < \infty$ then

$$\lim_{N\to+\infty} \widehat{I}_\pi^N(f) = I(f) \quad \text{almost surely}$$

  i.e., it converges...

- Central limit theorem: If $\sigma^2(f) \triangleq \boldsymbol{V}_\pi(f) < \infty$, then

$$\sqrt{N}\left(\widehat{I}_\pi^N(f) - I_\pi(f)\right) \Rightarrow \mathcal{N}(0, \sigma^2(f))$$

Inverse sampling

Assume(!!!) a perfect random number generator able to sample i.i.d. from $\mathcal{U}_{[0,1]}$.

The *distribution function* for the density $\pi(x)$ is defined as

$$\Pr(x \leq x_0) = \int \cdots \int_{-\infty}^{x_0} \pi(x)\, dx \triangleq F_\pi(x_0)$$

Define its inverse $F_\pi^{-1}(\cdot)$ such that $F_\pi^{-1}(u) = \inf\{x : F_\pi(x) \geq u\}$.

Inverse sampling algorithm: Sample $u \sim \mathcal{U}_{[0,1]}$ and set $x = F_\pi^{-1}(u)$.

Rejection sampling

Target distribution $\pi(x) \propto \pi^*(x)$. Proposal distribution $q(x) \propto q^*(x)$ (easy to sample from) and for all x, $\pi^*(x) \leq cq^*(x)$.

Rejection sampling algorithm

1. Sample $\tilde{x} \sim q(x)$ and $u \sim \mathcal{U}_{[0,1]}$.

2. If $u \leq \frac{\pi^*(\tilde{x})}{cq^*(\tilde{x})}$ then output $\tilde{x}$, otherwise go to step 1

Rejection sampling for Bayesian Estimation

Target distribution $\pi(x) = p(x \,|\, y) \propto p(y \,|\, x)p(x)$

Proposal distribution $q(x) = p(x)$.

Assume $p(y \,|\, x) \leq M$ then one can apply rejection sampling with $\pi^*(x) = p(y \,|\, x)p(x)$, $q^*(x) = p(x)$ and $c = M$.

Note that the acceptance probability Pr($\tilde{x}$ accepted) $= M^{-1}p(y)$ typically is unknown.

Rejection sampling – Practical Problems

- How to contruct the proposal $q(x)$?

- $\pi^*(x) \leq cq^*(x)$ is a strong requirement, where are the modes of $\pi(x)$?

- c typically increases exponentially with $\dim(x)$

Variance Reduction Methods

Does sampling from $\pi(x)$ really do the job?

Property of estimate:

$$\boldsymbol{V}\left(\widehat{I}_{\pi}^{N}(f)\right) = \frac{\boldsymbol{V}_{\pi}(f)}{N}$$

By reducing $\boldsymbol{V}_{\pi}(f)$ we can increase the speed of convergence.

- Rao-Blackwellisation: Integrate out anything you can analytically.

- Startified sampling: For mixture models, perform Monte Carlo integration on each term in the mixture.

The need for alternative methods

Sampling from π is a natural strategy when one is interested in computing $I_\pi(f)$ for *various* f.

If one wants to evaluate $I_\pi(f)$ for *a given* f, sampling from π can be very inefficient

In fact, for a given f, sampling from π is almost never optimal (doesn't use any knowledge about f!).

Importance Sampling

For any importance distribution $q(x)$ (such that $\pi(x) > 0 \Rightarrow q(x) > 0$)

$$I_\pi(f) = \int f(x)\pi(x)\, dx = \int f(x) \underbrace{\frac{\pi(x)}{q(x)}}_{w(x)} q(x)\, dx = I_q(fw)$$

where $w(x)$ is the so called imporance weight $w(x) = \frac{\pi(x)}{q(x)}$

Monte Carlo estimate, N i.i.d. samples $x^{(i)} \sim q(x)$

$$\widehat{I}_q^N(fw) = \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)}) w(x^{(i)})$$

