CS281 Section #9 Notes, v 1.1

1 Importance Sampling

To estimate $I = Ef = \int f(x)p(x)dx$, importance sampling samples from any proposal $q(x)$, then

$$Ef = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{S}\sum_{s=1}^{S} w_s f(x^s) = \hat{I}_1$$

$\hat{I}_1$ is unbiased, i.e. $E(\hat{I}_1) = I$.

If we only know $p(x)$ and $q(x)$ up to a normalizing constant, say, $p(x) = p_u(x)/Z_p$ and $q(x) = q_u(x)/Z_q$, we could instead compute:

$$Ef = \int f(x)p(x)dx = \frac{\int f(x)p_u(x)dx}{\int p_u(x)dx} = \frac{\int f(x)\frac{p_u(x)}{q_u(x)}q(x)dx}{\int \frac{p_u(x)}{q_u(x)}q(x)dx} \approx \frac{\sum_{s=1}^{S} w_s f(x^s)}{\sum_{s=1}^{S} w_s} = \hat{I}_2$$

$\hat{I}_2$ is biased but as we collect more and more samples, bias goes to zero and the biased term is $O(1/S)$, whose square is smaller than variance, so we could ignore bias if considering mean squared error.

A good proposal distritbution should concentrate on the region where both $f(x)$ and $p(x)$ are large. Importance sampling can be super efficient, which means that it needs less samples to achieve the same variance than sampling directly from $p(x)$. Let's compute the variance of $\hat{I}_1$:

$$Var(\hat{I}_1) = E_q(f^2(x)w^2(x)) - I^2 = \int f(x)^2\frac{p^2(x)}{q(x)}dx - I^2$$

where $w(x) = \frac{p(x)}{q(x)}$. By Jensen's inequality, we have

$$E_q(f^2(x)w^2(x)) \geq (E_q[|f(x)|w(x)])^2 = \left(\int |f(x)|p(x)dx\right)^2$$

obtained by $q(x) = \frac{|f(x)|p(x)}{\int |f(x)|p(x)dx}$, which is optimal.

The asymptotic variance of $\hat{I}_2$ is

$$\int (f(x) - I)^2\frac{p^2(x)}{q(x)}dx$$

The optimal proposal is $q(x) = \frac{|f(x)-I|p(x)}{\int |f(x)-I|p(x)dx}$. Even if we know the normalizing constants, in some cases, the variance of $\hat{I}_2$ is smaller than $\hat{I}_1$, using the second form is still better. Take an extreme example, suppose $f(x) \equiv 1$, then $Var(\hat{I}_2) = 0$ but $Var(\hat{I}_1) > 0$; another case,$f(x) = \frac{q(x)}{p(x)}$, then $I = 1$ and $Var(\hat{I}_1) = 0$ but $Var(\hat{I}_2) > 0$.

The variance above depends on $f(x)$, which is case-specific and hard to evaluate a proposal distribution. Another concept called "Effective Sample Size" is the equivalent number of samples if sampling from true distribution. By Taylor expansion, $Var(\hat{I}_2) \approx Var_p(f(x))(1 + Var_q(\frac{w(x)}{E_q(w(x))}))$, so

$$ESS := \frac{S}{1 + Var_q(\frac{w(x)}{E_q(w(x))})} = \frac{S(E_q[w(x)])^2}{E_q(w(x)^2)} \approx \frac{1}{\sum_{s=1}^{S} W_s^2}$$

