Lecture 1: Basics of Monte Carlo Methods

Tianpei Xie

Sep. 28th., 2022

Contents

1 Basic Concepts

1.1 Random variable generation

- The most fundamental methods for random variable generation is the ***uniform pseudo-random number generator***.

  Definition A *uniform pseudo-random number generator* is an algorithm which, starting from an initial value u_0 and a transformation D, produces a sequence $(u_i) = (D^i(u_0))$ of values in $[0, 1]$. For all n, the values $(u_1, \ldots, u_n)$ reproduce the behavior of an *iid* sample $(V_1, \ldots, V_n)$ of uniform random variables when compared through a usual set of tests.

- This definition is clearly restricted to testable aspects of the random variable generation, which are connected through the deterministic transformation $u_i = D^i(u_{i-1})$. Thus, the validity of the algorithm consists in the verification that the sequence $U_1, \ldots, U_n$ leads to acceptance of the hypothesis

$$H_0 : \quad U_1, \ldots, U_n \text{ i.i.d. } \mathcal{U}[0, 1]$$

- Definition above is therefore ***functional***: An algorithm that generates uniform numbers is acceptable if it is not rejected by a set of tests.

- This methodology is not without problems, however. Consider, for example, particular applications that might demand a large number of iterations, as the theory of large deviations, or particle physics, where algorithms resistant to standard tests may exhibit fatal faults. In particular, algorithms having hid- den periodicities (e.g. Ising models) or which are not uniform for the smaller digits may be difficult to detect.

- Given uniform distribution $\mathcal{U}[0, 1]$, and a known c.d.f. $F(x)$, we can generate samples via the ***inverse transform*** method. [Robert and Casella, 1999]

  Definition For a ***non-decreasing*** function F on $\mathbb{R}$, the ***generalized inverse*** of F, F^-, is the function defined by

$$F^-(u) = \inf\{x \in \mathcal{R}(F) : F(x) \geq u\}. \tag{1}$$

  where $\mathcal{R}(F)$ is the range of F. If F is the cumulative distribution function $F_X(x) = P(X \leq x)$, then F_X^{-1} is called ***quantile function***.

- We then have the following lemma, sometimes known as the **probability integral transform**, which gives us a representation of any random variable as a transform of a uniform random variable.

  Lemma 1.1 *[Robert and Casella, 1999]*
  If $U \sim \mathcal{U}[0, 1]$, then the random variable $F^-(U)$ has the distribution F.

- Examples for some quantile functions [Robert and Casella, 1999]

  - **Exponential distribution**: $F^-(u) = -\log(1-u)$. Since U and $1-U$ are both uniform in $[0, 1]$, $X = \log(U) \sim \exp(1)$ is exponential distributed with $\lambda = 1$;

  - **Gamma distribution** and χ^2 distribution can be generated from the exponential dis-

tribution exp(1)

$$Y = \beta \sum_{j=1}^{a} X_j \sim \Gamma(a, \beta), ; a \in \mathbb{N}+$$
$$Y = 2 \sum_{j=1}^{\nu} X_j \sim \chi^2_{2\nu}; \nu \in \mathbb{N}+$$

– **Normal distribution**:

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$$
$$X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

where U_1, U_2 are i.i.d from $\mathcal{U}[0,1]$. Then X_1, X_2 are i.i.d from $\mathcal{N}(0,1)$.

- Theoretically, any random variable can be generated using the inverse transform. But in practice, it is only available when we can compute the c.d.f and its inverse F^- explicitly.

1.2 Why Monte Carlo ?

- The most fundamental operations in probabilistic modeling and statistics involve ***integration*** on **high dimensional spaces**:

$$I := \int_{\mathcal{X}} g(\boldsymbol{x}) d\boldsymbol{x}$$

For example,

– ***Expectation***:

$$\mathbb{E}_{\boldsymbol{X}\sim p}\left[h(\boldsymbol{X})\right] = \int_{\mathcal{X}} h(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \tag{2}$$

where $\boldsymbol{x} := [x_1, \ldots, x_d]$

– ***Marginalization***:

$$P(x_1, \ldots, x_{i-1}, x_{i+1} \ldots, x_d) = \int_{x_i \in \mathcal{X}_i} P(x_1, \ldots, x_{i-1}, x_i, x_{i+1} \ldots, x_d) dx_i$$

– ***Conditioning***:

$$P(x_1, \ldots, x_{i-1}, x_{i+1} \ldots, x_d \,|\, X_i = x_i) = \frac{P(x_1, \ldots, x_{i-1}, x_i, x_{i+1} \ldots, x_d)}{\int_{\boldsymbol{x}_{-i} \in \boldsymbol{X}_{-i}} P(x_1, \ldots, x_{i-1}, x_i, x_{i+1} \ldots, x_d) d\boldsymbol{x}_{-i}}$$

where $\boldsymbol{x}_{-i} = [x_1, \ldots, x_{i-1}, x_{i+1} \ldots, x_d]$.

