Machine Learning 521289S
Spring 2019
Exercise 3: Bayesian Networks
Exercises 2 and 3 give exam points!

Exercises

1. *Bayes Nets*

   Take a look a the Bayes net presented in Figure 1 and se also the revised Section 2.11[1] of the course book (or alternatively the page 51 in Chapter 2 of the old lecture notes, in Finnish). Using this Bayes net:

   (a) Write out some of the conditional independence assumptions explicit in Figure 1, i.e. simplify $P(C|X, A)$ and so on.

   (b) Calculate the probability that a dark and thin sea bass will be caught in the summer in the north Atlantic, i.e. $P(a_3, b_1, x_2, c_3, d_2)$.

   (c) Find out what is the a priori probability of catching a salmon. In other words, calculate $P(X = x_1)$. What does this tell us about the a priori probability of catching a sea bass?

   (d) How does the result of the part (c) change when it is known that the fish will be caught at the south Atlantic (b_2)?

   (e) What would the result of (d) be if it was *also* known that it is fall (a_4)?

i	1	2	3	4
$P(a_i)$	0.25	0.25	0.25	0.25

i	1	2
$P(b_i)$	0.6	0.4

i	1	2	3
$P(c_i\|x_1)$	0.6	0.2	0.2
$P(c_i\|x_2)$	0.2	0.3	0.5

i	1	2
$P(d_i\|x_1)$	0.3	0.7
$P(d_i\|x_2)$	0.6	0.4

i, j	$P(x_1\|a_i, b_j)$	$P(x_2\|a_i, b_j)$
1, 1	0.5	0.5
1, 2	0.7	0.3
2, 1	0.6	0.4
2, 2	0.8	0.2
3, 1	0.4	0.6
3, 2	0.1	0.9
4, 1	0.2	0.8
4, 2	0.3	0.7

[1] `ftp://ftp.wiley.com/public/sci_tech_med/pattern/DHS2.11Revised.pdf`

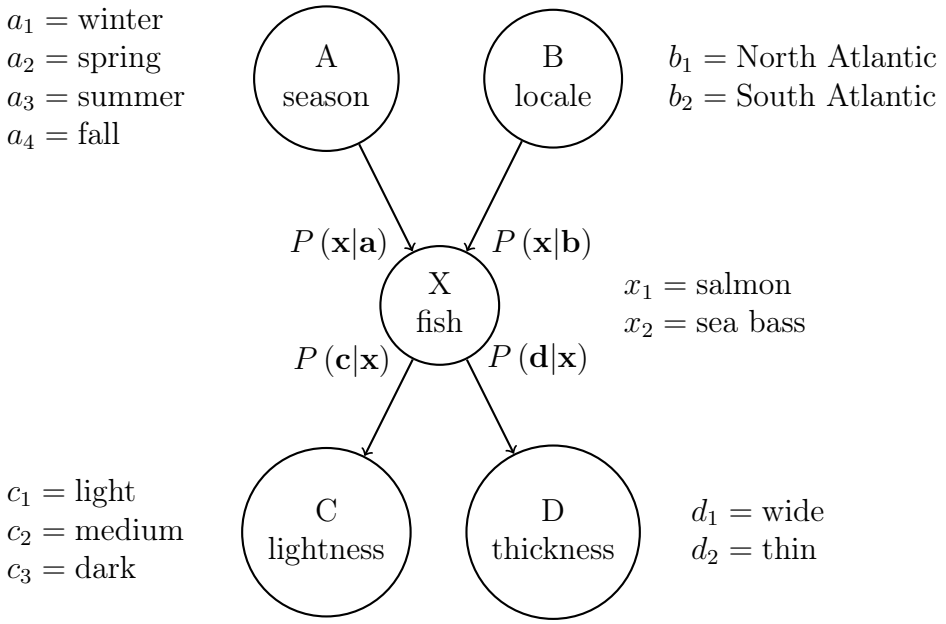


Figure 1: A belief network for fish. Adapted from the Duda, Hart & Stork, "Pattern classification," 2001.

