Project 2 Monte Carlo Methods

EE 511 – Section Thursday 9 am

Name: Junquan Yu

Student ID #: 3372029142

1. Problem Statement

Some fairly straightforward Monte Carlo evaluation projects.

1. Estimate π by the area method including confidence intervals on your estimate. Draw a graph of the successive values of the estimator as the number of samples increases.

How many points do you need to use for your estimate to be within $\pm 1\%$ of the true value of π (with probability 0.95)?

2. Consider a deck of cards (for simplicity numbered 1… N). Use a uniform random number generator to pick a card and record what card it is (if you were using actual cards, you would replace the card back into the deck – that is not necessary here since we never really take the card out of the deck). Repeat this *N* times, recording the number of times that each of the cards is selected. Some cards may not show up (actually, it is very likely that several card numbers will not show up) and some will show up more than once. You can use this data to estimate the following probabilities:

p_i =Pr{a card will be selected times in the selections}

It is unlikely that any card will show up more than about 10 times. Run this for N =10, N=52, N=100, N=1000, N=10,000 and verify that $p_0 \simeq 1/e$. Can you also find values for the other p_j based on a mathematical analysis?

3. Use the method discussed in class to find $\hat{y}$, an estimate for Y and find a 95% confidence interval for the value of the integral.

$$Y = \int_0^{\pi} \frac{\sin(x)}{x} dx$$

2. Theoretical Exploration or Analysis

For the problem 1:

Consider a quadrant with center at the origin and radius r=1 and a square enclosing this quadrant with length l=1.

Generate n pairs of random variables (X_i, Y_i) which are both obey the uniform distribution U(0,1). Each pair of random variables (X_i, Y_i) can represent one point falling into the area of square mentioned above. Part of these points may fall into the quadrant with the condition that $X_i^2 + Y_i^2 \leqslant 1$. We know that the probability for a point, (X_i, Y_i) ($0<X_i<1$, $0<Y_i<1$), to fall into that quadrant is the ratio of areas of two geometric figures, which is $\frac{\pi}{4}$ in theory.

Let $P_i=\begin{cases}0 & \text{if } (X_i, Y_i) \text{ is in the quadrant} \\ 1 & \text{otherwise}\end{cases}$

Then the sequence P_i are a sequence of samples of a Bernoulli distributed DRV, P, with

$\Pr\{P=1\}=p=\frac{\pi}{4}$

$E[P]=p=\frac{\pi}{4}$

$VAR[P]= \sigma_p^2=p(1-p)=\frac{\pi}{4}(1-\frac{\pi}{4})$

We are generating an estimator for the parameter p and hence $\frac{\pi}{4}$ by:

$$\hat{p} = \frac{\sum_{i=1}^{n} P_i}{n}$$

Then we can use this specific value of estimator $\hat{p}$ to estimate π for this trial, so the π', which is the estimator of π, is:

$$\pi' = \frac{4\sum_{i=1}^{n} P_i}{n}$$

Theoretically speaking, $\hat{p}$ is itself a Gaussian distributed random variable with

mean p and variance $\sigma^2_{\hat{p}}$, which is asymptotically valid for large n.

$$E[\hat{p}] = E[\frac{1}{n}\sum_{i=1}^{n} P_i] = p$$

$$\sigma^2_{\hat{p}} = VAR[\frac{\sum_{i=1}^{n} P_i}{n}] = \frac{1}{n^2}(\sum_{i=1}^{n}\sigma^2_{P_i}) = \frac{1}{n}\sigma^2_{P_i} = \frac{p(1-p)}{n}$$

Therefore, by referring to the standard table of values for the Normal Distribution, the confidence interval where we can expect $\hat{p}$ to fall with 95% probability is:

$$Pr\{p - 1.96\sigma_{\hat{p}} \le \hat{p} \le p + 1.96\sigma_{\hat{p}}\} = 95\%$$

However, for we don't have any knowledge about π and P_i are a series of independent and identical distributed Bernoulli trial, we need to use the sample variance $S^2_{\hat{p}} = \frac{\hat{p}_1(1-\hat{p}_1)}{n}$ as the estimate for the variance of $\hat{p}$.