- This is especially important for ***Bayesian inference*** and analysis. Since it replies on computing *posterior distribution* given data using **the Bayes rule**

$$P(\boldsymbol{\Theta}|\mathcal{D}) = \frac{P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})}{P(\mathcal{D})}$$

$$= \frac{P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})}{\int_{\boldsymbol{\Theta}} P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})d\boldsymbol{\Theta}}$$

$$\Rightarrow \log P(\boldsymbol{\Theta}|\mathcal{D}) = \log P(\mathcal{D}|\boldsymbol{\Theta}) + \log P(\boldsymbol{\Theta}) - \log \int_{\boldsymbol{\Theta}} P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})d\boldsymbol{\Theta}$$

where $\mathcal{D} := \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ are i.i.d. samples and $\boldsymbol{\Theta}$ is a set of parameters for model $p(\mathcal{D}|\boldsymbol{\Theta})$.

- The main challenge for Bayesian methods in high dimensional space is to compute the normalization factor

$$\int_{\boldsymbol{\Theta}} P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})d\boldsymbol{\Theta}.$$

- The ***log-partition function*** in **exponential families** play a critical role since it defines a *bijective mapping* from *natural parameters* to *mean parameters* and is related to **negative entropy** via *variational principles*. The log-partition function for exponential families is a convex function as well.

$$A(\boldsymbol{\eta}) = \log \int_{\Omega} \exp\left(\langle \boldsymbol{\eta}\,,\, \boldsymbol{\phi}(\boldsymbol{x})\rangle\right) d\boldsymbol{x}$$

- It is natural to propose using a sample $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m)$ generated from the density p to *approximate integration* by the *empirical average*. This approach is referred to as the ***Monte Carlo methods***. Specifically from (2),

$$\overline{h}_m = \frac{1}{m}\sum_{i=1}^{m} h(\boldsymbol{X}_i). \tag{3}$$

since $\overline{h}_m$ converges almost surely to $\mathbb{E}_{\boldsymbol{X}\sim p}[h(\boldsymbol{X})]$ by the *Strong Law of Large Numbers*. Moreover, when $\overline{h}_m$ has a finite expectation under p, the speed of convergence of $\overline{h}_m$ can be assessed since the variance

$$\mathrm{Var}\left(\overline{h}_m\right) = \frac{1}{m}\int_{\mathcal{X}} \left(h(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{X}\sim p}[h(\boldsymbol{X})]\right)^2 p(\boldsymbol{x})d\boldsymbol{x}$$

can also be estimated from the sample $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m)$ through

$$v_m = \frac{1}{m}\sum_{i=1}^{m} \left(h(\boldsymbol{X}_i) - \overline{h}_m\right)^2$$

For m large,

$$\frac{h(\boldsymbol{X}) - \overline{h}_m}{\sqrt{v_m}} \to \mathcal{N}(0,1).$$

This leads to the construction of a ***convergence test*** and of ***confidence bounds*** on the approximation of $\mathbb{E}_{\boldsymbol{X}\sim p}[h(\boldsymbol{X})]$.

2 Rejection and Weighting

2.1 Reject Sampling

- There exists a fundamental idea underlying the ***Accept-Reject sampling method***:

  Theorem 2.1 *(**Fundamental Theorem of Simulation**) [Robert and Casella, 1999] Simulating $X \sim F(x)$ is equivalent to simulating*

  $$(X, U) \sim \mathcal{U}\{(x, u) : 0 < u < F(x)\}.$$

  In essence, If f is the density of interest, on an arbitrary space, we can write this theorem as

  $$f(x) = \int_0^{f(x)} du.$$

  Thus, f appears as the marginal density (in X) of the joint distribution, $(X, U) \sim \mathcal{U}\{(x, u) : 0 < u < f(x)\}$.

- Suppose $l(\boldsymbol{x}) = c\, f(\boldsymbol{x})$ is computable, where f is the probability density and c is unknown. If we can find a sampling distribution $g(\boldsymbol{x})$ and a ***covering constant*** M so that $Mg(\boldsymbol{x}) \geq l(\boldsymbol{x})$ (i.e. the envelop property) is satisfied for all $\boldsymbol{x}$. We can apply the following procedure:

  Accept-Reject sampling sampling [Liu, 2001]

  1. Draw sample $\boldsymbol{x}$ from $g(\,)$ and compute the **ratio**

     $$r = \frac{l(\boldsymbol{x})}{M\, g(\boldsymbol{x})} \quad (\leq 1)$$

  2. Flip a coin with **success probability** r. If the head turns up, accept $\boldsymbol{x}$; otherwise, reject $\boldsymbol{x}$

  Then the accepted samples follow the distribution $f(\boldsymbol{x})$.

  Proof: Let $I \in \{0, 1\}$ be the event when head turns up $(= 1)$ vs. tail turns up $(= 0)$.

  $$P(I = 1) = \int P(I = 1|\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x} = \int \frac{c\, f(\boldsymbol{x})}{M\, g(\boldsymbol{x})} g(\boldsymbol{x}) d\boldsymbol{x} = \frac{c}{M}$$

  Then

  $$P(\boldsymbol{x}|I = 1) = \frac{g(\boldsymbol{x}) \frac{c\, f(\boldsymbol{x})}{M\, g(\boldsymbol{x})}}{P(I = 1)} = f(\boldsymbol{x}). \quad \blacksquare$$

- This is equivalent to say

  1. Draw sample $\boldsymbol{X}$ from $g(\boldsymbol{x})$ and U from $\mathcal{U}[0, 1]$
  2. Accept $\boldsymbol{Y} = \boldsymbol{X}$ if $U \leq f(\boldsymbol{x})/(M\, g(\boldsymbol{x}))$;
  3. Otherwise, return to step 1.

