

# Homework 2 STAT5376

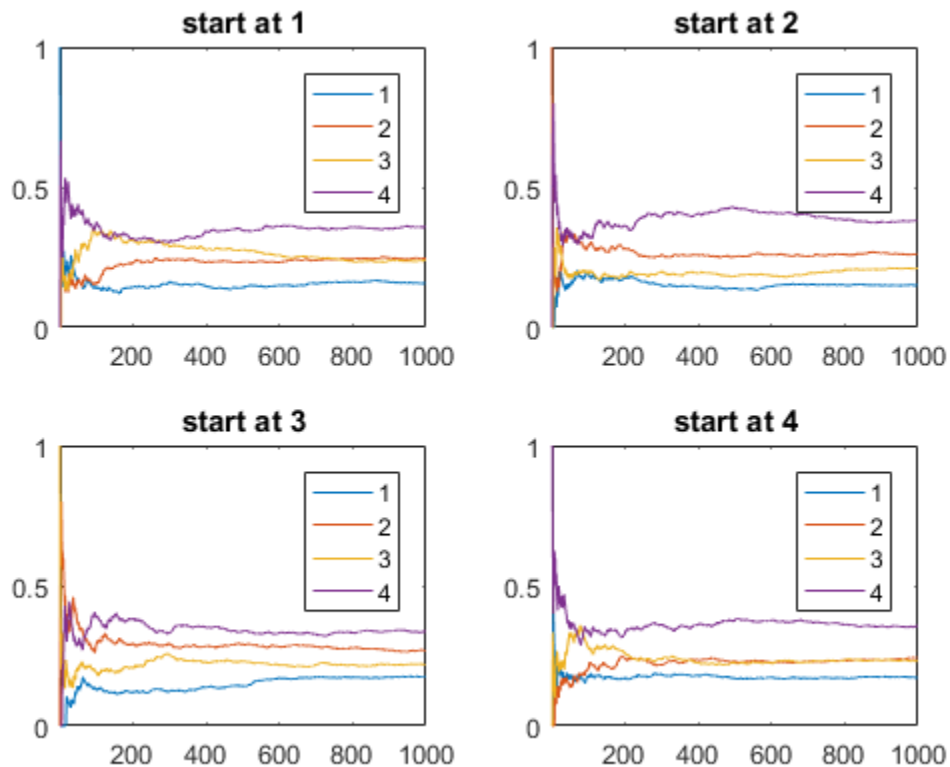
Li Sun

September 19, 2016

1. Write codes to simulate a discrete-time finite-state MC process with following transition Matrix:

$$\Pi = \begin{bmatrix} 0.2 & 0.2 & 0.1 & 0.5 \\ 0.1 & 0.3 & 0.4 & 0.2 \\ 0.3 & 0.2 & 0.3 & 0.2 \\ 0.1 & 0.3 & 0.1 & 0.5 \end{bmatrix}$$

Generates 4 sample path of MC, put relative frequency with which the path visits the four states for i up to 1000.



