University of Southern California EE511

Integrals and Intervals

Project #4

Name: Wu Jiawei
USCID: 9600392575

Abstract

In the project of integrals and intervals, three experiments are conducted using Matlab. The core method of the project is running simulation to approximate the integrals and intervals and comparing them with the theoretical values. The theories of Monte Carlo simulation, empirical distribution function, and the confidence interval are applied comprehensively in the lab. The experiments are repeated multiple times and the outcomes are shown in diagrams and calculations.

Introduction

Three experiments are conducted in the lab. All the samples are generated by random selection and the experiments are repeated multiple times. The goal of the first trial is to approximate the integrals using Monte Carlo Simulations and the results are compared with the exact values. The aim of the second experiment is to find the cumulative distribution function of giving samples by generating empirical distribution function. The objective of the third experiment is running simulation and sampling from Z-distribution and X-distribution to obtain the statistical confidence interval and verify the results by comparing them with the bootstrap confidence interval.

Methodology & Results

Experiment No.1

Description of Algorithm:

The Monte Carlo simulation is utilized in this experiment.

1).In the first part, to compute the value of $\int_a^b g(x)dx$.

Let $y = \frac{x-a}{b-a}$, Then

$$dy = \frac{dx}{b-a}$$

Therefore, the integral is equal to

$$\int_0^1 g(a+(b-a)y) * (b-a)dy$$

2).In the second part, to compute the value of $\int_0^\infty g(x)dx$

Let $y = \frac{1}{x+1}$, then

$$dy = -\frac{dx}{(x+1)^2} = -y^2dx$$

Therefore, the integral is equal to

$$\int_0^1 \frac{g(\frac{1}{y}-1)}{y^2}dy$$

3).In the third part, to compute the value of $\int_0^1 \cdots \int_0^1 g(x_1, \cdots x_n)dx_1 \cdots dx_n$

First generate k sets of $u \sim U[0,1]$, then evaluate each set

$g(u_1, \cdots, u_n)$

Therefore, the integral is equal to

$$\frac{\sum_{i=1}^{k} g(u_1^i, \cdots, u_n^i)}{k}$$

Description of Method:

The experimental values of integrals are approximated through Mento Carlo simulation. Firstly, multiple uniformly random variables are generated and stored in the array x and y, then the integral intervals are transformed according to the Mento Carlo theory, and the independent random samples are utilized to compute the outcome. The trials are repeated multiple times and the mean of values are calculated to finish the simulation. The theoretical values are generated using integral and integral2 function. The comparison between experimental and theoretical values are conducted.

Code:

```
function f=integ(N)
x=rand(N,1);
y=rand(N,1);
Exp_a=sum(4.*exp(4.*y-2+(4.*y-2).^2)./N);
Exp_b=sum(exp(-(1-1./y).^2)./(y.^2)./N)+sum(exp(-(1./y-1).^2)./y.^2./N);
Exp_c=sum(exp(-(x.^2+y.^2)))./N;
%Calculate the experimental values of integrals
disp([Exp_a Exp_b Exp_c]);

syms x y;
Theo_a=integral(@(x)exp(x+x.^2),-2,2);
Theo_b=integral(@(x)exp(-x.^2),-inf,inf);
Theo_c=integral2(@(x,y) exp(-((x+y).^2)),0,1,0,1);
%Calculate the theoretical values of integrals
disp([Theo_a Theo_b Theo_c]);
```

Simulation Result:

```
Command Window
>> integ(10000)
   92.1824    1.7974    0.5570

   93.1628    1.7725    0.4118
```

Finding:

The trials are repeated 10000 times to approximate the values of integrals.

For the problem a, the experimental value is 92.1824, the exact value is 93.1628.

For the problem b, the experimental value is 1.7974, the exact value is 1.7725.

For the problem c, the experimental value is 0.5570, the exact value is 0.4118.

Experiment No.2

Description of Algorithm:

The empirical distribution function for samples $\{x_1, x_2, \cdots, x_n\}$ is defined by

$$F_n^*(x) = \frac{1}{n}\sum_{i=1}^{n} I_{(-\infty,x]}(x_i)$$

And, I_A is the indicator function.

