Examination in Machine Intelligence

Thomas Dyhre Nielsen

January 6, 2017

On the next six pages you will find six questions covering different aspects of the course. The questions differ in their level of difficulty, and for each correctly answered question you will get a certain amount of points as indicated by each question. When solving the questions you are allowed to use all available material such as books, pocket calculator, etc., however, laptops/tablets and other networking devices are *not* allowed.

Before you answer a question make sure that you have read the question carefully. Moreover, make sure that you argue for your answers (e.g. include intermediate results) so that it is possible to follow your line of thought. Finally, it is important that your solutions are presented in a readable form. The answers to the questions should be written in English.

In addition to the six pages with questions, you are also provided with 10 pages that you can use when writing your answers to the questions.

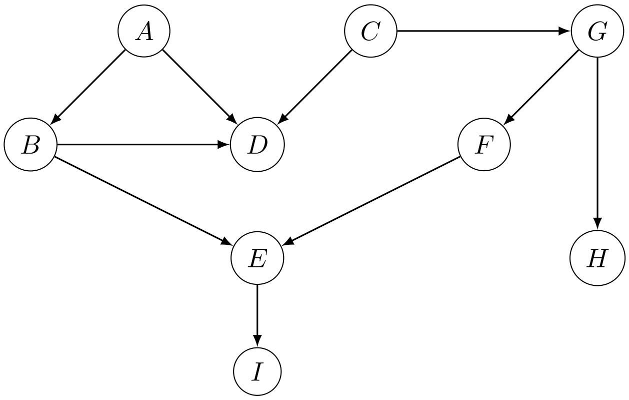
- Two of the pages contain the game tree found in Question 6. You can use these game trees when answering the question.
- For each sheet of paper containing your response to the questions, please include your name, study number, current page number, and the total number of pages.
- If you need more paper, simply raise your hand to contact one of the guards in the examination room.

Good luck with the questions

Thomas Dyhre Nielsen

Question 1 - 10 points

Consider the graph below:



1. List all probability distributions (on the form $P(X|Y_1, \ldots, Y_n)$) that should be specified in order to obtain a Bayesian network from the graph?
2. Which variables are d-separated from A?
3. Which variables are d-separated from A given hard evidence on I and F?
4. Which variables are d-separated from I given hard evidence on B and F?

Solution:

1. $P(A)$, $P(B|A)$, $P(C)$, $P(D|A, B, C)$, $P(E|B, F)$, $P(F|G)$, , $P(G|C)$, $P(H|G)$, $P(I|E)$
2. $\{C, F, G, H\}$
3. $\{C, G, H\}$
4. $\{A, C, D, G, H\}$

Question 2 - 20 points

Consider the three variables A, B, and C with state spaces $sp(A) = \{a_1, a_2\}$, $sp(B) = \{b_1, b_2, b_3\}$, and $sp(C) = \{c_1, c_2\}$. Let the joint probability distribution over the three variables be defined by the probability distributions $P(A)$, $P(B|A)$, and $P(C|B)$:

$P(A) =$

A	
a_1	a_2
0.3	0.7

$P(B|A) =$

		A	
		a_1	a_2
B	b_1	0.4	0.2
	b_2	0.5	0.6
	b_3	0.1	0.2

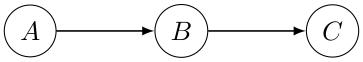
$P(C|B) =$

		B		
		b_1	b_2	b_3
C	c_1	0.7	0.4	0.1
	c_2	0.3	0.6	0.9

1. Show the structure of the Bayesian network representation for the variables A, B, and C.
2. Calculate the probability table $P(A, B, C = c_1)$.
3. Calculate the probability table $P(A, C = c_1)$.
4. Calculate the conditional probability distribution $P(A|C = c_1)$.
5. Is the variable A conditionally independent of C given B (explain why)?

Solution:

Sub-problem 1



Sub-problem 2

$$
\begin{aligned}
P(A, B, C = c_1) &= P(A)P(B|A)P(C = c_1|B) \\
&= \begin{array}{|c|ccc|}\hline
 & & \multicolumn{2}{c|}{\text{A}} \\
 & & a_1 & a_2 \\ \hline
 & b_1 & 0.084 & 0.098 \\
\text{B} & b_2 & 0.06 & 0.168 \\
 & b_3 & 0.003 & 0.014 \\ \hline
\end{array}
\end{aligned}
$$

Sub-problem 3

$$
\begin{aligned}
P(A, C = c_1) &= \sum_B P(A, B, C = c_1) \\
&= \sum_B \left(\begin{array}{|c|ccc|}\hline
 & & \multicolumn{2}{c|}{\text{A}} \\
 & & a_1 & a_2 \\ \hline
 & b_1 & 0.084 & 0.098 \\
\text{B} & b_2 & 0.06 & 0.168 \\
 & b_3 & 0.003 & 0.014 \\ \hline
\end{array} \right) \\
&= (0.147, 0.28)
\end{aligned}
$$

Sub-problem 4

$$
\begin{aligned}
P(A|C = c_1) &= \frac{P(A, C = c_1)}{P(C = c_1)} = \frac{P(A, C = c_1)}{\sum_A P(A, C = c_1)} \\
&= \frac{(0.147, 0.28)}{0.147 + 0.28} = (0.34, 0.66)
\end{aligned}
$$

Sub-problem 5

Yes. Check, e.g., the d-separation properties of the Bayesian network representation.

Question 3 - 20 points

The steel company *Constraint inc.* needs to schedule the starting times for its next four production jobs (labeled J_1, J_2, J_3, and J_4). The possible starting times for the jobs are divided into seven time slots: $\{1, 2, 3, 4, 5, 6, 7\}$. The jobs are, however, interdependent, and these dependencies impose the following constraints on when the jobs can start:

1. J_2 must start at least two time slots after J_1
2. J_3 must start before or at the same time as J_1.
3. J_4 must start after J_2 and J_3.

Furthermore, in order finish on time, J_4 should start before time 6.

1. Model the problem as a constraint satisfaction problem, i.e., identify the variables, their domains, and the constraints.
2. Represent the problem as a constraint network.
3. Make the network arc-consistent.
4. Find a satisfying solution to the problem (if one exists) using variable elimination with the elimination ordering J_4, J_2, J_1, J_3.

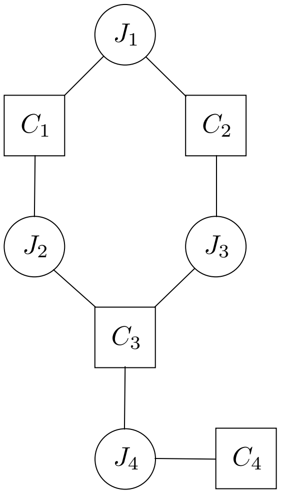
Solution:

Sub-problem 1 and 2

There is one variable for each job, and the state space of each of these variables corresponds to the time slots $\{1, 2, 3, 4, 5, 6, 7\}$. The constraints for the problem can be specified as:

1. C_1: $J_2 \geq J_1 + 2$
2. C_2: $J_3 \leq J_1$
3. C_3: $J_4 > max(J_2, J_3)$
4. C_4: $J_4 < 6$

The constraint network for this problem is shown in the figure below. Note that an equivalent representation could have been obtained by encoding C_3 using the two constraints, $C_3^1 : J_4 > J_2$ and $C_3^2 : J_4 > J_3$.



Sub-problem 3

After making the network arc-consistent we end up with the following domains:

- $dom(J_1) = \{1, 2\}$
- $dom(J_2) = \{3, 4\}$
- $dom(J_3) = \{1, 2\}$
- $dom(J_4) = \{4, 5\}$

Sub-problem 4

Eliminating J_4

$$C_5 = C_3^{\downarrow J_2, J_3} = \left(\begin{array}{c|c|c} J_2 & J_3 & J_4 \\ \hline 3 & 1 & 4 \\ 3 & 2 & 4 \\ 3 & 1 & 5 \\ 4 & 1 & 5 \\ 4 & 2 & 5 \end{array} \right)^{\downarrow J_2, J_3} = \begin{array}{c|c} J_2 & J_3 \\ \hline 3 & 1 \\ 3 & 2 \\ 4 & 1 \\ 4 & 2 \end{array}$$

Eliminating J_2

First we join the constraints with J_2 in the domain:

$$C_6' = C_1 \bowtie C_5 = \left(\begin{array}{c|c} J_1 & J_2 \\ \hline 1 & 3 \\ 2 & 4 \end{array}\right) \bowtie \left(\begin{array}{c|c} J_2 & J_3 \\ \hline 3 & 1 \\ 3 & 2 \\ 4 & 1 \\ 4 & 2 \end{array}\right) = \begin{array}{c|c|c} J_1 & J_2 & J_3 \\ \hline 1 & 3 & 1 \\ 1 & 3 & 2 \\ 2 & 4 & 1 \\ 2 & 4 & 2 \end{array}$$

Next we eliminate J_2:

$$C_6 = C_6'^{\downarrow J_1, J_3} = \left(\begin{array}{c|c|c} J_1 & J_2 & J_3 \\ \hline 1 & 3 & 1 \\ 1 & 3 & 2 \\ 2 & 4 & 1 \\ 2 & 4 & 2 \end{array}\right)^{\downarrow J_1, J_3} = \begin{array}{c|c} J_1 & J_3 \\ \hline 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{array}$$

Eliminating J_1

First we join the constraints with J_1 in the domain:

$$C_7' = C_2 \bowtie C_6 = \left(\begin{array}{c|c} J_1 & J_3 \\ \hline 1 & 1 \\ 2 & 1 \\ 2 & 2 \end{array}\right) \bowtie \left(\begin{array}{c|c} J_1 & J_3 \\ \hline 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{array}\right) = \begin{array}{c|c} J_1 & J_3 \\ \hline 1 & 1 \\ 2 & 1 \\ 2 & 2 \end{array}$$

Projecting down to J_3 gives:

$$C_7 = \begin{array}{c} J_3 \\ \hline 1 \\ 2 \end{array}$$

By backtracking we find that there are multiple solutions to the problem (only subset shown here):

J_1	J_2	J_3	J_4
1	3	1	4
1	3	1	5
2	4	1	5
2	4	2	5

Question 4 - 15 points

The book store *Smart books* wants to make book recommendations for its online customers. The store has recorded customer feedback (*Likes* with the states *yes* and *no*) for prior book purchases. For these purchases, the book store has recorded the following information about the books: The genre of the book (*Genre* with states *action*, *biography*, and *romance*), the format of the book (*Format* with states *paperback* and *hardcover*), and the length of the book (*Length* with states *long* and *short*). The data that has been collected is shown in the table below.