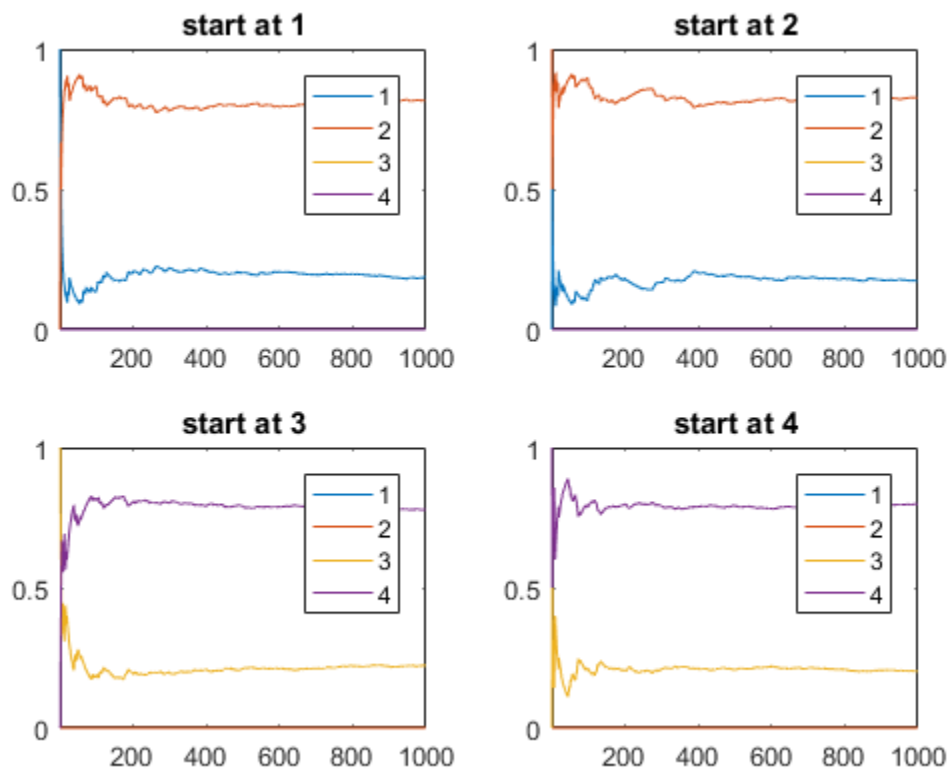
The re-scaled

dominant eigenvector is  $[0.1607 \ 0.2616 \ 0.2231 \ 0.3546]$

The simulated frequency starts from '1' is  $[0.157 \ 0.246 \ 0.24 \ 0.357]$ (if starts from other states, results are similar)

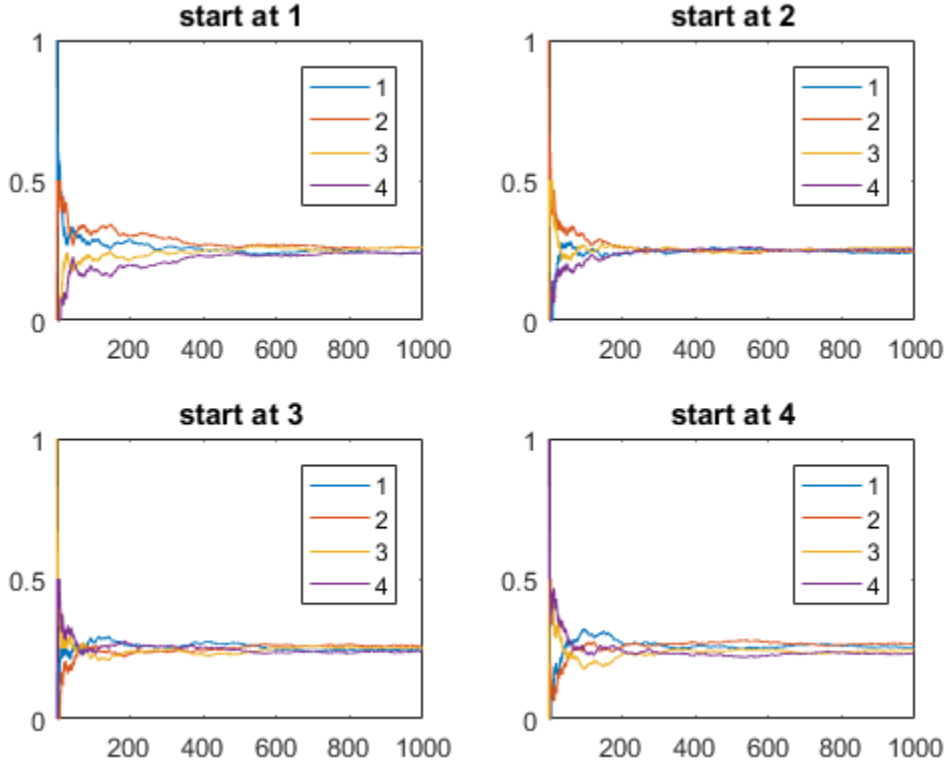
They are very close.

2. Do similar thing for another 2  $\Pi$   
 For the first  $\Pi$  which is reducible. we got



1. This  $\Pi$  is reducible because states 1, 2 are not communicating with states 3 and 4. So with different starting value we will get different stationary pmf. Theoretically we can divide this  $\Pi$  into 2 small  $\pi$  with dimension 2 by 2. We can calculate the theoretical stationary pmf of this two MC. If we start with 1 or 2. pmf1 is (0.1667, 0.8333, 0, 0), if we start with 3 or 4, the pmf2 is (0, 0, 0.2222, 0.7778). As shown in the 1000 step convergence, which is very close to theoretical ones calculated from the eigenvectors

For the second  $\Pi$  whose periodicity is 2. we got



The second  $\Pi$  is periodic with period 2 for all states. So at each step, the pmf is always different if the current state is different. This is like a process of moving a block in a 2 by 2 grid, the 4 possible location of the block is coded as 1,2,3,4 and the block has to move and only can move 1 step to one of its two neighbor locations. The probability of moving this block to either one is always equal. After a number of steps, we can find the probability for each location the block stays are equal. The simulated stationary pmf is roughly 0.25 for each state and the pmf calculated from eigenvector is (0.25 0.25 0.25 0.25), they match.