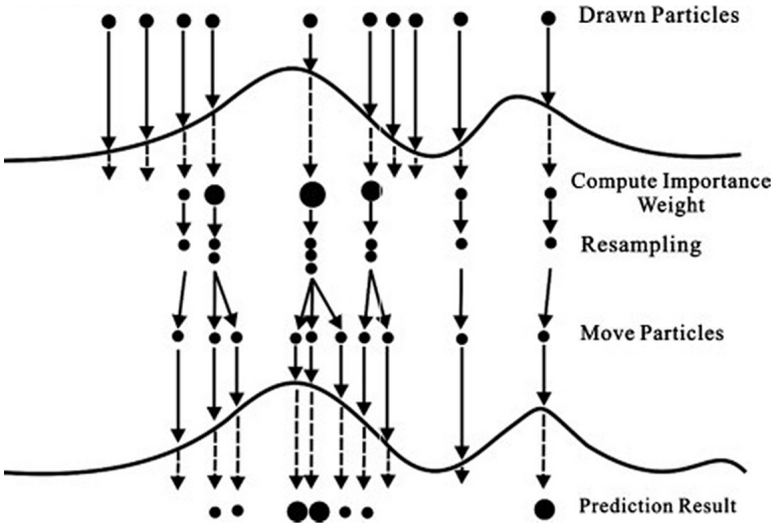
where W_s is normalized weight.

2 Sequential Monte Carlo (Particle Filter)

Consider a state-space model where x_t is the latent variable and y_t is the observations. Observing $y_{1:T}$, we are interested in the posterior $\pi(x_{1:T}|y_{1:T})$. Instead of sampling directly the whole path, we sample from a series of distribution (e.g. $\pi(x_1|y_1), \pi(x_{1:2}|y_{1:2}), ..., \pi(x_{1:T}|y_{1:T})$) gradually approaching our target distribution. For most cases, posteriors of x_t don't have an analytic form (except Gaussian and discrete cases), so we use sample ("particles") distribution to approximate posteriors at each step. Particle Filter algorithm is as follows:

- At time 0, generate N particles from the prior $\pi(x)$ and set weight to $1/N$.
- At time t+1, sample $x_{t+1}^i \sim q(x|x_t^i, y_{t+1}),\ i = 1, 2, ..., N$.
- Reweight each particle by $w_{t+1}^i = W_t^i \frac{\pi(x_{1:(t+1)}^i, y_{1:(t+1)})}{\pi(x_{1:t}^i, y_{1:t}) q(x_{t+1}^i|x_t^i, y_{t+1})} = W_t^i \frac{p(y_{t+1}|x_{t+1}^i) p(x_{t+1}^i|x_t^i)}{q(x_{t+1}^i|x_t^i, y_{t+1})}$ (up to normalizing constant), then normalize the weight by $W_{t+1}^i = \frac{w_{t+1}^i}{\sum_{i=1}^N w_{t+1}^i}$.
- Compute ESS=$\frac{1}{\sum_{i=1}^N (W_{t+1}^i)^2}$. If ESS $<$ cN, resampling according to W_{t+1}^i and set weight to $1/N$; otherwise record the current weight and move on.
- $\sum_{i=1}^N W_t^i \delta(x_t^i)$ (before resampling) is a approximation to $\pi(x_t|y_{1:t})$.

If $q(x_{t+1}|x_t, y_{t+1}) = p(x_{t+1}|x_t)$, then $w_{t+1}^i = W_t^i p(y_{t+1}|x_{t+1}^i)$.
If $q(x_{t+1}|x_t, y_{t+1}) = p(x_{t+1}|y_{t+1}, x_t)$, then $w_{t+1}^i = W_t^i \int p(y_{t+1}|x) p(x|x_t^i) dx$. This is optimal but hard to compute and sample.



Why we should do resampling? Without resampling, as time goes on, the weights accumulate,

$$w_t \propto \prod_{s=1}^{t} \frac{\pi(x_{1:s}, y_{1:s})}{\pi(x_{1:(s-1)}, y_{1:(s-1)}) q(x_s|x_{s-1}, y_s)} = \frac{\pi(x_{1:t}, y_{1:t})}{\prod_{s=1}^{t} q(x_s|x_{s-1}, y_s)}$$

If the proposal is transition probability of latent state, the weights

$$w_t \propto \prod_{s=1}^{t} p(y_s|x_s)$$

become highly skewed, the effective sample size is small as we wastes many particles at low probability regions. Therefore, we need to do resampling when the weights are too skewed. The particle with largest weight, which has higher probability up to current time point, will have higher chance to be replicated and reproduce in the following time points. After resampling, every particle will have sample weight and ESS will becomes N.

