# EQ2341 Pattern Recognition and Machine Learning Assignment 3

Ayushi Shah
ayushi@kth.se

Sri Janani Rengarajan
sjre@kth.se

May 8, 2020

In this assignment a Matlab function to perform the Forward Algorithm is implemented and verified.

## Forward Algorithm

Implementation of *forward* function from *MarkovChain* class

```
function [alfaHat, c]=forward(mc,pX)
T = size(pX, 2);   %Number of observations
N = mc.nStates;
A = mc.TransitionProb;
% forward step iteration
for t = 1 : T
    if t ==1
        alpha_temp = mc.InitialProb.*pX(:, 1);
        c = sum(alpha_temp);
        alfaHat = alpha_temp/c;
    else
        alpha_temp = pX(:, t).*(alfaHat(:, t - 1)'*A(:, 1 : N))';
        ct = sum(alpha_temp);
        c = [c ct];
        alfaHat = [alfaHat alpha_temp/ct];
    end
end

if finiteDuration(mc)
    c = [c alfaHat(:, end)'*A(:, end)];
end
end
```

## Log Prob

Implementation of *logprob* function from *HMM* class

```
function logP=logprob(hmm,x)
hmmSize=size(hmm);%size of hmm array
T=size(x,2);%number of vector samples in observed sequence
logP=zeros(hmmSize);%space for result
for i=1:numel(hmm)%for all HMM objects
    if hmm(i).DataSize == size(x,1)
        [p, logS] = prob(hmm(i).OutputDistr, x);
        scaling_factor = exp(logS).*ones(size(p));
        [~, c] = forward(hmm(i).StateGen, p.*scaling_factor);
        logP(i) = sum(log(c));
    else
        logP(i) = -inf;
    end

end;
```

# Verification

Forward algorithm implementation is verified for the following Markov chains using:

```
q = [1;0];
A = [0.9 0.1; 0.9 0.1];
mc=MarkovChain(q, A);%State generator
g1=GaussD("Mean",0,"StDev",1); %Distribution for state=1
g2=GaussD("Mean",3,"StDev",2); %Distribution for state=2
h=HMM(mc, [g1; g2]);
x = [-0.2,2.6,1.3];
pX = prob([g1,g2],x);
[alfaHat, c]=forward(mc,pX);
logP = logprob(h,x);
```

Finite duration:

$$\text{q}=\begin{pmatrix}1\\0\end{pmatrix};\ \text{A}=\begin{pmatrix}0.9 & 0.1 & 0\\0 & 0.9 & 0.1\end{pmatrix};\ \text{B}=\begin{pmatrix}\mathcal{N}(0,1)\\\mathcal{N}(3,2)\end{pmatrix}$$

Following is the Output From the codes

```
Values from the code for finite duration

alfaHat =

    1.0000    0.3847    0.4189
         0    0.6153    0.5811
c =

    1.0000    0.1625    0.8266    0.0581

Output of logprob function

logP = -9.1877
```

| $t$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\hat{\alpha}_{1,t}$ | 1 | 0 | 0 | 0 | 0 |
| $\hat{\alpha}_{2,t}$ | 0 | 1 | 1/7 | 1/79 | 0 |
| $\hat{\alpha}_{3,t}$ | 0 | 0 | 6/7 | 78/79 | 1 |
| $c_t$ | 1 | 0.35 | 0.35 | 79/140 | 0.0994 |

Figure 1: Answer table for infinite duration HMM

Verification code for infinite duration HMM (from Excercise 5.1):

```
q=[1 0 0]';
A = [0.3 0.7 0; 0 0.5 0.5; 0 0 1];
B = [1 0 0 0; 0 0.5 0.4 0.1; 0.1 0.1 0.2 0.6];
mc = MarkovChain(q,A);
pD=DiscreteD(B);%eqivalent to
h=HMM(mc,pD);
nStates=h.nStates;
z =[1 2 4 4 1];
T=length(z);
pZ=prob(pD,z);
[alfaHat,c]=forward(mc,pZ);
```

$$q=\begin{pmatrix}1\\0\\0\end{pmatrix};\ A=\begin{pmatrix}0.3 & 0.7 & 0\\0 & 0.5 & 0.5\\0 & 0 & 1\end{pmatrix};\ B=\begin{pmatrix}1 & 0 & 0 & 0\\0 & 0.5 & 0.4 & 0.1\\0.1 & 0.1 & 0.2 & 0.6\end{pmatrix}$$

Following are the Values from the code for infinite duration. Values from the manual calculations are shown in Figure 1

```
alfaHat =

    1.0000         0         0         0         0
         0    1.0000    0.1429    0.0127         0
         0         0    0.8571    0.9873    1.0000
c =

    1.0000    0.3500    0.3500    0.5643    0.0994
```