	Attributes			Target
	Genre	*Format*	*Length*	*Likes*
1	*bio*	*paperback*	*long*	*no*
2	*bio*	*paperback*	*short*	*yes*
3	*action*	*hardcover*	*long*	*yes*
4	*romance*	*paperback*	*long*	*yes*
5	*romance*	*hardcover*	*short*	*yes*
6	*romance*	*hardcover*	*long*	*no*

1. Calculate the entropy of the attribute *Genre*.[1]
2. Show the decision/classification tree that would be learned by the decision tree algorithm assuming that it is given the training examples above and uses information gain for selecting the attributes.
3. Show the value of the information gain for each candidate attribute at each step in the construction of the tree.

Solution:

Sub-problem 1

$$Ent(Genre) = -\frac{2}{6}\log_2(2/6) - \frac{1}{6}\log_2(1/6) - \frac{3}{6}\log_2(3/6) = 1.459$$

Sub-problem 2 and 3

First we calculate the expected entropy of for the three attributes:

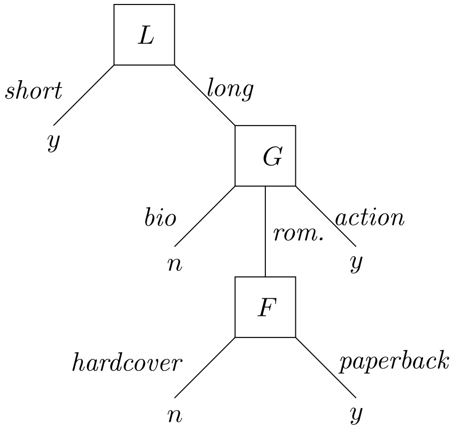
$$\begin{aligned} E\text{-}Ent(G) &= 2/6 \cdot 1 + 1/6 \cdot 0 + 3/6 \cdot 0.918 = 0.79 \\ E\text{-}Ent(F) &= 3/6 \cdot 0.918 + 3/6 \cdot 0.918 = 0.918 \\ E\text{-}Ent(L) &= 2/6 \cdot 0 + 4/6 \cdot 1 = 4/6 \end{aligned}$$

[1] Note that $\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$.

Thus, we put *Length* at the root. For *Length=short* there is nothing more to do, but for *Length=long* we need to expand further. Thus we calculate the expected entropy for the remaining two attributes conditional on *Length=long*:

$$\begin{aligned}\textit{E-Ent}(G|\textit{Length} = \textit{long}) &= 1/4 \cdot 0 + 1/4 \cdot 0 + 2/4 \cdot 1 = 1/2\\ \textit{E-Ent}(F|\textit{Length} = \textit{long}) &= 2/4 \cdot 1 + 2/4 \cdot 1 = 1\end{aligned}$$

We pick *Genre* to succeed *Length=long*. If *Genre* is either *bio* or *action*, there is nothing to do. For *Genre=romance*, we need to make a last check wrt. the attribute *Format*. The resulting decision is shown below.



Question 5 - 15 points

Consider the following five data points living in $\mathbb{R}^2$:

d_1	$(10, 4)$
d_2	$(50, 3)$
d_3	$(80, 10)$
d_4	$(90, 11)$
d_5	$(100, 8)$

Using the Euclidean distance to measure distances:

1. Find the data point closest to d_2 in the data set, i.e., $(50, 3)$.
2. Normalize the data using Z-score normalization. Find the data point closest to d_2 using the transformed data set.
3. Let the data points $(10, 4)$ and $(80, 10)$ be initial cluster centers for the k-means algorithm. Perform one more k-means iteration by updating these cluster centers using the non-normalized data set above.

Solution:

Sub-problem 1

The data point $(80, 10)$ is the one closets to $(50, 3)$ with a distance of 30.81.

Sub-problem 2

The normalized data set is given by:

$$(-1.72, -1.00), (-0.49, -1.32), (0.43, 0.88), (0.74, 1.19), (1.04, 0.25)$$

The data point closest to $(-0.49, -1.32)$ (corresponding to $(50, 3)$) is $(-1.72, -1.00)$ (corresponding to $(10, 4)$).

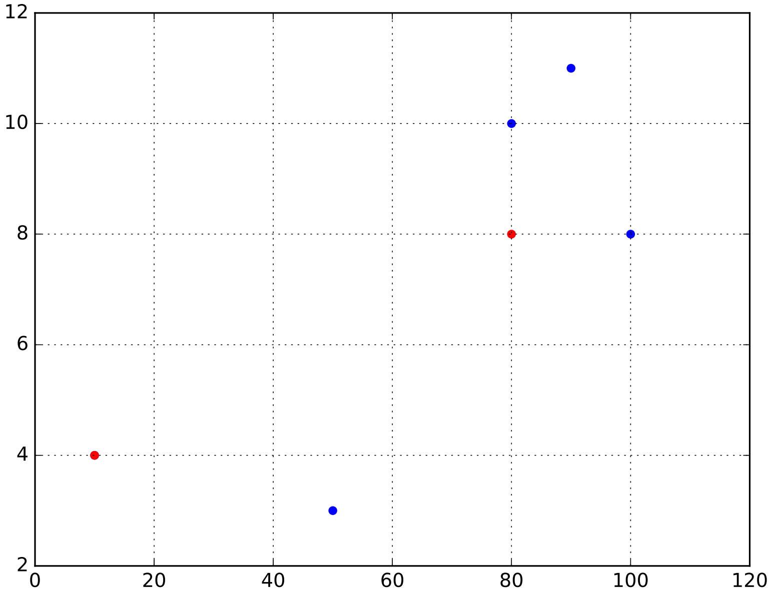
Sub-problem 3

First we find the points that belong to the two clusters defined by the initially chosen cluster centers:

dist	d_2	d_4	d_5
d_1	40.01	80.31	90.09
d_3	30.81	10.05	20.10

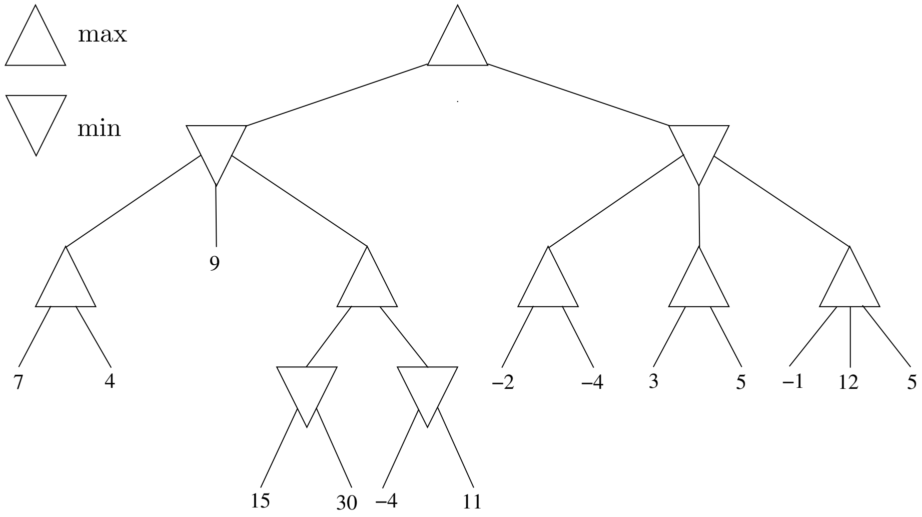
Thus, the data points d_2, d_4, and d_5 all move to the cluster defined by d_3; d_1 now defines a cluster with a single element.

Since no points are assigned to the cluster defined by d_1 we keep that data point as a cluster center. The cluster center defined by the mean of the four remaining points becomes $(80, 8)$.



