TDT4171 EXERCISE 2

Due: 22th Feb @ 0800

Name: Mathilde Hotvedt
Mail: mathildh@stud.ntnu.no

PART A: Describe the "Umbrella World" as an HMM

1) What is the set of unobserved variables for a given time-slice t (denoted X_t in the book)?

Answer:

The security guard of the umbrella world would like to know whether it is raining today but cannot go outside for himself.

unobserved variable is Rain: $X_t = Rain_t = R_t$

the timeslice is one day t = one day.

X_t takes on two values, true or false.

$X_t = \{R_{0:t}\}$

2) What is the set of observable variables for a given time-slice t (denoted E_t in the book)?

Answer:

The security guard is observing one evidence variable, wether the director is coming in with or without an umbrella.

$E_t = Umbrella_t = U_t$

$E_t = \{U_{1:t}\}$

E_t takes on two values true or false

3) Present the dynamic model $P(X_t | X_{t-1})$ and the observation model $P(E_t | X_t)$ as matrices

Answer:

$P(X_t | X_{t-1})$ $\rightarrow$ probability of rain or not rain on day t given the weather the day before

From figure 15.2 in the book we have given the table:

R_{t-1}	$P(R_t)$
t	0.7
f	0.3

And from chapter 15.3.1 it is given, given that we have a single discrete state variable X_t which we do, that the transition model may be written as

$T_{ij} = P(X_t = j | X_{t-1} = i)$

for instance $T_{11} = P(X_t = true | X_{t-1} = true) = 0.7$

$T_{12} = P(X_t = false | X_{t-1} = true) = 1 - 0.7 = 0.3$

$T_{21} = P(X_t = true | X_{t-1} = false) = 0.3$

$T_{22} = P(X_t = false | X_{t-1} = false) = 1 - 0.3 = 0.7$

$$\Rightarrow \mathbb{T} = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$$

The observation model will be drawn out from the 2nd table in figure 15.2

R_t	$P(U_t)$
t	0.9
f	0.2

And from chapter 15.3.1 we have that $P(e_t \mid X_t = i)$ will be given as a diagonal matrix with the entry $P(e_t \mid X_t = i)$ given on the diagonal

so if $U_t = true$:

$$O_t = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.2 \end{bmatrix}$$

if $U_t = false$:

$$O_t = \begin{bmatrix} 1-0.9 & 0 \\ 0 & 1-0.2 \end{bmatrix} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.8 \end{bmatrix}$$

4) which assumptions are encoded in this model? Are the assumptions reasonable for this particular domain?

<u>Answer</u>:

- First assumption $P(X_t \mid X_{0:t-1}) = P(X_t \mid X_{t-1})$ that is the current value of $X = X_t$ is only dependent on the previous state $X = X_{t-1}$ and none of the earlier states. This is called a first-order Markov process.

$$X_{t-2} \to X_{t-1} \to X_t \to X_{t+1} \to X_{t+2}$$

- 2nd assumption is that the probability of rain $P(X_t \mid X_{t-1})$ is the same for all timeslice t. Hence we view the world as stationary, "that the process of change that is governed by laws do not themselves change" [p. 568]

- 3rd assumption, the observation model is only dependent on the current state

  $$P(E_t \mid X_{0:t}, E_{0:t-1}) = P(E_t \mid X_t)$$

  and not on earlier evidence variables nor earlier state variables.

- 4th assumption - for making a HMM we need X_t to be a single discrete variable.

The assumptions are in this domain only approximate. For instance, rain often seem to persist in the real world, that is if it rained yesterday the probability that it is going to rain today is greater than if it was sunshine yesterday.

Further, rain depends on location. it does rain more in Norway than in Egypt.

Rain will also depend on season, it rains more often during fall than summer etc.
Rain will also depend on temperature, humidity and pressure

However, if the model doesnt suffice we can add more variables to take this into account.

PART B Implementing filtering

- According to section 15.3.1 the forward operation may be implemented as

$$f_{1:t+1} = \alpha O_{t+1} T^T f_{1:t}$$

where $\alpha = P(e_{1:t+1})$ when using Bayes Rule

$$P(A|B) = \frac{P(B,A)P(A)}{P(B)}$$

α = normalization constant,

NB: I have published all code and put it as appendix to this assignment, but it can also be found in zip.