According to the Glivenko-Cantelli theorem,

$$\|F_n^* - F\|_\infty = \sup_{x\epsilon R} |F_n^*(x) - F(x)| \to 0$$

Description of Method:

Firstly, the uniformly random variables are generated

$U \sim \mathrm{Uniform}([0,1])$

The inverse transform method is utilized to create samples

$X = F^{-1}(U)$

Second of all, the samples are sorted in an ascending order to plot the step function with “stairs()”, and overlaid with the theoretical cdf of $x \sim X^2(4)$.

Finally, the maximum difference at each sample is computed

$$\max_{x_i} |F_N^* - F(x_i)|$$

And the values of 25th, 50th and 90th percentiles are found from empirical distribution and compared with the theoretical percentile values for $X^2(4)$.

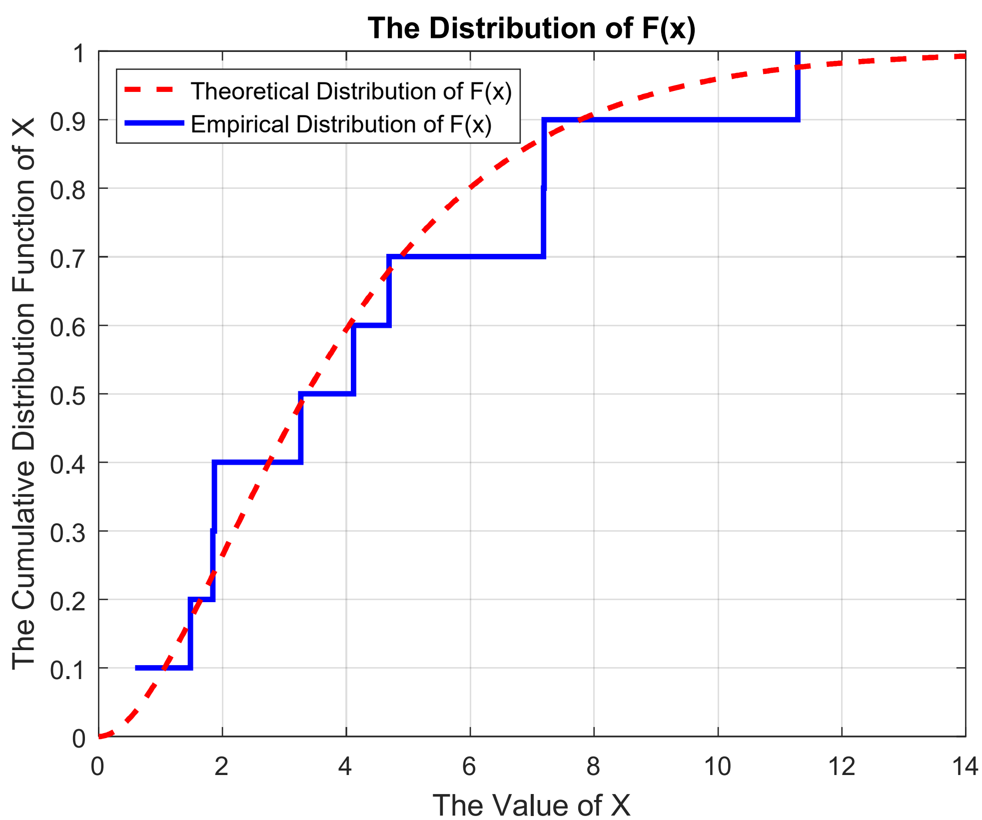
Code:

```
function f=emp(N)
arr_diff=zeros(N,1);
%Record the differences between empirical and theoretical distribution
arr_X=zeros(N,1);
arr_X_sorted=zeros(N,1);
%Record the value of X and sorted X
for i=1:N
Z1=randn;
Z2=randn;
Z3=randn;
Z4=randn;
arr_X(i)=Z1.^2+Z2.^2+Z3.^2+Z4.^2;
end
arr_X_sorted = sort(arr_X);

p=[0.25,0.5,0.9];
theo_value=chi2inv(p,4)
expe_value=arr_X_sorted((ceil(N*p)))'
%Find the 25th,50th and 90th percentiles using empirical distribution
for j=1:N
arr_diff(j)=chi2cdf(arr_X_sorted(j),4)-(1/N)*j;
end
Differ=max((abs(arr_diff)))
%Compute the maximum difference at each of samples
T2=stairs(arr_X_sorted,1/N:1/N:1,'b','linewidth',2);
hold on
t = 0:10^-4:14;
T1=plot(t,chi2cdf(t,4),'r--','linewidth',2);
hold off
grid on
hold off
title('The Distribution of F(x)');
xlabel('The Value of X');
ylabel('The Cumulative Distribution Function of X');
legend([T1,T2],'Theoretical Distribution of F(x)','Empirical Distribution of
F(x)');
```