Importance Sampling for Bayesian Estimation

Target distribution $\pi(x) \triangleq p(x \,|\, y)$

Proposal distribution $q(x) \triangleq p(x)$

Thus $w(x) = \frac{p(x \,|\, y)}{p(x)} \propto p(y \,|\, x)$

Unnormalized importance weights

$$\hat{I}_{\mathrm{IS}}^{N}(f) = \sum_{i=1}^{N} \tilde{w}^{(i)} f(x^{(i)}) \qquad \tilde{w}^{(i)} = \frac{w(x^{(i)})}{\sum_{j=1}^{N} w(x^{(j)})}$$

Importance Sampling – Practical Problems

- If $w(x) = \frac{\pi(x)}{q(x)}$ not is upper bounded, the estimator $\widehat{I}_{\mathrm{IS}}^{N}(f)$ has typically very poor behavior in practice.

- How to construct the proposal $q(x)$? Where are the modes of $\pi(x)$?

- Importance sampling is typlically limited to $n_x < 50$.

- Advantage: Easy to implement, parallelizable, sequential version exist.

Markov Chain Monte Carlo Methods – Justification

- Inverse sampling and transformation techniques requre to much analytical knowledge of the distribution.

- Rejection methods requires the definition of a good proposal distribution. Not efficient in high dimensional problems.

- Importance sampling techniques suffer from the same problems.

- Need for more powerful methods to tackle high dimensional problems.

Markov Chains

A Markov Chain is a sequence of random variables $\{x_t : t \in \mathbf{N}\}$ such that for any A

$$\Pr\left(x_t \in A \,|\, x_{t-1}, \ldots, x_0\right) = \Pr\left(x_t \in A \,|\, x_{t-1}\right)$$

The *transition kernel* $K(x_{t-1}, A) \triangleq \Pr\left(x_t \in A \,|\, x_{t-1}\right)$.

π is **an** *invariant distribution* for the transition kernel if

$$\int \pi(x) K(x, y)\, dx = \pi(y)$$

Properties needed for convergence results

Invariant An invariant distribution π is a stationary distribution of a Markov Chain.

Irreducible Any state can be reached form any other state in a finite number of iterations.

Irreducibillity and invariant w.r.t. π implies that π is unique.

Aperiodicity The greatest common divisor of the return time to a given state s is 1. Periodic, say period 2, means that the chain comes to the state at intervals necessarily multiples of 2.

Metropolis-Hastings Algorithm

1. Initiate by setting $t = 0$ and choosing x_0.

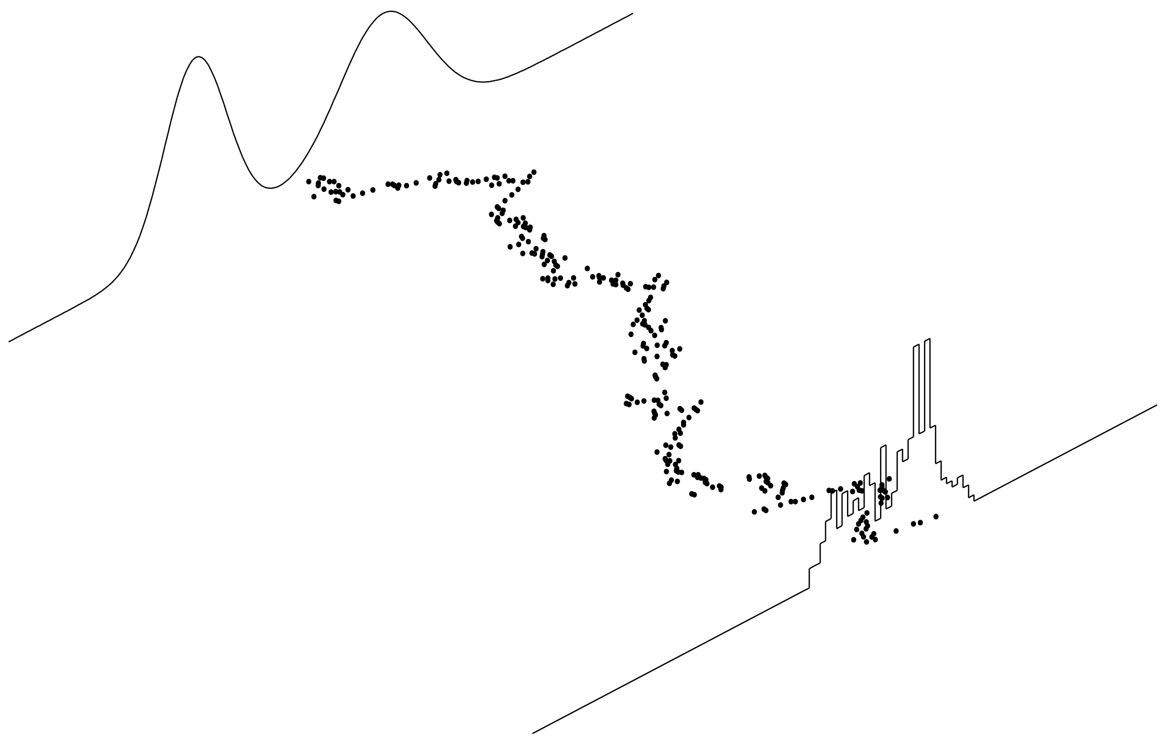
2. Sample $z \sim q(z \mid x_t)$ and $u \sim \mathcal{U}_{[0,1]}$.