- The key for a successful application of rejection sampling method is to choose a good trial distribution $g(\cdot)$ so that M is **small**. In high dimensional settings, it is usually hard to find a good trial distribution.

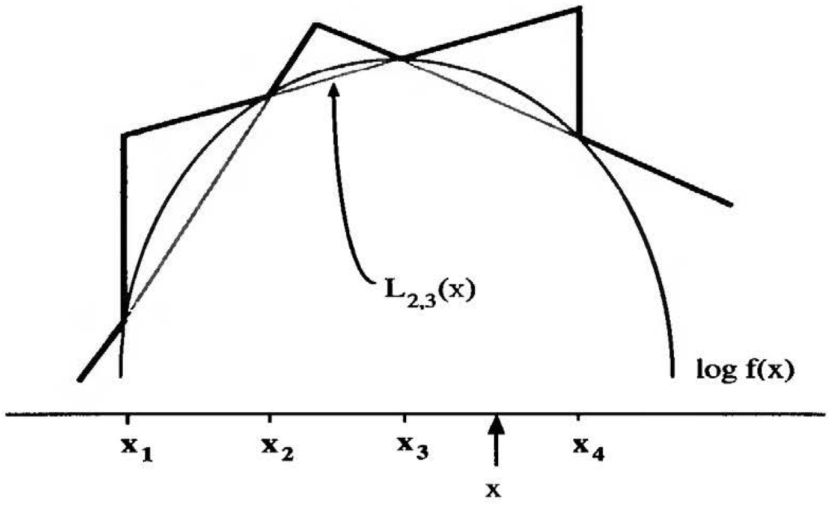

Fig. 2.5. Lower and upper envelopes of $h(x) = \log f(x)$, f a log-concave density (*Source:* Gilks et al. 1995).

Figure 1: The log-concave function h and its upper and lower bounds [Robert and Casella, 1999]

- **Lemma 2.2** *If there exist a density $g_m(\boldsymbol{x})$, a function $g_l(\boldsymbol{x})$ and a constant M such that*

$$g_l(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq M\, g_m(\boldsymbol{x})$$

then the algorithm **Envelope Accept-Reject**

1. *Draw sample $\boldsymbol{X}$ from $g_m(\boldsymbol{x})$ and U from $\mathcal{U}[0,1]$*
2. *Accept $\boldsymbol{X}$ if $U \leq g_l(\boldsymbol{x})/(M\, g_m(\boldsymbol{x}))$;*
3. *Otherwise, accept $\boldsymbol{X}$ if $U \leq f(\boldsymbol{x})/(M\, g_m(\boldsymbol{x}))$ else return to step 1.*

produces random variables that are distributed according to f.

- By the construction of a lower envelope on f, based on the function g_l, the number of evaluations of f is potentially decreased by a factor

$$\frac{1}{M} \int g_l(\boldsymbol{x}) d\boldsymbol{x},$$

which is the probability that f is not evaluated. This method is called the ***squeeze principle***.

- For the exponential families, the density function

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\left(\langle \boldsymbol{\theta}\,,\, \boldsymbol{x} \rangle - A(\boldsymbol{\theta})\right)$$

is *log-concave* in terms of $\boldsymbol{x}$. An envelop accept-reject sampling method is discussed in [Robert and Casella, 1999]. The method is called ***adaptive rejection sampling (ARS)*** and it provides a sequential evaluation of lower and upper envelopes of the density f when $h = \log f$ is concave. Consider a set of n samples $\mathcal{D}_n = \{\boldsymbol{x}_i\}$. For each sample $\boldsymbol{x}_i$, $h(\boldsymbol{x}_i) = \log f(\boldsymbol{x}_i)$ is concave, so the following upper bound and lower bound can be found using linear segments $L_{i,i+1}$ through $(\boldsymbol{x}_i, h(\boldsymbol{x}_i))$ and $(\boldsymbol{x}_{i+1}, h(\boldsymbol{x}_{i+1}))$. Define

$$\overline{h}_n(\boldsymbol{x}) := \min\left\{L_{i-1,i}(\boldsymbol{x}), L_{i+1,i+2}(\boldsymbol{x})\right\};$$

$$\underline{h}_n(\boldsymbol{x}) := L_{i,i+1}(\boldsymbol{x}).$$

Then by concavity of h, for $\boldsymbol{x} \in [\boldsymbol{x}_i, \boldsymbol{x}_{i+1}]$

$$\underline{h}_n(\boldsymbol{x}) \le h(\boldsymbol{x}) \le \overline{h}_n(\boldsymbol{x})$$

Therefore, for $\underline{f}_n(\boldsymbol{x}) = \exp \underline{h}_n(\boldsymbol{x})$ and $\overline{f}_n(\boldsymbol{x}) = \exp \overline{h}_n(\boldsymbol{x})$ implies that

$$\underline{f}_n(\boldsymbol{x}) \le f(\boldsymbol{x}) \le \overline{f}_n(\boldsymbol{x}) = \omega_n \, g_n(\boldsymbol{x})$$

where ω_n is a normalization factor so that $g_n(\boldsymbol{x})$ is a density.