Simulation Result:

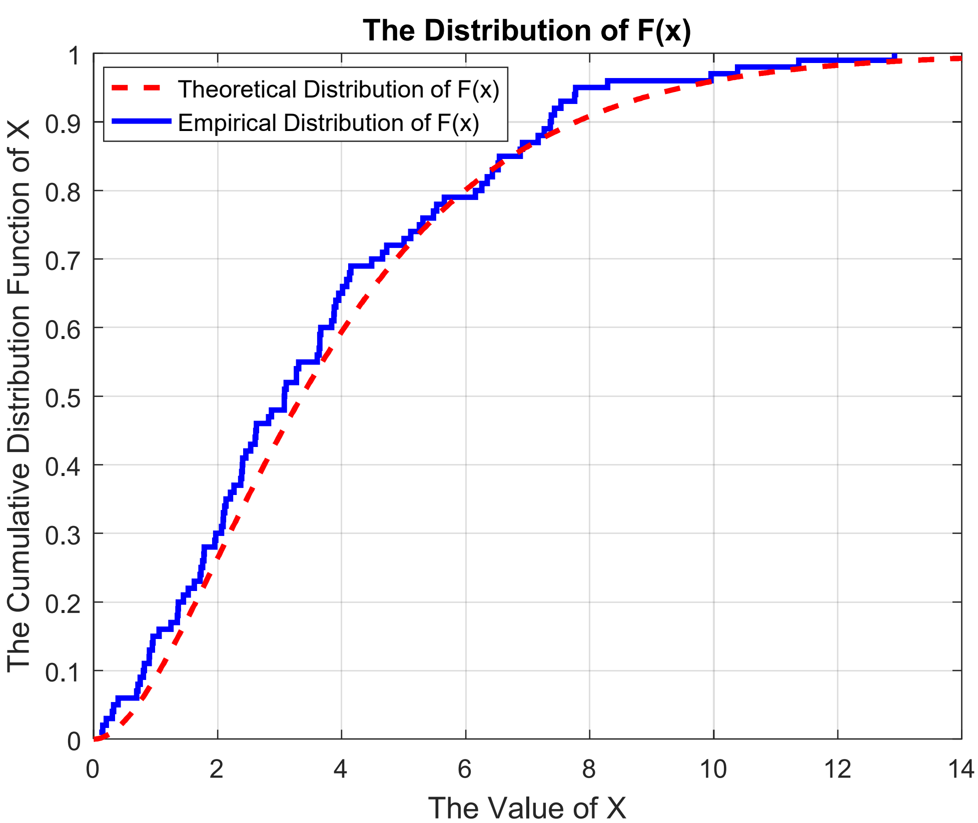
1).N=10

```
Command Window
>> emp(10)

theo_value =

    1.9226    3.3567    7.7794


expe_value =

    1.8468    3.2656    7.1942


Differ =

    0.1597
```



