Tutorial on Monte Carlo Techniques

Gabriel A. Terejanu
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
terejanu@buffalo.edu

1 Introduction

Monte Carlo (MC) technique is a numerical method that makes use of random numbers to solve mathematical problems for which an analytical solution is not known. The first article, "The Monte Carlo Method" by Metropolis and Ulam, has appeared for the first time in 1949 [9], even though well before that certain statistical problems were solved using random numbers. Since the simulation of random numbers is very time consuming, MC has became practical only with the advent of computers.

A simple MC simulation is the determination of π. Suppose we have a circle with radius $r = 1$ inscribed within a square. Then the area ratio is:

$$\frac{A_{\text{circle}}}{A_{\text{square}}} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4} \tag{1}$$

To determine π we randomly select n points, $p_i = (x_i, y_i)$ for $i = 1 \dots n$, in the square and approximate the area ratio by:

$$\frac{A_{\text{circle}}}{A_{\text{square}}} = \frac{m}{n} \tag{2}$$

where m is the number of random points in the circle, they must satisfy $x_i^2 + y_i^2 \leq 1$. Hence $\pi = 4 \times \frac{m}{n}$.

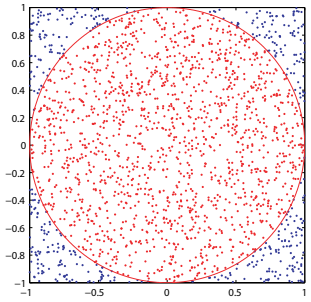

Figure 1: Compute π

This method is called **hit-or-miss Monte Carlo** since the estimate is computed as the actual ratio of hits to random tries. It is the least efficient MC method.

2 Random Variables

The "Monte Carlo" name is derived from the city, with the same name, in the Principality of Monaco, well known for its casinos. This was because the roulette wheel was the simplest mechanical device for generating random numbers [10].

Even though "random variable" implies that one cannot predict its value, the **distribution** may well be known. The distribution of a random variable gives the probability of a given value [7].

The probability distribution of a **discrete random variable** is a list of probabilities associated with each of its possible values. It is also sometimes called the **probability function** or the **probability mass function**. (Valerie J. Easton and John H. McColl's Statistics Glossary v1.1).

A **continuous random variable**, x, takes any values in a certain interval (a, b). It is defined by **probability density function (pdf)** $p(x)$ and the given interval. The probability of x_i falling in an arbitrary interval (a', b') is given by:

$$\mathrm{P}\{a' \leq x \leq b'\} = \int_{a'}^{b'} p(x)dx \tag{3}$$

The pdf has to satisfy the following two conditions:

1. $p(x) \geq 0$ for any $x \in (a, b)$
2. $\int_a^b p(x)dx = 1$

The **expected value** or mean, the **second moment** and the **variance** or central second moment of the random variable are respectively defined by:

$$E[x] = \mu = \int_a^b xp(x)dx \tag{4}$$

$$E[x^2] = \int_a^b x^2 p(x)dx \tag{5}$$

$$Var[x] = \sigma^2 = E[(x-\mu)^2] = \int_a^b (x-\mu)^2 p(x)dx = E[x^2] - \mu^2 \tag{6}$$

where σ is the **standard deviation**.

Consider now two continuous random variables x and y. We say that x and y are statistically **independent** if the distribution of x does not depend on y and vice-versa. Hence the joint probability density function $f_{xy}(x, y) = f_x(x)f_y(y)$.

The **covariance** and the **correlation** of two random variables:

$$Cov[x, y] = E\left[(x - E[x])(y - E[y])\right] = E[xy] - E[x]E[y] \tag{7}$$

$$Corr[x, y] = \frac{Cov[x, y]}{\sqrt{Var[x]Var[y]}} \tag{8}$$

Note: If x and y are uncorrelated then their covariance and the correlation coefficient is zero, hence $E[xy] = E[x]E[y]$. "Statistically independent random variables are always uncorrelated, but uncorrelated random variables can be dependent. Let x be uniformly distributed over $[-1, 1]$ and let $y = x^2$. The two random variables are uncorrelated but are clearly not independent" [8].

3 Generate Random Variables

The "Monte Carlo" name is derived from the city, with the same name, in the Principality of Monaco, well known for its casinos. This was because the roulette wheel was the simplest mechanical device for generating random numbers [10].

"A sequence of **truly** random numbers is unpredictable and therefore unreproducible. Such a sequence can only be generated by a random physical process" [7]. Practically it is very difficult to construct such a physical generator, which has to be fast enough, and to connect it to a computer. To overcome this problem one can use **pseudo-random** numbers, which are computed according to a mathematical formulation, hence are reproducible and appear random to someone who does not know the algorithm [12].