Then the ***adaptive rejection sampling (ARS)*** is

1. Initiate $\mathcal{D}_n = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$. Compute the line segments $L_{i,i+1}$, $i = 0, \ldots, n-1$.
2. Draw sample $\boldsymbol{X}$ from $g_n(\boldsymbol{x})$ and U from $\mathcal{U}[0,1]$
3. Accept $\boldsymbol{X}$ if $U \le \underline{f}_n(\boldsymbol{x})/(\omega_n \, g_n(\boldsymbol{x}))$;
4. Otherwise, accept $\boldsymbol{X}_n = \boldsymbol{X}$ if $U \le f(\boldsymbol{x})/(\omega_n \, g_n(\boldsymbol{x}))$ and update $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{\boldsymbol{X}_n\}$.

An interesting feature of this algorithm is that the set $\mathcal{D}_n$ is only updated when $f(\boldsymbol{x})$ has been previously computed. As the algorithm produces variables the lower bound $\underline{f}_n(\boldsymbol{x})$ and upper bound $\overline{f}_n(\boldsymbol{x})$ become increasingly accurate and, therefore, we progressively reduce the number of evaluations of f.

2.2 Variance Reduction

We introduce several techniques that reduce variance while maintain the unbiasness of estimator.

- **Stratified sampling**: We partition the region $\mathcal{X}$ into k sub-regions $\mathcal{X}^i$ and suppose that the probability distribution f in each sub-regions is ***relative homogeneous***. Then we can generate samples $\boldsymbol{X}^i_1, \ldots, \boldsymbol{X}^i_{m_i}$ from each region $\mathcal{X}^i$ and compute the region-wise sample average

$$\bar{\mu}^i_{m_i} = \frac{1}{m_i} \sum_{j=1}^{m_i} f(\boldsymbol{X}^i_j)$$

  Then the final result is the average of region-wise sample average $\bar{\mu} = \sum_{i=1}^k \bar{\mu}^i_{m_i}$ and the variance $\bar{\sigma}^2 = \sum_{i=1}^k \frac{\sigma_i^2}{m_i}$, which is reduced from original.

  Clearly, if the probability distribution is not homogeneous within each region, the stratified sampling would increase the bias and makes the estimate less accurate.

- **Rao-Blackwellization**: An approach to reduce the variance of an estimator is to use the *conditioning inequality*

$$\operatorname{var}(\mathbb{E}\left[\delta(\boldsymbol{X})|\boldsymbol{Z}\right]) \le \operatorname{var}(\mathbb{E}\left[\delta(\boldsymbol{X})\right]) \tag{4}$$

  sometimes called ***Rao-Blackwellization*** [Liu, 2001] because the inequality is associated with the *Rao-Blackwell Theorem* [Lehmann and Casella, 2006], although the conditioning is

not always in terms of sufficient statistics. This method reflects a **basic principle**: ***one should carry out analytical computation as much as possible***. The more information available for sampler, the less variance it would have.

Suppose the sample can be decomposed into two parts $(\boldsymbol{X}, \boldsymbol{Z})$ and the conditional expectation can be carried out explicitly $\mathbb{E}\left[h(\boldsymbol{X})|\boldsymbol{Z}\right]$ for each sub-population $(\boldsymbol{X}, \boldsymbol{Z}_i)$, then the estimator

$$\hat{I}_m = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}\left[h(\boldsymbol{X})|\boldsymbol{Z}_i\right]$$

is unbias and has lower variance than the direct sample average of $h(\boldsymbol{X}_j)$. In statistic, $\hat{I}_m$ is often called ***mixture estimator*** since it combine a mixture of distributions for each sub-population.

- **Control Variates Methods**: In this method, one uses a ***control variate*** C that is highly correlated with sample $\boldsymbol{X}$ to reduce the variance. Suppose that $\mu_C = \mathbb{E}\left[C\right]$ is known and the sample estimator

$$h(X) := X(b) = X + b\left(C - \mu_C\right),$$

which has the same mean as $\boldsymbol{X}$. Suppose that the optimal weight $b^* = \mathrm{Cov}(X\,C)/\mathrm{var}(C)$ can be computed explicitly, the variance of $\boldsymbol{X}(b)$

$$\begin{aligned}\mathrm{var}(X(b)) &= \mathrm{var}(X) - 2\,b\,\mathrm{Cov}(X\,C) + b^2\,\mathrm{var}(C)\\ &= (1-\rho_{X,C}^2)\,\mathrm{var}(X) \le \mathrm{var}(X).\end{aligned}$$

Note that the technique of control variates is manageable only in very specific cases: the control function $\mu_C = \mathbb{E}\left[C\right]$ must be available, as well as the optimal weight b^*.

- **Antithetic Variates Methods**: This method describes a way of **creating negative correlated samples**. Suppose $X_1 = F^{-1}(U)$ for some random variable $U \sim \mathcal{U}[0,1]$. Note that $X_2 = F^{-1}(1-U)$ follows the same distribution as X_1. X_2 is called ***antithetic variables***. More generally, let g be a monotone function on $[0,1]$, so that for any $u_1, u_2 \in [0,1]$

$$\left(g(u_1) - g(u_2)\right)\left(g(1-u_1) - g(1-u_2)\right) \le 0.$$

Then for two i.i.d random variables $U_1, U_2 \sim \mathcal{U}[0,1]$, $X_1 = g(U)$ and $X_2 = g(1-U)$ we have

$$\mathbb{E}\left[\left(g(U_1) - g(U_2)\right)\left(g(1-U_1) - g(1-U_2)\right)\right] = \mathrm{Cov}(X_1\,X_2) \le 0$$

Thus the variance of $\frac{1}{2}(X_1 + X_2)$ is less than the variance of two independent samples from Monte Carlo simulator.