3. Let  $X_t$  be a MC generated using some initial probability  $P[1]$  and the transition matrix  $\Pi$  is:

$$\Pi = \begin{bmatrix} 0.1 & 0.3 & 0.4 & 0.2 \\ 0.2 & 0.1 & 0.3 & 0.4 \\ 0.4 & 0.2 & 0.1 & 0.3 \\ 0.3 & 0.4 & 0.2 & 0.1 \end{bmatrix}$$

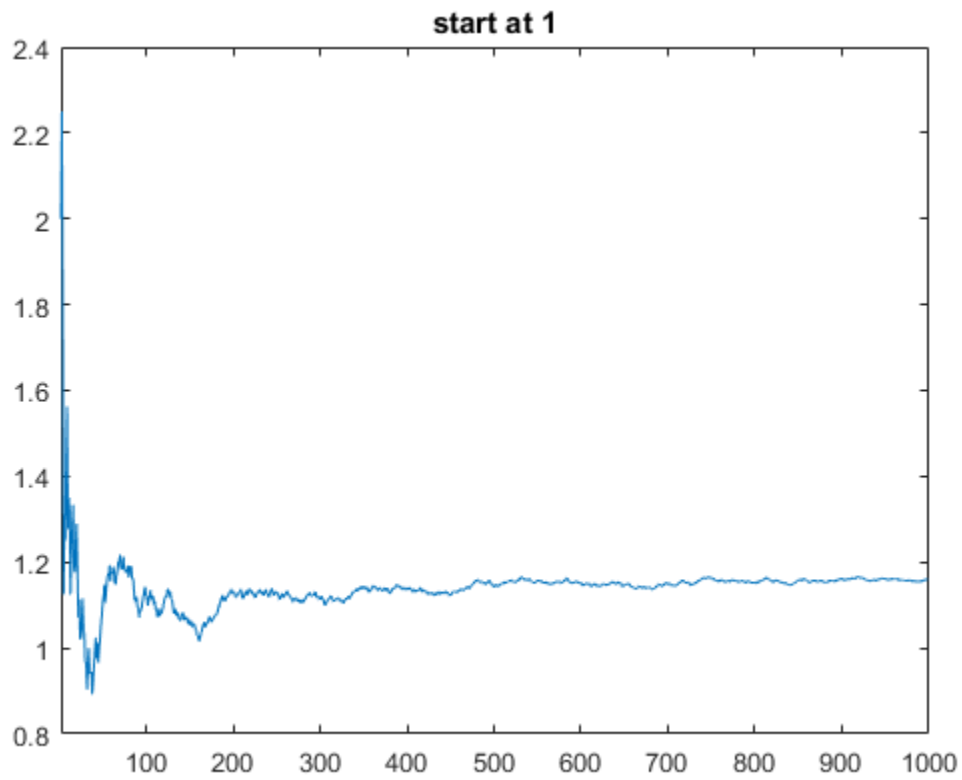
- i) the above matrix is irreducible because all elements are communicating with each other
- ii) the above matrix is aperiodic because  $\text{period}(a_1) = 1$  and all states share the same period because of i), so it is aperiodic. The stationary probability is found by re-scale the right eigenvector with eigenvalue is 1. It is (0.25,0.25,0.25,0.25).

For the second part, the right side can be calculated as

$$\sum_{j=1}^4 f(a_j)P(a_j) = E(f(a_j)) = 1.125$$

So this is to prove the mean of  $f(X_i)$  where  $X_i$  is MC with stationary pmf  $P$ , is asymptotically equal to the  $E(f(a_i))$  where  $a_i$  are from pmf  $P$ .

Simulation:



This shows that expected value is converging to 1.125 as expected.

All code please see <https://github.com/rikku1983/STAT5376>

Thanks!