John von Neumann has constructed the first pseudorandom generator called **mid-square method**. Suppose we have a 4-digit number $x_1 = 0.9876$. We square it, $x_1^2 = 0.97535376$, obtaining this way a 8-digit number. We obtain x_2 by taking out the middle four digits, hence $x_2 = 0.5353$. Now square x_2 and so on. Unfortunately, the algorithm tends to produce disproportionate frequency of small numbers [10].

One popular algorithm is the **multiplicative congruential method** suggested by D.H. Lehmer in 1949. Given a modulus m, a multiplier a and a starting point x_1, the method generates successive pseudo-random numbers by the formula [7]:

$$x_i = ax_{i-1}Mod(m) \tag{9}$$

4 MC integration

Let $f(x)$ be an arbitrary continuous function and $y = f(x)$ the new random variable. Then the expected value and the variance of y:

$$E[y] = E[f(x)] = \int_a^b f(x)p(x)dx \tag{10}$$

$$Var[y] = Var[f(x)] = \int_a^b \left(f(x) - E[f(x)]\right)^2 p(x)dx \tag{11}$$

Note: $E[f(x)] \neq f(E[x])$.

Our goal is to calculate the expectation of $f(x)$ without computing the integral. This can be achieved by using a MC simulation.

Crude Monte Carlo: A simple estimate of the integral (10) can be obtained by generating n samples $x_i \sim q(x)$, for $i = 1 \dots n$, and computing the estimate $I = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$. To determine the accuracy of method we first have to introduce two theorems.

The law of large numbers implies that the average of a sequence of random variables, of a known distribution, converge to its expectation as the numbers of samples goes to infinity. Let us select n numbers x_i randomly with probability density $p(x)$. Hence:

$$I = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \quad \rightarrow \quad E[f(x)] = \int_a^b f(x)p(x)dx \tag{12}$$

In the statistical context $A(n)$ is a **consistent estimator** of B if "$A(n)$ is said to **converge** to B as n goes to infinity if for any probability $P[0 < P < 1]$, and any positive quantity δ, a k can be found such that for all $n > k$ the probability that $A(n)$ will be within δ of B is greater that P" [7].

The central limit theorem: the sum of a large number of independent random variables is approximately normally distributed. A sum of identical and independent distribution (i.i.d) random variables z_i with mean μ an finite variance σ^2:

$$\frac{\sum_{i=1}^{n} z_i - n\mu}{\sigma\sqrt{n}} \rightarrow \mathcal{N}(0,1) \text{ as } n \rightarrow \infty \tag{13}$$

The error of the MC method [2]:

$$\begin{aligned} error &\equiv \left|I - E[f(x)]\right| \\ &= \left|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - E[f(x)]\right| \\ &= \frac{\sqrt{Var[f(x)]}}{\sqrt{n}}\left|\frac{\sum_{i=1}^{n} f(x_i) - nE[f(x)]}{\sqrt{Var[f(x)]}\sqrt{n}}\right| \\ &= \frac{\sqrt{Var[f(x)]}}{\sqrt{n}}\left|\mathcal{N}(0,1)\right| \end{aligned} \tag{14}$$

Mathematical properties [7]

1. If the variance of $f(x)$ is finite, the MC estimate is consistent
2. The MC estimate is asymptotically unbiased
3. The MC estimate is asymptotically normally distributed
4. The standard deviation of the MC estimate is given by $\sigma = \frac{\sqrt{Var[f(x)]}}{\sqrt{n}}$

Also note that the MC methods are immune to the curse of dimensionality. The accuracy of the method can be improved by increasing the number of samples, n, but the convergence is very slow. A better way to improve the accuracy is to decrease the variance, $Var[f(x)]$. This methods are called **variance-reducing techniques**.

5 Variance Reducing Techniques

5.1 Stratified Sampling

This technique divides the full integration space into subspaces, for which a MC integration is performed. The final result is the sum of all partial results. We divide the integration domain R into k regions R_i for $i = 1 \dots k$.

$$E[f(x)] = \int_R f(x)p(x)dx = \sum_{j=1}^{k} \int_{R_j} f(x)p(x)dx \tag{15}$$

The MC estimate of the expectation becomes:

$$I = \sum_{j=1}^{k} \frac{Vol(R_j)}{n_j} \sum_{i=1}^{n_j} f(x_i) \tag{16}$$

where n_j is the number of points used for MC integration on R_j, and $Vol(R_j)$ is the subdomain volume.

