Sampling from Arbitrary Distributions using the Rejection Method
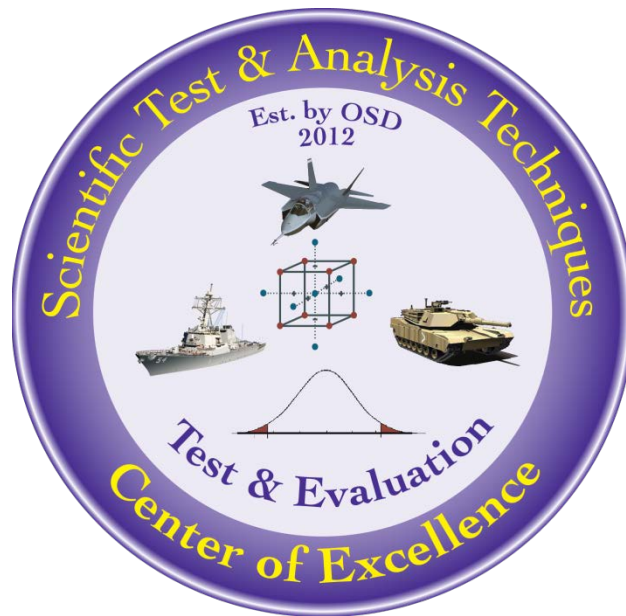
Authored by:

Nathaniel Choo

Kyle Kolsti, PhD

5 November 2018

The goal of the STAT COE is to assist in developing rigorous, defensible test strategies to more effectively quantify and characterize system performance and provide information that reduces risk. This and other COE products are available at www.afit.edu/STAT.

Table of Contents

Executive Summary

Many computer programs and programming languages have built-in functions to randomly sample from common probability distributions (uniform, normal, etc.). This capability is particularly useful when performing Monte Carlo simulations. However, what is one to do if actual data are used to build the model, and none of those common distributions adequately represent the data? The rejection method is a process that can be used to sample from any distribution – even those with complex functional forms from which direct sampling may be difficult. This paper discusses how to use the rejection method to sample from an arbitrary distribution and provides advice on how to apply the method.

Keywords: sampling, MCMC, Monte Carlo, rejection, acceptance-rejection, Bayesian

Introduction

Monte Carlo simulation is an analysis technique that requires random sampling from distributions. For example, consider a case where the analyst wishes to predict a physical parameter, y, where $y = x/(x + w)$ and x and w are random variables. The analyst has decided that x comes from a uniform distribution between the values a and b, and w comes from a normal distribution with mean μ and standard deviation σ. In statistical notation, where upper-case letters represent populations and lower-case letters represent a particular instance of a population, these assumptions can be written as $X \sim \mathrm{Unif}(a, b)$ and $W \sim \mathrm{Normal}(\mu, \sigma)$. Thus x is an observation drawn from X and w is an observation drawn from W. Monte Carlo simulation involves repeatedly calculating y via the formula $y = x/(x + w)$ using different instances of x and w each time. In this manner, a large set of values for y can be obtained, displayed as a histogram, and used to make inferences and decisions.

To perform this Monte Carlo simulation in Excel, the fundamental function is RAND(), which produces a random sample from a uniform distribution within the range [0,1]. Thus, every value from zero to one has the same probability of occurrence. In this example, a value for x can be sampled by typing in the function =RAND()*(a-b)+b, and a value for w can be obtained by using the equation =NORM.INV(RAND(),μ, σ) with the variables replaced by the corresponding cell references. Programming languages like Python, R, and MATLAB have built-in functions that efficiently perform the same operations for common distributions.