1) verifying implementation by calculating $P(X_2 | e_{1:2})$

$$P(X_0) = \langle 0.5, 0.5 \rangle$$

- As Matlab is 1-indexed I must remember to include day "zero" in my vectors
- See Matlab script oving2-partB.m which implements FORWARD as stated in 15.12. with matrix multiplications
- To normalize f vectors at each timestep t we must divide by the sum of the elements at $f_{1:t+1}$.
- The result was

$$f = \begin{array}{c} \begin{bmatrix} 0.5 & 0.8182 & 0.8834 \\ 0.5 & 0.1818 & 0.1166 \end{bmatrix} \begin{array}{l} \leftarrow \text{True} \\ \leftarrow \text{False} \end{array} \\ \begin{array}{ccc} \uparrow \text{ initial value "day zero"} & \uparrow \text{ day 1} & \uparrow \text{ day 2} \end{array} \end{array}$$

Hence, my algorithm found

$P(X_2 | e_{1:2}) = 0.8834 \approx 0.883$

which was given in the assignment and it seems that the algorithm works

- Now including several more evidence variables

$e_{1:5} = \{Umbrella_1 = true, Umbrella_2 = true,$

$Umbrella_3 = false, Umbrella_4 = true, Umbrella_5 = true\}$

we are to find $P(X_5 | e_{1:5})$.

see same document aing2_partB.m

Results became:

Day	1	2	3	4	5
True	0.8182	0.8834	0.1907	0.7308	0.8673
False	0.1818	0.1166	0.8093	0.2692	0.1327

PART C Implementing FORWARD - BACKWARD

- According to section 15.3.1 we have the two equations:

15.12 - FORWARD

$$f_{1:t+1} = \alpha O_{t+1} T^{T} f_{1:t}$$

15.13 - BACKWARD

$$b_{k+1:t} = T O_{k+1} b_{k+2:t}$$

which should be implemented.

- for the backwards part I will start from t and decrement down to 1.

- the Matlab script + result are found in file oving2-partC.m

- seen from the result: $P(X_1 \mid e_{1:2}) = \langle 0.8833, 0.1166 \rangle \sim \langle 0.883, 0.117 \rangle$ as expected. Code and result published are also found in the back of this exercise as well as code in oving2.zip

- when including $e_{1:5}$ I still

$P(X_1 \mid e_{1:5}) = \langle 0.8673, 0.1327 \rangle$

and the backwards messages became

$$b_{1:5} = \begin{bmatrix} 0.0444 & 0.0661 & 0.0906 & 0.4593 & 0.6900 \\ 0.0242 & 0.0455 & 0.1503 & 0.2437 & 0.4100 \end{bmatrix}$$

DESCRIBING / IMPLEMENTING VITERBI ALGORITHM

- The main idea behind this algorithm is that one want to find the most likely sequence of states that explains the evidence = observable variables that you have obtained.

- At the end of simulation, it will have the probability
- for the most likely sequence reaching each of the final states, and one may then choose the most likely overall by going backwards

- The algorithm is identical to 15.5 - filtering or FORWARD only that the forward message $f_{1:t} = P(X_t \mid e_{1:t})$ is replaced by

$$m_{1:t} = \max_{x_1 \dots x_{t-1}} P(x_1, \dots, x_{t-1}, X_t \mid e_{1:t})$$

which is the probability of the most likely path to each state x_t.

- we had that O_t represented $P(e_t \mid X_t = i)$ which was the observation model.

Hence:

$$\max_{x_1 \dots x_t} P(x_1, \dots, x_t, X_{t+1} \mid e_{1:t+1})$$

$$= \alpha \underbrace{P(e_{t+1} \mid X_{t+1})}_{\text{observation model}} \max_{x_t} \left(\underbrace{P(X_{t+1} \mid x_t)}_{\text{transition-model.}} \max_{x_1 \dots x_{t-1}} P(x_1, \dots, x_{t-1}, x_t \mid e_{1:t} \right)$$

This is how I implemented Viterbi algorithm

% Initialize different vectors / matrices

set $m_{1:1}$ = forward alg. result for day 1.

for $i = 2$ to end

Find if umbrella showed up or not and use vector $\begin{bmatrix} 0.9 \\ 0.2 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ 0.8 \end{bmatrix}$ accordingly

for $j = 1$ to 2 which is true, then false state

use the correct transition vector $\begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}$ $\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix}$

multiply t-vector with m_{i-1} (elementwise)

Find max. entry

$m_i = \underbrace{\text{observationvector}(j)}_{\text{true or false value in vector}} * \text{maximum}$.

end

end

Traceback from end to 1 to find max entry in m_i and save if it is on position true or false.