2.3 Importance Sampling

- The vanilla rejection sampling method suffers from ***low sample efficiency and slow convergence*** since it wasted a lot effort evaluating random samples located in regions where the target function value $h(\boldsymbol{x})p(\boldsymbol{x})$ is almost zero.
- The ***importance sampling*** idea suggests that one should **focus on regions of "importance"** so as to save computational resources. It is important to note that in high dimensional space, the support of the target distribution is exponentially small as compared to the entire region $\mathcal{X}$. In high dimensional setting, the **vanilla Monte Carlo methods are bound to fail**.

- Consider the expectation estimation

$$\begin{aligned}\mu = \mathbb{E}_{\boldsymbol{X}\sim p}\left[h(\boldsymbol{X})\right] &= \int_{\mathcal{X}} h(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \\ &= \mathbb{E}_{\boldsymbol{X}\sim g}\left[\frac{p(\boldsymbol{X})}{g(\boldsymbol{X})}h(\boldsymbol{X})\right] := \mathbb{E}_{\boldsymbol{X}\sim g}\left[w(\boldsymbol{X})h(\boldsymbol{X})\right] \\ &= \int_{\mathcal{X}} w(\boldsymbol{x})h(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x} \\ \text{where } w(\boldsymbol{x}) := \frac{p(\boldsymbol{x})}{g(\boldsymbol{x})} &\text{ is the } \textbf{\textit{importance weight.}}\end{aligned} \tag{5}$$

- The ***Importance Sampling Algorithm*** [Liu, 2001, Robert and Casella, 1999] is as below:
  1. Draw $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m \sim g(\boldsymbol{x})$ where g is trial distribution;
  2. Calculate *the importance weight* $w(\boldsymbol{x})$:

$$w_i := w(\boldsymbol{X}_i) = \frac{p(\boldsymbol{X}_i)}{g(\boldsymbol{X}_i)}, \quad i = 1, \ldots, m$$

  3. Approximate μ by

$$\hat{\mu} = \frac{\sum_{i=1}^m w_i\, h(\boldsymbol{X}_i)}{\sum_{i=1}^m w_i} \tag{6}$$

  Another way to approximate μ is via the unbiased estimator

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^m w_i\, h(\boldsymbol{X}_i) \tag{7}$$

The normalized estimator (6) is ***biased*** but as $m \to \infty$ $\hat{\mu} \to \mu$, i.e. it is ***asymptotically*** *unbiased*.

- While the (7) converges to $\mathbb{E}_p\left[h\right]$ almost surely, its variance

$$\mathbb{E}_g\left[\frac{p^2(\boldsymbol{X})}{g^2(\boldsymbol{X})}h^2(\boldsymbol{X})\right] = \mathbb{E}_p\left[\frac{p(\boldsymbol{X})}{g(\boldsymbol{X})}h^2(\boldsymbol{X})\right] = \int_{\mathcal{X}} \frac{p^2(\boldsymbol{x})}{g(\boldsymbol{x})}h^2(\boldsymbol{x})d\boldsymbol{x} < \infty \tag{8}$$

need to be finite. Thus $\text{supp}(g) \supset \text{supp}(p)$. In practice, the importance sampling works **poorly** (high-amplitude ***jumps***, ***instability*** of the path of the average, ***slow convergence***) when

$$\int_{\mathcal{X}} \frac{p^2(\boldsymbol{x})}{g(\boldsymbol{x})}d\boldsymbol{x} = \infty.$$

A "*Rule of thumb*" estimate of the variance is $\text{var}(\hat{\mu}) \approx \text{var}_g(h)\,\mathbb{E}_g\left[w^2\right]$. Geweke (1989) mentions two types of *sufficient conditions* for finite variance estimator:

1. $p(\boldsymbol{x})/g(\boldsymbol{x}) < M$, $\forall \boldsymbol{x} \in \mathcal{X}$ and $\text{var}(h) < \infty$;
2. $\mathcal{X}$ is compact, $p(\boldsymbol{x}) < F$ and $g(\boldsymbol{x}) > \epsilon$, $\forall \boldsymbol{x} \in \mathcal{X}$.

Note that the first condition implies that the rejection sampling also applies.

On the other hand, (6) can achieve finite variance and converges to $\mathbb{E}_p[h]$ when $m \to \infty$.

- The ***relative efficiency*** is defined as ratio between variance of vanilla Monte Carlo sampler and importance sampler:

$$\text{RE} = \frac{\frac{1}{m}\sum_{i=1}^{m} h(\boldsymbol{Y}_i)}{\frac{1}{m}\sum_{i=1}^{m} w_i\, h(\boldsymbol{X}_i)} \to 1,$$

if $w_i \to 1, \forall\, i$ or $\text{var}_g(w) \to 0$. Note that $\mathbb{E}_g[w] = \mathbb{E}_g\left[\frac{p}{g}\right] = 1$. Therefore $\text{RE} \leq 1$. This is the price we pay for sampling via simpler instrumental distribution g. If we use the normalized weight estimator (6), the relative efficiency is about

$$\text{RE} \approx \frac{1}{1 + \text{var}_g(w)} \tag{9}$$

The ***effective sample size (ESS)*** is thus

$$\text{ESS} \approx \frac{m}{1 + \text{var}_g(w)}.$$

These approximations do not involve h. Here we assume that

$$\mathbb{E}_p\left[(w(\boldsymbol{X}) - \mathbb{E}_p[w(\boldsymbol{X})])(h(\boldsymbol{X}) - \mu)\right] \text{ is small.}$$

