

Monte Carlo inference

Jiali Lin

Virginia Tech

December 4, 2016

Outline

Introduction

Sampling from inverse cdf

Rejection sampling

Importance sampling

Particle filtering

Introduction

- ▶ We typically can draw sample $\boldsymbol{x}^s \overset{i.i.d}{\sim} f(\boldsymbol{x})$.
- ▶ Now, we are unable to have this step.
- ▶ **Idea:** based on non-iterative algorithms of **Monte Carlo approximation**.
 - Generate some (unweighted) samples from the posterior, $\boldsymbol{x}^s \sim p(\boldsymbol{x}|\mathcal{D})$ (where s means “sample”).
 - Use these to compute any quantity of interest, such as a posterior marginal, $p(x_1 - x_2|\mathcal{D})$.
- ▶ Useful when we have complicated posterior distribution.

Outline

Introduction

Sampling from inverse cdf

Rejection sampling

Importance sampling

Particle filtering

Sampling from inverse cdf

- ▶ Let F be a cdf of target distribution and F^{-1} be its inverse.
- ▶ We can sample from any univariate distribution, by evaluating its inverse cdf.
- ▶ Let u represent the height up the y axis. $u \sim U(0, 1)$ using a pseudo random number generator.
- ▶ **Idea:** “slide along” the x axis until intersecting the F curve, “drop down” to return the corresponding x .
- ▶ **Problems:** impossible to derive cdf $F(x)$.

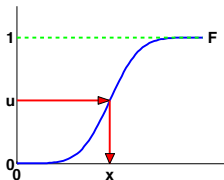


Figure: Sampling using an inverse CDF. Figure generated by `SampleCdf`.

Outline

Introduction

Sampling from inverse cdf

Rejection sampling

Importance sampling

Particle filtering

Rejection sampling

- ▶ **Goal:** reject samples from $q(x)$ such that they are sampled from $p(x)$.
- ▶ $q(x)$ must “cover” or envelop the distribution $p(x)$ (i.e. $cq(x) > p(x)$ for all x).
- ▶ The samples are accepted if $\frac{p(x)}{cq(x)} > u$ where $u \sim \text{Unif}(0, 1)$, and rejected otherwise.
- ▶ If the ratio is close to one, then $p(x)$ must have a large amount of probability mass around x and that sample should be more likely accepted.
- ▶ If the ratio is small, then it means that $p(x)$ has low probability mass around x and we should be less likely to accept the sample.

Illustration of rejection sampling

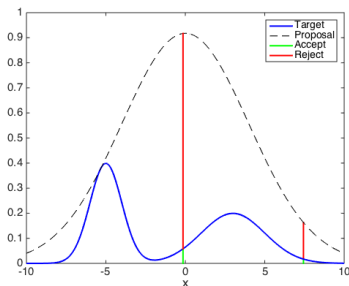


Figure: Rejection sampling from a mixture of two normal distributions using a proposal of normal distribution, where $c = \max(f(x) ./ q(x))$. Figure generated by `rejectionSampling`.

Comments:

- ▶ Hard to define proposal $q(x)$ before we know $p(x)$.
- ▶ $q(x)$ has to be closer to $p(x)$. Otherwise, it would be bad efficiency (low acceptance ratio).
- ▶ Fails in high dimensional space.

Outline

Introduction

Sampling from inverse cdf

Rejection sampling

Importance sampling

Particle filtering

Importance sampling

- ▶ We can evaluate $p(\mathbf{x})$ easily for any given value of \mathbf{x} .
- ▶ Proposal distribution $q(\mathbf{x})$ is easy to draw samples.
- ▶ **Ideas:** uses these samples to estimate the integral

$$I = E[f] = \int f(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} \approx \frac{1}{S} \sum_{s=1}^S w_s f(\mathbf{x}^s) = \hat{I}$$

where $w_s = \frac{p(\mathbf{x}^s)}{q(\mathbf{x}^s)}$ are the importance weights that can be computed.