The MC variance:

$$\sigma^2 = \sum_{j=1}^{k} \frac{Vol^2(R_j)}{n_j} Var_{R_j}[f(x)] \tag{17}$$

$$\text{where} \quad Var_{R_j}[f(x)] = \frac{1}{Vol(R_j)} \int_{R_j} \left(f(x) - \frac{1}{Vol(R_j)} \int_{R_j} f(x)p(x)dx \right)^2 p(x)dx \tag{18}$$

By selecting carefully the number of points this can lead to a lower variance compared with the crude MC. If not it can perform worse than the crude MC.

5.2 Importance Sampling

The pdf under the integral, $p(x)$, may not be the best pdf for MC integration. In this case we want to use a different and simpler pdf $q(x)$ from which we can draw the samples. $q(x)$ is called the **importance density** or the **proposal distribution**. Therefore we can write:

$$E[y] = E[f(x)] = \int_a^b f(x)p(x)dx \tag{19}$$

$$= \int_a^b \frac{f(x)p(x)}{q(x)} q(x)dx \tag{20}$$

$$= E\left[\frac{f(x)p(x)}{q(x)}\right] \tag{21}$$

By generating n samples $x_i \sim q(x)$, for $i = 1 \dots n$, the estimate becomes:

$$I = \frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)p(x_i)}{q(x_i)} = \frac{1}{n} \sum_{i=1}^{n} W(x_i)f(x_i) \tag{22}$$

where $W(x_i) = \frac{p(x_i)}{q(x_i)}$ are the **importance weights**. Since the normalizing factor $p(x_i)$ is not know, we have to normalize the weights such that $\sum_{i=1}^{n} W(x_i) = 1$:

$$I = \frac{\frac{1}{n}\sum_{i=1}^{n} W(x_i) f(x_i)}{\frac{1}{n}\sum_{i=1}^{n} W(x_i)} = \frac{1}{n}\sum_{i=1}^{n} W_n(x_i) f(x_i) \tag{23}$$

where $W_n(x_i) = \frac{W(x_i)}{\sum_{i=1}^{n} W(x_i)}$ are the **normalized importance weights**. This estimate is biased but the bias vanishes asymptotically as $n \to \infty$ [4].

The variance of the importance sampler estimate is given by [4]:

$$Var_q[f(x)] = \frac{1}{n}\int \left[\frac{(f(x)p(x))^2}{q(x)}\right] dx - \frac{E_p^2[f(x)]}{n} \tag{24}$$

To reduce the variance $q(x)$ should be chosen to match the shape of $p(x)$ or $|f(x)|\,p(x)$ [4].

5.3 Control Variates

Suppose that we have a function $g(x)$ for which $E[g(x)]$ is known. Let:

$$h(x) = f(x) - c\big(g(x) - E[g(x)]\big) \tag{25}$$

where $h(x)$ can have a smaller variance than $f(x)$. Our estimate is given now by:

$$I = \frac{1}{n}\sum_{i=1}^{n}\big(f(x_i) - cg(x_i)\big) + cE[g(x)] \tag{26}$$

The variance of $h(x)$:

$$Var[h(x)] = Var[f(x)] + c^2 Var[g(x)] - 2cCov[f(x), g(x)] \tag{27}$$

where $Cov[f(x), g(x)] = E\left[(f(x) - E[f(x)])(g(x) - E[g(x)])\right]$.

We want to minimize the variance $Var[h(x)]$ with respect to c:

$$\frac{\mathrm{d}Var[h(x)]}{\mathrm{d}c} = 2cVar[g(x)] - 2Cov[f(x), g(x)] \tag{28}$$

By setting this to zero, the optimal value for c is then given by:

$$c = \frac{Cov[f(x), g(x)]}{Var[g(x)]} \tag{29}$$

Then the variance $Var[h(x)]$ becomes:

$$Var[h(x)] = Var[f(x)] - \frac{Cov^2[f(x), g(x)]}{Var[g(x)]} = Var[f(x)]\left(1 - Corr^2[f(x), g(x)]\right) \tag{30}$$

where $Corr[f(x), g(x)] = \frac{Cov[f(x),g(x)]}{\sqrt{Var[f(x)]}\sqrt{Var[g(x)]}}$ is the correlation coefficient. Hence the quality of the estimation depends on the correlation between $f(x)$ and $g(x)$.