Now the 95% confidence interval which uses sample statistics can be found:

$$Pr\{\frac{\pi}{4} - 1.96S_{\hat{p}} \le \hat{p} \le \frac{\pi}{4} + 1.96S_{\hat{p}}\} = 95\%$$

So the 95% confidence interval for the estimator of π, which is π', is:

$$Pr\{\pi - 4\times 1.96S_{\hat{p}} \le \pi' \le \pi + 4\times 1.96S_{\hat{p}}\} = 95\%$$

Repeat the above process for successive values of estimator π' as the number of samples increases and draw the graph.

For the problem 2:

Let's assume $X_1, X_2, ..., X_N$ be a series of DRV from uniform distribution, each of which take positive integer value from 1 to N. The probability for a card being selected 0 times in N selections with replacement can be regarded as the probability for the event that all of these DRV are not equal to 1. In theory, we can know that the probability is:

$$P_r\{No1\} = P_r\{X_1 \ne 1, X_2 \ne 1, ..., X_N \ne 1\} = (1 - \frac{1}{N})^N$$

And

$$\lim_{N\to+\infty}(1-\frac{1}{N})^N=\frac{1}{e}\simeq 0.368$$

In the simulation, we just generate the series of $\{X_i : i=1,...,N\}$ and repeat this process until N series of $\{X_i : i=1,...,N\}$ are produced. Then we record the number of series where number 1 was not seen, which is M. Therefore, the probability for a card being selected 0 times in N selections with replacement is $p_0=\frac{M}{N}$. By comparing p_0 with $\frac{1}{e}$, we can verify whether $p_0\simeq\frac{1}{e}$.

For the problem 3:

We need to estimate $Y=\int_0^{\pi}\frac{\sin(x)}{x}dx$

Let $y=\frac{\pi}{x}$, y is RV which obeys uniform distribution U(0,1), and we can get:

$$Y=\int_0^{\pi}\frac{\sin(x)}{x}dx=\int_0^{1}\frac{\sin(\pi y)}{y}dy=\int_0^{1}g(y)dy\quad(g(y)=\frac{\sin(\pi y)}{y})$$

According to the strong law of large numbers, we can get:

$$\hat{y}=\frac{\sum_{i=1}^{n}g(y_i)}{n}\to E(g(y_i))=Y\qquad n\to+\infty$$

Therefore, we can produce a sequence of sample of RV $y_i(i=1,...,n)$ which obey the uniform distribution U(0,1), we can compute the sample mean of these $g(y_i)$,which is $\hat{y}$, and use it as the estimated value of Y when the size of sample is large enough.

Normally we do not know the details of the distribution of g(y), but for large enough sample size, the underlying distribution doesn't matter for the CLT to be asymptotically true.

We can also compute the sample variance of $g(y_i)$, which is $S_{g(y_i)}{}^2$:

$$S_{g(y_i)}^{\ 2} = \frac{\sum_{i=1}^{n}[g(y_i) - \hat{y}]^2}{n-1}$$

According to CLT, we can get:

$$E[\hat{y}] = E[\frac{\sum_{i=1}^{n} g(y_i)}{n}] = E(g(y_i)) = Y$$

$$S_{\hat{y}}^{\ 2} = \frac{S_{g(y_i)}^{\ 2}}{n}$$

So the 95% confidence interval for the estimator of Y, which is $\hat{y}$, is:

$$\Pr\{Y - 1.96S_{\hat{y}} \le \hat{y} \le Y + 1.96S_{\hat{y}}\} = 95\%$$

3. Simulation Methodology

For the problem 1:

I used the "for" loop structure to generate 20000 points which fall into the square enclosing the quadrant with center at the origin and radius r=1. Every time through the "for" expression, a pair of U(0,1) random numbers was generated which corresponds one point in the square. Then I used "if (x_1(n)^2+y_1(n)^2<=1)" conditional statement to judge whether this point fall into the quadrant with center at the origin and radius r=1. If the result of judgment is true, the accumulator p plus 1. After 20000 times of loops, the value of p is actually the number of points falling into the quadrant mentioned above. We can easily use the ratio of the value of p and 20000 as the estimator of π. After that, I used this estimator of π to calculate the variance of this estimator and thus used this variance to compute the upper bound, which is max, and lower bound, which is min, of 95% confidence interval of the estimator of π for this trial.

In order to draw a graph of the successive values of the estimator as the number of samples increase, I used loop nesting structure to meet the requirements provided. I can repeat this experiment for 39000 times with the total number of points from 1001 all the way to 40000 by the external loop. In the internal loop, the accumulator p saves the

number of points falling into the quadrant for each trial. Then I assign the estimated value of π for each trial into the array k[ ]. When the whole loop finished, I can get 39000 estimated values of π, each of which corresponds to one trial of this experiment. Finally, I drew the graph with the number of random points as x-axis and its corresponding estimated value of π as y-axis with the function plot( ). I also added two red reference lines indicating $\pm 1\%$ of true value of π to help me figure out how many points i needed to use for my estimate.

For the problem 2:

I used the function input( ) to enable user to assign any specific number to the variable N(N=10, N =52, N=100, N=1000,N=10000). Then I created the "for" loop structure in order to repeat the following experiment for N times. Every time through the "for" expression, N discrete random variables from uniform distribution are generated and assigned to the array a[ ], each of which can take one of positive integer values from interval [1,N]. Then I used the "while" loop structure to loop through all the elements of the array a[ ] in sequence and used the conditional statement "if (a(1,j)==1)" to judge if there is number 1 in this array. Once the result of judgment is true, just jump out of the "while" loop, otherwise the variable j (initial value is 1) plus 1. When the "while" loop finished, the conditional statement "if (j==N+1)" is applied to check the value of j. If the result of judgment is true, this means that this program looped all the way to the last elements of the array a[ ], which means none of elements in the array a[ ] is number 1 in this trial, and then the accumulator p (initial value is 1) plus 1. If the result of judgment is false, this means this program looped to a certain element of the array a[ ] and jump out of the "while" loop, which means number 1 exists in the array a[ ] in this trial. When the whole "for" loop finished, the value of accumulator p is actually number of times that number 1 was not seen. Because this experiment is repeated for N times in total, I can use the ratio of the value of p and N as the estimated value of p_0, which is the probability for the event that a card will be selected 0 times in N selections.

By comparing the value of $\frac{p}{N}$ and the value of $\frac{1}{e}$, we can verity that $p_0 \simeq \frac{1}{e}$.

For the problem 3:

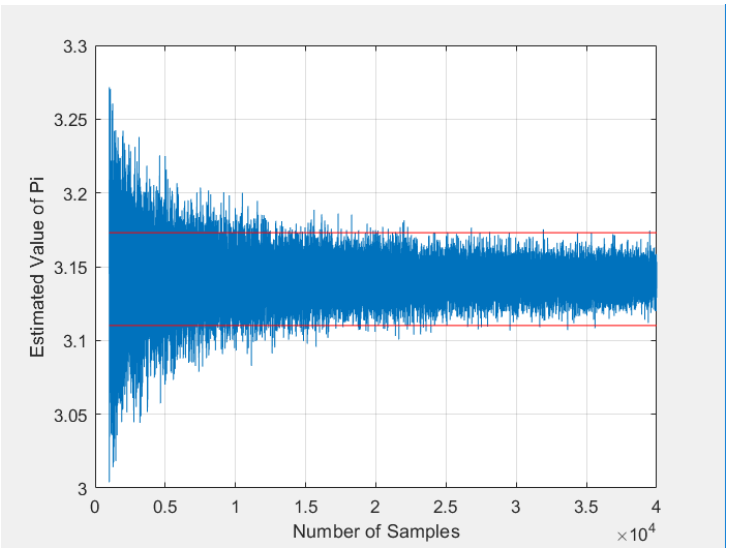
I used the function input( ) to enable user to assign any specific number to the variable N and generated N samples of y with function rand( ), which obeys the uniform distribution U(0,1). Then I used the first "for" loop structure to compute the N values of function g(y) which corresponds to each of samples of y. Afterwards, I used the second "for" loop structure to compute the sum of these samples of g(y). When the second loop finished, the accumulator p saves the sum of these samples of g(y), so I can use the ratio of the value of p and N as the estimated value of the integral as long as N is large enough according to the mathematical analysis in Theoretical Explanation or Analysis Section. After that, I used the function int( ) to calculate the true value of Y. Finally, I computed the upper bound, which is max, and the lower bound, which is min, of 95% confidence interval of the estimator of Y in the same way as for the estimator of π in the problem 1.

4. Experiments and Results

For the problem 1:

I chose loop variable i=20000 (I think this is large enough) as the number of random points generated in the square with the length l=1. After running the program, it would return the estimated value of π, which is π'=3.1462, and the upper bound and lower bound of its corresponding 95% confidence interval, which are max=3.1643 and min=3.1189. Therefore, the 95% confidence interval of the estimated value of π for this trial is $\Pr\{3.1189 \le \pi' \le 3.1643\} = 95\%$.

Then I let the value of loop variable i take on from 1001 to 4000, which means repeating this experiment for 39000 times with the total number of random points from 1001 all the way to 40000. After running the program, the graph displays like this:



It can be inferred from this graph that we need about 1.4W points for the estimates to be within the $\pm 1\%$ of the value of π.

In theory, we can calculate the number of points required from the following equation:

$$\pi+4\times1.96\times\sqrt{\frac{\frac{\pi}{4}\times(1-\frac{\pi}{4})}{n}}=1.01\pi$$

$\therefore n=10497$

Therefore, we can say that the result from stimulation matches well with the theoretical value from the mathematical analysis.

For the problem 2:

The problem requires me to the experiment for N=10, N=52, N=100, N=1000, N=10000, so I run the program for one time for each of these values. The results are as follows:

N	10	52	100	1000	10000
p_0	0.3	0.3654	0.3600	0.3650	0.3657

It can be seen from the above sheet that values of p_0 for all the five trials is very close to the $\frac{1}{e}$ and the larger the sample size N is, the closer the value of p_0 is to the $\frac{1}{e}$, which means $p_0 = \frac{1}{e}$ is verified.

From the mathematical analysis, we can also get the value for the other p_j:

$$p_j = C_N^j (\frac{1}{N})^j (1-\frac{1}{N})^{N-j}$$

For the problem 3:

After running the program, I input N=20000 (I think this is large enough) as the number of random numbers generated from the uniform distribution U(0,1). Then it would return the estimated value of Y, which is $\hat{y}$=1.8595, and the upper bound and lower bound of its corresponding 95% confidence interval, which are max=1.8660 and min=1.8379. Therefore, the 95% confidence interval of the estimated value of Y for this trial is $\Pr\{1.8379 \le \hat{y} \le 1.8660\} = 95\%$. After that, I also tried N=5000 and N=40000 in order to find some trend or regularity. For N=5000, the estimated value of Y is $\hat{y} = 1.8647$ and its 95% confidence interval is $\Pr\{1.8238 \le \hat{y} \le 1.8800\} = 95\%$ while for N=40000, the estimated value of Y is 1.8518 and its 95% confidence interval is $\Pr\{1.8420 \le \hat{y} \le 1.8619\} = 95\%$. Therefore, the larger the sample size N is, the closer the value of $\hat{y}$ is to the true value of Y, which is 1.8519, and the narrower the 95% confidence interval of $\hat{y}$ is.

