Chapter 1

Sampling Algorithms

1.1 Inverse Method

Consider a distribution $p(x)$ which we want to obtain samples from. Now, let be z a value inside $[0, 1]$ such as:

$$z = h(y) = P(y < X) = \int_{-\infty}^{y} p(x)dx \tag{1.1}$$

where $h(y)$ is the value of the cumulative distribution in point y. Thus, $y = h^{-1}(z)$. Consider the case of a exponential distribution.

$$p(x) = \lambda \exp(-\lambda x)$$

And we know that

$$\begin{aligned} z = P(y < X) &= 1 - P(y \geq X) \\ &= 1 - \int_{y}^{\infty} p(x)dx \\ &= 1 - \exp(-\lambda y) \end{aligned}$$

Then, the samples are obtained using the inverse transform $h^{-1}(z)$ which is equal to

$$y = -\frac{1}{\lambda} \ln(1 - z)$$

The problem with this method is that sometimes is impossible to obtain the inverse of the cumulative distribution we want to get samples from and other times can be computationally expensive.

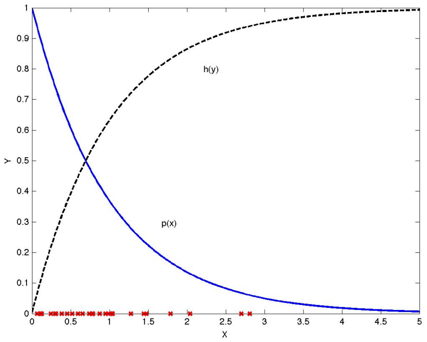

Figure 1.1: 30 samples (red points) of the beta distribution obtained with the inverse transform algorithm.

1.2 Rejection Sampling

Be $p(x)$ a Gaussian distribution in the interval $[a, b]$ with mean μ and variance σ^2. The objective is to draw samples of this distribution. Let be $g(x)$ a uniform distribution in the same interval. We have to choose a value k in such a way that $kg(x)$ is above $p(x)$ in the support of $p(x)$. See figure 1.2.

The Rejection Sampling algorithm works in the following way, take a random point z_1 in the range $[a, b]$, that point will be accepted as a valid sample of the Gaussian distribution with probability $p(z_1)/kg(z_1)$. Thus, to know if z_1 is accepted just take a random number z_2 in the range $[0, kg(z_1)]$ if $z_2 \leq f(z_1)$, then is accepted, otherwise, is rejected. This process is repeated until the desired number of samples are accepted.

The probability for a point to be accepted as a sample of $f(x)$ is equal to:

$$p(accepted) = \frac{\int_a^b p(x)dx}{\int_a^b kg(x)dx} = \frac{1}{k\int_a^b g(x)dx} = \frac{1}{k}$$

Thus, bigger the value of k, less the probability of a point of being accepted. According to the geometric distribution, expected number of rejected samples before accepting the first one, is given by:

$$\mathbb{E}[\#rejected\_samples] = \frac{1-p}{p}$$

where p is the probability of being accepted. For our example the probability of being accepted can be seen as the area bellow the Gaussian divided by the

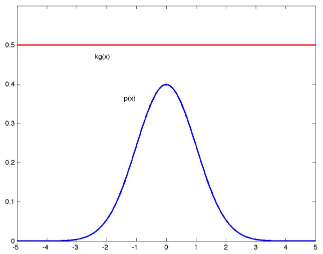

Figure 1.2: $f(x)$ is Gaussian distribution in the interval $[-55]$, using $k = 5, \mu = 0, \sigma^2 = 1$. Meanwhile $g(x)$ is represented by an uniform distribution.

area of the rectangle with width $b - a$ and height $k/(b - a)$.

$$\begin{aligned} p(accepted) &= \frac{1}{k \int_a^b g(x)dx} \\ &= \frac{1}{k \int_a^b 1/(b-a)dx} \\ &= \frac{1}{k(b-a)/(b-a)} \\ &= \frac{1}{k} \end{aligned}$$

Thus, the number of rejected points before accept the first one, is given by

$$\#rejected\_samples = \frac{1 - 1/k}{1/k} = k - 1$$

In figure 1.3 we can see the accepted and rejected points for our example using $a = -5, b = 5, k = 5, \mu = 0$ and $\sigma^2 = 1$. There are 30 accepted points and 119 rejected points, that represent a 20.13% of accepted samples, which is very close to the expected value $1/k$.