5.4 Rejection Sampling (or Acceptance Rejection Sampling)

Suppose that we know an upper bound for the underlying pdf $p(x)$ and we use a proposal distribution $q(x)$ as in importance sampling. So, there is $c < \infty$ such that $p(x) < cq(x)$ [4].

Algorithm 1 Rejection Sampling

1: Select a uniform random variable $u \sim \mathcal{U}(0, 1)$
2: Draw a sample $x \sim q(x)$
3: **if** $u < \frac{p(x)}{cq(x)}$ **then**
4: accept x
5: **else**
6: reject it and repeat the process
7: **end if**

One should carefully select the value for c. If it is too small the rejection rate will be low; if it is too big the acceptance rate will be low.

5.5 Antithetic Variates

Consider that we have an even number of samples, $2n$, drawn from $p(x)$. One approach is to generate correlated samples to reduce the variance by cancellations in their sum. The estimate:

$$I = \frac{1}{2n}\sum_{i=1}^{2n} f(x_i) \tag{31}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_{2i-1}) + f(x_{2i})}{2} \tag{32}$$

$$= \frac{1}{n}\sum_{i=1}^{n} g(x_{2i}) \tag{33}$$

where $g(x_{2i}) = \frac{f(x_{2i-1})+f(x_{2i})}{2}$.

The variance $Var[g(x_{2i})]$:

$$Var[g(x_{2i})] = Var\left[\frac{f(x_{2i-1}) + f(x_{2i})}{2}\right] \tag{34}$$

$$= \frac{1}{4}\big(Var[f(x_{2i-1})] + Var[f(x_{2i})] + 2Cov[f(x_{2i-1}), f(x_{2i})]\big) \tag{35}$$

Variance analysis:

1. If $Cov[f(x_{2i-1}), f(x_{2i})] = 0$ or $f(x_{2i-1}), f(x_{2i})$ are independent then $Var[g(x_{2i})] = \frac{1}{2}Var[f(x_{2i-1})]$, hence the variance of the estimator remains the same.
2. If $Cov[f(x_{2i-1}), f(x_{2i})] < 0$ then the estimate is improved.
3. If $Cov[f(x_{2i-1}), f(x_{2i})] > 0$ the actual performance is worse.

5.6 Partial averaging (or Rao-Blackwellization)

The method is based on the idea that "one should carry out analytical computation as much as possible" [3]. So, if we can analytically compute the integral over a subspace, this yields the same expected value but with a smaller variance.

From the Law of Iterated Expectations we can compute the conditional expectation of $f(x)$ for some sigma algebra $\mathcal{F}$:

$$E[f(x)] = E\big[E[f(x)|\mathcal{F}]\big] \tag{36}$$

From the Law of Total Variance:

$$Var[f(x)] = Var\big[E[f(x)|\mathcal{F}]\big] + E\big[Var[f(x)|\mathcal{F}]\big] \tag{37}$$

Therefore $Var[f(x)] \geq Var\big[E[f(x)|\mathcal{F}]\big]$. In practice we can factor $f(x)$ such that some of the parts to be computed analytical and the rest can be solved by using Monte Carlo simulation. The method can be very problem dependent due to this factorization and sometimes may be more computational expensive than just simulating the entire system [3].

6 Markov Chain Monte Carlo

MCMC deals with generating samples from some complex probability distribution $p(x)$ that is difficult to directly sample from. The approach is to use a sequence of samples $\{x_k\}$ from a Markov chain, such that the sequence converges on $p(x)$ as $k \to \infty$.

In the following section only the case of **discrete-time, continuous state space Markov process** is discussed.

6.1 Markov chain

A Markov process is a stochastic process that has the **Markov property**. A stochastic process has the Markov property if the conditional probability distribution of the future states given all the past states depends only on the current state:

$$p(x_{k+1}|x_0, x_1 \dots x_k) \quad = \quad p(x_{k+1}|x_k) \tag{38}$$

A **Markov chain** is a sequence of random variables $\{x_k\}$ generated by a Markov processes. The value of x_k is called the **state** of the process at time k. The **one step transition probability density (transition kernel)** $p(y|x)$ is the probability density that the process moves from x to y. The transition kernel must satisfy:

$$\int p(y|x)dy \quad = \quad 1 \tag{39}$$

Definition: The Markov chain is said to be **homogeneous/stationary** if the transition kernels do not depend on the time.

