Stochastic Process Project

晏浩洋 81913758

2020 年 12 月 1 日

1 D1-2

Using Metropolis-Hasting algorithm with approximately transforming the continuous distribution to discrete.

the steps are as follows:

1.generate a irreducible Markov transition probability matrix Q

in this part I choose Q with all elements $\frac{1}{n}$

2.generate the matrix α with $\alpha_{ij} = \min(\frac{f(j)Q_{ji}}{f(i)Q_{ij}}, 1)$

3.generate the first state x_1 for random

4.choose a state from the transition matrix with probability $Q_{x_1,j}$

5.generate a radom number in range(0,1),

if it is less than $\alpha_{x_1,j}$， state transfer to j, else, it stay in x_1

repeat steps 4-5

For experiment, I choose step size $a = 0.01$ and run the markov chain with length $t = 10000$, statistic the long-run frequency. The result is as follow.

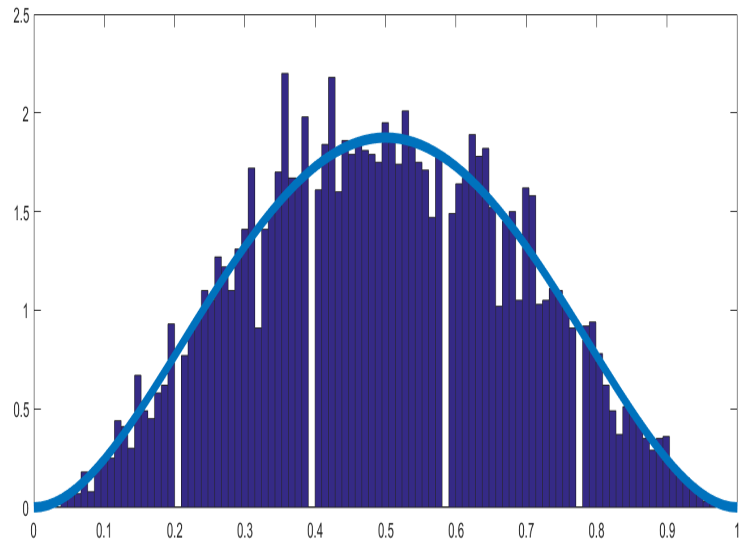

图 1: frequency distribution of the markov chain,a=0.01,t=10000

the matlab code is as follows

```
a = 0.01;
n = 1/a+1;
Q = ones(n)/n;
for i = 1:n
        for j = 1:n
                ii = (i-1)*a;
                jj = (j-1)*a;
                fi = 30*(ii^2-2*ii^3+ii^4)*a;
                fj = 30*(jj^2-2*jj^3+jj^4)*a;
                alpha(i,j) = min(fj*Q(j,i)/(fi*Q(i,j)),1);
        end
end

x1 = randi(n);
xx = [x1];
```

```
t = 10000;
for k = 1:t
        q1 = rand(1);
        q0 = 0;
        for j = 1:n
                q0 = q0+Q(j);
                if q0 > q1
                        break
                end
        end
        alpha1 = rand(1);
        if alpha1 < alpha(x1,j)
                x1 = j;
        else
                x1 = x1;
        end
        xx = [xx,x1];
end
xx = (xx-1)*a;
hist(xx,n)
hold on
x = 0:a:1;
fx = 30*(x.^2-2*x.^3+x.^4)*a;
plot(x,fx*10000,'LineWidth',5)
```

2 D1

To generate the Markov chain, we firstly generate a transfer matrix

for a continuous density function of stationary distribution, there should be a 2-D continuous density function as the transfer distribution $P(ij)$,which obey the constraints that:

$$\int_0^1 f(x_i)P(ij)di = f(x_j) \qquad \forall x_j \in (0,1)$$

$$\int_0^1 P(ij)dj = 1 \qquad \forall i$$

$$P(ij) \geq 0 \qquad \forall i, j$$

However,a continious function is not processible for computer, so we discretization the transfer distribution to a transfer matrix P with step size of a and order of $n = \frac{1}{a} + 1$,$f(x) := f(x) * a$ The transfer matrix obey constraints that:

$$\sum_{i=1}^{n} f(x_i) P_{ij} = f(x_j) \qquad \forall x_j \in (0, 1)$$

$$\sum_{j=1}^{n} P_{ij} = 1 \qquad \forall i$$

$$P_{ij} \geq 0 \qquad \forall i, j$$

It is a problem with n^2 non-negative variable and $2n$ linear constraints, so it is underdetermined.