Sometimes, however, the underlying distributions cannot be adequately represented by one of these simple formulas. For example, Fig. 1 is a JMP-produced histogram of wave data from a buoy off the Oregon coast. In particular, it displays the occurrences of the average wave height in feet for each hour of December, 2004[1]. The histogram clearly indicates a bimodal distribution (i.e., the histogram has two peaks), so it is likely there will be simple equations or built-in functions that will allow random

---

[1] https://www.ndbc.noaa.gov/station_history.php?station=46029

sampling from it. The rejection method, also called the acceptance-rejection method, is a way to obtain those random samples.
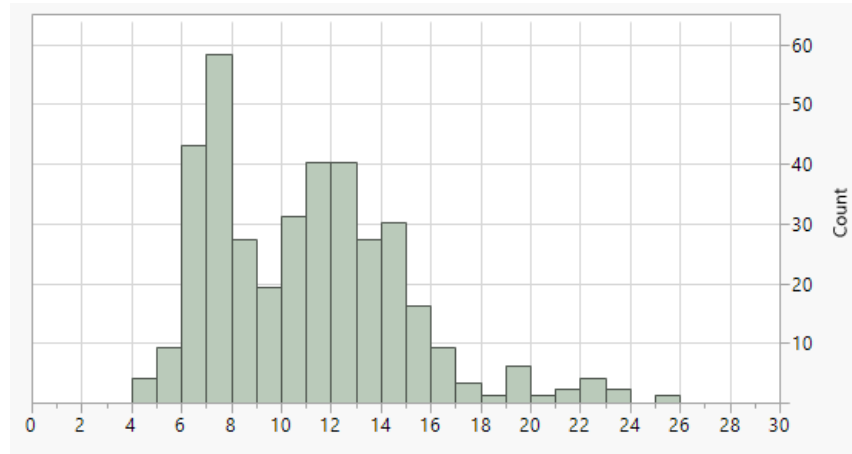
Figure 1: Histogram of hourly average wave heights (feet) from December, 2004

Another application is Bayesian analysis. In this setting, random sampling often provides a more computationally efficient alternative to numerical integration. However, the underlying distribution is typically complex, leading to the development of numerous Markov Chain Monte Carlo (MCMC) sampling methods. The rejection method is a simple algorithm that can be employed for this purpose.

Method

Overview

This section explains the process of randomly sampling from a one-dimensional distribution using the rejection method as described in Gamerman (2006). The concept is the same for multi-dimensional distributions. Suppose there is a need to sample a random value x_i from a target distribution, $\pi(x)$. The expression for $\pi(x)$ is known, so the value of the function can be calculated for any given value of x. Figure 2 illustrates an example of $\pi(x)$.

To capitalize on the built-in functions of the software, define the auxiliary distribution function, $q(x)$ from which a random value for x can easily be drawn by a formula. Practical, commonly-available choices include a uniform distribution or a normal distribution as mentioned in the introduction. Figure 2 depicts an example auxiliary distribution function.

Next, choose a value for the "envelope constant," A, such that $0 < A < \infty$. The envelope constant multiplies the auxiliary distribution to create the "blanket function," $Aq(x)$. The selections of q and A must be such that the blanket function satisfies the following condition:

$$Aq(x) \geq \pi(x) \text{ for all } x \qquad (1)$$
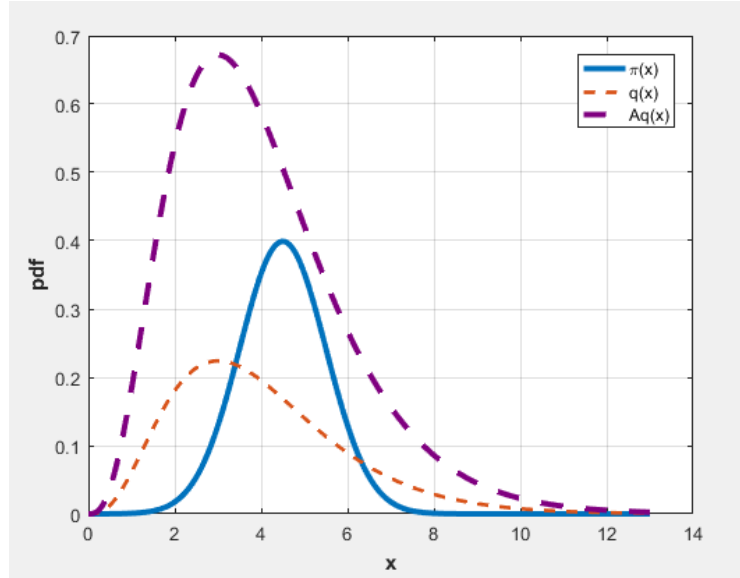
Figure 2: Target distribution (normal), auxiliary distribution (gamma), and blanket function where *A*=3

In other words, the blanket function must be at or above the target distribution everywhere. Later sections will provide advice for building the blanket function and checking if it violates the assumption. Figure 2 demonstrates a blanket function (purple dashed line) that is at or above the target distribution (solid blue line) everywhere, satisfying the requirement in Eq. (1).