Question 6 - 20 points

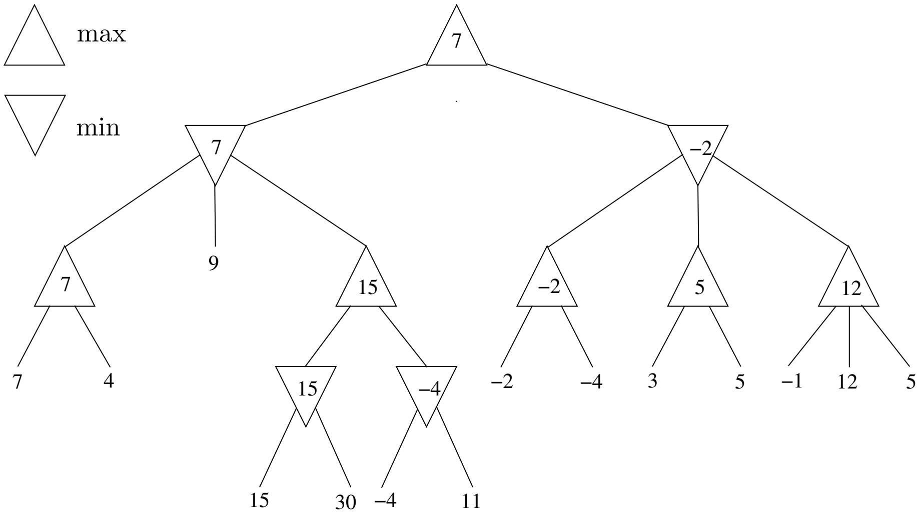
Consider the following zero-sum game tree:



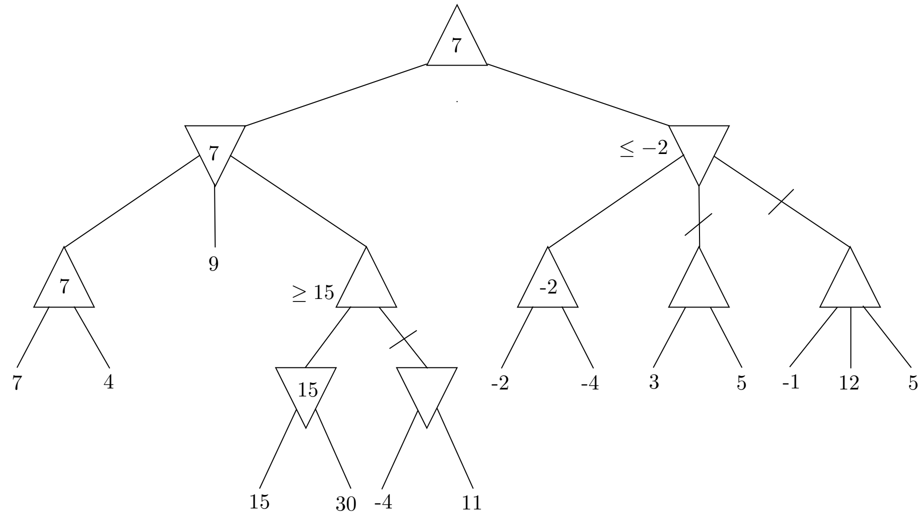
1. Compute the utility values for all the nodes.
2. Assume that the utility values are calculated in a depth-first order that always considers the branches in a left to right order. Mark the nodes that will not be visited when employing $(\alpha - \beta)$ pruning.

Solution:

Sub-problem 1



Sub-problem 2



Examination in Machine Intelligence

Thomas Dyhre Nielsen

January 8, 2018

On the next six pages you will find six questions covering different aspects of the course. The questions differ in their level of difficulty, and for each correctly answered question you will get a certain amount of points as indicated by each question. When solving the questions you are allowed to use all available material such as books, pocket calculator, etc., however, laptops/tablets and other networking devices are *not* allowed.

Before you answer a question make sure that you have read the question carefully. Moreover, make sure that you argue for your answers (e.g. include intermediate results) so that it is possible to follow your line of thought. Finally, it is important that your solutions are presented in a readable form. The answers to the questions should be written in English.

In addition to the six pages with questions, you are also provided with 10 pages that you can use when writing your answers to the questions.

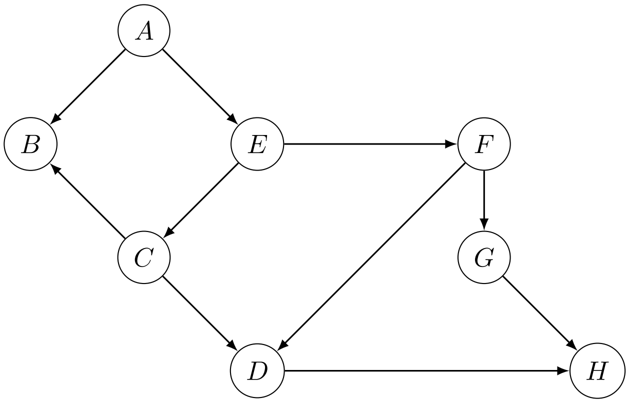
- Two of the pages contain the grid world found in Question 2. You can use these grids when answering this question.
- For each sheet of paper containing your response to the questions, please include your name, study number, current page number, and the total number of pages.
- If you need more paper, simply raise your hand to contact one of the guards in the examination room.

Good luck with the questions

Thomas Dyhre Nielsen

Question 1 - 10 points

Consider the graph below:



1. List all probability distributions (on the form $P(X|Y_1, \ldots, Y_n)$) that should be specified in order to obtain a Bayesian network from the graph?
2. Assume that the variable D has three states labeled $\{a, b, c\}$ and that the remaining variables (i.e., $\{A, B, C, E, F, G, H\}$) all have two states labeled t and f. Give a table representation of a valid conditional probability distribution for variable D.
3. Which variables are d-separated from H given hard evidence on D and E?
4. Which variables are d-separated from A given hard evidence on B and E?
5. Which variables are d-separated from C given hard evidence on E and H?

Solution:

1. $P(A)$, $P(B|A, C)$, $P(C|E)$, $P(D|C, F)$, $P(E|A)$, $P(F|E)$, , $P(G|F)$, $P(H|G, D)$
2. $\ldots$
3. $\{A\}$
4. $\{G, F\}$
5. $\{A\}$

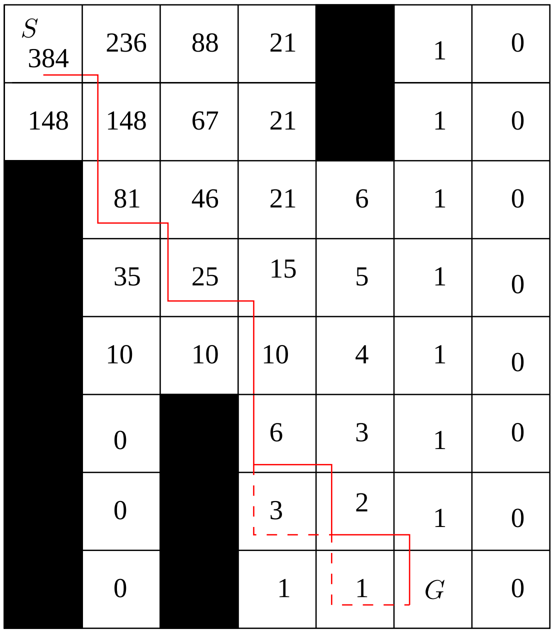
Question 2 - 15 points

Consider a robot that can move in the grid shown below. The robot can only move *right* or *down* and only one step at a time; no step can be made into the shaded areas or outside the grid.

S				■		
				■		
■						
■						
■						
■		■				
■		■				
■		■			G	

1. Use dynamic programming to calculate the number of paths leading to the goal cell G from each of the cells in the grid.
2. Specify the path the robot should take from S to G so that at each step the robot moves to the cell that has the maximum number of paths leading to G.

Solution:



S 384	236	88	21	■	1	0
148	148	67	21	■	1	0
■	81	46	21	6	1	0
■	35	25	15	5	1	0
■	10	10	10	4	1	0
■	0	■	6	3	1	0
■	0	■	3	2	1	0
■	0	■	1	1	G	0

Question 3 - 20 points

The software company *Macrosoft* needs to upgrade four of its key software systems. The software systems (labeled S_1, S_2, S_3, and S_4) should be upgraded over the course of a single day and, to keep things organized, seven possible starting times (labeled 1, 2, 3, 4, 5, 6, and 7) for the software updates have been identified. The software systems are, however, inter-connected, and these connections induce constraints on when the upgrades can start:

1. S_1 should start after S_2.
2. S_3 should start before S_2 and S_4.
3. S_4 should start before time 7.
4. S_4 must start at least two time slots after S_1.

You should:

1. Model the problem as a constraint satisfaction problem, i.e., identify the variables, their domains, and the constraints.
2. Represent the problem as a constraint network.
3. Make the network arc-consistent.
4. Find a satisfying solution to the problem (if one exists) using variable elimination with the elimination ordering S_1, S_2, S_3, S_4.

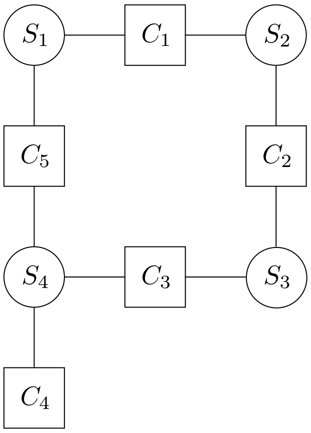
Solution:

Sub-problem 1 and 2

There is one variable for each system, and the state space of each of these variables corresponds to the time slots $\{1,2,3,4,5,6,7\}$. The constraints for the problem can be specified as:

1. C_1: $S_1 > S_2$
2. C_2: $S_3 < S_2$
3. C_3: $S_3 < S_4$
4. C_4: $S_4 < 7$
5. C_5: $S_1 + 2 \leq S_4$

The constraint network for this problem is shown in the figure below.