- ▶ Weights correct the bias introduced by sampling from the wrong distribution.
- ▶ Unlike rejection sampling, *all the samples are used and focused on the important parts of space.*
- ▶ The latent sequence, $\mathbf{x}_{1:t}$ is determined by weight $w_{1:t}$.

Importance sampling (cont'd)

- Usually $p(\mathbf{x}) = \tilde{p}(\mathbf{x})/Z_p$, where $p(\mathbf{x})$ can be evaluated easily, whereas Z_p is unknown. Then

$$E[f] = \frac{Z_q}{Z_p} \int f(\mathbf{x}) \frac{\tilde{p}(\mathbf{x})}{\tilde{q}(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} \approx \frac{Z_q}{Z_p} \frac{1}{S} \sum_{s=1}^S \tilde{w}_s f(\mathbf{x}^s)$$

where $\tilde{w}_s = \frac{\tilde{p}(\mathbf{x}^s)}{\tilde{q}(\mathbf{x}^s)}$ is the unnormalized importance weight.

- We can use the same set of samples to evaluate the ratio

$$\frac{Z_p}{Z_q} = \frac{1}{Z_p} \int \tilde{p}(\mathbf{x}) d\mathbf{x} = \int \frac{\tilde{p}(\mathbf{x})}{\tilde{q}(\mathbf{x})} \approx \frac{1}{S} \sum_{s=1}^S \tilde{w}_s$$

- Hence

$$\hat{I} = \frac{\frac{1}{S} \sum_s \tilde{w}_s f(\mathbf{x}^s)}{\frac{1}{S} \sum_s \tilde{w}_s} = \sum_{s=1}^S w_s f(\mathbf{x}^s)$$

where

$$w_s = \frac{\tilde{w}_s}{\sum_{s'} \tilde{w}_{s'}}$$

Outline

Introduction

Sampling from inverse cdf

Rejection sampling

Importance sampling

Particle filtering

Introduction

- ▶ **Particle filtering (PF)**: a recursive Bayesian inference Monte Carlo algorithm.
- ▶ Estimate the posterior density of the **state space model**.
- ▶ Observations arrive sequentially in time.
- ▶ Given $p(\mathbf{z}_1)$, $p(\mathbf{z}_t|\mathbf{z}_{t-1})$, $p(\mathbf{y}_t|\mathbf{z}_t)$.
- ▶ Goal: performing inference on-line $p(\mathbf{z}_{1:t}|\mathbf{y}_{1:t})$.

Particle filtering

- **Idea:** approximate the belief state (of the entire state trajectory) using a weighted set of particles

$$p(\mathbf{z}_{1:t}|\mathbf{y}_{1:t}) \approx \sum_{s=1}^S \hat{w}_t^s \delta_{\mathbf{z}_{1:t}^s}(\mathbf{z}_{1:t})$$

- Update the marginal distribution over $p(\mathbf{z}_t|\mathbf{y}_{1:t})$, by ignoring the previous parts of the trajectory, $\mathbf{z}_{1:t-1}$.
- If the proposal has the form $q(\mathbf{z}_{1:t}^s|\mathbf{y}_{1:t})$, then the importance weights are given by

$$w_t^s \propto \frac{p(\mathbf{z}_{1:t}^s|\mathbf{y}_{1:t})}{q(\mathbf{z}_{1:t}^s|\mathbf{y}_{1:t})}$$

which can be normalized

$$\hat{w}^s = \frac{w_t^s}{\sum_{s'} w_t^{s'}}$$

Particle filtering (cont'd)

- We can rewrite the numerator recursively

$$\begin{aligned} p(\mathbf{z}_{1:t}|\mathbf{y}_{1:t}) &= \frac{p(\mathbf{y}_t|\mathbf{z}_{1:t}, \mathbf{y}_{1:t-1})p(\mathbf{z}_{1:t}|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{y}_{1:t-1})p(\mathbf{z}_{1:t-1}|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \\ &\propto p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{z}_{t-1})p(\mathbf{z}_{1:t-1}|\mathbf{y}_{1:t-1}) \end{aligned}$$

where we have made the usual Markov assumptions.