2. *More Bayes Nets*

   Consider the Bayesian network in Figure 2 and the associated data in the tables on the next page. Solve the probability that

   (a) it is cloudy (without further evidence),

   (b) it does not rain given that it is cloudy,

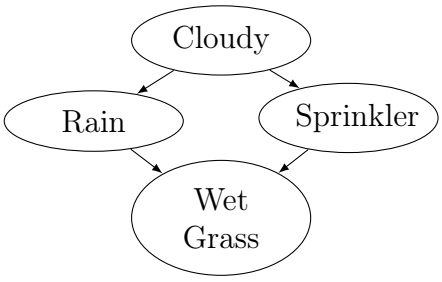
   (c) the grass is wet (without further evidence),

   (d) the sprinkler was on with the evidence that the grass is wet,

   (e) it rains given that the sprinkler is on.

Cloudy	
P(C=F)	P(C=T)
0.5	0.5

	Rain	
Cloudy	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8



	Sprinkler	
Cloudy	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

		Wet Grass	
Sprinkler	Rain	P(W=F)	P(W=T)
F	F	1.0	0.0
F	T	0.1	0.9
T	F	0.1	0.9
T	T	0.01	0.99

Figure 2: A Bayes Net. Adapted from the Bayesian Network Toolbox for Matlab documentation, originally Russell and Norvig, "Artificial Intelligence: a Modern Approach," 1995.

3. *Naive Bayes Nets*

   Use the following Table 1 and a **naive Bayes classifier** to classify the instance (Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong) to decide whether to play tennis or not. (Hint: The MLEs for the probabilities are their relative frequencies over the data.)

Table 1: Table contains information for fourteen days. Reproduced by kind permission of the author from "Machine Learning", Tom Mitchell, McGraw Hill, 1997.

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*Play Tennis*
1	Sunny	Hot	High	Weak	**No**
2	Sunny	Hot	High	Strong	**No**
3	Overcast	Hot	High	Weak	**Yes**
4	Rain	Mild	High	Weak	**Yes**
5	Rain	Cool	Normal	Weak	**Yes**
6	Rain	Cool	Normal	Strong	**No**
7	Overcast	Cool	Normal	Strong	**Yes**
8	Sunny	Mild	High	Weak	**No**
9	Sunny	Cool	Normal	Weak	**Yes**
10	Rain	Mild	Normal	Weak	**Yes**
11	Sunny	Mild	Normal	Strong	**Yes**
12	Overcast	Mild	High	Strong	**Yes**
13	Overcast	Hot	Normal	Weak	**Yes**
14	Rain	Mild	High	Strong	**No**

4. *Hidden Markov Models*

Hidden Markov Models (HMMs) are **a special case of Bayes nets** having the form shown in Figure 3. The basic idea behind the HMMs is that they model some sort of an underlying process that cannot be directly observed. At each (time) step, the only thing we can observe/measure are the values of the nodes at the output layer. It is assumed that each output node is directly affected by the internal state at that time but nothing else. We are then interested in what is happening behind the scenes, i.e. at the hidden nodes. To make the problem more tractable, a simplification called the *Markov assumption* is made assuming that the internal state nodes only depend on the previous one. Although the assumption is quite an oversimplification in most real cases, it has often been proven to be useful and to lead to systems with high accuracy in practice.