Sub-problem 3

After making the network arc-consistent we end up with the following domains:

- $dom(S_1) = \{3, 4\}$
- $dom(S_2) = \{2, 3\}$
- $dom(S_3) = \{1, 2\}$
- $dom(S_4) = \{5, 6\}$

Sub-problem 4

Eliminating S_1

$$C_6 = (C_1 \bowtie C_5)^{\downarrow S_2,S_4} = \left(\begin{array}{c|c} S_1 & S_2 \\ \hline 3 & 2 \\ 4 & 2 \\ 4 & 3 \end{array} \bowtie \begin{array}{c|c} S_1 & S_4 \\ \hline 3 & 5 \\ 3 & 6 \\ 4 & 6 \end{array} \right)^{\downarrow S_2,S_4} = \begin{array}{c|c} S_2 & S_4 \\ \hline 2 & 5 \\ 2 & 6 \\ 3 & 6 \end{array}$$

Eliminating S_2

$$C_7 = (C_2 \bowtie C_6)^{\downarrow S_3,S_4} = \left(\begin{array}{c|c} S_2 & S_3 \\ \hline 2 & 1 \\ 3 & 1 \\ 3 & 2 \end{array} \bowtie \begin{array}{c|c} S_2 & S_4 \\ \hline 2 & 5 \\ 2 & 6 \\ 3 & 6 \end{array} \right)^{\downarrow S_3,S_4} = \begin{array}{c|c} S_3 & S_4 \\ \hline 1 & 5 \\ 1 & 6 \\ 2 & 6 \end{array}$$

Eliminating S_3

$$C_8 = (C_3 \bowtie C_7)^{\downarrow S_4} = \left(\begin{array}{c|c} S_3 & S_4 \\ \hline 1 & 5 \\ 1 & 6 \\ 2 & 5 \\ 2 & 6 \end{array} \bowtie \begin{array}{c|c} S_3 & S_4 \\ \hline 1 & 5 \\ 1 & 6 \\ 2 & 6 \end{array} \right)^{\downarrow S_4} = \begin{array}{c} S_4 \\ \hline 5 \\ 6 \end{array}$$

By back tracking we find multiple solutions:

S_1	S_2	S_3	S_4
3	2	1	5
3	2	1	6
4	2	1	6
4	3	1	6
4	3	2	6

Question 4 - 15 points

Consider the variables A, B, and C with state spaces $sp(A) = \{a_1, a_2\}$, $sp(B) = \{b_1, b_2, b_3\}$, and $sp(C) = \{c_1, c_2\}$, respectively. The joint probability distribution over the three variables is defined by the conditional probability distributions assigned to the variables:

$P(A) =$

A	
a_1	a_2
0.6	0.4

$P(B) =$

	B	
b_1	b_2	b_3
0.4	0.5	0.1

$P(C|A, B) =$

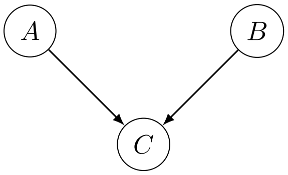
			B	
		b_1	b_2	b_3
A	a_1	$(0.9, 0.1)$	$(0.2, 0.8)$	$(0.4, 0.6)$
	a_2	$(0.7, 0.3)$	$(0.6, 0.4)$	$(0.1, 0.9)$

You should:

1. Show the structure of the Bayesian network representation for the variables A, B, and C.
2. Calculate the probability table $P(A, B, C = c_1)$.
3. Calculate the probability table $P(A, C = c_1)$.
4. Calculate the conditional probability distribution $P(A \,|\, C = c_1)$.
5. Calculate the conditional probability distribution $P(A \,|\, B = b_2, C = c_1)$.
6. Based on the calculated probability distributions, argue whether A is independent of B given $C = c_1$.

Solution:

Sub-problem 1



Sub-problem 2

$$P(A,B,C=c_1) = P(C=c_1 \mid A,B)P(A)P(B)$$

$$= \begin{array}{|cc|ccc|} \hline & & & \text{B} & \\ & & b_1 & b_2 & b_3 \\ \hline \text{A} & a_1 & 0.216 & 0.06 & 0.024 \\ & a_2 & 0.112 & 0.12 & 0.004 \\ \hline \end{array}$$

Sub-problem 3

$$P(A,C=c_1) = \sum_B P(A,B,C=c_1)$$

$$= \sum_B \left(\begin{array}{|cc|ccc|} \hline & & & \text{B} & \\ & & b_1 & b_2 & b_3 \\ \hline \text{A} & a_1 & 0.216 & 0.06 & 0.024 \\ & a_2 & 0.112 & 0.12 & 0.004 \\ \hline \end{array} \right)$$

$$= \begin{array}{|cc|} \hline \multicolumn{2}{|c|}{\text{A}} \\ a_1 & a_2 \\ \hline 0.3 & 0.236 \\ \hline \end{array}$$

Sub-problem 4

$$P(A \mid C=c_1) = \frac{P(A,C=c_1}{\sum_A P(A,C=c_1)}$$

$$\approx \begin{array}{|cc|} \hline \multicolumn{2}{|c|}{\text{A}} \\ a_1 & a_2 \\ \hline 0.5597 & 4403 \\ \hline \end{array}$$

Sub-problem 5

$$P(A \mid B=b_2, C=c_1) = \frac{P(A,B=b_2,C=c_1)}{\sum_A P(A,B=b_2,C=c_1)} = (1/3, 2/3)$$

Sub-problem 6

A is not independent of B given $C=c_1$ since $P(A \mid B=b_2, C=c_1) \neq P(A \mid C=c_1)$.

Question 5 - 20 points

The book store *Smart books* wants to make book recommendations for its online customers. The store has recorded customer feedback (*Likes* with the states *yes* and *no*) for prior book purchases. For these purchases, the book store has recorded the following information about the books: The genre of the book (*Genre* with states *action*, *biography*, and *romance*), the format of the book (*Format* with states *paperback* and *hardcover*), the length of the book (*Length* with states *long* and *short*), and the price (*Price*) of the book (in DKK). The data that has been collected is shown in the table below.

	Attributes				Target
	Genre	*Format*	*Length*	*Price*	*Likes*
1	*bio*	*paperback*	*long*	110	*no*
2	*bio*	*paperback*	*short*	100	*no*
3	*action*	*hardcover*	*long*	70	*yes*
4	*romance*	*paperback*	*long*	80	*yes*
5	*romance*	*hardcover*	*short*	90	*yes*
6	*romance*	*hardcover*	*long*	105	*no*

In order to compare previous book purchases, the book store has defined distances for the states of the features characterizing a book:

- For the *Price* feature, the book store defines the distance as the absolute difference in prices.
- For the three discrete features, the book store uses the following distances for the states of the features:

Genre	*bio*	*action*	*romance*
bio	0	2	1
action	2	0	1
romance	1	1	0

Format	*paperback*	*hardcover*
paperback	0	1
hardcover	1	0

Length	*Short*	*Long*
Short	0	1
Long	1	0

The total distance between two instances representing a book is the sum of the distances of the four features with the price distance weighted with 0.1.

1. A new book is characterized by the feature values $\langle \textit{bio}, \textit{hardcover}, \textit{short}, 90\rangle$. Calculate its distance to the six recorded instances in the table.
2. Classify the new book according to the 1-nearest-neighbor-rule.
3. Classify the new book according to the 3-nearest-neighbor-rule.

Solution:

Sub-problem 1

The distances are:

Instance	Distance
1	4
2	2
3	5
4	4
5	1
6	3.5

Sub-problem 2

The closets neighbor is instance 5. The new book will therefore be classified as *yes* according to the 1-NN.

Sub-problem 3

The three closets neighbors are instances 2, 5, and 6. The new book will therefore be classified as *no* according to the 3-NN.

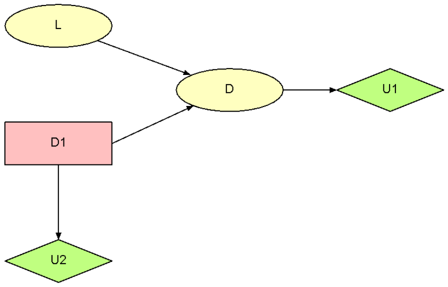
Question 6 - 20 points

On a vacation near the sea, you consider going on a guided tour in the hope of seeing dolphins. You have been told that there is a probability of 0.7 that there are dolphins at the location where the tour is heading, with probability 0.1 there are many dolphins, and with probability 0.2 there are no dolphins. However, even if there are dolphins at the location, there is no guarantee that you will actually see any dolphins: if there are some dolphins in the area you estimate that the probability of seeing a dolphin is 0.6, but if there are many dolphins then the probability increases to 0.9. The tour costs 200 DKK and you estimate that seeing a dolphin will be worth 500 DKK to you.

1. You should decide (D) whether you should *go* on the tour or *stay* at home.
   (a) Construct an influence diagram for your decision problem based on the description above. This include specifying the graphical structure and the probability and utility tables.
   (b) Calculate the expected utility of *go* and *stay*. Which decision should you take in order to maximize the expected utility.
2. Before deciding on whether to go, an experienced local sailor offers you accurate information about the dolphin population in the area (i.e, whether there *some*, *many*, or *no* dolphins). The cost of the information is 30 DKK. Determine whether it is worth to pay for the information by treating this information gathering decision as a value of information problem.