I received the result:

$$m_{1:5} = \begin{bmatrix} 0.8182 & 0.5155 & 0.0361 & 0.0334 & 0.0210 \\ 0.1818 & 0.0491 & 0.1237 & 0.0173 & 0.0024 \end{bmatrix}$$

as in textbook with

evidence = [true true false true true]

Scanned by CamScanner

The most likely sequence of events to explain observations become

seq = [true, true, false, true, true].

Table of Contents

Oving 2 - part B

```
%Initialization
f_0 = [0.5,0.5]'; %P(X_0) initial probability of rain and not rain
T = [0.7, 0.3;
    0.3, 0.7];    % Transition matrix, the dynamic model
O_true = [0.9, 0; % Observation model if the umbrella shows up
          0, 0.2];
O_false = eye(2) - O_true;  %Evidence model if the umbrella does not
 show up
```

First part, prediction for the second day X_2

```
Evidence = [1,1]; %1 if umbrella showed up at day i
final_time = 2;
f = zeros(length(f_0),final_time+1); %initialize space for the
 finished result
f(:,1) = f_0;

% FORWARD algorithm
for i=1:final_time
    O_temp = O_false;
    if Evidence(i)
        O_temp = O_true;
    end
    f(:,i+1) = O_temp*T'*f(:,i);
    %Normalizing
    f(:,i+1) = f(:,i+1)/(sum(f(:,i+1)));
end

%Displaying the probability of rain at day 2
disp('P(X_2 | e_{1:2}) = ');
disp(f(1,3));
disp('All normalized messages f_{1:2}');
disp(f(:,2:end));
```

```
P(X_2 | e_{1:2}) =
    0.8834

All normalized messages f_{1:2}
    0.8182    0.8834
    0.1818    0.1166
```

Second part, prediction for the fifth day X_5

```
Evidence = [1,1,0,1,1];
final_time = 5;
f = zeros(length(f_0),final_time+1); %initialize space for the
 finished result
f(:,1) = f_0;

% FORWARD algorithm
for i=1:final_time
    O_temp = O_false;
    if Evidence(i)
        O_temp = O_true;
    end
    f(:,i+1) = O_temp*T'*f(:,i);
    %Normalizing
    f(:,i+1) = f(:,i+1)/(sum(f(:,i+1)));
end

%Displaying the probability of rain at day 2
disp('P(X_5 | e_{1:5}) = ');
disp(f(1,5));
disp('All normalized messages f_{1:5}');
disp(f(:,2:end));
```

```
P(X_5 | e_{1:5}) =
    0.7308

All normalized messages f_{1:5}
    0.8182    0.8834    0.1907    0.7308    0.8673
    0.1818    0.1166    0.8093    0.2692    0.1327
```

Published with MATLAB® R2015a

Table of Contents

Oving2_partC - Smoothing

```
clear all
%Initialization
f_0 = [0.5,0.5]'; %P(X_0) initial probability of rain and not rain
T = [0.7, 0.3;
    0.3, 0.7];    % Transition matrix, the dynamic model
O_true = [0.9, 0; % Observation model if the umbrella shows up
          0, 0.2];
O_false = eye(2) - O_true;  %Evidence model if the umbrella does not
 show up
```

First part, prediction for the first day X_1

```
%but now with smoothed estimates from two evidence variables
Evidence = [1,1]; %1 if umbrella showed up at day i
final_time = 2;
f = zeros(length(f_0),final_time+1); %initialize space for the
 finished result of forward messages
b = ones(length(f_0),final_time+1); %initialize space for the backward
 messages
s = zeros(length(f_0),final_time); %initialize space for the smoothed
 states
f(:,1) = f_0;


% FORWARD algorithm
for i=1:final_time
    O_temp = O_false;
    if Evidence(i)
        O_temp = O_true;
    end
    f(:,i+1) = O_temp*T'*f(:,i);
    %f(:,i+1) = f(:,i+1)/(sum(f(:,i+1))); %normalizing the f, do not
 think it is necessary, but just a precation
end


%BACKWARD algorithm
for i=final_time:(-1):1
    O_temp = O_false;
    if Evidence(i)
        O_temp = O_true;
    end
```

```
    %Before normalizing
    non_norm = f(:,i+1).*b(:,i+1);
    %normalizing the smoothed estimates
    s(:,i) = non_norm/sum(non_norm);

    %backward message
    b(:,i) = T*O_temp*b(:,i+1);

end

%Displaying results for the smoothed estimate of day 1
disp('Smoothed estimate for day 1:');
disp('P(X_1) = ');
disp(s(:,1));

Smoothed estimate for day 1:
P(X_1) =
    0.8834
    0.1166
```