Define $p(x)$ the **total probability density** of x, and $p(x_0)$ the initial probability density of x_0. The total probability density of y is given by the **Chapman-Kolmogorov equation**:

$$p(y) = \int p(y|x)p(x)dx \tag{40}$$

Theorem: "If a Markov chain is **ergodic**, then there exists a **unique steady state distribution** π independent of the initial state" [4].

$$\pi(y) = \int p(y|x)\pi(x)dx \tag{41}$$

Reversible condition (detailed balance property): Guarantees the invariance of π under the transition kernel:

$$p(x|y)\pi(y) = p(y|x)\pi(x), \forall x, y \tag{42}$$

"The unconditional probability of moving y to x is equal to the unconditional probability of moving x to y, where x and y are both generated from $\pi(\cdot)$" [4].

"In MCMC sampling framework the samples are generated by a homogeneous, reversible, ergodic Markov chain with invariant distribution π" [4].

6.2 Metropolis-Hastings Algorithm

Published in 1953 by Nicholas Metropolis and generalized in 1970 by W.K.Hastings, the algorithm draws samples from the target density $\pi(x)$, for which the normalizing factor might be unknown.

A Markov chain is generated with a **proposal density (candidate target)** $q(y|x)$, where $\int q(y|x)dx = 1$. The proposal density generates a value for y when the process is at x [5]. If $q(y|x)$ satisfies the reversible condition then our search for a transition kernel is over.

Suppose, without loss of generality, that:

$$q(y|x)\pi(x) > q(x|y)\pi(y) \tag{43}$$

which means that the process moves from x to y much often than from y to x [5]. To reduce the number of moves from x to y one can introduce the **probability of move**, $\alpha(y|x)$, such that the transition kernel from x to y is given by:

$$p_{MH}(y|x) \equiv q(y|x)\alpha(y|x) \tag{44}$$

From (43) set $\alpha(x|y) = 1$, because we need as many moves as we can get from y to x. To determine $\alpha(y|x)$, one must require $p_{MH}(y|x)$ to have the detailed balance property:

$$p_{MH}(y|x)\pi(x) = p_{MH}(x|y)\pi(y) \tag{45}$$

$$\begin{aligned} q(y|x)\alpha(y|x)\pi(x) &= q(x|y)\alpha(x|y)\pi(y) \\ &= q(x|y)\pi(y) \end{aligned} \tag{46}$$

Thus the probability of move is given by:

$$\alpha(y|x) \quad = \quad min\left[\frac{q(x|y)\pi(y)}{q(y|x)\pi(x)}, 1\right] \tag{47}$$

Algorithm 2 Metropolis-Hasting

Draw x_0 from $p(x_0)$
for $i = 0$ to N **do**
  Generate $y \sim q(y|x_i)$
  Generate $u \sim \mathcal{U}(0, 1)$
  if $u \leq \alpha(y|x_i)$ **then**
    $x_{i+1} = y$
  else
    $x_{i+1} = x_i$
  end if
end for
return $\{x_{Nb}, x_{Nb+1}, \dots x_N\}$

The number of steps, N, determines the algorithm performance. We need a good amount of steps, denoted Nb, for the chain to approaches stationarity (burn-in period). Another important factor for the efficiency is to have many samples or a high acceptance rate for the values of y. The draws are regarded as samples from $\pi(x)$ only after the burn-in period Nb [5].

6.3 Gibbs Sampling

The method was introduced by German and German in 1984 and named after the physicist J.W. Gibbs. The algorithm is a special case of Metropolis-Hastings for generating a sequence of samples from a joint distribution $p(\mathbf{x})$ hard to sample, of two or more random variables.

Let $\mathbf{x}$ be a n-dimensional random vector: $\mathbf{x} = [x_1, x_2, \dots x_n]^T$. The Gibbs sampler uses only **univariate** conditional distributions, where all of the random variables but one are assigned fixed values.

Algorithm 3 Gibbs Sampling

Draw $\mathbf{x}^0$ from $p(\mathbf{x}^0)$, results : $\mathbf{x}^0 = [x_1^0, x_2^0, \dots x_n^0]^T$
for $i = 1$ to N **do**
  Draw $x_1^i \sim p(x_1|x_2^{i-1}, x_3^{i-1}, \dots x_n^{i-1})$
  ...
  Draw $x_n^i \sim p(x_n|x_1^{i-1}, x_2^{i-1}, \dots x_{n-1}^{i-1})$
  $\mathbf{x}^i = [x_1^i, x_2^i, \dots x_n^i]^T$
end for
return $\{\mathbf{x}_{Nb}, \mathbf{x}_{Nb+1}, \dots \mathbf{x}_N\}$

where Nb denotes the burn-in period.