%1000

```
clear all;
close all;
pai=[0.2 0.2 0.1 0.5; 0.1 0.3 0.4 0.2; 0.3 0.2 0.3 0.2; 0.1 0.3 0.1 0.5];
%generate MC chain with 4 different starting states
x=MCsim(pai, 1000);
%Calculate frequency versus i
y=@(n,i)[mean(x(i,1:n)==1), mean(x(i,1:n)==2),mean(x(i,1:n)==3),mean(x(i,1:n)==4)];
steps=1000;
for n=1:steps
    xper1(n,1:4)=y(n,1);
    xper2(n,1:4)=y(n,2);
    xper3(n,1:4)=y(n,3);
    xper4(n,1:4)=y(n,4);
end
subplot(221);plot(1:steps,xper1);title('start at 1');xlim([1,steps+1]);legend('1','2','3','4');
subplot(222);plot(1:steps,xper2);title('start at 2');xlim([1,steps+1]);legend('1','2','3','4');
subplot(223);plot(1:steps,xper3);title('start at 3');xlim([1,steps+1]);legend('1','2','3','4');
subplot(224);plot(1:steps,xper4);title('start at 4');xlim([1,steps+1]);legend('1','2','3','4');

%Calculate dominant eigenvector
[V,D,W]=eig(pai)
W(:,1)/sum(W(:,1))
%Simulated frequency
xper1(1000,:)

%question 2, for another pi which is reducible
clear all;
close all;
pai=[0.5 0.5 0 0; 0.1 0.9 0 0; 0 0 0.3 0.7; 0 0 0.2 0.8];
x=MCsim(pai, 1000);
%Calculate frequency versus i
y=@(n,i)[mean(x(i,1:n)==1), mean(x(i,1:n)==2),mean(x(i,1:n)==3),mean(x(i,1:n)==4)];
steps=1000;
for n=1:steps
    xper1(n,1:4)=y(n,1);
    xper2(n,1:4)=y(n,2);
    xper3(n,1:4)=y(n,3);
    xper4(n,1:4)=y(n,4);
end
subplot(221);plot(1:steps,xper1);title('start at 1');xlim([1,steps+1]);legend('1','2','3','4');
subplot(222);plot(1:steps,xper2);title('start at 2');xlim([1,steps+1]);legend('1','2','3','4');
subplot(223);plot(1:steps,xper3);title('start at 3');xlim([1,steps+1]);legend('1','2','3','4');
subplot(224);plot(1:steps,xper4);title('start at 4');xlim([1,steps+1]);legend('1','2','3','4');

%Calculate dominant eigenvector
[V,D,W]=eig(pai)
W(:,2)/sum(W(:,2))
```

```

W(:,4)/sum(W(:,4))
%Simulated frequency
xper1(1000,:)
xper2(1000,:)
xper3(1000,:)
xper4(1000,:)

% For the pi whose periodicity is 2
clear all;
close all;
pai=[0 0.5 0 0.5;0.5 0 0.5 0; 0 0.5 0 0.5; 0.5 0 0.5 0];
x=MCsim(pai, 1000);
%Calculate frequency versus i
y=@(n,i)[mean(x(i,1:n)==1), mean(x(i,1:n)==2),mean(x(i,1:n)==3),mean(x(i,1:n)==4)];
steps=1000;
for n=1:steps
    xper1(n,1:4)=y(n,1);
    xper2(n,1:4)=y(n,2);
    xper3(n,1:4)=y(n,3);
    xper4(n,1:4)=y(n,4);
end
subplot(221);plot(1:steps,xper1);title('start at 1');xlim([1,steps+1]);legend('1','2','3','4');
subplot(222);plot(1:steps,xper2);title('start at 2');xlim([1,steps+1]);legend('1','2','3','4');
subplot(223);plot(1:steps,xper3);title('start at 3');xlim([1,steps+1]);legend('1','2','3','4');
subplot(224);plot(1:steps,xper4);title('start at 4');xlim([1,steps+1]);legend('1','2','3','4');

%Calculate dominant eigenvector
[V,D,W]=eig(pai)
W(:,4)/sum(W(:,4))
%Simulated frequency
xper1(1000,:)
xper2(1000,:)
xper3(1000,:)
xper4(1000,:)

%Q3
clear all;
close all;
pai=[0.1 0.3 0.4 0.2;0.2 0.1 0.3 0.4;0.4 0.2 0.1 0.3; 0.3 0.4 0.2 0.1];
[V,D,W]=eig(pai);
W(:,1)/sum(W(:,1));
%simulation
x=MCsim(pai, 1000, 1);
%Calculate frequency versus i
f=[2,1,2.5,-1];
steps=1000;
for n=1:steps
    xper1(n)=mean(f(x(1:n)));

```

```
end
```

```
plot(1:steps,xper1);title('start at 1');xlim([1,steps+1]);
```

```

function x = MCsim(pai, n, start)

nstat=size(pai,1);
steps=n;
    if nargin < 3
        start=1:nstat;
    end
nsim=length(start);
for i=1:nsim
    x(i,1)=start(i); %start status at t1;
    for t=1:steps
        u=rand;
        if(u<pai(x(i,t),1))
            x(i,t+1)=1;
        elseif(u<sum(pai(x(i,t),1:2)))
            x(i,t+1)=2;
        elseif(u<sum(pai(x(i,t),1:3)))
            x(i,t+1)=3;
        else
            x(i,t+1)=4;
        end
    end
end
end

```