Having built the blanket function Aq, follow the next steps multiple times to obtain the desired sample size. Obtain values for the random variables x_i and u_i, where u_i is drawn from the population $U \sim \text{Unif}(0,1)$. Recall that x_i is drawn from the conveniently-chosen distribution q. Then, use the following decision rule to determine whether to accept or reject the current value of x_i:

$$\begin{cases} u_i Aq(x_i) \leq \pi(x_i), & \text{accept } x_i \\ u_i Aq(x_i) > \pi(x_i), & \text{reject } x_i \end{cases} \tag{2}$$

This process is repeated a set number of times, or until the number of accepted values of x_i is adequate for the analysis. A summary of the rejection method is listed below.

Steps

<u>INITIALIZATION</u>:

1. Identify the target distribution, $\pi(x)$.

2. Choose an auxiliary distribution function, $q(x)$.

3. Choose an envelope constant, A, such that the condition of Eq. (1) is satisfied.

<u>FOR ITERATION i</u>:

4. Draw u_i from $\text{Unif}(0,1)$.

5. Draw x_i from $q(x)$.

6. Calculate $\pi(x_i)$.

7. Calculate $q(x_i)$.

8. Use Eq. (2) to decide whether to accept or reject x_i.

Repeat steps $4 - 8$ either a set number of times or until a desired number of values for x have been accepted. When enough values have been obtained, they can be used to pull off the $p^{(th)}$ quantile to calculate a nonparametric estimate of the mean and variance.

Examples

Example 1: Demonstrating the method

The purpose of this first example is to demonstrate that the rejection method works by showing that after taking many samples it reproduces a well-known distribution. This procedure is also useful for verifying code to make sure it is working properly before using it on more complicated distributions.

For Step 1, let the target distribution to be sampled, $\pi(x)$, be a normal distribution with a mean of 4.5 and a standard deviation of 1. For Step 2, choose the auxiliary distribution, $q(x)$, to be a gamma distribution with a mean of 4 and a standard deviation of 2, which results in the parameters $shape = 4$ and $scale = 1$. (There is no particular reason to use the gamma distribution here – it was chosen primarily to distinguish it from the normal target distribution and to facilitate the examples to follow.) Though theoretically the normal distribution extends from $-\infty$ to ∞ and the gamma distribution extends from 0 to ∞, it is reasonable here to only consider values of x between 0 and 13. These target distribution and auxiliary distribution were shown in Fig. 2 as solid blue and dashed orange lines, respectively.

For Step 3, the envelope constant A must be chosen so that the blanket function $Aq(x)$ satisfies the condition of Eq. (1). Figure 2 shows the blanket function (dashed purple) for this example with $A = 3$. The blanket function is above the target function for all values of x, so it satisfies Eq. (1).

Having verified that the blanket function satisfies Eq. (1), the sampling process can begin. The steps of the method as described previously will be listed. Steps 1-3 were just completed in the previous paragraphs but are repeated here for completeness.

STEP 1: Let $\pi(x)$ be $\mathrm{Normal}(4.5,1)$

STEP 2: Choose $q(x)$ to be $\mathrm{Gamma}(4,1)$

STEP 3: Choose $A = 3$

The remaining steps are to randomly sample to obtain x_i and u_i, then test if Eq. (2) is satisfied. The following table shows numerical examples that would lead to accepting and rejecting the sampled x_i.

