

DM828 - Introduction to Artificial Intelligence

Exercise Sheet 3, Autumn 2011 [pdf format]

Prepare the following exercises for discussion in class on Thursday, November 24.

Exercises

1. Exercise 14.14 of the text book.
2. Exercise 14.15 of the text book.
3. Exercise 14.17 of the text book.
4. Exercise 15.2 of the text book.
5. This example is a slightly modified version of one taken from Wikipedia. Consider two friends, Alice and Bob, who live far apart from each other and who talk together daily over the telephone about what they did that day. Bob is only interested in two activities: walking in the park or staying at home and reading. The choice of what to do is determined exclusively by the weather on a given day. Alice has no definite information about the weather where Bob lives, but she knows general trends. Based on what Bob tells her he did each day, Alice tries to guess what the weather must have been like.

Alice believes that the weather of a day depends only on the weather of the previous day. There are three states, "Rainy", "Sunny" and "Foggy", but she cannot observe them directly. On each day, there is a certain chance that Bob will perform one of the following activities, depending on the weather: "walk" or "read". Bob tells Alice about his activities.

```
states = ['Rainy', 'Sunny', 'Foggy']

observations = ['walk', 'read', 'walk']

start_probability = {'Rainy': 0.5, 'Sunny': 0.4, 'Foggy': 0.1}

transition_probability = {
    'Rainy' : {'Rainy': 0.6, 'Sunny': 0.2, 'Foggy': 0.2},
    'Sunny' : {'Rainy': 0.05, 'Sunny': 0.8, 'Foggy': 0.15},
    'Foggy' : {'Rainy': 0.3, 'Sunny': 0.2, 'Foggy': 0.5},
}

emission_probability = {
    'Rainy' : {'walk': 0.2, 'read': 0.8},
    'Sunny' : {'walk': 0.9, 'read': 0.1},
    'Foggy' : {'walk': 0.7, 'read': 0.3},
}
```

```
}

```

Derive manually the computations for answering the following questions.

- Suppose that where Bob lives one day the weather was sunny. The next day, Bob tells Alice that he read. What is the probability that the next day it was rainy?
- Suppose that where Bob lives one day it is foggy (and he walked); Bob tells Alice that the successive day (day 2) he read, but that he went walking on day 3. What is the probability that it is foggy on day 3?
- Alice talks to Bob three days in a row and discovers that on the first day he went for a walk, on the second day he read, and on the third day he walked again. Alice has a question: what is the most likely sequence of rainy/foggy/sunny days that would explain these observations?

Check the correctness of your answer to the second and third questions running on the appropriate data the Forward and the Viterbi algorithm implemented below.

```
from numpy import *

aFirst = array([0.5,0.4,0.1])
a = array([[.6,.2,.2],[.05,.8,.15],[.3,.2,.5]]) ## trans
b = array([[.2,.8],[.9,.1],[.7,.3]]) ## emit
obs = array([0,1,0])

def Forward(start_p,trans_p,emit_p,obs):

    nStates = shape(b)[0]
    T = shape(obs)[0]

    alpha = zeros((nStates,T))

    alpha[:,0] = start_p*emit_p[:,obs[0]]

    for t in range(1,T):
        for s in range(nStates):
            ## elapse time
            alpha[s,t] = sum(alpha[:,t-1] * trans_p[:,s])
            ## observe
            alpha[s,t] = alpha[s,t] * emit_p[s,obs[t]]

    return alpha

def ViterbiSimple(start_p,trans_p,emit_p,obs):

    nStates = shape(b)[0]
    T = shape(obs)[0]

    path = zeros(T)
    m = zeros((nStates,T))
```

```

m[:,0] = start_p * emit_p[:,obs[0]]
path[0] = argmax(m[:,0])

for t in range(1,T):
    for s in range(nStates):
        m[s,t] = max( emit_p[s,obs[t]] * trans_p[:,s] * m[:,t-1] )
    path[t] = argmax(m[:,t])

    print m
print "Path:␣", path
#print m[path[T-1]]
return path,m

path,m = ViterbiSimple(aFirst,a,b,obs)
print [states[int(x)] for x in path]

```

Solution of 14.14.

a

$$\begin{aligned}
 P(B|j,m) &= \alpha P(B) \sum_e P(e) \sum_a P(a|b,e) P(j|a) P(m|a) \\
 &= \alpha P(B) \sum_e P(e) \left[.97 \begin{bmatrix} .95 & .29 \\ .94 & .001 \end{bmatrix} + .05.01 \begin{bmatrix} .05 & .71 \\ .06 & .999 \end{bmatrix} \right] \\
 &= \alpha P(B) \sum_e P(e) \begin{bmatrix} .598525 & .183055 \\ .59223 & .0011295 \end{bmatrix} \\
 &= \alpha P(B) \left[.002 \begin{bmatrix} .598525 \\ .183055 \end{bmatrix} + .998 \begin{bmatrix} .59223 \\ .0011295 \end{bmatrix} \right] \\
 &= \alpha \begin{bmatrix} .001 \\ .999 \end{bmatrix} \begin{bmatrix} .59224259 \\ .001493351 \end{bmatrix} \\
 &= \alpha \begin{bmatrix} .00059224259 \\ .0014918576 \end{bmatrix} \approx \langle .284, .716 \rangle
 \end{aligned}$$

b Including the normalization step, there are 7 additions, 16 multiplications, and 2 divisions. The enumeration algorithm has two extra multiplications.

c To compute $P(X_1|X_n = \text{true})$ using enumeration, we have to evaluate two complete binary trees (one for each value of X_1), each of depth $n - 2$, so the total work is $O(2n)$. Using variable elimination, the factors never grow beyond two variables. For example, the first step is

$$\begin{aligned}
 P(X_1|X_n = \text{true}) &= \alpha P(X_1) \dots \sum_{x_{n-2}} P(x_{n-2}|x_{n-3}) \sum_{x_{n-1}} P(x_{n-1}|x_{n-2}) P(X_n = \text{true}|x_{n-1}) \\
 &= \alpha P(X_1) \dots \sum_{x_{n-2}} P(x_{n-2}|x_{n-3}) \sum_{x_{n-1}} f_{X_{n-1}}(x_{n-1}, x_{n-2}) f_{X_n}(x_{n-1}) \\
 &= \alpha P(X_1) \dots \sum_{x_{n-2}} P(x_{n-2}|x_{n-3}) f_{X_{n-1}X_n}(x_{n-2})
 \end{aligned}$$

The last line is isomorphic to the problem with $n - 1$ variables instead of n ; the work done on the first step is a constant independent of n , hence (by induction on n , if you want to be formal) the total work is $O(n)$.

d Here we can perform an induction on the number of nodes in the polytree. The base case is trivial. For the inductive hypothesis, assume that any polytree with n nodes can be evaluated in time proportional to the size of the polytree (i.e., the sum of the CPT sizes). Now, consider a polytree with $n + 1$ nodes. Any node ordering consistent with the topology will eliminate first some leaf node from this polytree. To eliminate any leaf node, we have to do work proportional to the size of its CPT. Then, because the network is a polytree, we are left with independent subproblems, one for each parent. Each subproblem takes total work proportional to the sum of its CPT sizes, so the total work for $n + 1$ nodes is proportional to the sum of CPT sizes.