5. References

1. *Alberto Leon-Garcia. (2008). Probability, Statistics, and Random Processes for Electrical Engineering. Upper Saddle River, NJ 07458. Pearson Education, Inc.*

2. *Zhou Sheng, Shiqian Xie, Chengyi Pan. (2008). Probability Theory and Mathematical Statistics. No.4, Dewai Street, Xicheng District, Beijing. Higher Education Press.*

6. Source Code

The program for the problem 1:

```
i=20000;
p=0;
for n=1:20000                %generate 20000 random points in the square with length l=1
   x1(n)=rand;
```

```
    y1(n)=rand;
    if (x1(n)^2+y1(n)^2<=1)    %judge whether each of these points fall into the quadrant
                                with center at the origin and radius r=1
        p=p+1;                 %count the number of points falling into this quadrant
    end
end
l=4*p/n                        %estimate the value of pi
q=p/n;
s=q*(1-q)/n;
min=pi-4*1.96*s^(0.5)          %compute the 95% confidence interval for this trial
max=pi+4*1.96*s^(0.5)

m=1;
for i=1001:40000               %repeat this experiment for 39000 times with the total number
                               of points from 1001 all the way to 40000
    p=0;
   a(1,m)=i;
for n=1:i                      %generate i(i from 1001 to 40000) random points in the square
                                with length l=1
    x2(n)=rand;
    y2(n)=rand;
    if (x2(n)^2+y2(n)^2<=1)   %judge whether each of these points fall into the quadrant with
                               center at the origin and radius r=1
        p=p+1;                 %count the number of points falling into this quadrant
    end
end
    k(1,m)=4*p/n;              %estimate the value of pi for each trial
    m=m+1;
end                            %draw the graph with the number of random points as x-axis and
                               its corresponding estimated value of pi as y-axis
plot(a,k);
hold on;
grid on;
pi1=1.01*pi*ones(1,39000);
pi2=0.99*pi*ones(1,39000);
plot(a,pi1,'r');
plot(a,pi2,'r');
ylabel('Estimated Value of Pi')
xlabel('Number of Samples')
```

the program for the problem 2:

```
N=input('please enter the number of repetitions: '); %enter the number of trials, which is N
p=0;
for i=1:N                  %repeat following process for N times
```

```
    j=1;
    a=randi(N,[1,N]);      %generate N discrete random variables from uniform distribution,
                            each of which can take one of positive integer values from
                            interval [1,N]
    while (j<=N)           %judge whether number 1 can be seen in each trial
        if (a(1,j)==1)
            break;
        end
        j=j+1;
    end
    if (j==N+1)
        p=p+1;
    else
        p=p+0;
    end
end
k=p/N                   %compute the probability for a card being selected 0 times in N
                        selections
```

the program for the problem 3:

```
N=input('please enter the number of sample of y: '); %enter the number of sample of y, which
is N
a=rand([1,N]);         %generate N samples of y
p=0;
q=0;
for i=1:N              %generate N samples of g(y)
    g(1,i)=sin(pi*a(1,i))/a(1,i);
end
for i=1:N             %compute the sum of these samples of g(y)
   p=p+g(1,i);
end
k=p/N;                %compute the sample mean of these g(y), which is approximately true
                      value of the integral when N is large enough
for i=1:N
    q=q+(g(1,i)-k)^2;
end
s1=q/(N-1);
s2=s1/N;
syms x
r=int(sin(x)/x,x,0,pi);   %compute the true value of the integral
min=vpa(r)-1.96*s2^(0.5)  %compute the 95% confidence interval for this trial
max=vpa(r)+1.96*s2^(0.5)
```