Mini Project 3 - Viterbi Algorithm

Rishi Laddha, Section 2

Question 1

Code:

```
%Viterbi Algorithm Complexit - K^2 T
%(K -> Number of states, T -> Number of time steps)

%E (Emission probability)-> probability that a particular state corresponds to a particular
%observation e.g. probability that it is sunny if I had gone for a walk.

clc
clear all

S = ["rainy" "sunny"];
P = [0.2 0.8; 0.4 0.6];
PD = [0.43 0.57];
E = [0.2 0.4 0.4; 0.3 0.25 0.45];
Y=[1 1 2 1 3]; %say

%intialisation
for i=1:size(S,2)
    viterbi_prob(i,1)=PD(i)*E(i,Y(1)); %E(i,1) because first observation is fixed
    viterbi_path(i,1)=0;
end

%computing viterbi path
for j=2:size(Y,2)
    for i=1:size(S,2)
        [viterbi_prob(i,j), viterbi_path(i,j)] = max(E(i, Y(j)).*P(:, i).*viterbi_prob(:, j-1));
    end
end

[useless, Z(size(Y,2))]= max(viterbi_prob(:,size(Y,2)));
X(size(Y,2)) = S(Z(size(Y,2)));
%Retracking the most likely path X

for j=size(Y,2):-1:2
    Z(j-1)=viterbi_path(Z(j),j);
    X(j-1)=S(Z(j-1));
end

viterbi_path
viterbi_prob
X
Z
```

Output:

```
viterbi_prob =

    0.0860    0.0137    0.0049    0.0004    0.0002
    0.1710    0.0308    0.0046    0.0012    0.0003


viterbi_path =

     0     2     2     2     2
     0     2     2     1     2


X =

  1×5 string array

    "sunny"    "sunny"    "rainy"    "sunny"    "sunny"
```

```
Z =

     2     2     1     2     2
```

Predicted weather pattern for the required 5 days for the given observations is sunny, sunny, rainy, sunny and sunny.

Question 2

Code:

```
%Viterbi Algorithm Complexit - K^2 T
%(K -> Number of states, T -> Number of time steps)

%E (Emission probability)-> probability that a particular state corresponds to a particular
%observation e.g. probability that it is sunny if I had gone for a walk.

clc
clear all

S = ["normal" "alternate" "direct"];
P = [0.7 0.1 0.2; 0.4 0.5 0.1; 0.2 0.3 0.5];
PD = [0.8 0.1 0.1];
E = [0.6 0.4; 0.3 0.7; 0.2 0.8];
Y=[1 2 2 2 2 1 1 2 2 2 2];

%intialisation
for i=1:size(S,2)
    viterbi_prob(i,1)=PD(i)*E(i,Y(1)); %E(i,1) because first observation is fixed
    viterbi_path(i,1)=0;
end

%computing viterbi path
for j=2:size(Y,2)
    for i=1:size(S,2)
        [viterbi_prob(i,j), viterbi_path(i,j)] = max(E(i, Y(j)).*P(:, i).*viterbi_prob(:, j-1));
    end
end

[useless, Z(size(Y,2))]= max(viterbi_prob(:,size(Y,2)));
X(size(Y,2)) = S(Z(size(Y,2)));

%Retracking the most likely path X
for j=size(Y,2):-1:2
    Z(j-1)=viterbi_path(Z(j),j);
    X(j-1)=S(Z(j-1));
end

format shortE
viterbi_prob
viterbi_path
X
Z
```

Output:

```
viterbi_prob =

   4.8000e-01   1.3440e-01   3.7632e-02   1.0537e-02   2.9503e-03   1.2391e-03   5.2044e-04   1.4572e-04   4.0803e-05   1.1425e-05   3.1989e-06
   3.0000e-02   3.3600e-02   1.6128e-02   6.4512e-03   2.5805e-03   4.4237e-04   6.6355e-05   3.6431e-05   1.7487e-05   6.9947e-06   2.7979e-06
   2.0000e-02   7.6800e-02   3.0720e-02   1.2288e-02   4.9152e-03   4.9152e-04   4.9566e-05   8.3271e-05   3.3308e-05   1.3323e-05   5.3293e-06


viterbi_path =

     0     1     1     1     1     1     1     1     1     1     1
     0     1     3     3     3     3     2     1     3     3     3
     0     1     3     3     3     3     1     1     3     3     3


X =

  1×11 string array

    "normal"    "normal"    "normal"    "normal"    "normal"    "normal"    "normal"    "direct"    "direct"    "direct"    "direct"
```

```
Z =

     1     1     1     1     1     1     1     3     3     3     3
```

Predicted control laws for the given pitch data is normal, normal, normal, normal, normal, normal, normal, direct, direct, direct and direct.