Assignment 10

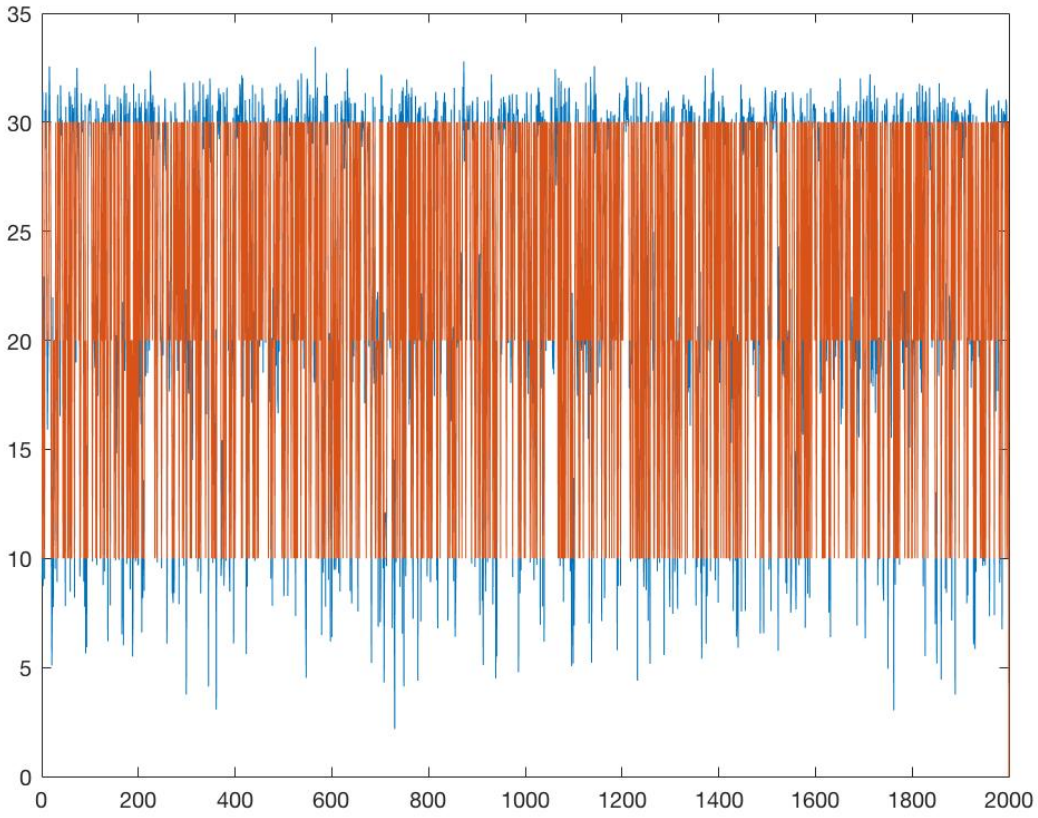
wl598

1. Simulate a 3-state HMM with iid Gaussian noise of zero mean and variance 1. Given 2000 data points of the HMM, illustrate using Mat- lab the performance of the EM algorithm for computing the maximum likelihood estimate of the transition matrix, state levels of the Markov chain and noise variance. Plot the likelihood at each iteration and show that it increases versus iteration number.

```
clear all;
close all;
clc;
n = 10000;
nu = [0,0,1];
Q = [0.8 0.1 0.1; 0.1 0.8 0.1;0.3 0.3 0.4];
g = [0.25 0.25 0.25 0.25; 0.25 0.25 0.25 0.25;0.05 0.05 0.45 0.45];
[x,y] = HMMsample(nu, Q, g, n);
n = length(y);
%filter
[phi, c] = HMMfilters(y, nu, Q, g);
%EM
tol = 0.01;
maxIt = 10;
[Q, g, l] = HMMem(y, nu, tol, maxIt, Q, g);


function [Q, g, l] = HMMem(y, nu, tol, maxIt, Q, g)
if nargin<4, maxIt = 100; end
if nargin<3, tol = 1e-4; end

k = length(nu); r = max(y); n = length(y);
Y = zeros(n, r); Y(sub2ind([n, r], 1:n, y))=1;

% if they are not provided, sample random initial transition and emission
matrices
if nargin<5, Q = rand(k); Q = Q ./ (sum(Q, 2)*ones(1, k)); end
if nargin<6, g = rand(k, r); g = g ./ (sum(g, 2)*ones(1, r)); end

it = 0; oldQ = Q; oldg = g+tol+1;
while ((norm(oldQ(:)-Q(:), 1) + norm(oldg-g, 1) > tol) && (it<maxIt))
  it = it + 1;
  % compute the posterior distribution for the current parameters
  [phi, c] = HMMfilters(y, nu, Q, g);
  beta = HMMsmoothers(y, Q, g, c);
  post = phi .* beta;

  % expectation of the number of transitions under the current parameters
  N =Q.*(phi(:, 1:(end-1))*(beta(:, 2:end).*g(:, y(2:end))./(ones(k,
1)*c(2:end)))');
  % expectation of the number of emissions
  M = post * Y;

  % re-estimation
  oldQ = Q; oldg = g;
```

```
  Q = N ./ (sum(N, 2) * ones(1, k));
  g = M ./ (sum(M, 2) * ones(1, r));