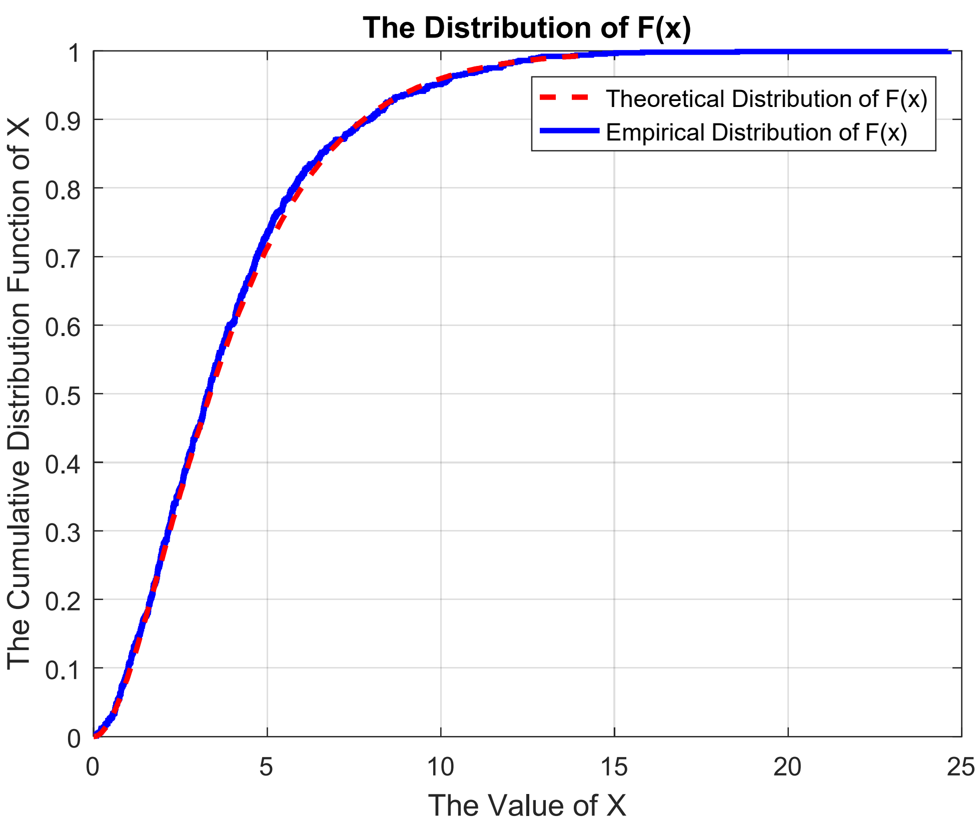
2). N=100

```
Command Window
>> emp(100)

theo_value =

    1.9226    3.3567    7.7794


expe_value =

    1.7337    3.0737    7.3639


Differ =

    0.0820
```



3). N=1000

```
Command Window
>> emp(1000)

theo_value =

    1.9226    3.3567    7.7794


expe_value =

    1.8954    3.2807    7.8993


Differ =

    0.0251
```



Finding:

1). When the sample size is 10,

The theoretical value of 25th, 50th and 90th percentiles are 1.9226, 3.3567 and 7.7794 respectively;

The empirical value of 25th, 50th and 90th percentiles are 1.8468, 3.2656, and 7.1942 respectively;

The lower bound for $\|F_n^* - F\|_\infty$ is 0.1597.

2). When the sample size is 100,

The theoretical value of 25th, 50th and 90th percentiles are 1.9226, 3.3567 and 7.7794 respectively;

The empirical value of 25th, 50th and 90th percentiles are 1.7337, 3.0737, and 7.3639 respevtively;

The lower bound for $\|F_n^* - F\|_\infty$ is 0.0820.

3). When the sample size is 1000,

The theoretical value of 25th, 50th and 90th percentiles are 1.9226, 3.3567 and 7.7794 respectively;

The empirical value of 25th, 50th and 90th percentiles are 1.8954, 3.2807, and 7.8993 respevtively;

The lower bound for $\|F_n^* - F\|_\infty$ is 0.0251.

It is shown from the results that the estimation gets better and better as the number of samples increasing.

Experiment No.3

Description of Algorithm:

This experiment is required to calculate the statistical confidence interval using n samples from total data samples.

When n>=30, the statistical confidence interval is computed using the Z-distribution,

$$(\overline{X} - \frac{s}{\sqrt{n}} z\alpha/_2 \,,\ \overline{X} + \frac{s}{\sqrt{n}} z\alpha/_2)$$

When n<30, the statistical confidence interval is computed using the T-distribution,

$$(\overline{X} - \frac{s}{\sqrt{n}} t\alpha/_2 \,,\ \overline{X} + \frac{s}{\sqrt{n}} t\alpha/_2)$$

Description of Method:

In the third experiment, the statistic confidence interval is generated by the following steps,

1. Select the data of “Waiting Time” from the “faithful.dat.txt”.
2. Calculate the mean and standard deviation of the first 15 data samples.
3. Construct a confidence interval of confidence level 95% under T-distribution.
4. Calculate the mean and standard deviation of the total 272 data samples.
5. Construct a confidence interval of confidence level 95% under Z-distribution.

The 95% bootstrap confidence interval is also generated using first 15 data samples and the 272 data totally. The comparison is conducted between the statistical and bootstrap confidence interval.