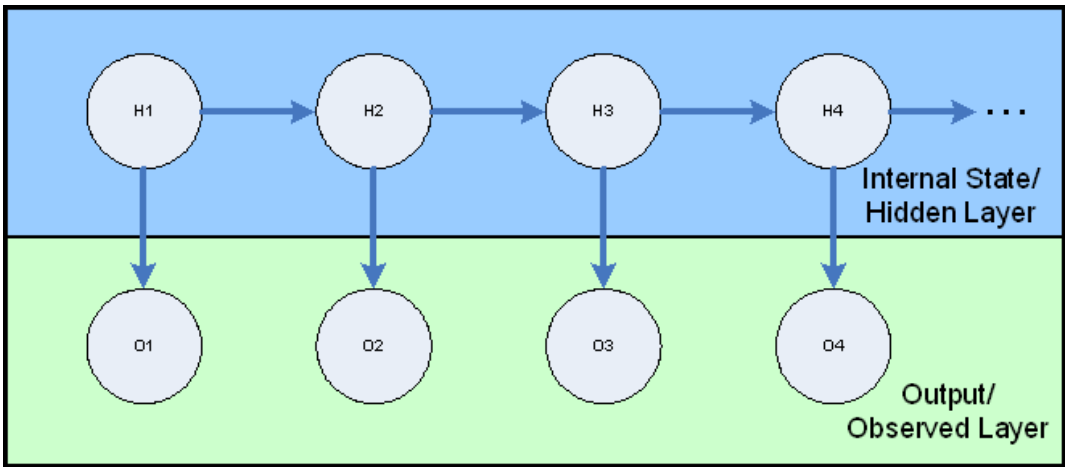


Figure 3: The structure of a HMM.

The HMMs are widely used in different fields. For example, they can be used in speech recognition to decide what are the subsequent phonemes the speaker has spoken when the given waveforms are observed with a microphone. In telecommunications, one observes modulation symbols that are covered in noise and tries to decide what was the actual bit sequence that was transmitted. In natural language processing (NLP), one might be interested in what are the correct parts-of-speech of words the user has typed in, i.e. one parses the sentence to uncover the grammatical structure behind the words. In yet another NLP scenario, a user has made an typing error (observed nodes are typed characters), and we try to decide what is the correct word (hidden nodes are the correct characters of the word the user intended to write). In this case, the vertical arrows relate to the probabilities $P(O_i|H_i)$ which describe how probable it is to mistype the character O_i in place of the character H_i. (This, of course, depends on the keyboard layout, the pronunciational and/or visual similarities between characters etc.) Horizontal arrows then describe the probabilities $P(H_{i+1}|H_i)$, i.e. how likely it is that the character H_i is followed by the character H_{i+1} in a correctly written word.

(a) The general approach to any classification task is that we classify each instance to the class that gives the highest posterior probability according to the Bayes

decision rule. In order to do this, calculate the (simplified symbolic formula for) probability of a supposed hidden sequence ($H_1 = h_1$, $H_2 = h_2$, ..., $H_N = h_n$) when the sequence ($O_1 = o_1$, $O_2 = o_2$, ..., $O_N = o_n$) has been observed by examining the dependencies (or lack thereof) depicted in Figure 3.

(b) To find the most probable (a posteriori) classification, the probability of each possible hidden state sequence has to be evaluated. Would it be feasible to try to find the correct sequence using this type of a brute force approach, i.e. trying out all the possible hidden node combinations and selecting the one with largest probability? How many combinations there would be in a HMM with 15 steps assuming that each hidden node could have 5 states on average; and if we could calculate 1000 posteriors per second, how long it would take? Is there a better way to find the best combination?

Answers:
1. a) Find a simplified formula for $P(A, B, X, C, D)$ by using conditional independence assumptions. Use conditional probability. b) Use the simplified formula from (a). $0.018 \approx 0.02$ c) 0.445 and 0.555 d) 0.475 and 0.525 e) 0.3 and 0.7
2. a) 0.5 b) 0.2 c) 0.6471 d) $0.42976 \approx 0.43$ e) 0.30
3. 0.2 vs. 0.8 (0.0053 vs. 0.0206)
4. a) $P(h_1, h_2, \ldots, h_N | o_1, o_2, \ldots, o_N) = \alpha\ P(h_1)\ P(o_1|h_1) \prod_{i=2}^{N} P(h_i|h_{i-1})\ P(o_i|h_i)$, where $\alpha = P(o_1, o_2, \ldots, o_N)^{-1}$. b) 0.967 71 *a*