end
l = sum(log(c));

function beta = HMMsmoothers(y, Q, g, c)
n = length(y);
beta = ones(size(Q, 1), n);
for t=(n-1):-1:1
  beta(:, t) = Q * (g(:, y(t+1)) .* beta(:, t+1)) / c(t+1);
end

function [phi, c] = HMMfilters(y, nu, Q, g)
n = length(y);
phi = zeros(size(Q, 1), n);
c = zeros(1, n);

Z = nu'.*g(:, y(1));
c(1)= sum(Z);
phi(:, 1) = Z/c(1);

for t=2:n
  Z = (phi(:, t-1)' * Q)' .* g(:, y(t));
  c(t) = sum(Z);
  phi(:, t) = Z / c(t);
end

clear all;
close all;
%hmm
clear all;
close all;
N = 2000;
noise = normrnd(0,1,[1,N]);
%noise = randn(N,1);
Y = zeros(N,1);
O = zeros(N,1);
%noise = 5*randn(N,1);
% transition
P = [0.3 0.3 0.4;0.2 0.7 0.1;0.5 0.4 0.1];
C = [10 20 30];
D = [3 2 1];
% 3 States Markov Chain Simulation
% 4 kinds of observations
pi = zeros(3,N);
pi(:,1) = [0.3;0.1;0.6];
dice = zeros(1,N);
state = zeros(1,N);
X = zeros(1,N);
% sate simulation
for i = 2:1:N
    dice(i-1) = rand;
    if dice(i-1) < pi(1,i-1)
        state(i-1) = 1;
```

```
    else if dice(i-1) < pi(1,i-1) + pi(2,i-1)
            state(i-1) = 2;
        else
            state(i-1) = 3;
        end
    end
    pi(:,i) = P*pi(:,i-1) ;
end
% yk
for i = 1:1:N-1
    ind = state(i);
    Y(i) = C(ind) + D(ind)*noise(i);
    O(i) = C(ind);
end
figure(1)
plot(Y)
hold on
plot(O)
hold off
```



I =

1.0e+04 *

0 -1.3764

0.798034487366878 0.0997543109208597 0.102211201712263
0.0997543109208597 0.798034487366878 0.102211201712263
0.296764244650522 0.296764244650522 0.406471510698957

2. The un-normalized HMM filter can be used to evaluate the likelihood of a model given a sequence of observations. Use this with a general purpose optimizer in Matlab to compute the maximum likelihood estimate of a HMM for the same data sequence as Problem 1. Compare the performance with the EM algorithm in Problem 1.