- We will restrict attention to proposal densities

$$q(\mathbf{z}_{1:t}|\mathbf{y}_{1:t}) = q(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{y}_{1:t})q(\mathbf{z}_{1:t-1}|\mathbf{y}_{1:t-1})$$

- We can “grow” the trajectory by adding the new state \mathbf{z}_t to the end

$$\begin{aligned} w_t^s &\propto \frac{p(\mathbf{y}_t|\mathbf{z}_t^s)p(\mathbf{z}_t^s|\mathbf{z}_{t-1}^s)p(\mathbf{z}_{1:t-1}^s|\mathbf{y}_{1:t-1})}{q(\mathbf{z}_t^s|\mathbf{z}_{1:t-1}^s, \mathbf{y}_{1:t})q(\mathbf{z}_{1:t-1}^s|\mathbf{y}_{1:t-1})} \\ &= w_{t-1}^s \frac{p(\mathbf{y}_t|\mathbf{z}_t^s)p(\mathbf{z}_t^s|\mathbf{z}_{t-1}^s)}{q(\mathbf{z}_t^s|\mathbf{z}_{1:t-1}^s, \mathbf{y}_{1:t})} \end{aligned}$$

Particle filtering (cont'd)

- We further assume that $q(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}) = q(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{y}_t)$

$$w_t^s \propto w_{t-1}^s \frac{p(\mathbf{y}_t | \mathbf{z}_t^s) p(\mathbf{z}_t^s | \mathbf{z}_{t-1}^s)}{q(\mathbf{z}_t^s | \mathbf{z}_{t-1}^s, \mathbf{y}_t)}$$

- Hence we can approximate the posterior filtered density using

$$p(\mathbf{z}_t | \mathbf{y}_{1:t}) \approx \sum_{s=1}^S \hat{w}_t^s \delta_{\mathbf{z}_t^s}(\mathbf{z}_t)$$

Potential problems of Particle filtering

- ▶ **Comment # 1:** Choice of proposal distribution.
 - *Solutions:* Let $q(z_t | z_{1:t-1}, y_t) = p(z_t | z_{1:t-1})$ (known as **Condensation Filter**).
- ▶ **Comment # 2: Particle degeneracy:** Pfs fails after a few steps because most of the particles will have negligible weight.
 - *Solutions:* Introducing re-sampling: break those big particle into smaller ones, from the “re-sampling” step.

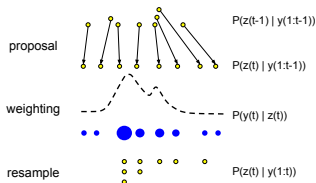


Figure: Illustration of particle filtering.

Particle filtering algorithm

The basic algorithm is now very simple: for each old samples, propose an extension using $\mathbf{z}_t^s \sim q(\mathbf{z}_t | \mathbf{z}_{t-1}^s, \mathbf{y}_t)$, and give this new particle weight w_t^s .

for $s = 1 : S$ **do**

1. Draw $\mathbf{z}_t^s \sim q(\mathbf{z}_t | \mathbf{z}_{t-1}^s, \mathbf{y}_t)$.

2. Compute weight $w_t^s \propto w_{t-1}^s \frac{p(\mathbf{y}_t | \mathbf{z}_t^s) p(\mathbf{z}_t^s | \mathbf{z}_{t-1}^s)}{q(\mathbf{z}_t^s | \mathbf{z}_{t-1}^s, \mathbf{y}_t)}$.

given x^k, λ^{k-1} , and parameter $\beta \in (0, 1)$.

Normalize weights: $\hat{w}_t^s = \frac{w_t^s}{\sum_{s'} w_t^{s'}}$.

Compute $\hat{S}_{\text{eff}} = \frac{1}{\sum_{s=1}^S (w_t^s)^2}$

If $\hat{S}_{\text{eff}} < S_{\text{min}}$ **then**

1. Resample S indices $\pi \sim w_t$.

2. $\mathbf{z}_t = \mathbf{z}_t^\pi$.

3. $w_t^s = 1/S$.

return $p(\mathbf{z}_{1:t} | \mathbf{y}_{1:t})$.