For intuitive, I write the problem to the matrix formation with $a = 0.5$(this must be inaccurate, actually I choose $a = 0.01$ in practice)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \mathrm{f(0)} & 0 & 0 & \mathrm{f(0.5)} & 0 & 0 & \mathrm{f(1)} & 0 & 0 \\ 0 & \mathrm{f(0)} & 0 & 0 & \mathrm{f(0.5)} & 0 & 0 & \mathrm{f(1)} & 0 \\ 0 & 0 & \mathrm{f(0)} & 0 & 0 & \mathrm{f(0.5)} & 0 & 0 & \mathrm{f(1)} \end{bmatrix} \begin{bmatrix} \mathrm{P}_{0,0} \\ \mathrm{P}_{0,0.5} \\ \mathrm{P}_{0,1} \\ \mathrm{P}_{0.5,0} \\ \mathrm{P}_{0.5,0.5} \\ \mathrm{P}_{0.5,1} \\ \mathrm{P}_{1,0} \\ \mathrm{P}_{1,0.5} \\ \mathrm{P}_{1,1} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \mathrm{f(0)} \\ \mathrm{f(0.5)} \\ \mathrm{f(1)} \end{bmatrix}$$

$$P_{ij} \geq 0 \forall i, j$$

Obviously, this underdetermined problem has more than one solution. AMPL which can solve large-scale problem has no open source, so I can only use MATLAB to solve this problem. However, because of the non-negative constraint, I can only use a small constraint $0.0001 P_{ij} = 0.0001$ to approximately control the sign, then get the transfer matrix: $P = \begin{bmatrix} \mathrm{P}_{0,0} & \mathrm{P}_{0,0.5} & \mathrm{P}_{0,1} \\ \mathrm{P}_{0.5,0} & \mathrm{P}_{0.5,0.5} & \mathrm{P}_{0.5,1} \\ \mathrm{P}_{1,0} & \mathrm{P}_{1,0.5} & \mathrm{P}_{1,1} \end{bmatrix}$

To generate the markov chain, the algorithm is as follows:

1. generate a random state x between 1 and n
2. generate a random number q between 0 and 1
3. if the first k item of the transfer matrix P's xth row is larger than q, state $x := k$, and break the circulation; else, $k := k + 1$ till it obey the condition above.

For experiment, I choose step size $a = 0.01$ and run the markov chain with length $t = 10000$, statistic the long-run frequency. The result is as follow.

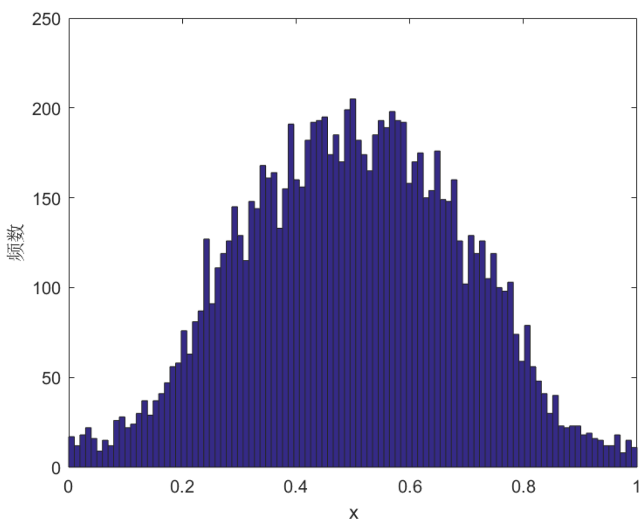

图 2: frequency distribution of the markov chain,a=0.01,t=10000

Comparing with the function graph of $f(x) = 30(x^2 - 2x^3 + x^4)$, we can see that they are very similar