Table 1: Example calculations

STEP	Example for accept	Example for reject
4 – Draw u_i from $Unif(0,1)$	$u_i = 0.1500$	$u_i = 0.7300$
5 – Draw x_i from $q(x)$	$x_i = 3.0643$	$x_i = 1.4199$
6 – Calculate $\pi(x_i)$	$\pi(x_i) = 0.1423$	$\pi(x_i) = 0.0035$
7 – Calculate $q(x_i)$	$q(x_i) = 0.2239$	$q(x_i) = 0.1153$
8 – Decide using Eq. (2)	$0.1008 \leq 0.1423$? Yes→ Accept x_i	$0.2526 \leq 0.0035$? No→ Reject x_i

Steps 4-8 were performed until 20,000 values of x were accepted. Figure 3 overlays the histogram of those 20,000 accepted points along with a plot of $\pi(x)$. By inspection, the histogram of values sampled from $\pi(x)$ reproduce $\pi(x)$, demonstrating that the rejection method successfully drew random samples from the target distribution π. Importantly, note that the method was able to do this without using a random number-generating expression for the target distribution, which was a normal distribution in this case.
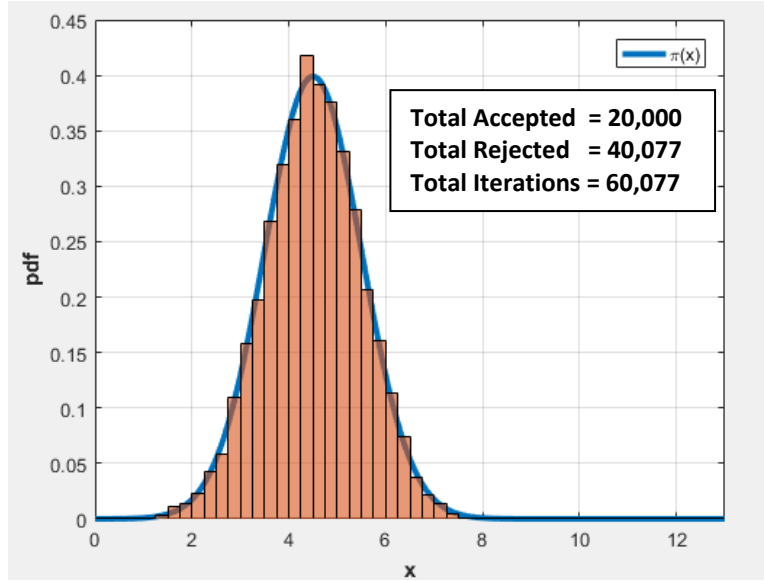
Figure 3: Histogram of 20,000 values drawn from $\pi(x)$ with *A* = 3

## Example 2: When A is excessively large

The choices of A and q affect the computational efficiency of the algorithm. Any blanket function Aq that satisfies the condition of Eq. (2) will work; however, a blanket function excessively larger than the target distribution will require substantially more iterations to collect a certain number of accepted points. This inefficiency can be seen by manipulating Eq. (2) to the following form:

$$u_i \leq \frac{\pi(x_i)}{Aq(x_i)} \tag{3}$$

According to Eq. (3), for a proposed point x_i to be accepted, u_i must be less than the ratio of the amplitudes of the target and blanket functions at x_i. As the amplitude of the blanket function grows, the ratio becomes smaller and it's less likely that u_i will be less than the ratio. Thus, more proposed points will be rejected. This could be problematic because as the number of rejections increases, the total amount of time and cost to reach a desired number of accepted values also increases.