Solution:

Sub-problem 1



The probability and utility tables are given by: $P(L) = (0.2, 0.7, 0.1)$, $C(D) = (-200_{yes}, 0_{no})$, $U(F) = (500_{yes}, 0_{no})$, and

$$P(DF|D,L) = \begin{array}{r|ccc} & L = \mathit{no} & L = \mathit{some} & L = \mathit{many} \\ \hline D = \mathit{go} & (0,1) & (0.6,0.4) & (0.9,0.1) \\ D = \mathit{stay} & (0,1) & (0,1) & (0,1) \end{array}$$

First we marginalize out L:

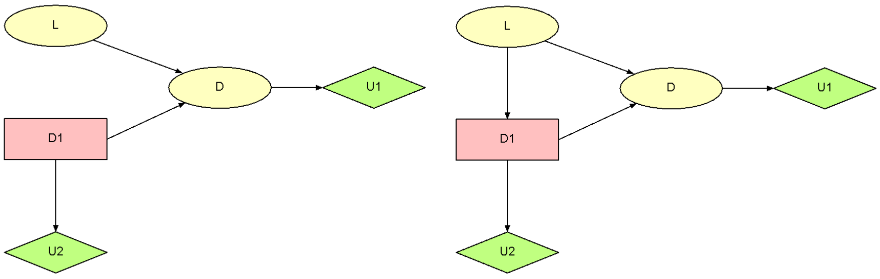
$$\begin{aligned} P(DF|D) &= \sum_L P(DF, L \mid D) \\ &= \sum_L P(DF \mid L, D)P(L) \\ &= \begin{array}{r|cc} & DF = \mathit{yes} & DF = \mathit{no} \\ \hline D = \mathit{go} & 0.51 & 0.49 \\ D = \mathit{stay} & 1 & 0 \end{array} \end{aligned}$$

This yields the expected utilities:

$$\begin{aligned} EU(\mathit{go}) &= 0.51 \cdot 500 + 0.49 \cdot 0 - 200 \\ &= 55 \\ EU(\mathit{stay}) &= 0 \end{aligned}$$

Sub-problem 2

The value of information problem basically boils down to calculating the differences in maximum expected utility between the two models:



The expected utility of the left-most model was calculated above.

The expected utility of the right model is given by:

$$EU = \sum_L P(L) \max_D (\sum_{DF} P(DF \mid D, L)U(DF) + U(C))$$

For the first part ($\sum_{DF} P(DF \mid D, L)U(DF) + U(C)$) of the calculations we get the expected utility $EU(L, D)$:

$$EU(L, D) = \begin{array}{r|ccc} & L = no & L = some & L = many \\ \hline D = go & -200 & 100 & 250 \\ D = stay & 0 & 0 & 0 \end{array}$$

By marginalizing out L and D we find:

$$EU = 95$$

and by also taking the cost (30DKK) of the information into account, we have a value of information given by $95 - 55 - 30 = 10$. Since the value is positive you should pay for the information.

Exercises for MI

Exercise sheet 1

Thomas Dyhre Nielsen

Exercise 1

You want to design an agent for playing tic-tac-toe (see e.g. `http://boulter.com/ttt/`).

- What is the state space the agent needs to reason with?
- How many states are there in the state space?
- Design 2 different feature-based representations of this state space.
- Design one relational representation of this state space.

Solution:

- The relevant states are all configurations of the 3×3 grid
- The 9 cells of the grid can be marked with any of the symbols "X","O","empty". This gives $3^9 = 19683$ states. However, this includes many states that can never be reached in an actual game (for example, the number of Xs and Os can differ by at most one).
- (a) Number the grid cells 1,. . . ,9. Define the features *mark of cell 1*,. . . ,*mark of cell 9* with values "X", "O", "empty". (b) Both players can have at most 5 moves in the game. Let X1,. . . ,X5 and O1,. . . ,O5 represent the position that the X-player (respectively the O-player) places his mark in move 1,. . . ,5. The possible values for each feature are the grid cells 1,. . . ,9, and "not played".
- One can use a binary relation state_of. Then state_of(cell 3,X), for example is the boolean feature that says whether cell 3 is marked with an X.

Exercise 2 You want to design a soccer playing robot (see e.g. `http://www.robocup.org`).

- Compared to the tic-tac-toe problem, is there a single "right" state space?
- Design one possible feature-based hierarchical state representation for the robot.

Solution: The soccer playing robot operates in a much more complicated environment than a tic-tac-toe playing agent. The current situation in a soccer match can not be fully represented by a small, fixed number of attributes, as in the tic-tac-toe problem. In principle, there are infinitely many possible positions the robots can have on the playing field, and ideally the soccer playing robot can distinguish them all, and adapt his actions to the exact scenario. Even though the robot may need to have a rather detailed state space representation to enable it to perform certain actions, it is useful to have a more abstract top-level representation on the basis of which only high-level goals are formulated. For example, we could design the robots so that they can operate in two modes: defend and attack. Then, at the highest level the robot only needs to decide in which mode it needs to play. For that decision, only some high-level features of the environment are required. Examples for features at the top-level could be:

ball_possession {*own team,opponent*}
ball_location$\{ownhalf, opponentshalf\}$
current_score$\{leading, equal, trailing\}$

Once the robot has made the decision whether to play attack or defend mode, it must plan its further actions (for example *attack_via_left_field* or *attack_via_right_field*) based on lower-level features. For example, now a more precise feature for the ball location would be needed, as well as more information on the current position of all robots on the playing field. For this, the playing field might be divided into n regions, and the current situation would be more precisely represented by:

ball_location {*region_1,. . . ,region_n*}
teammate_1 {*region_1,. . . ,region_n*}
...
teammate_5 {*region_1,. . . ,region_n*}
opponent_1 {*region_1,. . . ,region_n*}
...
opponent_5 {*region_1,. . . ,region_n*}
ball_in_possession_of {*teammate_1,. . . , teammate_5, opponent_1,. . . , opponent_5*}

Finally, at the lowest level of the hierarchy, the robot will need to plan and execute concrete movements. For example, when it has decided to try a shot at the goal, it needs still more precise information, as expressed by features like

direction_to_goal {*1,2,. . . ,360*}
distance_to_goal {*1,2,. . . ,10*}
own orientation relative to goal {*1,2,. . . ,360*}

Exercise 3 Discuss the differences between the problem domains above according to the dimensions of complexity summarized in Section 1.5.10:

Dimension	Values
Modularity	flat, modular, hierarchical
Representation scheme	states, features, relations
Planning horizon	non-planning, finite stage, indefinite stage, infinite stage
Sensing uncertainty	fully observable, partially observable
Effect uncertainty	deterministic, stochastic
Preference	goals, complex preferences
Learning	knowledge is given, knowledge is learned
Number of agents	single agent, multiple agents
Computational limits	perfect rationality, bounded rationality

Exercises for MI

Exercise sheet 2

Thomas Dyhre Nielsen

The exercises below can roughly be grouped into two categories. Those focusing on making a formal state-space representation of a problem and those aimed at analyzing or applying a particular search algorithm. As such, all exercises (except those relying on computer access) could be examples of questions for the exam. The questions are, however, of varying difficulty (with the latter group being the more difficult), and that would also be reflected in the 'number of points' each exercise would give at the exam.

I have marked the possible exam questions with an *.

Exercise 1 * Formalize the following problems as state-space problems: define a suitable state space, a start state, set of actions, the action function, and a goal test.

a. In the movie *Die hard: With a vengeance*, Bruce Willis and Samuel L. Jackson are given a water jug riddle, which they need to solve in order to disarm a bomb: There is a water fountain and two jugs that can hold 3 and 5 liters of water, respectively. How do you measure up 4 gallons of water (to disarm the bomb) using only the two jugs?

b. (From Russel & Norvig, Exercise 3.9): The *missionaries and cannibals problem* is usually stated as follows: Three missionaries and three cannibals are on one side of a river, along with a boat that can hold either one or two people. Find a way to get everyone to the other side of the river without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place.

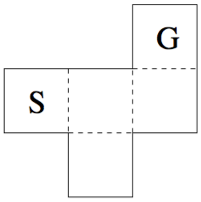
Solution:

a. States are the possible filling amounts (in full gallons) of the two jugs. For example, $(2, 0)$ is the state corresponding to the case where the first jug contains 2 gallons and the second jug contains 0 gallons. The start state is $(0, 0)$. The 'fill second jug' action leads to the state $(0, 5)$ from which the state $(3, 2)$ can be reached by the action 'fill first jug from second jug'. A goal state is any state where the second jug contains 4 gallons.

b. Here is one possible representation: A state is a six-tuple of integers listing the number of missionaries, cannibals, and boats on the first side, and then the second side of the river. The goal is a state with 3 missionaries and 3 cannibals on the second side. The cost function is one per action, and the successors of a state are all the states that move 1 or 2 people and 1 boat from one side to another.

Exercise 2 Solve Exercise 3.2 in **PM**.