However, resampling will reduce the diversity of samples and when looking back in time, all the particles will collapse to a single point within a few iterations. This is known as sample impoverishment. [1] is a comprehensive review for particle filter.

3 Markov Chain Monte Carlo

Again in order to estimate $I = Ef = \int f(x)p(x)dx$, instead of independent sampling, we could generate samples from the Markov Chain whose stationary distribution is the target distribution ($p(x)$) and estimate I using samples from the Markov Chain by $\frac{1}{T}\sum_{t=1}^{T} f(x^t)$. Its variance depends on the auto-correlation of the Markov Chain:

$$Var(\hat{I}_{MCMC}) = Var_p(f(x)) + 2\sum_{t=2}^{\infty} Cov(f(x_1), f(x_t))$$

3.1 Metroplis-Hastings vs. Gibbs

Gibbs sampling gives a Markov Chain whose stationary distribution is the target; while Metroplis-Hastings is a more generic way of doing that (in most cases, the conditional distribution is hard to derive). Suppose $q(x'|x)$ is the proposal distribution, then for MH algorithm, it will accept the new proposal with probabilty

$$r = \min(1, \frac{p(x')q(x|x')}{p(x)q(x'|x)})$$

otherwise save the original x.

First, we will prove that, the stationary distribution of this Markov chain is the target distribution ($p(x)$), i.e. $p(x) = \int p(x')K(x|x')dx'$, where $K(x'|x)$ is the Markov kernel defined by MH algorithm. Detailed balance which states that

$$p(x)K(x'|x) = p(x')K(x|x')$$

is stronger condition. If a Markov chain satisfies detailed balance, its stationary distribution is $p(x)$. We will prove MH satifies detailed balance. The Markov kernel for MH is $K(x'|x) = q(x'|x)\min(1, \frac{p(x')q(x|x')}{p(x)q(x'|x)})$ for $x' \neq x$ (for $x' = x$, it satifies detailed balance obviously).

$$\begin{aligned} p(x)K(x'|x) &= p(x)q(x'|x)\min(1, \frac{p(x')q(x|x')}{p(x)q(x'|x)}) \\ &= \min(p(x)q(x'|x), p(x')q(x|x')) = p(x')q(x|x')\min(1, \frac{p(x)q(x'|x)}{p(x')q(x|x')}) \\ &= p(x')K(x|x') \end{aligned}$$

Second, show that Gibbs sampling is a specially case of MH. For Gibbs,

$$q(x'|x) = p(x'_i|x_{-i})I(x'_{-i} = x_{-i})$$

and

$$\frac{p(x')q(x|x')}{p(x)q(x'|x)} = \frac{p(x'_i|x_{-i})p(x_{-i})p(x_i|x_{-i})}{p(x_i|x_{-i})p(x_{-i})p(x'_i|x_{-i})} = 1$$

The fact that the acceptance rate is 100% does not necessarily mean that Gibbs will converge rapidly, since it only updates one coordinate at a time.

Finally, if we don't have a good proposal, we could choose $q(x'|x)$ as a random walk, i.e. $q(x'|x) = N(x'|x, \sigma^2)$. The autocorrelation of the Markov chain depends on σ^2. If σ^2 is too large, the new proposal always get rejected so the Markov chain stays at the same place, resulting in large correlation; on the other hand, if σ^2 is too small, the samples will also have large correlation and it takes a long time for the Markov chain to traverse across the target space. A rule of thumb is to select σ^2 such that the acceptance ratio is between 23% to 50%. [4] show demos for different MCMC methods sampling from various shapes of target distribution.

Particle MCMC combines MCMC with particle filtering [2]. Some new methods (e.g. [3]) can alleviate path dengenecy problem of PMCMC.

