aMI Handin assignments

Sander Jespersen, Mathias Melgaard Andersen, Anders Roland Nielsen,
Andreas Berre Eriksen, Kent Munthe Caspersen & Mikkel Alexander Madsen

March 30, 2015

1 Water tank

1.1 Baysian network

Indsæt figur fra Genie image.

1.2 Suitable conditional probability distribution

Hvis ventil er åben fanger den stortset altid der er flow, hvis den er lukket laver den oftere fejl.

1.3 Simulation

We assume that only the controller has access to the water level sensor, and thus we can not manually read the water level through the sensor.

2 Matlab

2.1 Forward-backward algorithm

The changes we did to the given code was as follows:

- In the HMM class we added a backwardMessages
- We initialised it to NaN when a HMM object was created

```
backwardMessages;

function obj = HMM(priorModel, transModel, sensorModel)
  ...
  obj.backwardMessages = NaN;
end
```

The code below is our implementation of the backward part of the algorithm. The linebreak in the code is not present in the actual code but was done to fit on the page.

```
function obj = backward(obj, data)
  totalTime = length(data);

  obj.backwardMessages=zeros(obj.noHidden,totalTime+1);

  obj.backwardMessages(:,totalTime+1) = 1;
    for t=totalTime:-1:1,
      obj.backwardMessages(:,t)
      = obj.transModel*obj.sensorModel{data(t)}*obj.backwardMessages(:,t+1);
      obj.backwardMessages(:,t)
      = obj.backwardMessages(:,t)./sum(obj.backwardMessages(:,t));
    end
end
```

The result of running the our function on the given demo that forward was run on gives the following result:

```
0.6469   0.5923   0.3763   0.6533   0.6273   1.0000
0.3531   0.4077   0.6237   0.3467   0.3727   1.0000
```

2.2 HMM for exercise 1

```
Trans = [ 0.8, 0.2;
  0.2, 0.8 ];
Prio = [ 0.6, 0.4 ]';
Sens = [ 0.02, 0.21;
 0.18, 0.49;
 0.08, 0.09;
 0.72, 0.21 ]';

% 1=yes+red, 2=yes+not red,  3=no+red, 4=no+not red
Dat = [ 4, 2, 1 ];

newhmm = HMM(Prio, Trans, Sens);
newhmm = newhmm.forward(Dat);
newhmm = newhmm.backward(Dat);

disp('Forward:');
disp(newhmm.forwardMessages);
disp('Backward:');
disp(newhmm.backwardMessages);
```

2.3 Implementation of HMM

- Forward:

```
0.8372   0.4643   0.0804
0.1628   0.5357   0.9196
```

- Backward:

```
0.5325   0.2661   0.2522   1.0000
0.4675   0.7339   0.7478   1.0000
```

3 Exercise 3

3.1 Umbrella

- By calculating the likelihood of the models correctness and the one with the highest likelihood is the most reliable model:
  - $0.7 \cdot 0.7 \cdot 0.7 \cdot 0.3 \cdot 0.7 \cdot 0.7 \cdot 0.3 \cdot 0.3 \cdot 0.7 = 0.003176523$
  - $0.6 \cdot 0.6 \cdot 0.6 \cdot 0.4 \cdot 0.8 \cdot 0.8 \cdot 0.2 \cdot 0.4 \cdot 0.8 = 0.003538944$

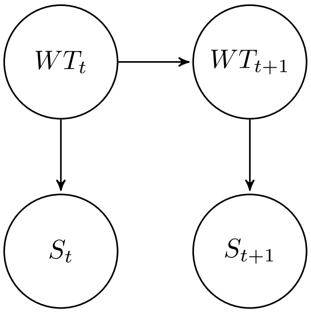
- Matlab

```
function p = SSP(obj, sequence)
  p = 1;
  for t=2:length(sequence),
    transition = obj.transModel(sequence(t-1),sequence(t));
    p = p * transition;
  end
end
```

- MATLAB gave us the same results as the manual calculations of the likelihood.

3.2 Water tank

- Kalman Filter



1. $WT_{t+1} = \mathcal{N}(WT_t, 1)$
2. $S_t = \mathcal{N}(WT_t, 1.5)$

- Filtered estimates:

4 Exercise 4

4.1 Incomplete observations

- $P(R_0) = (0.5, 0.5)$
- $P(R_t \mid R_{t-1} = t) = (0.7, 0.3)$
- $P(R_t \mid R_{t-1} = f) = (0.3, 0, 7)$
- $P(U_t \mid R_t = t) = (0.9, 0.1)$
- $P(U_t \mid R_t = f) = (0.2, 0.8)$

$S_1 = U \;\; \neg U \;\; U$
$S_2 = U \;\; \neg U \;\; \neg U$
Forward og backward hvor C er normaliseringskonstant:

- $P(R_0 = i \mid S) = \alpha_i(i)\beta_i(i) \cdot C$
- $P(R_0 \mid S_1) = (0.4827, 0.0745) \cdot C$
- $P(R_0 \mid S_2) = (0.4692, 0.0775) \cdot C$
- $P(R_0 \mid S) = (0.9519, 0.1520) \cdot C = (0.8623, 0.1377)$