- For optimal choice of g in (7), we have the following theorem

Theorem 2.3 *[Robert and Casella, 1999] The choice of g that minimizes the variance of the estimator* (7) *is*

$$g^*(\boldsymbol{z}) = \frac{|h(\boldsymbol{z})|\, p(\boldsymbol{z})}{\int_{\mathcal{X}} |h(\boldsymbol{z})|\, p(\boldsymbol{z}) d\boldsymbol{z}}.$$

- There are several **advantages** using importance sampling:
  - For (6), one only need to know target distribution $p(\cdot)$ ***up to a constant*** when calculating the importance weight. This is very convenient for distributions such as the exponential families. Also (6) has lower variance with higher bias as compared to (7).
  - Importance sampling allows us to **approximate** the expectation of an unknown complex target distribution using **simple trial distribution** and then ***correcting the bias*** via ***reweighting***. Similar to rejection sampling, the performance of importance sampling depends on the **closeness** of $g(\boldsymbol{x})$ to the target $h(\boldsymbol{x})p(\boldsymbol{x})$. In particular, the trial density $g(\boldsymbol{x})$ should have longer tail than $p(\boldsymbol{x})$ (i.e. $\text{supp}(g) \supset \text{supp}(p)$) so that the importance weight is well-defined in all support of target distribution. In high dimensional setting, it would still be challenging to find such a good trial distribution g.

3 Sequential Importance Sampling

3.1 Sequential Importance Sampling (SIS)

- It is nontrivial to find a good trial distribution g in high dimensional space. One of the most useful strategies in these problems is to *build up the trial density* ***sequentially***.

- Denote $\boldsymbol{x} := [x_1, \ldots, x_d]$. The trial distribution g can be *factorized* as

$$g(\boldsymbol{x}) = g(x_1) \prod_{i=2}^{d} g(x_i | x_1, \ldots, x_{i-1})$$

by which we hope to obtain some information on target density while building up the trial density. Note that the target density $p(\boldsymbol{x})$ can also be factorized as

$$p(\boldsymbol{x}) = p(x_1) \prod_{i=2}^{d} p(x_i | x_1, \ldots, x_{i-1}).$$

So the importance weight is calculated as

$$w(\boldsymbol{x}) = \frac{g(x_1) \prod_{i=2}^{d} g(x_i | x_1, \ldots, x_{i-1})}{p(x_1) \prod_{i=2}^{d} p(x_i | x_1, \ldots, x_{i-1})}. \tag{10}$$

From (10), we can reformulate the importance weight **recursively**

$$\begin{aligned} w_t(\boldsymbol{x}_t) &= w_{t-1}(\boldsymbol{x}_{t-1}) \frac{p(x_t | \boldsymbol{x}_{t-1})}{g(x_t | \boldsymbol{x}_{t-1})}, \quad t = 2, \ldots, d \\ w_1(\boldsymbol{x}_1) &= \frac{p(x_1)}{g(x_1)} \end{aligned} \tag{11}$$

where $\boldsymbol{x}_t = (x_1, \ldots, x_t)$.

- Note that computing $p(x_t | \boldsymbol{x}_{t-1})$ may be nontrivial since it require marginalization to find $p(\boldsymbol{x}_{t-1})$. Suppose we can find a sequence of ***auxiliary distributions*** $p_t(\boldsymbol{x}_t), t = 1, \ldots, d$ so that $p_t(\boldsymbol{x}_t)$ is a reasonable approximation of the *marginal* $p(\boldsymbol{x}_t)$ and $p_d(\boldsymbol{x}) = p(\boldsymbol{x})$. Note that $p_t(\boldsymbol{x}_t)$ only need to be known up to a constant and only serves as a "guide" to construction of whole samples.

  The ***Sequential Importance Sampling (SIS)*** algorithm is described as below:

  1. Draw $X_t = x_t$ from $g(x_t | \boldsymbol{x}_{t-1})$ and let $\boldsymbol{x}_t \leftarrow [x_t, \boldsymbol{x}_{t-1}]$.
  2. Compute the ***incremental weight***:

$$u_t = \frac{p_t(\boldsymbol{x}_t)}{p_{t-1}(\boldsymbol{x}_{t-1})\, g(x_t | \boldsymbol{x}_{t-1})} \tag{12}$$

  3. Compute $w_t \leftarrow w_{t-1}\, u_t$

  It is easy to show that $\boldsymbol{x}_t$ is properly weighted by w_t given that $\boldsymbol{x}_{t-1}$ is properly weighted by w_{t-1}. Thus the whole sample $\boldsymbol{x}$ obtained sequentially is properly weighted by the final importance w_d w.r.t. to target $p(\boldsymbol{x})$.

- The **benefits** for using sequential importance sampling include:
  - We can make use of the **characteristics of local factor** $p(x_t | \boldsymbol{x}_{t-1})$ in target density when designing trial density $g(x_t | \boldsymbol{x}_{t-1})$. An example is the *local Markov property* $p(x_t | \boldsymbol{x}_{t-1}) = p(x_t | x_{t-1})$ for probabilistic graphical models. By sequential sampling, we break the complex problem into smaller pieces.