4 An example of Particle Filter

4.1 Model

$$y_t = u_t + \epsilon_t,\ \epsilon_t \sim i.i.d.N(0, \sigma_\epsilon^2)$$
$$u_t = u_{t-1} + v_t,\ v_t \sim i.i.d.t_3(\sigma_v^2)$$

$y_t = 400\Delta \ln(\text{OPHNFB}_t)$ which is the logarithmic approximation to the quarterly percentage growth of productivity at an annual rate (Productivity in the US nonfarm business sector (OPHNFB) is from FRED https://fred.stlouisfed.org/series/OPHNFB). u_t is a drift in the growth rate, y_t is the drift plus indepedent noises:

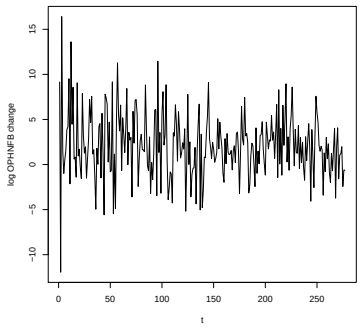


Figure 1: Plot of Y_t.

y_t is quite noisy, so the problem is to estimate long-term trend of the growth rate of productivity (u_t).

To get posterior of the parameter σ_v^2 and σ_ϵ^2, we used MH to sample σ_v^2 and σ_ϵ^2, where the marginal likelihood $P(\mathbf{y}|\sigma_v^2,\ \sigma_\epsilon^2)$ is computed by particle filter. Fig. 2 is the filtering and smoothing distribution plugging in posterior modes of $\hat{\sigma}_v^2 = 0.004$ and $\hat{\sigma}_\epsilon^2 = 12.61$.

The figure on the right shows the trajectories of u_t^i when using 5 particles (for illustration purpose), shown as gray dots. If we trace back the ancestries of the 5 particles at time T (showed as colored lines), they actually come from a common ancestor at some time in the middle, and thus half of the pathes are identical. We will only get few samples of x_t when t is a early time point, and thus couldn't get a good estimate of $\pi(x_t|y_{1:T})$ nevertheless the posterior of whole path.

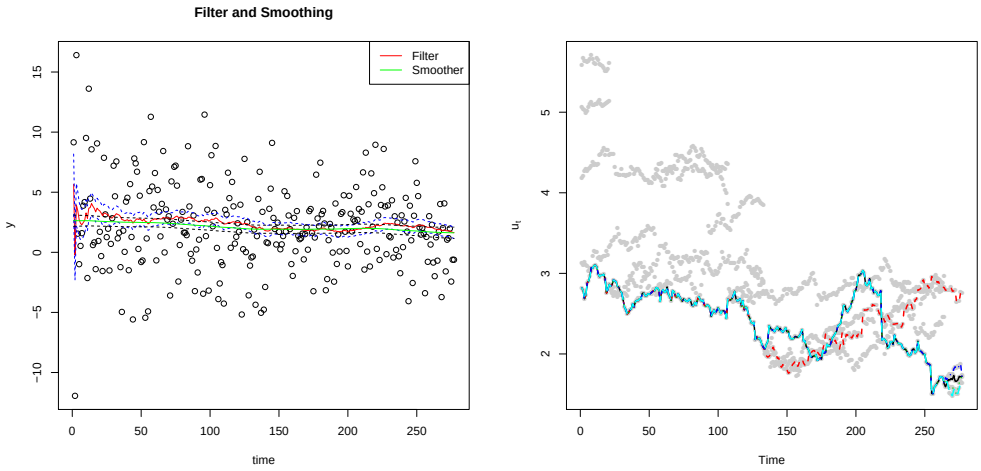


Figure 2: Left: Filtering and Smoothing of y_t. Right: Trace back the ancestries of of y_T^i

5 References

1. A Tutorial on Particle Filtering and Smoothing: Fifteen years later (2012) Arnaud Doucet, Adam M. Johansen
2. Particle Markov chain Monte Carlo methods (2010), Christophe Andrieu, Arnaud Doucet, Roman Holenstein
3. Particle Gibbs with Ancestor Sampling (2014), Fredrik Lindsten, Michael I. Jordan, Thomas B. Schon
4. https://chi-feng.github.io/mcmc-demo/