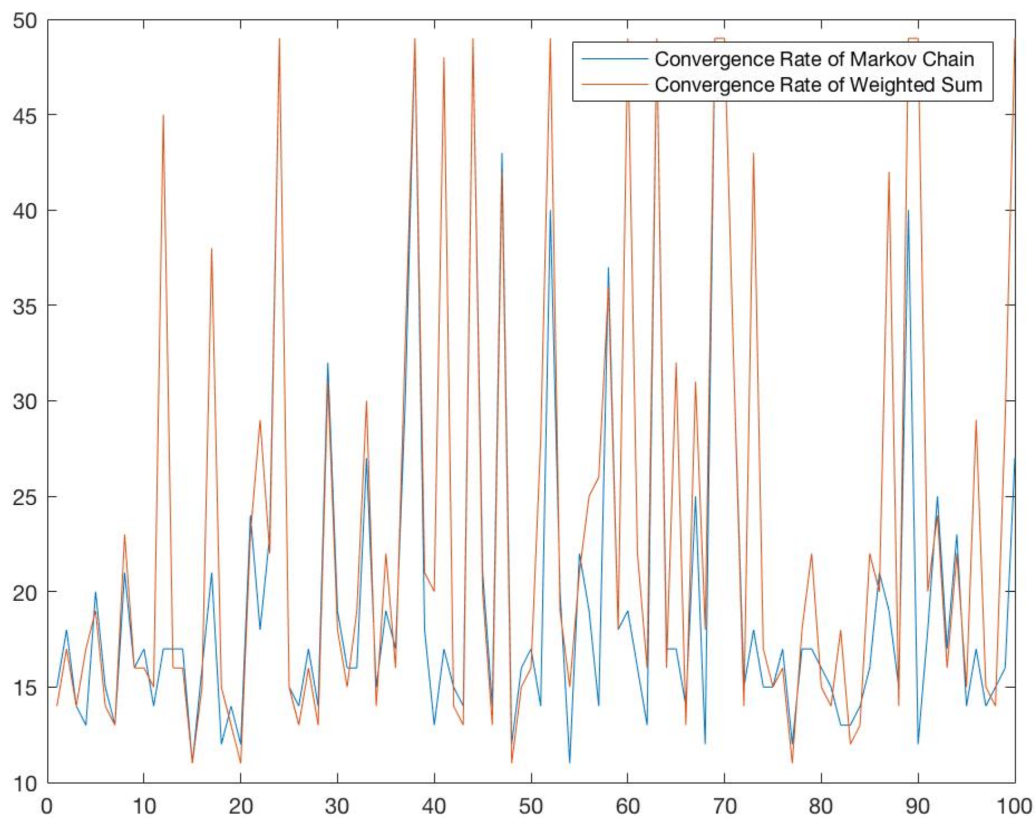
HW 8 & 9
Bayesian Network

4. DeGroot Model

The *DeGroot Model* is very similar to a Markov chain, by switching from different states alpha(0) coverage to alpha (infinity) with a stationary distribution.
I used *weighted model* to simulate the same problem, used T instead of P

$$T^d_{ij} = \frac{d_j}{\sum_{j \in N_i} d_j + d_i} \text{ and } T^d_{ii} = \frac{d_i}{\sum_{j \in N_i} d_j + d_i}, \tag{2.2}$$

Reference:
https://web.stanford.edu/~arungc/CLX.pdf



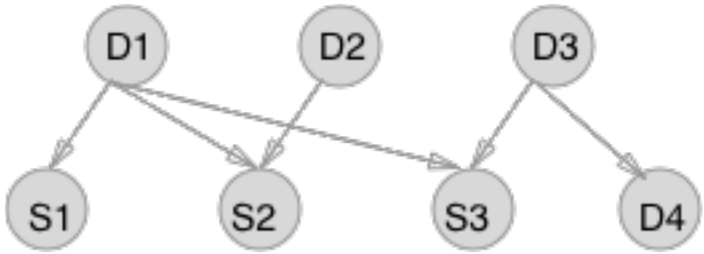
Comment:

The result shows *weighted method* converge less compared to *DeGroot* algorithm due to the modification of the transitional matrix, which is more realistic model compared to *DeGroot*

```
clear all;
close all;
%Simulate for over 100 times
```

```
% counter m count convergence of markov
% counter w count convergence of weighted
counter_m = zeros(1,100);
counter_w = zeros(1,100);
for m = 1:1:100
% P transtion
n = 10;
P = rand(n,n);
P = bsxfun(@times,P,1./sum(P,2));
% Weight matrix
T = rand(n,n);
P = bsxfun(@times,P,1./sum(P,2));
%
di = zeros(n,1) ;
dj = zeros(n,1) ;
%
di = sum(P,1);
dj = sum(P,2);
%
tij_p = (sum(dj)+di)';
for j = 1:1:10
T(j,:) = dj'/tij_p(j);
end
for i = 1:1:10
T(i,i) = di(i)/tij_p(i);
end
w = T./sum(T,2);
%
%Simulation on state transition
%T is not normalized some how add up to 0.9
t = 50;
Wei = zeros(n,t);
alpha = rand(10,1);
Wei(:,1) = alpha;
Markov(:,1) = alpha;
for i = 2:1:t
Wei(:,i) = w*Wei(:,i-1);
if Wei(:,i)==Wei(:,i-1)
    break
end
end
counter_w(m) = i-1;
for j = 2:1:t
Markov(:,j) = P*Wei(:,j-1);
if Markov(:,j)==Markov(:,j-1)
    break
end
end
counter_m(m) = j-1;
end
plot(counter_m)
hold on
plot(counter_w)
hold off
```

3. Bayesian Network, disease D1 D2 and D3, and we have symptom S1 S2 S3 and S4.
a)



b) P(D1,D2,D3,S1,S2,S3,S4) = P(D1) P (D2) P(D3) P(S1|D1) P(S2|D1,D2) P(S3|D1,D3) P(S4|D3)

c) 1+1+1+2+3+3+2 = 15

d) If not based on Bayesian, total number is 2^7-1 = 127

e) If S4 = true , we gain information of D3

f) If S2 = true , S4 = true, we gain information of D3 as well, because we can't confirm on D1 or D2, and D1 to D3 is independent, so we definitely gain information of D3

2. & 1
Will Submit Later with previous missing questions

1. Bearing only Target Tracking

```
Extended Kalman Filter
clear all;
close all;
% P transtion
n = 3;
P = rand(n,n);
P = bsxfun(@times,P,1./sum(P,2));
%Simulate for over 100 times
t = 100;
a = zeros(1,t);
u = rand(1,t);
yy = ones(1,t);
y = floor(n*rand(1,t))+1;
for i =2:1:t
    p = P(yy(i-1),:);
    a(i)= n*max(p);
    if u(i) < p(y)/(a/n)
        yy(i) = y(i);
    end
end
%observation
for i = 1:1:t
    if yy(i) == 1
        yy(i) = 5;
```

```
    else
    if yy(i) == 2
       yy(i) = 10;
    else
       yy(i) = 15;
        end
    end
end
```