Exercise 3 In this exercise we experiment with the Graph Searching applet on http://www.aispace.org/search/index.shtml. A robot needs to find a path from the start position S to the goal position G in the following map:



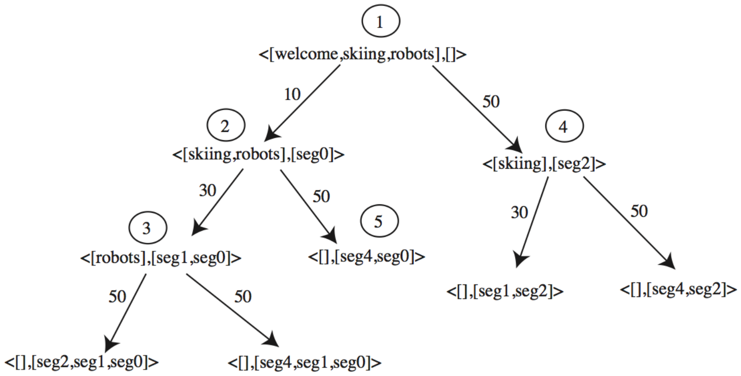
- Create in the Graph Searching applet a graph representing this problem.
- Before you continue, you may want to save your graph into a file, since the applet sometimes experiences crashes that destroy all unsaved input ...
- Use the 'Fine Step' function under the 'Solve' tab to simulate depth-first search for this problem
- Does depth-first search terminate? If not, is depth-first search unable in principle to find a solution for this problem? How could the implementation in the applet be changed, so that depth-first can run successfully?
- Try depth-first search again with 'Search Options:Pruning:Loop Detection' activated.
- Now try breadth-first search. Is a solution found? What do you observe about the Frontier?

Solution: Depth-first search will work if neighbors of nodes are enumerated in the right order. In this problem, Node 3 should be the last neighbor of Node 1 to be enumerated (i.e. be the one put on the stack last), and Node 4 must be the last neighbor of Node 3 to be enumerated. The order can be influenced in the applet via the Neigbor ordering strategies in Search options.

Exercise 4 * Solve Exercise 3.4 in **PM**. Disregard the question about monotone restriction.

Solution:

a. The nodes are expanded according to the circled numbers. The circled (5) is the first goal node found - this is the shortest presentation that covers all the topics.



- Here are two solutions:

  Solution 1: For each topic, t, let $s(t)$ be the length of the smallest segment that covers topic t. Let $h(\langle TC, Segs\rangle) = \max_{t \in TC} s(t)$. That is, we find the topic t for which $s(t)$ is maximum.

  Solution 2: For each segment let the contribution of the segment be the time of the segment divided by the number of topics the segment covers. For each topic, t, let $s(t)$ be the smallest contribution for all of the segments that covers the topic. Let $h(\langle TC, Segs\rangle) = \sum_{t \in TC} s(t)$. That is, we sum $s(t)$ for all of the topics t in TC. The intuition is that each topic t requires at least $s(t)$ time. Note that we need to divide by the number of topics the segment covers to make sure that we do not double count the time for segments added that cover multiple topics.

  Both of these solution require one pass through the segment database to build the $s(t)$ function, but once this is built, the heuristic function can be computed in time proportional to the length of the To cover list.

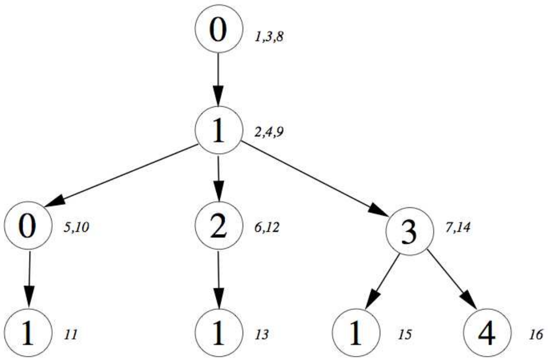
Exercise 5 * For the robot navigation problem of the previous exercise:

- Draw the search tree for this problem (or as much of the search tree as is needed to answer the following:

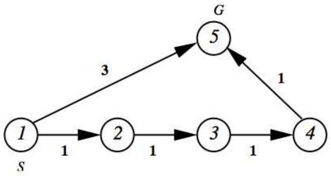
- Show how iterative deepening search will solve this problem: show the order in which nodes of the search tree will be selected and expanded.

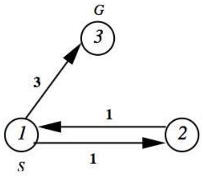
Solution:

The numbers inside the nodes correspond to the node numbers in the solution to the previous exercise. We assume that branches are explored from left to right. The small numbers beside the nodes then give the order in which nodes are expanded when performing iterative deepening search.

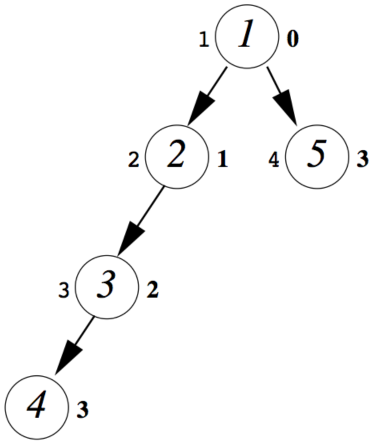


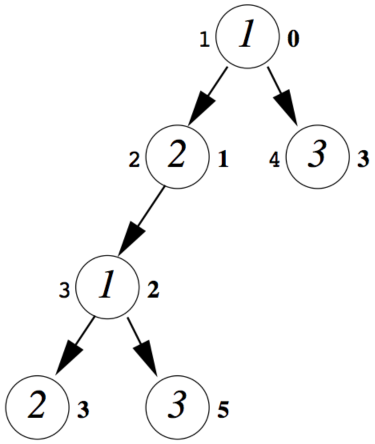
Exercise 6 * Show how Lowest-Cost-First Search will find for the two problems below a minimial cost path from start state S to goal state G (draw the relevant part of the search tree, and indicate in which order nodes are expanded).





Solution: Bold numbers to the right of the nodes show the cost associated with the node. Typewriter font numbers to the left show the order in which the nodes are chosen for expansion. Since the cost of both the frontier nodes at the end is 3, the choice between them is arbitrary. The algorithm might also choose to expand the left branch by one more node, before it finds the optimal path by going down the right branch.





Exercise 7 *

- For each node n in the two graphs of the exercise above determine the cost $opt(n)$ of an optimal solution starting from that node.
- Show how A^* finds the optimal solutions for these two problems when $h(n) = opt(n)$.

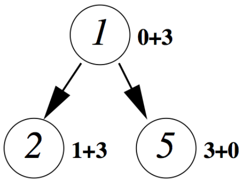
Solution: Costs for the left graph (node number $n : opt(n)$):

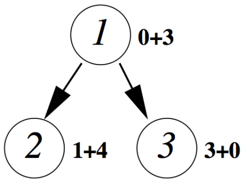
$$1 : 3, 2 : 3, 3 : 2, 4 : 1, 5 : 0$$

Costs for the right graph (node number $n : opt(n)$):

$$1 : 3, 2 : 4, 3 : 0$$

The bold numbers now show the sum $f(n) = cost(n) + opt(n)$. A^* chooses the node with the lowest $f(n)$ value. Thus, the optimal path (down the right branch) is immediately found.





Exercise 8 * You are planning a dinner for three guests. The menu should consist of at least one appetizer, exactly one main dish, and at least one desert (multiple appetizers or deserts are o.k.). You have a list of candidate dishes, and for each guest you know whether they like that dish or not:

Item	Cost	Guest 1	Guest 2	Guest 3
Appetizer 1	5			o.k.
Appetizer 2	5		o.k.	
Appetizer 3	15		o.k.	o.k.
Appetizer 4	30	o.k.		
Main dish 1	90		o.k.	o.k.
Main dish 2	100	o.k.		o.k.
Dessert 1	30		o.k.	
Dessert 2	50	o.k.	o.k.	

Your menu must contain for each guest at least one item that they like. Use A^* to find a minimal cost solution:

- Define the underlying state space problem
- Define a heuristic function that underestimates the true optimal cost function *opt*.
- Show how A will find the minimal cost solution using this heuristic function.

Solution:

- The states are all possible subsets of dishes: $\emptyset, \{A1\}, \{A2\}, \ldots, \{A1, A2, D1\}$, ... The start state is $\emptyset$. Actions are of the form 'add dish X to the menu'. For example, 'add dish D1 to the menu' applied in state $\{A2\}$ leads to state $\{A2, D1\}$. Goal states are all states that contain at least one appetizer, exactly one main dish, at least one desert, and that contain for each guest one dish that guest likes. Thus, for example, a goal state must contain at least one of $A4$, $M2$, or $D3$ (to satisfy guest 1). Note that enumerating all goal states would be quite tedious, but all we need is a goal test, i.e. for a given state we must be able to decide whether it is a goal state.
- If a state is not yet a goal state, then at least one more dish must be added to the state. The cost to reach a goal state, thus is at least the cost of the cheapest dish not yet in the state. This would be a first possible heuristic function. A better heuristic function is obtained by considering what type of dishes are still lacking. Types can be 'appetizer', 'main', 'desert', 'liked by G1', 'liked by G2', 'liked by G3'. If the current state, for example, does not yet contain a desert, nor any dish liked by Guest 3, then the cost of reaching a goal state is at least the cheapest dish that is a desert, or liked by Guest 3. This heuristic function is better than the first, because it underestimates the true function opt not as much as the first function. Further refinements that provide yet closer approximations of opt can also be defined.

