Monte Carlo MAP 5615 HW2

Sen Zhang

2

measures the distance between functions and .

measures the maximum distance between functions and at the left end point of the interval .

measures the maximum distance between functions and at the right end point of the interval .

3

5

I tried RANDU generator and the LCG .

RANDU is bad random number generator since the triplets of random numbers RANDU generated lie on no more than 15 planes in R3. This doesn’t look random at all.

The other LCG generator is bad since it generates integers from 1 to 10000. This is not a sequence of random numbers.

Following is the result for RANDU. See code at appendix [1]

**RANDU**

>> chiKS

c =

14.2000 8.2400 9.5000 12.1400 14.0800 6.0400 8.5200 9.7400 14.8000 15.0000

DN =

0.3898

**Since 0.3898<0.41, we accept that c follows chi square distribution with degree 9 at the 5% significance level.**

Note: 0.41 comes from the table of KS test with N=10 and 5% significant level. See appendix [2]

**LCG**

>> chiKS

c =

Columns 1 through 8

1000 1000 1000 1000 1000 1000 1000 1000

Columns 9 through 10

1000 1000

DN =

1

**Since 1>0.41, we reject that c follows chi square distribution with degree 9 at the 5% significance level.**

By this test RANDU are good random number generators while the LCG is a bad one. This also means the test is weak.

7

In the chi square test, we have 2 assumptions. The random experiment has mutually exclusive and exhaustive outcomes. The nPi > 5 for all i.

Assuming we have the probability of run up of length i Pi, we still cannot make sure nPi > 5 if i is very large. So I don’t think we can use chi square test directly. We may set the probability of run up of length t or more as a outcome. Then we can use chi square test.

9

Design a test:

Given a sequence of random numbers. Consider the interval L=(0, 0.5) and R= (0.5, 1) , the equ-head-tail of length p is by giving an example. Consider the numbers 0.3,0.6,0.7,0.4,0.8,.1,0.9,0.8,0.3,0.2,0.7. Say a random number lie in L is tail and one lie in R is head. Put a vertical line when the total number of heads and tails are equal.

0.3,0.6 | 0.7,0.4 | 0.8,0.1 | 0.9,0.8,0.3,0.2 | 0.7

Count the numbers in each block. We have a equ-head-tail length 2,2,2,4.

Apply chi square test as following:

Pick a value t. Let u(i) be the number of equ-head-tail of length 2i for i = 1, 2, 3. . . t-1; let u(t) be the number of length 2t or more. The outcomes of the test are: “equ-head-tail length of 2”, “equ-head-tail length of 4”…… “equ-head-tail length of 2t-2” and “equ-head-tail length of 2t or more”. We know the probability of each outcome occurring, write as Pi. Pick a value n such that nPi > 5. Given a sequence of random number, go through the sequence to identify blocks as “equ-head-tail length of i”, for some i. Stop when there are a total of n length found. Count the number of equ-head-tail length of i, and t or more. These counts are the Yi value in chi square test.

For example, apply the test on RANDU generator.

Pick t = 4, n = 100. See code in appendix [3]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Equ-head-tail length | 2 | 4 | 6 | 8 or more |
| Pi | 0.5 | 0.125 | 0.0625 | 0.3125 |
| Yi | 50 | 15 | 5 | 30 |
| nPi | 50 | 12.5 | 6.25 | 31.25 |

Q3=0.8, P(x<0.8)=0.1505, so accept RANDU generator at the significant level 5%. We can see that this test is a weak test since RANDU is a bad generator.

# Appendix

[1] code of problem 5

function chiKS

x=RANDU(1,10000);

%x=[1:10000];

M=10;

k=10;

p=1/k;

c = zeros(1,M);

for m = 1:M

for i = 1:k

s=x(1000\*(m-1)+1:1000\*m);

Y(i)=0;

for j = 1:1000

if ((i-1)/k) <= s(j) && s(j) < (i/k)

Y(i) = Y(i) +1;

end

end

c(m)=c(m)+(Y(i)-1000\*p)^2/(1000\*p);

end

end

for i = 1:M

DU(i)=i/M-chi2cdf(c(i),9);

DD(i)=chi2cdf(c(i),9)-(i-1)/M;

end

DU=sort(DU);

DD=sort(DD);

DN=max(DU(M),DD(M));

c

DN

[2] table of KS test

<http://www.eridlc.com/onlinetextbook/appendix/table7.htm>

[3] code of problem 9

function updown

s=RANDU(1,10000);

c=0;

U=0;

D=0;

for j = 1:10000

if s(j) < 0.5

U = U +1;

else

D = D +1;

end

j = j+1;

if U == D

c=[c, j-1];

n = size(c);

if n(2)>100

break

end

end

end

count=zeros(1,4);

C=[];

for i = 1:100

C(i)= c(i+1)-c(i);

end

for i = 1:100

if C(i)==2

count(1)=count(1)+1;

elseif C(i) == 4

count(2)=count(2)+1;

elseif C(i)== 6

count(3) = count(3)+1;

else

count(4)=count(4)+1;

end

end

count

end