The previous example will now be revisited to demonstrate the drop in efficiency with a large-magnitude blanket function. This time, the target distribution and auxiliary distribution are the same as in Example 1; however, the envelope constant, A, is chosen to be 8. This scenario is shown in Fig. 4. Clearly the ratios of the amplitudes (target/blanket) are very small.
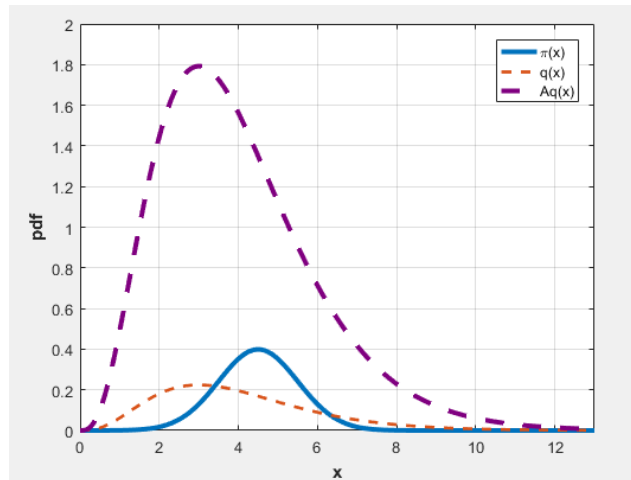
Figure 4: Distributions and blanket function with *A* = 8
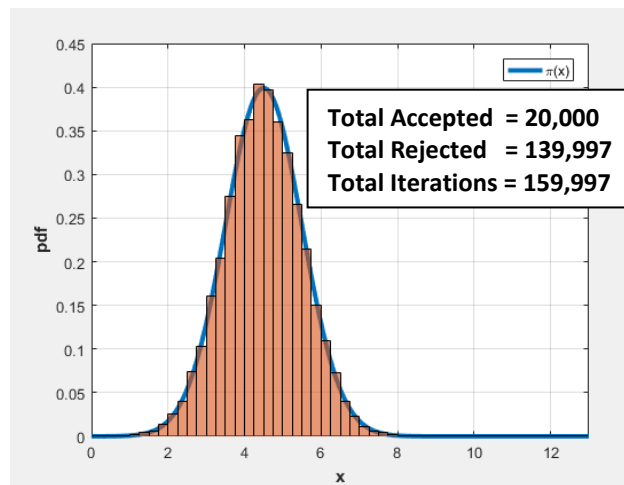
Figure 5: Histogram of 20,000 values drawn from $\pi(x)$ with *A* = 8

As before, the rejection method was used to obtain 20,000 accepted values. The results in Fig. 5 demonstrate that since the blanket function meets the condition, the target distribution was again successfully reproduced. However, with $A = 8$ instead of $A = 3$, the number of iterations required to accept 20,000 values increased from approximately 60,000 to 160,000. By calculating efficiency as the proportion of proposed values that were accepted, the efficiency dropped from 33.3% to 12.5%. Note that both values are equal to $1/A$ – since q has an area of one by definition, the blanket function has an area of A. Thus, the target distribution area makes up "$1/A$ th" the area of the blanket function, a direct indication of how difficult it is for a proposed value to be accepted. Though handy here for conceptual purposes, this prediction would be difficult to utilize in complex problems.

For complex, multi-dimensional problems the inefficiency of large values for A can lead to excessive computation times. A clever choice for q can permit smaller values of A. Clearly it is beneficial from a

computational perspective for a blanket function to satisfy the condition of Eq. (1) by only a small margin.

## Example 3: When the blanket function violates its condition

In the last section, it was shown that it is computationally beneficial to satisfy the condition of Eq. (1) by as small a margin as possible. However, for complex problems, the choices of A and q are not obvious. What happens if in the pursuit of efficiency, A and q are chosen such that the condition of Eq. (1) is inadvertently violated? This example demonstrates a way to recognize that such a situation has occurred so the method can be re-executed with a new blanket function.

Once again, the same target (normal) and auxiliary (gamma) distributions will be used. However, this time, the envelope constant is set to $A = 1$. Equation (1) is not satisfied as shown in Fig. 6 since the blanket function is not greater than the target distribution for all values of x. Consequently, Fig. 7 shows that the method failed to randomly draw from the target distribution.
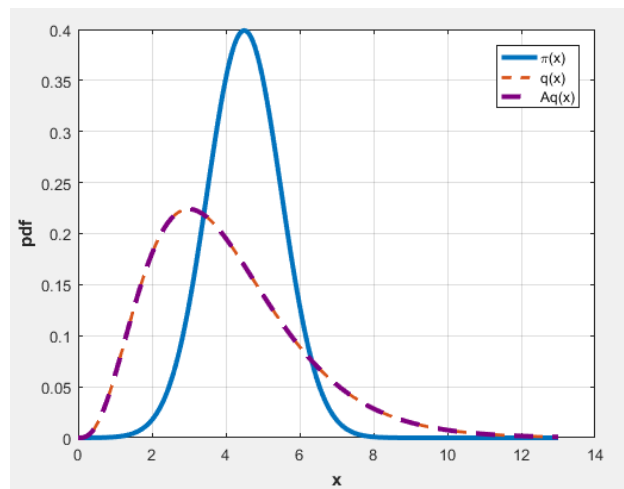