Code:

```
function f=confi(N)
Z=norminv(0.975);
```

```
T=tinv(0.975,14);
%Compute the value of T(alpha/2) and Z(alpha/2)
s = importdata('faithful.dat1.txt');
%Import the data table
waiting_T = s(:,3);
%Record the values of waiting time
ss_t=std(waiting_T(1:15));
Mu_t=mean(waiting_T(1:15));
Theo_Leftconfit_T=Mu_t-(ss_t/(sqrt(15)))*T
Theo_Righconfit_T=Mu_t+(ss_t/(sqrt(15)))*T
Theo_Width_T=abs(Theo_Leftconfit_T-Theo_Righconfit_T)
%Compute a 95% statistical confidence interval using the first 15 data
arr_Mu=zeros(1,N);
Rand_Sel=[];
for i=1:N
Rand_Ind=randi([1 15],15,1);
Rand_Sel=waiting_T(Rand_Ind,:);
sample_Mu=mean(Rand_Sel);
arr_Mu(i)=sample_Mu;
end
c=sort(arr_Mu);
Exp_Leftconfit_T=prctile(arr_Mu,2.5)
Exp_Righconfit_T=prctile(arr_Mu,97.5)
Expe_Width_T=abs(Exp_Leftconfit_T-Exp_Righconfit_T)
%Compute a 95% bootstrap confidence interval using the first 15 data
ss_z=std(waiting_T(1:272));
Mu_z=mean(waiting_T(1:272));
Theo_Leftconfit_Z=Mu_z-(ss_z/(sqrt(272)))*Z
Theo_Righconfit_Z=Mu_z+(ss_z/(sqrt(272)))*Z
Theo_Width_Z=abs(Theo_Leftconfit_Z-Theo_Righconfit_Z)
%Compute a 95% statistical confidence interval using all of the data
arr_Mu1=zeros(1,N);
Rand_Sel1=[];
for i=1:N
Rand_Ind1=randi([1 272],272,1);
Rand_Sel1=waiting_T(Rand_Ind1,:);
sample_Mu1=mean(Rand_Sel1);
arr_Mu1(i)=sample_Mu1;
end
c1=sort(arr_Mu1);
Exp_Leftconfit_Z=prctile(arr_Mu1,2.5)
Exp_Righconfit_Z=prctile(arr_Mu1,97.5)
Expe_Width_Z=abs(Exp_Leftconfit_Z-Exp_Righconfit_Z)
%Compute a 95% bootstrap confidence interval using all of the data
```

Simulation Result:

Command Window

```
>> confi(10000)

Theo_Leftconfit_T =

   62.5571


Theo_Righconfit_T =

   79.3096


Theo_Width_T =

   16.7525


Exp_Leftconfit_T =

   63.3333


Exp_Righconfit_T =

   78.1333


Expe_Width_T =

   14.8000
```

```
Theo_Leftconfit_Z =

   69.2814


Theo_Righconfit_Z =

   72.5127


Theo_Width_Z =

    3.2313


Exp_Leftconfit_Z =

   69.2482


Exp_Righconfit_Z =

   72.4816


Expe_Width_Z =

    3.2335
```

Finding:

In the experiment, the sample space for the bootstrap simulation is set as 10000.

1). When the first 15 data of waiting time are utilized for T-distribution.

The 95% statistical confidence interval for the waiting times is [62.5571, 79.3096].

The width of statistical confidence interval is 16.7525.

The 95% bootstrap confidence interval for the waiting times is [63.3333, 78.1333].

The width of bootstrap confidence interval is 14.8000.

2). When the total 272 data of waiting time are utilized for Z-distribution.

The 95% statistical confidence interval for the waiting times is [69.2814, 72.5127].

The width of statistical confidence interval is 3.2313.

The 95% bootstrap confidence interval for the waiting times is [69.2482, 72.4816].

The width of bootstrap confidence interval is 3.2335.

It is shown from the simulation results that when the sample space increase from first 15 data to the total 272 data samples, the width of the confidence interval decreases from 16.7525 to 3.2313.

Conclusion

Overall, the simulations of three experiments focus on computing integrals and intervals. The Mento Carlo simulations are completed in the first experiment to approximate the value of different integrals. The second trial simulates the empirical distribution function for the giving samples and obtains the value of different percentiles. The third experiment uses X-distribution and Z-distribution comprehensively to compute the statistical confidence intervals and the results are compared with bootstrap confidence intervals. In conclusion, the project uses mathematical methods to process independent samples, and solve integral and interval problems.