"The Gibbs sampler is a nice solution to estimation of hierarchical or structured probabilistic model. Gibbs sampling can be viewed as a Metropolis method in which the proposal distribution is defined in terms of the conditional distributions of the joint distribution and every proposal is always accepted" [4].

7 Sequential Monte Carlo

In this section we focus our attention on sequential state estimation using sequential Monte Carlo (SMC). SMC is known also as bootstrap filtering, particle filtering, the condensation algorithm, interacting particle approximations and survival of the fittest [1].

7.1 Sequential Importance Sampling (SIS)

Consider the following nonlinear system, described by the difference equation and the observation model:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \quad (48)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_{k-1}) \quad (49)$$

Denote by $\mathbf{Z}_k = \{\mathbf{z}_i | 1 \leq i \leq k\}$ the set of all observations up to time k, conditionally independent given the process with distribution $p(\mathbf{z}_k|\mathbf{x}_k)$. Also, assume that the state sequence $\mathbf{x}_k$ is an unobserved (hidden) Markov process with initial distribution $p(\mathbf{x}_0)$ and transition distribution $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ [6].

Our aim is to estimate recursively the posterior distribution $p(\mathbf{x}_k|\mathbf{Z}_k)$ and expectations of the form[6]:

$$E[f(\mathbf{x}_k, \mathbf{w}_k)] = \int f(\mathbf{x}_k, \mathbf{w}_k) p(\mathbf{x}_k|\mathbf{Z}_k) d\mathbf{x}_k \quad (50)$$

Suppose that we cannot sample from the posterior distribution and use an importance sampling approach to sample from a **proposal distribution** $q(\mathbf{x}_k|\mathbf{Z}_k)$. Hence:

$$E[f(\mathbf{x}_k, \mathbf{w}_k)] = \int f(\mathbf{x}_k, \mathbf{w}_k) \frac{p(\mathbf{x}_k|\mathbf{Z}_k)}{q(\mathbf{x}_k|\mathbf{Z}_k)} q(\mathbf{x}_k|\mathbf{Z}_k) d\mathbf{x}_k \quad (51)$$

$$= \int f(\mathbf{x}_k, \mathbf{w}_k) \frac{p(\mathbf{Z}_k|\mathbf{x}_k)p(\mathbf{x}_k)}{p(\mathbf{Z}_k)q(\mathbf{x}_k|\mathbf{Z}_k)} q(\mathbf{x}_k|\mathbf{Z}_k) d\mathbf{x}_k \quad (52)$$

$$= \int f(\mathbf{x}_k, \mathbf{w}_k) \frac{W_k(\mathbf{x}_k)}{p(\mathbf{Z}_k)} q(\mathbf{x}_k|\mathbf{Z}_k) d\mathbf{x}_k \quad (53)$$

where $W_k(\mathbf{x}_k) = \frac{p(\mathbf{Z}_k|\mathbf{x}_k)p(\mathbf{x}_k)}{q(\mathbf{x}_k|\mathbf{Z}_k)} \propto \frac{p(\mathbf{x}_k|\mathbf{Z}_k)}{q(\mathbf{x}_k|\mathbf{Z}_k)}$ are the unnormalized importance weights. One can get rid of the unknown normalizing density $p(\mathbf{Z}_k)$ by deriving the expectation as:

$$E[f(\mathbf{x}_k, \mathbf{w}_k)] = \frac{\int f(\mathbf{x}_k, \mathbf{w}_k) W_k(\mathbf{x}_k) q(\mathbf{x}_k|\mathbf{Z}_k) d\mathbf{x}_k}{\int p(\mathbf{Z}_k|\mathbf{x}_k) p(\mathbf{x}_k) \frac{q(\mathbf{x}_k|\mathbf{Z}_k)}{q(\mathbf{x}_k|\mathbf{Z}_k)} d\mathbf{x}_k} \quad (54)$$

$$= \frac{E_q[W_k(\mathbf{x}_k) f(\mathbf{x}_k, \mathbf{w}_k)]}{E_q[W_k(\mathbf{x}_k)]} \quad (55)$$

By drawing N samples $\{\mathbf{x}_k^i\} \sim q(\mathbf{x}_k|\mathbf{Z}_k)$, $i = 1 \dots N$, one can approximate the expectations:

$$E[f(\mathbf{x}_k, \mathbf{w}_k)] \approx \frac{\frac{1}{N}\sum_{i=1}^N W_k(\mathbf{x}_k^i) f(\mathbf{x}_k^i, \mathbf{w}_k)}{\frac{1}{N}\sum_{i=1}^N W_k(\mathbf{x}_k^i)} \tag{56}$$

$$= \sum_{i=1}^N \bar{W}_k(\mathbf{x}_k^i) f(\mathbf{x}_k^i, \mathbf{w}_k) \tag{57}$$

where $\bar{W}_k(\mathbf{x}_k^i) = \frac{W_k(\mathbf{x}_k^i)}{\sum_{i=1}^N W_k(\mathbf{x}_k^i)}$ are the normalized weights.