Figure 6: Distributions and blanket function with *A* = 1

The reason the method fails when Eq. (1) is not satisfied can be demonstrated by inspecting the distribution of the rejections that occur. In Fig. 7, look at the region where the condition is violated, when x is between 3.4 and 6.3. Because the amplitude of the blanket function Aq is less than that of the target distribution π, and the random value u cannot exceed one, uAq is always less than π. Therefore, every point will be accepted and none will be rejected in this interval. This result is confirmed in Fig. 8, which shows the histogram of rejected values. When there are unexpected regions of the rejection histogram where no points were rejected, it is likely that the blanket function is not covering the arbitrary distribution.
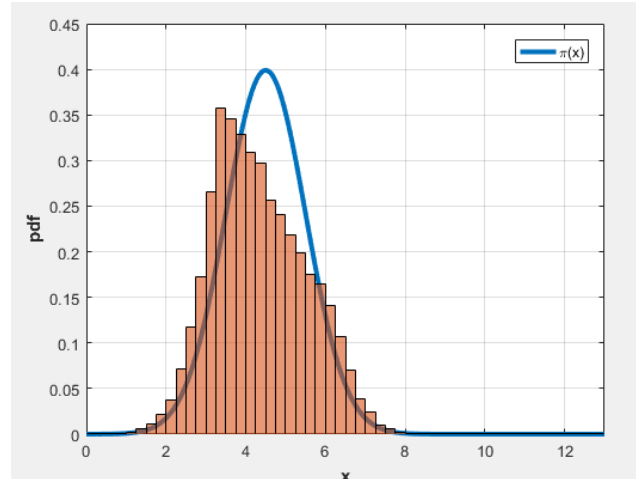
Figure 7: Result of 20,000 random draws from $\pi(x)$ with A = 1

Referring back to the predictor of efficiency, one would expect an efficiency of 100% since $1/A = 1$. This makes sense because if $A = 1$, the only way the condition of Eq. (1) can be satisfied is if π and q are the same distribution. Then, it will always be true that $uAq \leq \pi$ at a given x. This same logic also leads to the conclusion that A must be greater than or equal to one. For this example, it turns out that only about 60% of the attempts were accepted, indicating a problem with the blanket function.
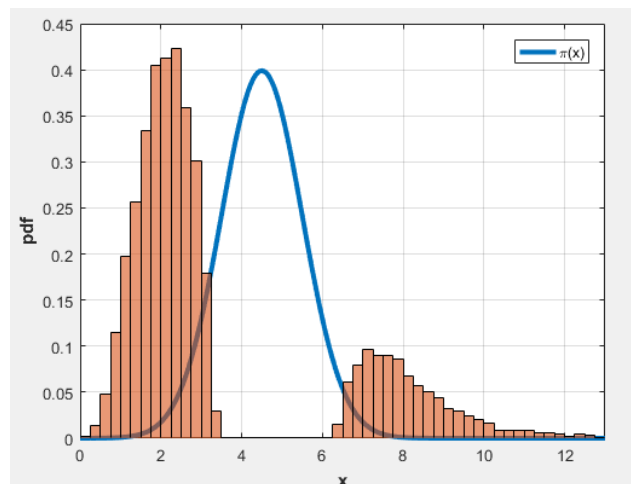
Figure 8: Histogram of rejected values with A = 1

Example 4: Sampling from a complicated distribution

As a final demonstration of the rejection method, consider a deliberately complicated piece-wise target distribution, shown in Fig. 9 by the solid blue line. There are no known convenient functions that can randomly sample from this distribution. The auxiliary distribution, q, was chosen to be the uniform

distribution from 0 to 8. The envelope constant was chosen to be $A = 2.4$. The auxiliary distribution and the blanket function are also shown in Fig. 9 (dashed orange line and dashed purple line, respectively). By visual inspection, the condition of Eq. (1) is satisfied.