3. Compute the acceptance probability $\alpha(x_t, z) = \min\left(1, \frac{\pi(z)q(x_t \mid z)}{\pi(x_t)q(z \mid x_t)}\right)$

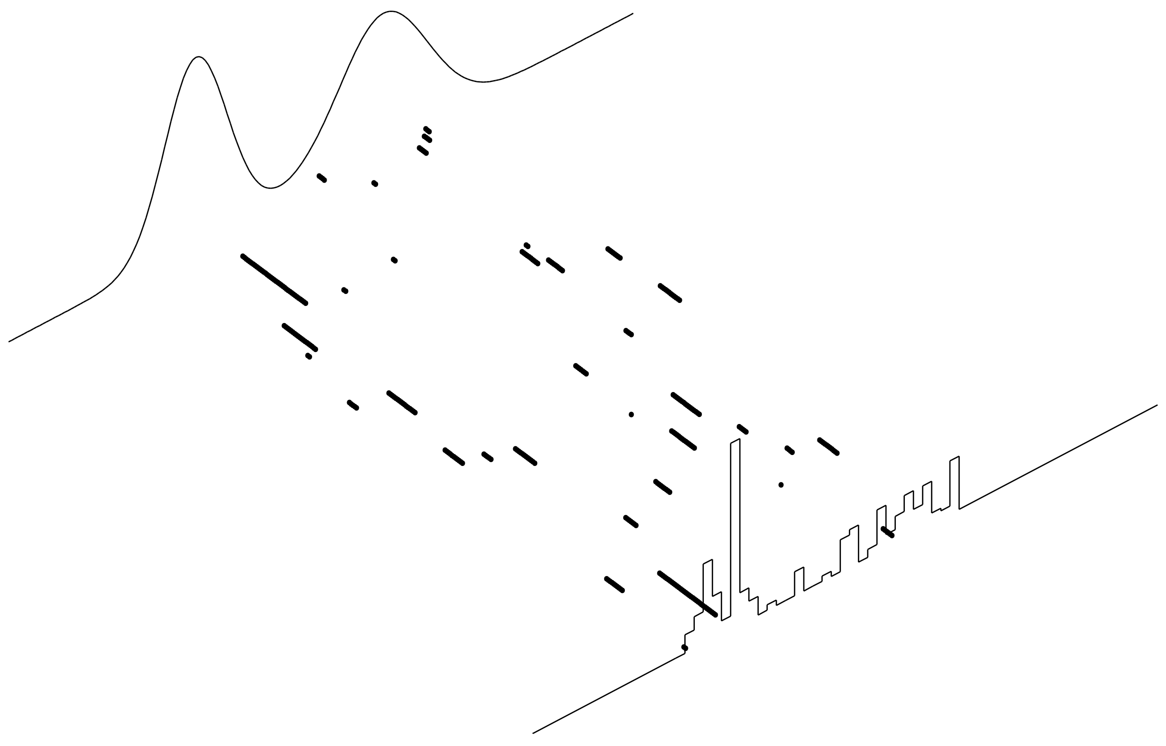
4. If $u \leq \alpha(x_t, z)$ set $x_{t+1} = z$. Otherwise set $x_{t+1} = x_t$.