Since it is very hard to find a good proposal distribution in a high dimensional space, one can sequentially construct it. Suppose we can factorize the proposal distribution:

$$q(\mathbf{x}_k|\mathbf{Z}_k) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}_k) q(\mathbf{x}_{k-1}, \mathbf{Z}_{k-1}) \tag{58}$$

The posterior distribution can be factorized as:

$$p(\mathbf{x}_k|\mathbf{Z}_k) = p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}) \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \tag{59}$$

$$\propto p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}) p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{60}$$

Hence we can recursively update the weights:

$$W_k(\mathbf{x}_k^i) \propto \frac{p(\mathbf{x}_k^i|\mathbf{Z}_k)}{q(\mathbf{x}_k^i|\mathbf{Z}_k)} \tag{61}$$

$$= \frac{p(\mathbf{x}_{k-1}^i|\mathbf{Z}_{k-1}) p(\mathbf{z}_k|\mathbf{x}_k^i) p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{Z}_k) q(\mathbf{x}_{k-1}^i, \mathbf{Z}_{k-1})} \tag{62}$$

$$\propto W_{k-1}(\mathbf{x}_{k-1}^i) \frac{p(\mathbf{z}_k|\mathbf{x}_k^i) p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{Z}_k)} \tag{63}$$

Algorithm 4 Sequential Importance Sampling

1: Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{Z}_k)$ for $i = 1 \dots N$

2: Compute the importance weights $W_k^i = W_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i) p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{Z}_k)}$

3: Normalize the importance weights $\bar{W}_k^i = \frac{W_k^i}{\sum_{i=1}^N W_k^i}$

Degeneracy problem: After a few iterations all but one particle will have negligible weights. Hence the algorithms allots time to update a large number of weights with no effect in our sampling.

Proposition (Kong et al. 1994): The unconditional variance (that is, when the observations are regarded as random) of the importance ratios increases over time, where the **importance ratio** is given by:

$$W_k(\mathbf{x}_k^i) \propto \frac{p(\mathbf{x}_k^i|\mathbf{Z}_k)}{q(\mathbf{x}_k^i|\mathbf{Z}_k)} \tag{64}$$

This problem can be overcome by adding a resampling strategy to SIS algorithm.

7.2 Sequential Importance Resampling (SIR)

The resampling step eliminates the samples with low importance weights and multiplies the ones with high importance weights. "Many of the ideas on resampling have stemmed from the work of Efron 1982, Rubin 1988, and Smith and Gelfand 1992" [11].

To detect the weights degeneracy (or **sample impoverishment**), one can use the **effective sample size** N_{eff} given by [4]:

$$N_{eff} = \frac{N}{1 + Var_q[\bar{W}_k(\mathbf{x}_k)]} \tag{65}$$

$$= \frac{N}{E_q[\bar{W}_k(\mathbf{x}_k^2)]} \tag{66}$$

An estimate of the effective sample size:

$$\hat{N}_{eff} = \frac{N}{\sum_{i=1}^{N}(\bar{W}_k^i)^2} \tag{67}$$

If $\hat{N}_{eff}$ is less than an given threshold then run the resampling step.

One approach is to draw N particles from the current set of particles with probabilities proportional to their weights and set the new weights to $\frac{1}{N}$. This can be done by generating a new set of indices from a CDF that maps the random measure $\{\mathbf{x}_k^i, \bar{W}_k^i\}$ into an equally weighted random measure $\{\mathbf{x}_k^j, \frac{1}{N}\}$ [11].