The histograms of accepted and rejected values are shown in Fig. 10. There are no regions where zero rejections occurred, confirming that Eq. (1) was not inadvertently violated. The histogram of accepted points matches the target distribution, confirming the validity of the sampling method.
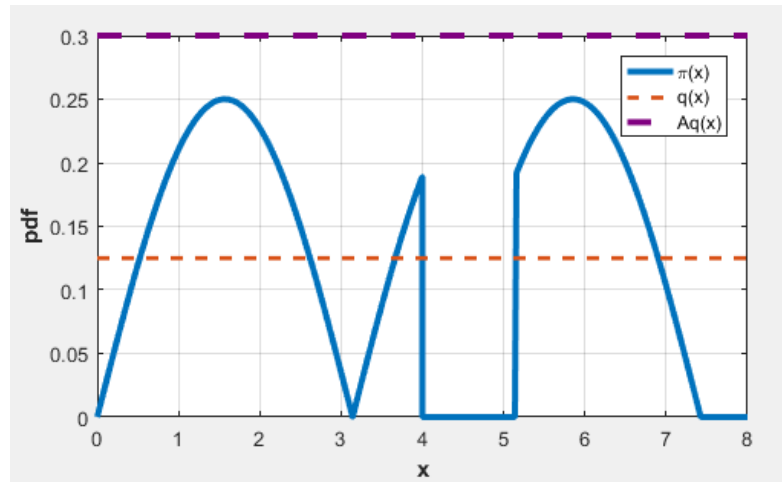

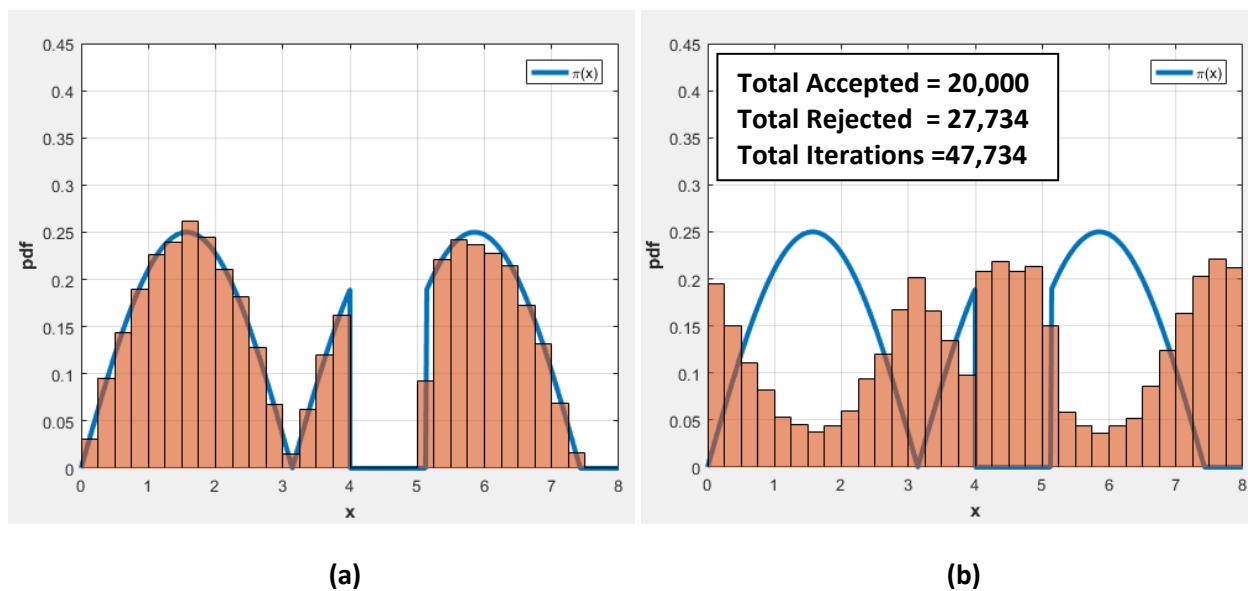**Figure 9: Distributions and blanket function with *A* = 2.4**

| (a) | (b) |

Figure 10: Histograms of accepted values (9a) and rejected values (9b).

Conclusion

This paper explained how to perform the MCMC rejection method to randomly sample from an arbitrary distribution. One useful application of this method is performing simulations where data do not fit a distribution from which software can readily provide random values. A series of examples demonstrated the validity of the method and the pitfalls of inefficient or invalid blanket functions. Sample code from MATLAB is available in Appendix A.

References

Gamerman, Dani, and Lopes, Hedibert. *Markov Chain Monte Carlo – Stochastic Simulation for Bayesian Inference.* 2nd ed., Chapman & Hall/CRC, 2006.