5. Increase t and return to 2.

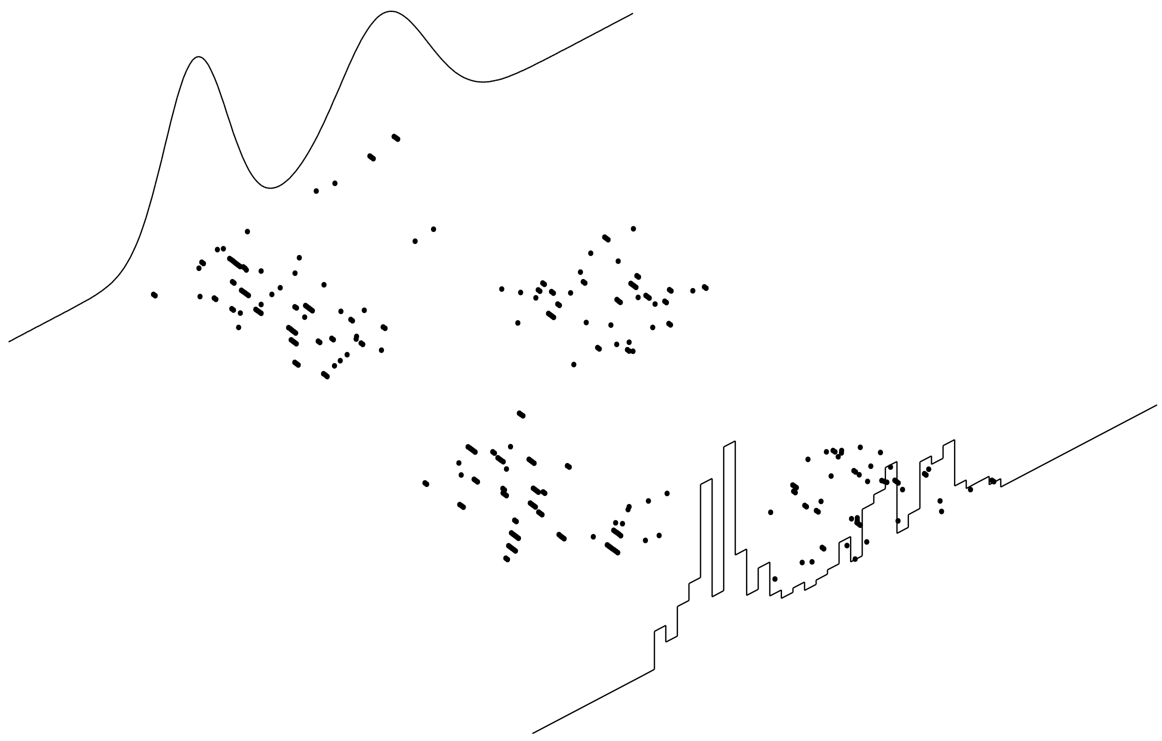
The Choice of Proposal distribution – too small steps



The Choice of Proposal distribution – too many rejections



The Choice of Proposal distribution – good choice



Proposal distribution – common choices

Random walk Set $z = x_t + y$, with $y \sim \varphi(y)$.

If $\varphi(\cdot)$ is even $\alpha(x_t, z) = \min\left\{1, \frac{\pi(z)}{\pi(x_t)}\right\}$ (accept all steps into the peaks of $\pi(\cdot)$, and some others.)

Independent Choose $z \sim \varphi(z)$, independently of current state.

$$\alpha(x_t, z) = \min\left\{1, \frac{\pi(z)\varphi(x_t)}{\pi(x_t)\varphi(z)}\right\}$$

To work well π and φ should be as similar as possible.