– We can **stop** generating further components of $\boldsymbol{x}$ if the ***partial weights*** w_k that derived from the sequentially generated the ***partial samples*** $\boldsymbol{x}_k$ are *too small*. We can also **reject sample with small weight** and restart again. This way we avoid wasting effort generating samples with little effect on final estimation. This rejection process would introduce additional bias which should be corrected [Robert and Casella, 1999].

– The SIS algorithm is **attractive** since we can use a sequence of auxiliary distributions to construct more efficient sampling algorithm.

3.2 Nonlinear filtering

- Assume $\boldsymbol{\xi}_k \in \mathbb{R}^d$ is the state at time k and $\boldsymbol{y}_k$ is the observation at time k. $\boldsymbol{\xi}_{1:t} = [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_t] \in \mathbb{R}^{d\times t}$ is the trajectory of states up to t. The ***state-space model*** has the following equations

$$\begin{aligned} \boldsymbol{\xi}_t &\sim q_t(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1}, \theta) \quad \text{(state equation)} \\ \boldsymbol{y}_t &\sim f_t(\boldsymbol{y}_t|\boldsymbol{\xi}_t, \phi) \quad \text{(observation equation)} \end{aligned}$$

- A main challenge is to find the efficient methods for *online* ***estimation*** *and* ***prediction*** ***(filtering)*** of the state $\boldsymbol{\xi}_t$ given sequential observations $\boldsymbol{y}_{1:t} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t]$.

- By Bayes rule, the optimal online estimation of $\boldsymbol{\xi}_t$ given $\boldsymbol{y}_{1:t}$ is the *conditional mean estimator*

$$\begin{aligned} \widehat{\boldsymbol{\xi}}_t &= \mathbb{E}\left[\boldsymbol{\xi}_t \,|\, \boldsymbol{y}_{1:t}\right] \\ &= \int \boldsymbol{\xi}_t \, p(\boldsymbol{\xi}_t \,|\, \boldsymbol{y}_{1:t}) d\boldsymbol{\xi}_t \end{aligned} \tag{13}$$

  where the **posterior distribution** of state

$$\begin{aligned} p_t(\boldsymbol{\xi}_t) := p(\boldsymbol{\xi}_t \,|\, \boldsymbol{y}_{1:t}) &= \frac{f_t(\boldsymbol{y}_t \,|\, \boldsymbol{\xi}_t, \phi)\, p(\boldsymbol{\xi}_t \,|\, \boldsymbol{y}_{1:(t-1)})}{p(\boldsymbol{y}_t \,|\, \boldsymbol{y}_{1:(t-1)})} \\ &= \frac{\int \left[f_t(\boldsymbol{y}_t \,|\, \boldsymbol{\xi}_t, \phi)\, q_t(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1}, \theta)\right]\, p(\boldsymbol{\xi}_{t-1} \,|\, \boldsymbol{y}_{1:(t-1)}) d\boldsymbol{\xi}_{t-1}}{p(\boldsymbol{y}_t \,|\, \boldsymbol{y}_{1:(t-1)})} \\ &\propto \int \left[q_t(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1}, \theta) f_t(\boldsymbol{y}_t|\boldsymbol{\xi}_t, \phi)\right] p_{t-1}(\boldsymbol{\xi}_{t-1}) d\boldsymbol{\xi}_{t-1} \end{aligned} \tag{14}$$

  where $p_{t-1}(\boldsymbol{\xi}_{t-1}) := p(\boldsymbol{\xi}_{t-1} \,|\, \boldsymbol{y}_{1:(t-1)})$ is the "current" posterior distribution on state $\boldsymbol{\xi}_{t-1}$ and $p(\boldsymbol{y}_t \,|\, \boldsymbol{y}_{1:(t-1)})$ is a constant.

- There are several special cases

  – If q_t and f_t are both *linear Gaussian condition distributions*, the state-space model is called ***linear state-space model*** or ***dynamic linear model***. The recursive update (14) can be computed analytically since the current posterial distribution $p(\boldsymbol{\xi}_{t-1} \,|\, \boldsymbol{y}_{1:(t-1)})$ is also Gaussian. The resulting algorithm is the basis of ***Kalman filter*** [Liu, 2001].

  – If $\boldsymbol{\xi} \in \Xi$ is discrete with finite element, the state-space model is called ***Hidden Markov Model (HMM)***. The recursive update in (14) is close to the dynamic programming algorithm (*forward-backward algorithm or Baum-Welch*).

- Besides these two cases, exact computation the optimal online estimation $\widehat{\boldsymbol{\xi}}_t = \mathbb{E}\left[\boldsymbol{\xi}_t \,|\, \boldsymbol{y}_{1:t}\right]$ is impossible since the computation of normalization constant becomes infeasible when t grows.