- First steps of A^* using the second heuristic function. The start state has 8 neighbors:

State n	$f(n) = cost(n) + h(n)$	State n	$f(n) = cost(n) + h(n)$
$\{A1\}$	$5 + 5$	$\{M1\}$	$90 + 5$
$\{A2\}$	$5 + 5$	$\{M2\}$	$110 + 5$
$\{A3\}$	$15 + 5$	$\{D1\}$	$30 + 5$
$\{A4\}$	$30 + 5$	$\{D2\}$	$50 + 5$

The heuristic function value is 5 for all states, because all states lack either an appetizer, or a dish liked by Guest 2, or a dish liked by Guest 3, all of which can be added in the form of $A1$ or $A2$, i.e. at the cost of 5.

A^* will now pick a state with minimal f value, which is either $\{A1\}$ or $\{A2\}$. Suppose we pick $\{A1\}$, we have the following neighbors:

State n	$f(n) = cost(n) + h(n)$	State n	$f(n) = cost(n) + h(n)$
$\{A1, A2\}$	$10 + 30$	$\{A1, M1\}$	$95 + 30$
$\{A1, A3\}$	$20 + 30$	$\{A1, M2\}$	$115 + 5$
$\{A1, A4\}$	$35 + 5$	$\{A1, D1\}$	$35 + 30$
		$\{A1, D2\}$	$50 + 5$

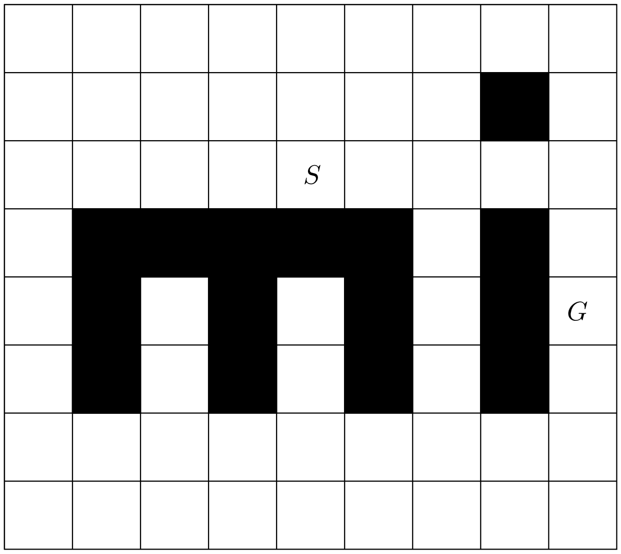
For $n = \{A1, A2\}$ we now have $cost(n) = 10$, and $h(n) = 30$, because n lacks a main dish, a desert, and a dish liked by Guest 1, and the cheapest dishes that correct one of these defects are $A4$ and $D1$, at the price of 30. In the next step $\{A1, A2\}$ or $\{A1, A4\}$ will be further expanded.

Exercise 9

Consider the problem of finding a path from position S to G in the grid shown below. You can only move horizontally and vertically and only one step at a time; no step can be made into the shaded areas or outside the grid. The cost of the path between two positions is the number of steps on the path.

Show how to solve the problem using dynamic programming. Mark each node/position on the grid with the *cost_to_goal* (see **PM** 3.8.3) and show which path is found.

Solution: Following the procedure outlined in **PM** we node annotation shown in the figure below from which a lowest cost path can be directly deduced.



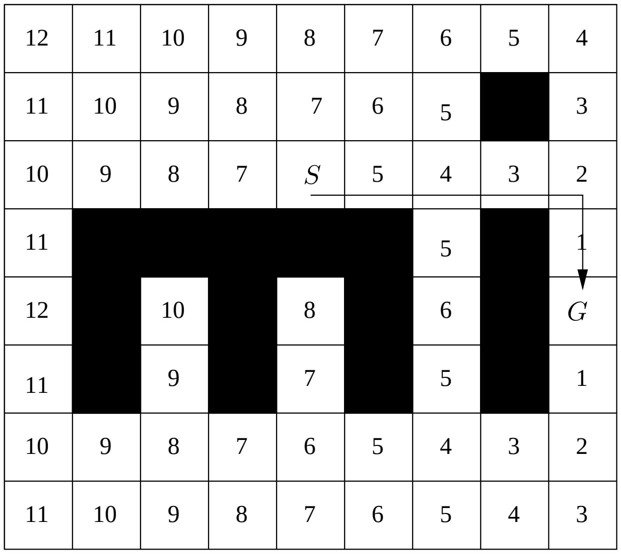
Exercise 10 *

Previous exam question: Consider the problem of finding a path from position S to G in the grid shown below. You can only move horizontally and vertically and only one step at a time; no step can be made into the shaded areas or outside the grid. The cost of the path between two positions is the number of steps on the path.

1. Number the positions (cells in the grid) in the order in which they are added to the frontier in a lowest-cost-first search for G starting at S. Assume that the ordering of the operators is *right*, *down*, *left*, and *up*, and that there is multiple path pruning. When multiple lowest-cost paths exists, choose the path first added to the frontier.
2. Define a heuristic function that underestimates the cost of the lowest-cost path.
3. Number the positions (cells in the grid) in the order in which they are added to the frontier according to an A^* search using your heuristic function. Resolve ambiguities using the operator ordering and the procedure above.

NB: When numbering the positions it may be helpful to also annotate the positions with the cost/function values calculated during the search.

Solution:



12	11	10	9	8	7	6	5	4
11	10	9	8	7	6	5	■	3
10	9	8	7	*S*	5	4	3	2
11	■	■	■	■	■	5	■	1
12	■	10	■	8	■	6	■	*G*
11	■	9	■	7	■	5	■	1
10	9	8	7	6	5	4	3	2
11	10	9	8	7	6	5	4	3

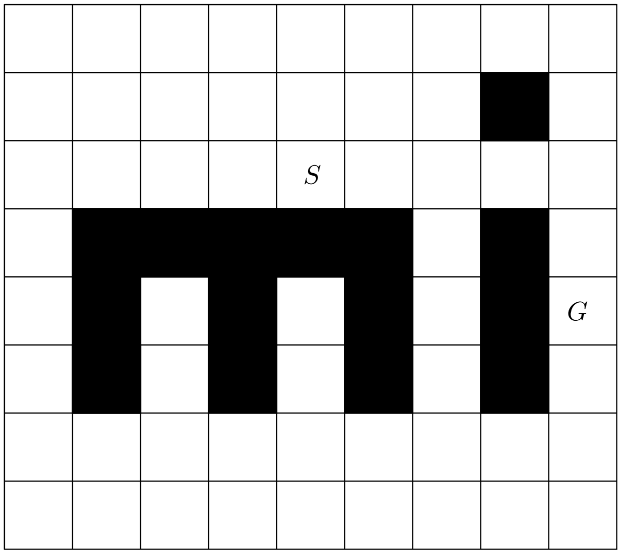
Answer 1

The red numbers specify the order in which the cells are added to the frontier. The blue numbers are the costs associated with the cells.

Answer 2

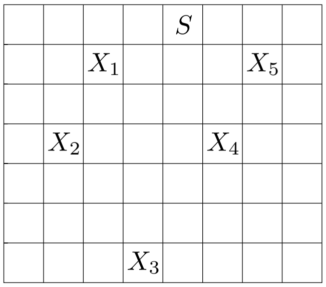
One possible heuristic function simply calculates the cost of the lowest-cost path ignoring the obstacles (shaded regions).

Answer 3



Exercise 11 *

Previous exam question: Consider a robot that can move in the grid below. The cost of moving from one cell to another is equal to the number of steps required for the move, where by a single step the robot can move horizontally or vertically to an adjacent cell (no diagonal steps allowed).

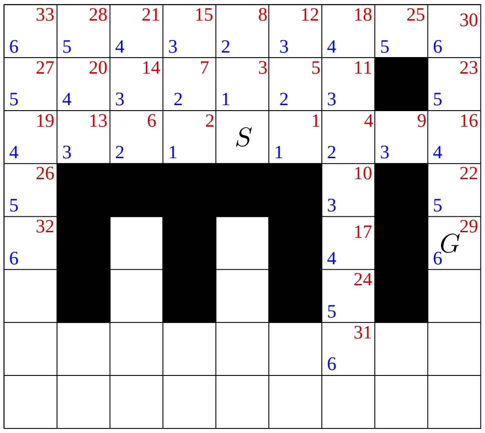


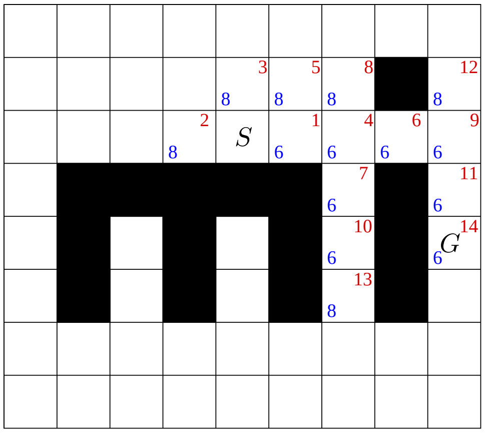
The cells marked X_1, X_2, X_3, X_4, and X_5 are *cells of interest*, each holding one or two of the symbols a, b, and c:

- X_1: (a, c)
- X_2: (b)
- X_3: (b, c)
- X_4: (c)
- X_5: (a)

That is, X_1 holds the symbols a and c. When visiting a cell the robot collects the symbols located at that cell.

Starting in cell S, the robot's task is to visit cells of interest and collect exactly





one copy of each of the three symbols (a, b, c) before returning to cell S. The robot is *not* allowed to move to a cell containing, e.g., an a if the robot has previously visited a cell holding an a.