Metropolis-Hastings for Bayesian Estimation

In Bayesian estimation we are interested in generating samples from $\pi(x) \propto p(y \,|\, x)p(x)$, the acceptance probability is

$$\alpha(x_t, z) = \min\left(1, \frac{p(y \,|\, z)p(z)q(x_t \,|\, z)}{p(y \,|\, x_t)p(x_t)q(z \,|\, x_t)}\right)$$

The normalizing constant of $\pi(\cdot)$ cancels.

A common choice for proposal is the independent MH using the prior $p(x)$, the acceptance probability then reduces to

$$\alpha(x_t, z) = \min\left(1, \frac{p(y \,|\, z)}{p(y \,|\, x_t)}\right)$$

Metropolis Hastings – Practical Problems

- Universal methods but suffers from problems common to rejection sampling and importance sampling
- Hard to get the chain to “mix” in large state spaces
- This is not a black box tool
- The choice of proposal is unlimited, e.g., one can choose the proposal randomly at each iteration...

Other Algorithms

Gibbs sampler – updating only one element at a time, fixing all the other. Can yeild unity acceptance probability.

Reversible jump MCMC – for performing moves between state spaces of different dimension.

A number of variations and other approaches... See literature

Sequential Bayesian Estimation

Estimation model

$$p(X_t, Y_t) = p(y_t \,|\, x_t) \prod_{i=0}^{t-1} p(x_{i+1} \,|\, x_i) p(y_i \,|\, x_i) p(x_0)$$

where $Y_t \triangleq \{y_i\}_{i=0}^{t}$ and $X_t \triangleq \{x_i\}_{i=0}^{t}$.

Bayesian solution One of the marginals of

$$p(X_t \,|\, Y_t) = \frac{p(X_t, Y_t)}{p(Y_t)} = \frac{p(X_t, Y_t)}{\int p(X_t, Y_t)\, dX_t}$$

usually the filter density $p(x_t \,|\, Y_t)$ determined *sequentially*.

Sequential density update

$$p(x_t \,|\, Y_t) = \frac{p(y_t \,|\, x_t)p(x_t \,|\, Y_{t-1})}{p(y_t \,|\, Y_{t-1})} \tag{M.U.}$$

$$p(x_{t+1} \,|\, Y_t) = \int p(x_{t+1} \,|\, x_t)p(x_t \,|\, Y_t)\, dx_t \tag{T.U.}$$

where $p(x_0 \,|\, Y_{-1}) = p(x_0)$ and

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

Given an observation Y_t, the density $p(x_t \,|\, Y_t)$ condenses all information about the state x_t.

Let $L(x_t^\star, x_t)$ be a *cost function* penalizing any given parameter candidate $x_t^\star$. A *Bayesian estimator* minimizes the expected cost:

$$\widehat{x}_t(Y_t) = \arg\min_{x_t^\star} \int L(x_t^\star, x_t) p(x_t \,|\, Y_t) \, dx_t$$

Example: Quadratic cost

$$L(x_t^\star, x_t) = (x_t - x_t^\star)^T Q (x_t - x_t^\star) \qquad \text{yields for any } Q > 0$$
$$\widehat{x}_t^{\text{MMSE}}(Y_t) = \int x_t \, p(x_t \,|\, Y_t) \, dx_t = \mathbf{E}(x_t \,|\, Y_t)$$

Example: 0/1 cost

$$L(x_t^\star, x_t) = \mathbf{1}_{\{\|x_t^\star - x_t\| \geq \varepsilon\}} \qquad \text{yields as } \varepsilon \to 0$$
$$\widehat{x}_t^{\text{MAP}}(Y_t) = \arg\max_{x_t} p(x_t \,|\, Y_t)$$

Computational Methods

- Express $p(x_t \,|\, Y_t)$ using a finite number of parameters
- Approximate the sequential update of this density

This invloves numerical integration $I_f \triangleq \int f(x_t)p(x_t \,|\, Y_t)\, dx_t$