- Instead, we can approximate $\widehat{\boldsymbol{\xi}}_t = \mathbb{E}\left[\boldsymbol{\xi}_t \,|\, \boldsymbol{y}_{1:t}\right]$ using sequential importance sampling (SIS). Define the trial distribution $g(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{1:(t-1)}, \boldsymbol{y}_{1:t}) = q_t(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1})$. We choose the ***auxiliary distribution*** $p_t(\boldsymbol{\xi}_{1:t}) = p(\boldsymbol{\xi}_{1:t} \,|\, \boldsymbol{y}_{1:t})$ and *importance weight* is updated via

$$
\begin{aligned}
w_t &= \frac{p(\boldsymbol{\xi}_{1:t}|\boldsymbol{y}_{1:t})}{g(\boldsymbol{\xi}_{1:t}|\boldsymbol{y}_{1:t})} \qquad (15)\\
&= \frac{p(\boldsymbol{\xi}_t, \boldsymbol{\xi}_{1:(t-1)}|\boldsymbol{y}_t, \boldsymbol{y}_{1:(t-1)})}{g(\boldsymbol{\xi}_t, \boldsymbol{\xi}_{1:(t-1)}|\boldsymbol{y}_t, \boldsymbol{y}_{1:(t-1)})} = \frac{p(\boldsymbol{\xi}_t, \boldsymbol{y}_t, \boldsymbol{\xi}_{1:(t-1)}|\boldsymbol{y}_{1:(t-1)})}{g(\boldsymbol{\xi}_t, \boldsymbol{\xi}_{1:(t-1)}|\boldsymbol{y}_{1:t})} \frac{1}{p(\boldsymbol{y}_t \,|\, \boldsymbol{y}_{1:(t-1)})}\\
&\propto \frac{p(\boldsymbol{\xi}_t, \boldsymbol{y}_t|\boldsymbol{\xi}_{1:(t-1)}, \boldsymbol{y}_{1:(t-1)})}{g(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{1:(t-1)}, \boldsymbol{y}_{1:t})} \frac{p(\boldsymbol{\xi}_{1:(t-1)}|\boldsymbol{y}_{1:(t-1)})}{g(\boldsymbol{\xi}_{1:(t-1)}|\boldsymbol{y}_{1:(t-1)})}\\
&= \frac{f_t(\boldsymbol{y}_t|\boldsymbol{\xi}_t, \phi)\, q_t(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1})}{g(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{1:(t-1)}, \boldsymbol{y}_{1:t})}\, w_{t-1}\\
&= \frac{f_t(\boldsymbol{y}_t|\boldsymbol{\xi}_t, \phi)\, q_t(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1})}{q_t(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1})}\, w_{t-1} = f_t(\boldsymbol{y}_t|\boldsymbol{\xi}_t, \phi)\, w_{t-1} := u_t\, w_{t-1}. \qquad (16)
\end{aligned}
$$

  The *incremental weight* $u_t = f_t(\boldsymbol{y}_t|\boldsymbol{\xi}_t, \phi)$.

- Note that normalization on $w_{t+1}^{(j)} = w_1^{(j)} \prod_{k=1}^{t} f_k(\boldsymbol{y}_{k+1}|\boldsymbol{\xi}_{k+1}^{(j)}, \phi)$ is equivalent to applying ***softmax*** operation on $\left\{\prod_{k=1}^{t} f_k(\boldsymbol{y}_{k+1}|\boldsymbol{\xi}_{k+1}^{(j)}, \phi)\right\}_j$. As $k \to \infty$, we will see $w_{t+1}^{(j)} \to 1$ but all other $w_{t+1}^{(i)} \to 0\ \forall i \neq j$. This phenomenon is called ***particle degeneracy***.

- A simple method called ***particle filter*** (or ***boostrap filter***) is proposed to fix the particle degeneracy by ***weight resampling*** [Liu, 2001].

  The ***Sampling-Importance-Resampling (SIR)***

  1. Draw $\boldsymbol{\xi}_{t+1}^{(*,j)}$ from the state equation $q_t(\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_t^{(j)}, \theta)$ for $j = 1, \ldots, m$
  2. Weight each draw by $w_{t+1}^{(j)} = f_t(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1}^{(*,j)}, \phi)\, \widetilde{w}_t^{(j)}$
  3. Normalize $w_{t+1}^{(j)}$ as $\widetilde{w}_{t+1}^{(j)}$
  4. **Resample** from $\left\{\boldsymbol{\xi}_{t+1}^{(*,1)}, \ldots, \boldsymbol{\xi}_{t+1}^{(*,m)}\right\}$ according to multinomial distribution with probability $\left\{\widetilde{w}_{t+1}^{(j)}\right\}_{j=1}^{m}$ to produce a random sample $\left\{\boldsymbol{\xi}_{t+1}^{(1)}, \ldots, \boldsymbol{\xi}_{t+1}^{(m)}\right\}$ for time $t+1$

  Averaging $\left\{\boldsymbol{\xi}_{t+1}^{(1)}, \ldots, \boldsymbol{\xi}_{t+1}^{(m)}\right\}$ will obtain the approximate conditional posterior mean estimator at $t+1$.

- The sequence $(\boldsymbol{\xi}_1^{(j)}, \ldots, \boldsymbol{\xi}_t^{(j)})$ is called j-th ***particle trajectory***. The particle filter maintains m particles in the system at each time stamp.

- It is easy to see that if the $\boldsymbol{\xi}_t^{(j)}$ follow the "current" posterior distribution $p_t(\boldsymbol{\xi}_t)$ and if m is large enough, then the new random samples $\left\{\boldsymbol{\xi}_{t+1}^{(1)}, \ldots, \boldsymbol{\xi}_{t+1}^{(m)}\right\}$ follows the updated posterior $p_{t+1}(\boldsymbol{\xi}_{t+1})$ ***approximately***.

References

Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

Jun S Liu. *Monte Carlo strategies in scientific computing*, volume 10. Springer, 2001.

Christian P Robert and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.