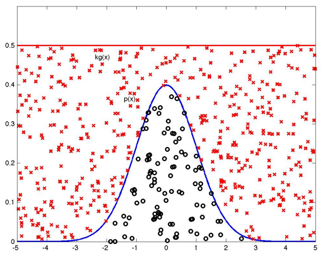

Figure 1.3: Red points represent rejected samples, and blue black points represent accepted samples.

1.2.1 Adaptive Rejection Sampling

The Adaptive Rejection Sampling (ARS) algorithm consists on updating the envelope function $kg(x)$ every time a sample is rejected. The goal is that the envelope function to be closer as possible to the distribution we want to get samples from. Is important to mention that this algorithm only works for distributions that are log concave.

The first step consists on defining three initial points randomly and find its corresponding value in $\log f(x)$, then trace a line between consecutive points and chose those places where the lines are above $\log f(x)$. See figure 1.4.

Each line segment in $\log f(x)$ represent an exponential in $f(x)$. the next step is to obtain samples from those exponential functions $k_i g_i(x)$, defined by

$$k_i g_i(x) = \begin{cases} 0 & x < z_i \\ \exp(m_i + c_i) & z_i \leq x \leq z_{i+1} \\ 0 & x > z_{i+1} \end{cases}$$

where z_i represent the i^{th} point in envelope. Thus. if there are n points in the envelope there will be $n-1$ exponential functions. Now, we need that $g(x)$ to be a valid probability distribution, such as

$$\int_{z_i}^{z_{i+1}} k_i^{-1} \exp(m_i x + c_i) dx = 1$$

Solving for k_i, we obtain

$$k_i = \frac{1}{m_i} \left[\exp(m_i z_{i+1} + c_i) - \exp(m_i z_i + c_i)\right]$$

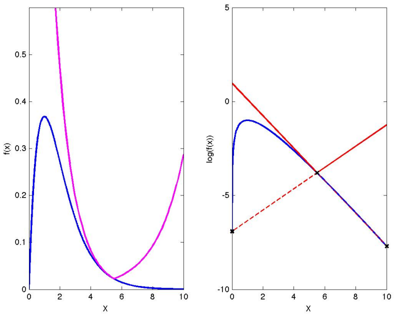

Figure 1.4: At left the function we want to sample from is a gamma distribution with $\alpha = 2, \beta = 1$ and the initial envelope function. At right the log concave function and the initial three points and the lines above that function.

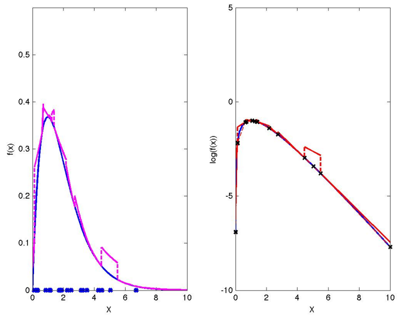

Figure 1.5: At left the final envelope function formed with 12 rejected points. At right the chosen lines above the log concave function.

Now we can see the distribution $g(x)$ as mixture of exponentials defined as

$$g(x) = \sum_{i=1}^{n-1} w_i k_i^{-1} g_i(x)$$

where w_i is the mixture proportion and is given by

$$w_i = \frac{k_i}{\sum_{i=1}^{n-1} k_i}$$

The value of w_i represent the area bellow the exponential $k_i g_i(x)$ divided by the total area. The second step of the ARS algorithm is to choose a exponential function to sample from, where the i^{th} function has a probability of being chose of w_i. Sampling from an exponential exponential function can be done by the **inverse method**, choosing a random number $u \in [0, 1]$, we have:

$$t = \frac{1}{m_i} \left\{ \ln \left[u k_i m_i + \exp(m_i z_i + c_i) \right] - c_i \right\}$$

Just as in RS algorithm, a sample t is accepted with probability

$$p(t_{acc}) = \frac{f(t)}{k_i g_i(t)}$$

If the point is rejected, then that point is used to refine the envelope function. Figure 1.5 shows the final result of the algorithm, with 30 accepted samples and 12 rejected samples (including the initial three points). A proportion of 71.4% of accepted samples.