- Quadrature integration resulting in point-mass densities
- Simulation based integration with random grid approximations

Traditional numerical integration by means of a quadrature formula

$$I_f = \int f(x_t) p(x_t \,|\, Y_t)\, dx_t \approx \widehat{I}_N \triangleq \sum_{i=1}^{N} \alpha_i f(x_t^i) p(x_t^i \,|\, Y_t)$$

The most simplisitc appoach based on a uniform grid and $\alpha_i = \frac{1}{N}$ yields a *curse of dimensionality*

$$\frac{I_f - \widehat{I}_N}{I_f} = O(N^{-1/n_x}) \qquad n_x = \mathrm{dim}(x_t)$$

If we could generate N i.i.d. candidate vectors $\{x_t^i\}_{i=1}^N \sim p(x_t \,|\, Y_t)$

$$\hat{I}_N = \frac{1}{N}\sum_{i=1}^{N} f(x_t^i) \xrightarrow{\text{a.s.}} I_f = \mathbf{E}\left(f(x_t) \,|\, Y_t\right) = \int f(x_t)p(x_t \,|\, Y_t)\, dx_t$$

And if

$$\sigma_f^2 \triangleq \mathbf{E}\Big((f(x_t) - \mathbf{E}f(x_t))^2 \,|\, Y_t\Big) < \infty$$

then

$$\sqrt{N}\left(\hat{I}_N - I_f\right) \Rightarrow \mathcal{N}(0, \sigma_f^2) \qquad \text{as } N \to \infty$$

Assume that we can sample from the *importance density* $q(x_t \,|\, Y_t)$, then any integral w.r.t. the posterior can be written

$$I_f = \int f(x_t) \underbrace{\frac{p(x_t \,|\, Y_t)}{q(x_t \,|\, Y_t)}}_{w(x_t)} q(x_t \,|\, Y_t)\, dx_t$$

and an estimate formed as the weighted empirical mean

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} w(x_t^i) f(x_t^i)$$

However, in our Bayesian framework we have

$$w(x_t) = \frac{p(x_t \,|\, Y_t)}{q(x_t \,|\, Y_t)} = \frac{p(y_t \,|\, x_t) p(x_t \,|\, Y_{t-1})}{p(y_t \,|\, Y_{t-1}) q(x_t \,|\, Y_t)}$$

which is impossible to evaluate since $p(y_t \,|\, Y_{t-1})$ is unknown.

In *Bayesian importance sampling* the weights are evaluated up to a normalizing constant

$$w(x_t) \propto \frac{p(y_t \,|\, x_t)p(x_t \,|\, Y_{t-1})}{q(x_t \,|\, Y_t)}$$

and the (for $N < \infty$ biased) estimate is computed as

$$\hat{I}_N = \frac{\sum_{i=1}^N w(x_t^i) f(x_t^i)}{\sum_{j=1}^N w(x_t^i)}$$

A SLLN and a CLT still holds asymptotically.

Assume we have a prior cloud of particles $\{x_{t-1}^i\}_{i=1}^N$ such that

$$p(x_{t-1} \,|\, Y_{t-1}) \approx \sum_{i=1}^{N} w(x_{t-1}^i)\, \delta_{x_{t-1}^i}(x_{t-1})$$

The Bayesian update we seek to sample from is

$$\begin{aligned} p(x_t \,|\, Y_t) &\propto p(y_t \,|\, x_t) \int p(x_t \,|\, x_{t-1}) p(x_{t-1} \,|\, Y_{t-1})\, dx_{t-1} \\ &\propto p(y_t \,|\, x_t) \sum_{i=1}^{N} p(x_t \,|\, x_{t-1}^i) w(x_{t-1}^i) \end{aligned}$$

Importance sampling with $q(x_t \,|\, Y_t) = p(x_t \,|\, x_{t-1}^i)$ for each term in this mixture density yields

$$p(x_t \,|\, Y_t) \approx \sum_{i=1}^{N} w(x_t^i) \delta_{x_t^i}(x_t) \qquad w(x_t^i) \propto p(y_t \,|\, x_t^i) w(x_{t-1}^i)$$