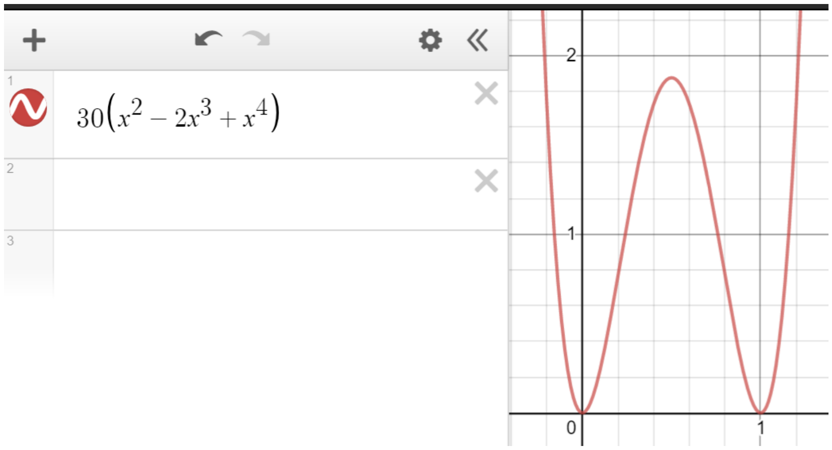

图 3: function graph of $f(x) = 30(x^2 - 2x^3 + x^4)$

The MATLAB source code is as follows:

```
a = 0.01;
n = 1/a+1;
x = 0:a:1;
fx = 30*(x.^2-2*x.^3+x.^4)*a;
A = [];
for i = 1:n
        b = [zeros(1,(i-1)*n),ones(1,n),zeros(1,n^2-n*i)];
        A = [A;b];
end
for i = 1:n
        b = [];
        for j = 1:n
                c = [zeros(1,i-1),fx(j),zeros(1,n-i)];
                b = [b,c];
        end
        A = [A;b];
end
A = [A;0.00001*eye(n^2)];
B = ones(n,1);
```

```
B = [B;fx'];
B = [B;0.00001*ones(n^2,1)];
p = A\B;
P = [];
for i = 1:n
        c=[];
        for j = 1:n
                c = [c,p((i-1)*n+j)];
        end
        P = [P;c];
end

x1 = randi(n);
xx = [x1];
t = 10000;
for j = 1:t
        q = rand(1,t);
        zz = 0;
        for i = 1:n
                zz = zz+P(x1,i);
                if zz >= q(j)
                        x1 = i;
                        xx = [xx,x1];
                        break
                end
        end
end
xx = (xx-1)/(n-1);
hist(xx,n)
```

3 D2

$$f(x) = 30(x^2 - 2x^3 + x^4)$$
$$\frac{f(x)}{dx} = 60x - 180x^2 + 120x^3$$

while $x = 0.5$,
there is $\frac{f(x)}{dx} = 0$,$f(x)_{max} = 1.875 \quad x \in (0, 1)$
let $g(x) = 1$

$$\frac{f(x)}{f(x)_{max} g(x)} = 16(x^2 - 2x^3 + x^4)$$

and thus the rejection procedure is as follows:
Step 1:Generate random numbers U_1 and U_2 ,
where U_1 and U_2 are uniform distribution in range [0,1]
Step 2:If $U_2 \leq 16(U_1^2 - 2U_1^3 + U_1^4)$, stop and set $X = U1$. Otherwise return to step 1

4 S1

The MATLAB code is as follows:

```
xx = [];
for j =1:10000
        for i = 1:9999
                u1 = rand(1);
                u2 = rand(1);
                if u2 <= 16*(u1^2-2*u1^3+u1^4)
                        x = u1;
                        break
                        end
        end
        xx = [xx,x];
end
hist(xx,100)
hold on
```

```
x = 0:0.01:1;
fx = 30*(x.^2-2*x.^3+x.^4)*0.01;
plot(x,fx*10000,'LineWidth',5)
```

the result is as follows:

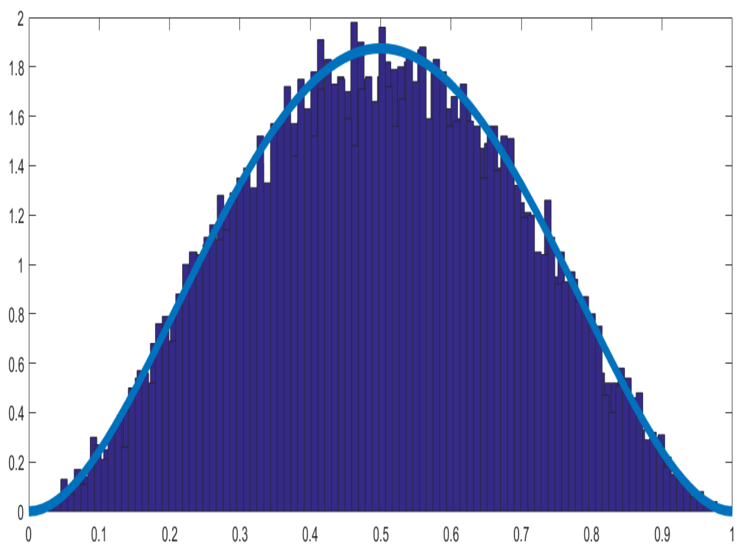

图 4: random sampling of $f(x) = 30(x^2 - 2x^3 + x^4)$