Næste skridt

- $P(R_{t-1}, R_t \mid S) = \alpha(t-1)P(R_t \mid R_{t-1})P(U_T \mid R_t)\beta(t)$
- $P(R_1, R_2 \mid S_1) = \alpha(1)P(R_2 \mid R_1)P(U_2 \mid R_2)\beta(2)$

$forward_0 \cdot backward_1 \cdot Trans \cdot Sensor$

- 0,8182 $\cdot$ 0.3695 $\cdot$ 0.7 $\cdot$ 0.1 = 0.0212
- 0,8182 $\cdot$ 0.6305 $\cdot$ 0.3 $\cdot$ 0.8 = 0.1256
- 0,1818 $\cdot$ 0.3695 $\cdot$ 0.3 $\cdot$ 0.1 = 0.0020
- 0,1818 $\cdot$ 0.6395 $\cdot$ 0.7 $\cdot$ 0.8 = 0.0642

$forward_1 \cdot backward_2 \cdot Trans \cdot Sensor$

- 0.1738 $\cdot$ 0.6273 $\cdot$ 0.7 $\cdot$ 0.9 = 0.0687
- 0,1738 $\cdot$ 0.3737 $\cdot$ 0.3 $\cdot$ 0.2 = 0.0039
- 0,8268 $\cdot$ 0.6273 $\cdot$ 0.3 $\cdot$ 0.9 = 0.1400
- 0,8268 $\cdot$ 0.3737 $\cdot$ 0.7 $\cdot$ 0.2 = 0.0433

New trans model

- $P(R_t = T \mid R_{t-1} = T) = 0.0212 + 0.0687 = 0.0899$
- $P(R_t = F \mid R_{t-1} = T) = 0.0687 + 0.0039 = 0.1295$
- $P(R_t = T \mid R_{t-1} = F) = 0.0020 + 0.1400 = 0.1420$
- $P(R_t = F \mid R_{t-1} = F) = 0.0642 + 0.0433 = 0.1075$

	T	F
T	0.0899	0.1295
F	0.1420	0.1075

New Sensor model
Umbrella is true:

- $P(R_1 \mid S_{1_{top}}) = 0.8182 \cdot 0.5900 = 0.4827$
- $P(R_1 \mid S_{1_{bot}}) = 0.1818 \cdot 0.4100 = 0.0745$
- $P(R_3 \mid S_{1_{top}}) = 0.7251 \cdot 0.6273 = 0.4549$
- $P(R_3 \mid S_{1_{bot}}) = 0.2749 \cdot 0.3727 = 0.2702$
- $P(R_1 \mid S_{2_{top}}) = 0.8182 \cdot 0.5735 = 0.4692$
- $P(R_1 \mid S_{2_{bot}}) = 0.1818 \cdot 0.4264 = 0.0775$

Summation of top and bottom respectively:

- 0.4827 + 0.4549 + 0.4692 = 1.4068
- 0.0745 + 0.2702 + 0.0775 = 0.4222

Umbrella is false:

- $P(R_2 \mid S_{1_{top}}) = 0.1738 \cdot 0.3695 = 0.0642$

- $P(R_2 \mid S_{1_{bot}}) = 0.8262 \cdot 0.6305 = 0.5209$
- $P(R_2 \mid S_{2_{top}}) = 0.1738 \cdot 0.3247 = 0.0564$
- $P(R_2 \mid S_{2_{bot}}) = 0.8262 \cdot 0.6753 = 0.5579$
- $P(R_3 \mid S_{2_{top}}) = 0.0683 \cdot 0.3444 = 0.0235$
- $P(R_3 \mid S_{2_{bot}}) = 0.9317 \cdot 0.6556 = 0.6108$

Summation of top and bottom respectively:

- 0.0642 + 0.0564 + 0.0235 = 0.1441
- 0.5209 + 0.5579 + 0.6108 = 1.6896

Normalisation of table:

	T	F
T	0.9071	0.0929
F	0.1999	0.8001

5 Exercise 5

5.1 Slide 19 example

The formula: $\gamma = 0.9$

- $U(x, y) = R(x, y) + \gamma \cdot max_{a \in \{left,right,up,down\}} \sum_{s' \in S} P(s' \mid a, s)U^2(s')$
- U^3(2,1) = -0.1 + 0.9 · max(0.7 · -0.63 + 0.1 · -5.17 + 0.1 · -0.19 + 0.1 · -0.19, 0.7 · -5.17 + 0.1 · -0.63 + 0.1 · -0.19 + 0.1 · -0.19, 0.7 · -0.19 + 0.1 · -0.63 + 0.1 · -5.17 + 0.1 · -0.19, 0.7 · -0.19 + 0.1 · -0.63 + 0.1 · -5.17 + 0.1 · -0.19)
- U^3(2,1) = -0.1 + 0.9 · max(-0.996,-3.72,-0.732,-0.732)
- U^3(2,1) = -0.7588

	1	2	3
1	3.42	6.23	10
2	-0.76	-1.07	5.24
3	-0.35	-0.77	2.79

5.2 Policy iteration