1. Generate N sample vectors $x_0^i \sim p(x_0)$ and set $w(x_0^i) := \frac{1}{N}$ and $t := 0$.

2. Update the weigths $w(x_t^i) := p(y_t \,|\, x_t^i) w(x_{t-1}^i)$

3. Normlize the weights $w(x_t^i) := \frac{w(x_t^i)}{\sum_{j=1}^N w(x_t^j)}$

4. Predict the samples $x_{t+1}^i \sim p(x_{t+1} \,|\, x_t^i)$ set $t := t + 1$ and repeat at item 2.

The MMSE estimate is the center of gravity $\widehat{x}_t = \sum_{i=1}^N w(x_t^i) x_t^i$

Problems with sample depletion is handled by *resampling* with replacement from the set $\{x_{t-1}^i\}_{i=1}^N$ to a new set $\{x_{t-1}^{i\star}\}_{i=1}^N$ where the probability of resampling particle i is proportional to $w(x_{t-1}^i)$. After resampling the weights are reset to $w(x_{t-1}^i) = \frac{1}{N}$,

$$\sum_{i=1}^{N} w(x_{t-1}^i)\, \delta_{x_{t-1}^i}(x_{t-1}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{x_{t-1}^{i\star}}(x_{t-1})$$

Even though efficient $O(N)$ algorithms exist, resampling is computationally costsome and may be used only when

$$\widehat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} w^2(x_{t-1}^i)}$$

falls below some threshold.

1. Generate N sample vectors $x_0^i \sim p(x_0)$ and set $w(x_0^i) := \frac{1}{N}$ and $t := 0$.

2. Update the weigths $w(x_t^i) := p(y_t \,|\, x_t^i) w(x_{t-1}^i)$

3. Normlize the weights $w(x_t^i) := \frac{w(x_t^i)}{\sum_{j=1}^N w(x_t^j)}$

4. If $\hat{N}_{\text{eff}} < N_{\text{thresh}}$ resample with replacement and reset the weights to $w(x_t^i) = \frac{1}{N}$.

5. Predict the samples $x_{t+1}^i \sim p(x_{t+1} \,|\, x_t^i)$ set $t := t + 1$ and repeat at item 2.

Alternatively, a particle filter can be based on resampling alone. Having a prior cloud of i.i.d. particles $\{x_t^i\}_{i=1}^N \sim p(x_t \,|\, Y_{t-1})$ induces the particle estimate

$$p(x_t \,|\, Y_{t-1}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{x_t^i}(x_t)$$

a sample set $\{x_t^{i\star}\}_{i=1}^N$ from $p(x_t \,|\, Y_t)$ is obtained by resampling with replacement from the original set where the probability of resampling particle i is proportional to the likelihood value $p(y_t \,|\, x_t^i)$. To conclude one iteration of the *Bayesian Bootstrap*, one sample is drawn from each of the terms in the resulting mixture

$$p(x_{t+1} \,|\, Y_t) \approx \frac{1}{N} \sum_{i=1}^{N} p(x_{t+1} \,|\, x_t^{i\star})$$

1. Generate N sample vectors $x_0^i \sim p(x_0)$ and set $t := 0$.

2. Compute the weigths $w(x_t^i) := p(y_t \,|\, x_t^i)$

3. Normlize the weights $w(x_t^i) := \frac{w(x_t^i)}{\sum_{j=1}^{N} w(x_t^j)}$

4. Resample with replacement according to the density defined by the weights.

5. Predict the samples $x_{t+1}^i \sim p(x_{t+1} \,|\, x_t^i)$ set $t := t+1$ and repeat at item 2.

Other approaches

Combining importance sampling and MCMC to refresh the sample population

Limiting the number of resampling iterations

Summary

- Monte Carlo methods provide a family of algorithms to tackle high dimensional estimation problems
- The methods are particulary suitable to handle Bayesian estimation problems
- The methods are also aplicable to sequential estimation problems
- There are no "black box" solutions but a toolbox where each tool needs to be tailored for the particular case