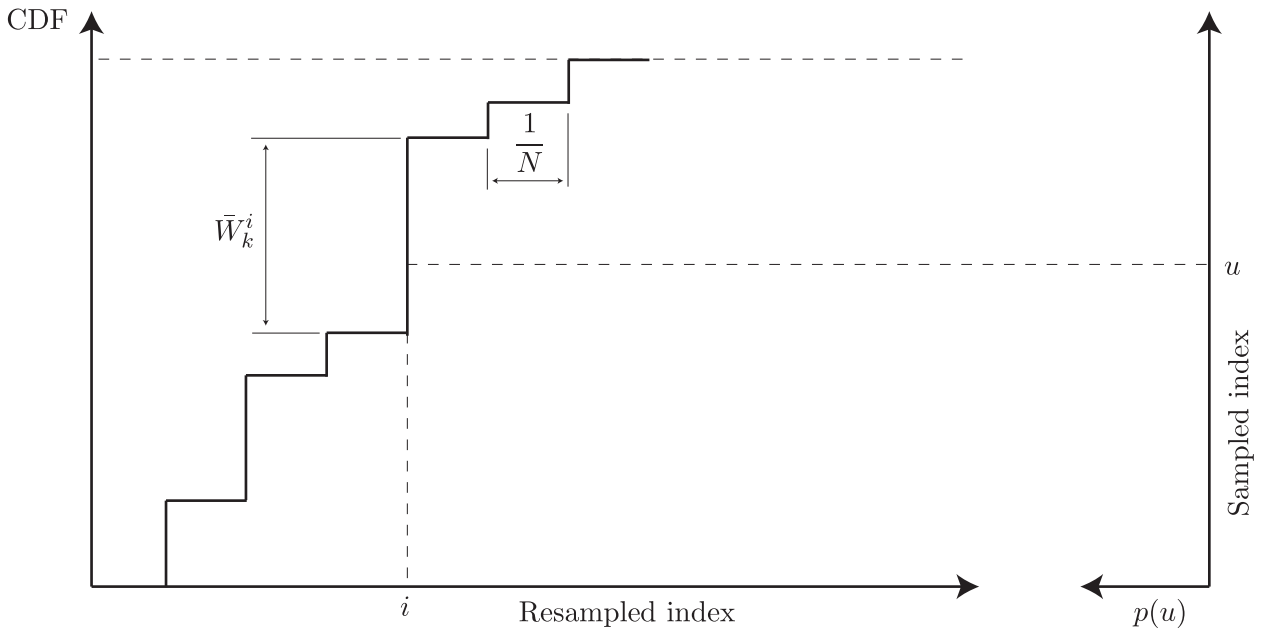

Figure 2: Resampling process [11]

Algorithm 5 Resampling Algorithm

1: Initialize CDF : $c_1 = 0$
2: Construct CDF : $c_i = c_{i-1} + \bar{W}_k^i$, for all $i = 1 \dots N$
3: **for** $j = 1$ to N **do**
4: Sampling index: $u \sim \mathcal{U}(0, 1)$
5: Resampling index: $i = CDF^{-1}(u)N$
6: Assign sample: $\tilde{\mathbf{x}}_k^j = \mathbf{x}_k^i$
7: Assign weight: $\bar{W}_k^j = \frac{1}{N}$
8: **end for**
9: **return** the new set $\{\tilde{\mathbf{x}}_k^j, \bar{W}_k^j\}$

where $CDF^{-1}(\cdot)$ is the inverse of the $CDF(\cdot)$ function.

This step reduces the sample impoverishment effect but introduces new practical problems: it limits the opportunity to have a parallel algorithm, particles with high weights are statistically selected many times and leads to a loss of diversity among the particles [1].

References

[1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Trans. on Signal Processing*, 50, 2002.

[2] Paul J. Atzberger. Strategies for Improving the Efficiency of Monte-Carlo Methods. http://www.cims.nyu.edu/ paulatz/class_finance/improvingMonteCarloMethods.pdf.

[3] K.R. Beevers. Monte Carlo integration. Lecture Notes for CSCI-6971, 2006.

[4] Zhe Chen. Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. Manuscript.

[5] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 1995.

[6] A. Doucet, S. Godsill, and C. Andrieu. On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. Technical report, Signal Processing Group, Department of Engineering, University of Cambridge, UK.

[7] F. James. Monte Carlo Theory and Practice. *Rep. Prog. Phys.*, 43:1147–1188, 1980.

[8] D. Johnson. Jointly Distributed Random Variables. http://cnx.org/content/m11248/latest/, 2005.

[9] N. C. Metropolis and S. M. Ulam. The Monte-Carlo Method. *J. Amer. Stat. Assoc.*, 44:335–341, 1949.

[10] Ilya M. Sobol'. *A Primer for the Monte Carlo Method*. CRC Press, 1994.

[11] Rudolph van der Merwe, Nando de Freitas, Arnaud Doucet, and Eric Wan. The unscented particel filter. In *Advances in Neural Information Processing Systems 13*, Nov 2001.

[12] S. Weinzierl. Introduction to Monte Carlo Methods. Technical report, NIKHEF Theory Group, 2000.