Second part, prediction for the first day X_1

```
%but with smoothed estimate from 5 evidence values
Evidence = [1,1,0,1,1]; %1 if umbrella showed up at day i
final_time = 5;
f = zeros(length(f_0),final_time+1); %initialize space for the
 finished result of forward messages
b = ones(length(f_0),final_time+1); %initialize space for the backward
 messages
s = zeros(length(f_0),final_time); %initialize space for the smoothed
 states
f(:,1) = f_0;


% FORWARD algorithm
for i=1:final_time
    O_temp = O_false;
    if Evidence(i)
        O_temp = O_true;
    end
    f(:,i+1) = O_temp*T'*f(:,i);
    %f(:,i+1) = f(:,i+1)/(sum(f(:,i+1))); %normalizing the f, do not
 think it is necessary, but just a precation
end

%BACKWARD algorithm
for i=final_time:(-1):1
    O_temp = O_false;
    if Evidence(i)
        O_temp = O_true;
    end
    %Before normalizing
```

```
    non_norm = f(:,i+1).*b(:,i+1);
    %normalizing the smoothed estimates
    s(:,i) = non_norm/sum(non_norm);

    %backward message
    b(:,i) = T*O_temp*b(:,i+1);

end

%Displaying results for the smoothed estimate of day 1
disp('Smoothed estimate for day 1:');
disp('P(X_1) = ');
disp(s(:,1));
disp('The backward messages:');
disp('b = ');
disp(b(:,1:final_time));
```

```
Smoothed estimate for day 1:
P(X_1) =
    0.8673
    0.1327

The backward messages:
b =
    0.0444    0.0661    0.0906    0.4593    0.6900
    0.0242    0.0455    0.1503    0.2437    0.4100
```

Published with MATLAB® R2015a

Viterbi

```
clear all
%Initialization
T_true = [0.7;0.3];     % Transition matrix for when X is true and
 false
T_false = [0.3; 0.7];
T = [T_true, T_false];
O_true = [0.9;0.2];     % Observation model if the umbrella shows up
O_false = [0.1;0.8];    % Evidence model if the umbrella does not show
 up
Evidence = [1,1,0,1,1]; % 1 if umbrella showed up at day i
final_time = 5;
m = zeros(length(T_true),final_time);
m(:,1) = [0.8182;0.1818];            % from the first step of FORWARD,
 just to have a starting point
obs = O_false;
sequence = zeros(1,final_time);     %storing the sequence of most
 likely states that gave the observations
```

Viterbi

```
for i=2:final_time
    if Evidence(i)              %Find whether umbrella showed up or
 not
        obs = O_true;
    else
        obs = O_false;
    end

    for j=1:2
        T_temp = T(:,j);        %first do true, then false
        maximum = max(T_temp.*m(:,i-1));
        m(j,i) = obs(j)*maximum;
    end
end

%Displaying the result of the messages
disp('m_{1:t} = ');
disp(m);

%Tracebacking to find most likely sequence of states
for i=final_time:-1:1;
    if m(1,i) > m(2,i)
        sequence(i) = 1;
    else
        sequence(i) = 0;
    end

end

%Displaying the most likely sequemce of states
```

```
for j=1:final_time
    fprintf('Rain at day: %g, ',j);
    if sequence(j)
        disp('is most likely true');
    else
        disp('is most likely false');
    end
end
```

```
m_{1:t} =
    0.8182    0.5155    0.0361    0.0334    0.0210
    0.1818    0.0491    0.1237    0.0173    0.0024

Rain at day: 1, is most likely true
Rain at day: 2, is most likely true
Rain at day: 3, is most likely false
Rain at day: 4, is most likely true
Rain at day: 5, is most likely true
```

Published with MATLAB® R2015a