Appendix A: MATLAB code

```
%{
Rejection Method Demo

Basic demonstration of MCMC rejection method. Written to reproduce
figures in the STAT Best Practice, "Sampling from Arbitrary Distributions
using the Rejection Method."

Primary Reference:
Gamerman and Lopes, "Markov Chain Monte Carlo - Stochastic Simulation for
Bayesian Inference," 2nd Edition, Chapman & Hall/CRC, 2006.

Created at the Scientific Test and Analysis Techniques (STAT) Center of
Excellence (COE). Contact info at www.AFIT.edu/STAT
DISCLAIMER: This code has been checked using known examples and functions
correctly to the best of our knowledge.

Nathaniel Choo and Dr. Kyle Kolsti
Version 3.0 10/30/2018
%}

% Clear and initialize
close all;
clear;
clc;

%% DEFINE THE DISTRIBUTIONS AND PARAMETERS
% Comment out unwanted lines to choose distributions

% Define the target distribution, pi(x), here called "p"
mu = 4.5;    % Mean for normal distribution
sigma = 1;   % Standard deviation for normal distribution
p = @(x) (1/(sigma.*sqrt(2*pi))).*exp(-1.*(x-mu).^2/(2*sigma.^2)); % Normal
% p = @(x) (abs(sin(x)) .* heaviside(x) - abs(sin(x)) .* heaviside(x - 4)+
abs(sin(x+2)) .* heaviside(x+2 - (4 + pi)) - abs(sin(x+2)) .* heaviside(x+2 -
3 * pi)) / (4);

% Define the auxiliary distribution, q(x), and function to randomly sample
% from it.
alpha = 4;   % Scale parameter for gamma distribution
beta = 1;    % Shape parameter for gamma distribution
q = @(x) (x.^(alpha-1).*exp(-1.*(x./beta)))./(beta.^alpha.*gamma(alpha));
randomx = @(x) gamrnd(alpha, beta); % Random draw from gamma distribution

% a = 0;  % left limit of uniform distribution
% b = 8;  % right limit of uniform distribution
% q = @(x) 1/(b-a) + x .* 0; % Uniform distribution from a to b
% randomx = @(x) a + (b-a)*rand(); % Random draw from uniform dist, a to b

% Choose value for A
A = 3;
```

```matlab
% Choose number of values desired (this many values will be accepted)
N = 20000;

% Define x axis
x = 0 : 0.01 : 13;
% x = 0 : 0.001/(2*pi) : 8;

%% Rejection Method

accept = zeros(N,1);
reject = zeros(1000*N,1);
i = 0;
j = 0;
tic
while i < N
    xi = randomx();
    u = rand();
    if u*A*q(xi) <= p(xi)
        i = i + 1;
        accept(i) = xi;
    else
        j = j + 1;
        reject(j) = xi;
    end
end
reject = reject(1:j);  % Chop off unused cells from array
toc  % This times how long it takes to accept "N" samples

%% PLOT CHOSEN DISTRIBUTIONS

% Plot target function, p(x)
figure
plot(x, p(x), 'linewidth', 3)

% Plot auxiliary distribution, q(x)
hold on
plot(x, q(x), 'linewidth', 2, 'linestyle', '--')

% Plot blanket function, Aq(x)
hold on
plot(x, A .* q(x), 'linewidth', 3, 'linestyle', '--', 'color', [0.5,0,0.5])

% Plot admin
legend('\pi(x)', 'q(x)', 'Aq(x)')
xlabel('x', 'Fontweight', 'bold')
ylabel('pdf', 'fontweight', 'bold')
grid on


%% PLOT RESULTS OF SAMPLING

% Plot original target function, p(x)
figure
```

```
plot(x, p(x), 'linewidth', 3)

% Plot histogram of sampled values. Should reproduce p(x).
% If histogram of rejects is esired, replace 'accept' with 'reject'
hold on
Nbins = 40;
histogram(accept, Nbins, 'normalization', 'pdf')

% Plot admin
legend('\pi(x)')
xlabel('x', 'Fontweight', 'bold')
ylabel('pdf', 'fontweight', 'bold')
grid on

%% Compare coefficient to proportion of proposed values that were accepted
predictedratio = 1/A
ratio = N/(N+j)
```