Determine which of the cells of interest to visit and in which order these cells should be visited by the robot, so that the cost of the path traveled (when starting and ending in cell S) is minimized.

1. Define a suitable state and action representation for this problem.
2. Define a heuristic function that at any given state provides an underestimate of the cost of reaching a goal state (i.e., a state where the robot is located in cell S with the symbols (a, b, c)).
3. Show the search tree for this problem as it would be expanded using A* search.

Solution:

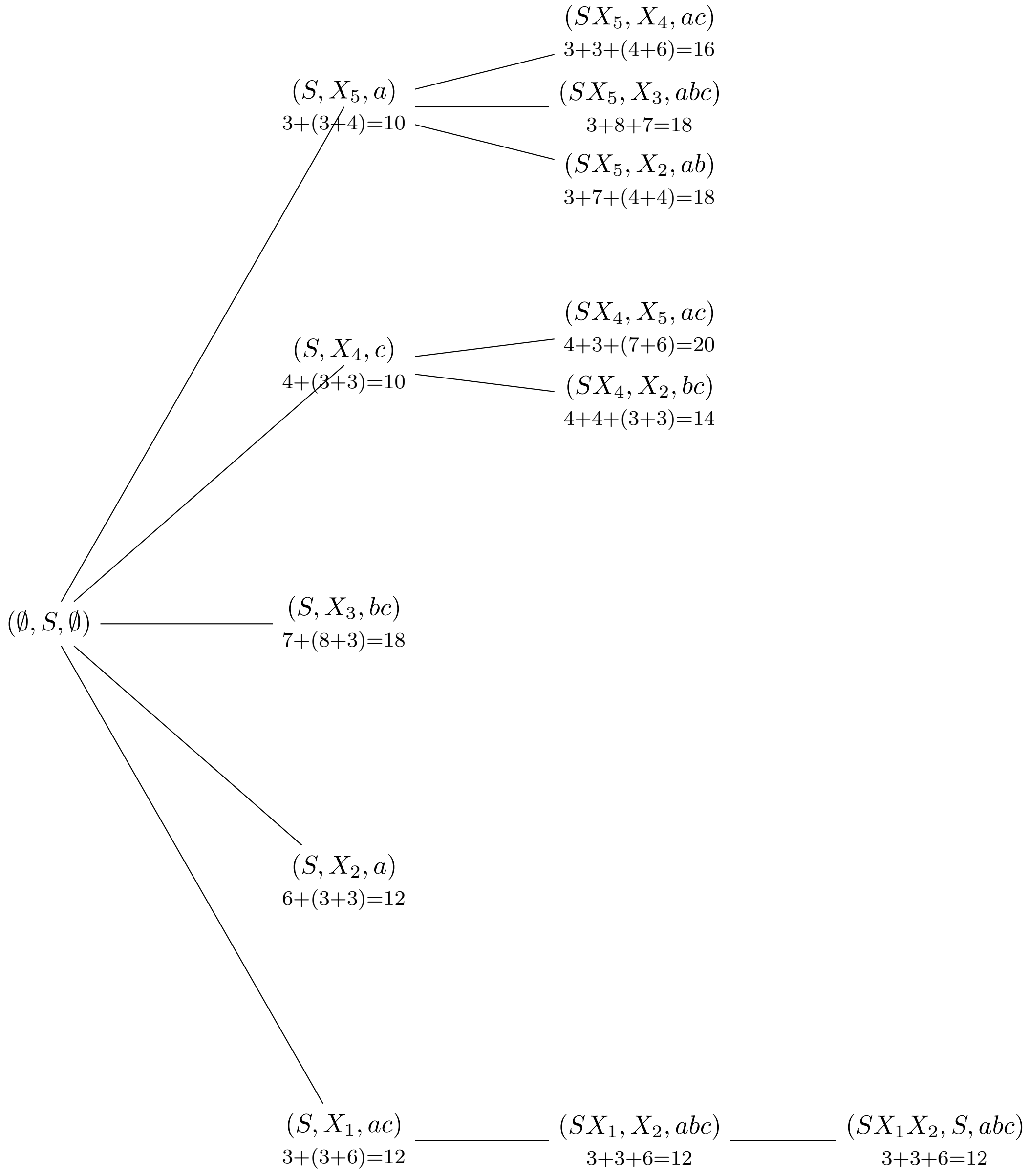
Sub-problem 1

One possible state representation could be (*cell history*, *current cell*, *symbols cllected*), where we keep track of the cells previously visited, the current cell the robot is in, and which symbols that have been collected so far. We can, however, also do without including the cell history in the state representation.

Sub-problem 2

One possible heuristic function could be the minimal cost incurred by visiting an admissible cell holding a symbol not yet collected and afterwards returning to S.

Sub-problem 3



Exercises for MI

Exercise sheet 3

Thomas Dyhre Nielsen

I have marked the possible exam questions with an *. The questions are, however, of varying difficulty (with the latter group being the more difficult), and that would also be reflected in the 'number of points' each exercise would give at the exam.

When you have finished with the exercises, you should continue with the exam sheet from the previous years, which can be found at the course's home page.

Exercise 1 * Consider the same situation as in Exercise 8 (from the last exercise sheet), but now you only want to know whether there is a menu that costs no more than 150. Express this problem as a constraint satisfaction problem:

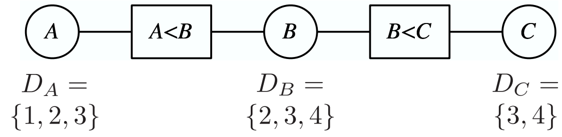
- what are suitable *variables* that describe the possible worlds?
- what are the constraints? For the specification it is sufficient to give them in intensional form.
- draw (a part of) the state space graph that you would use if you wanted to solve this problem using search.

Solution: One can describe the states (possible worlds) already used in Problem 2.8 using 8 boolean variables: *in_menu_A1*, *in_menu_A2*,..., *in_menu_D2*. The state (or possible world) $\{A1, M1, D2\}$ then corresponds to the value assignment *in_menu_A1* = *true*, *in_menu_A2* = *false*, etc. Constraints are

$$\mathit{in\_menu\_A1} = \mathrm{true} \lor \mathit{in\_menu\_A2} = \mathit{true} \lor \mathit{in\_menu\_A3} = \mathit{true} \lor \mathit{in\_menu\_A4} = \mathit{true}$$

(there must be at least one appetizer in the menu), and similarly for main dishes, deserts, and each of the guests.

Exercise 2 *



- Which arcs in the above constraint network are arc consistent?
- How can the whole network be made arc consistent, i.e., which changes should be made to the domains of the variables making the network arc-consistent without eliminating potential solutions?
- Check your result by implementing the model in the CSP-implementation found at `http://aispace.org/constraint/`.

Solution:

- The arc $\langle B, B < C \rangle$ is not consistent, because for $B = 4$ there is no value in D_C that satisfies the constraint. All other arcs are arc consistent.
- First the value 4 has to be removed from D_B. Now the arc $\langle A, A < B \rangle$ no longer is consistent, because for $A = 3$ now no suitable value for B exists. After also removing 3 from D_A the network is arc consistent.
- See the file ABC-constraint.xml.

Exercise 3

Consider the following mini-Sudoku:

		4	
2			
		1	
3			

The empty fields have to be filled with numbers 1,2,3, or 4, such that each row, each column, and each of the 2 × 2 sub-squares contain each of these numbers exactly once.

a. * Formalize this problem as a constraint satisfaction problem: define an appropriate set of variables, and a set of constraints that define the solutions of the sudoku (hint: instead of using constraints as on the lecture slide, define a smaller set of constraints, each constraint expressing the full condition for one row, one column, or one sub-square. We have not discussed off-the-shelf formal languages for expressing such constraints, so you should try to express them as formally as possible).

b. * Draw the constraint network for this problem. If this gets too large, draw only the part of the network that is sufficient to answer the next question.

c. Apply the generalized arc consistency algorithm to show that the top left square must contain a 1. Does the operation of the algorithm resemble how you would come to that conclusion yourself?

Solution:

Label the empty fields with variables like this:

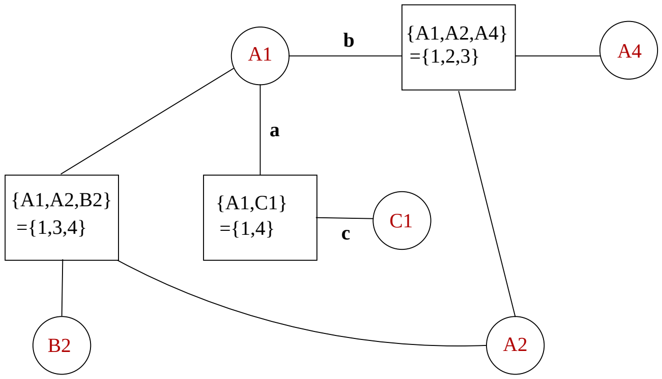
A1	A2	4	A4
2	B2	B3	B4
C1	C2	1	C4
3	D2	D3	D4

We now know that $A1, A2, A4$ combined must contain the values 1,2,3. We write this as a constraint in the form:

$$\{A1, A2, A4\} = \{1, 2, 3\}$$

Similarly, we define one constraint for each remaining row, column, and 2×2 sub-square.

A part of the constraint network now is:



Initially, we have $D_{A1} = D_{A2} = \ldots D_{D4} = \{1, 2, 3, 4\}$. We first find that the arc **a** is not arc consistent, because for the values $A1 = 2$ or $A1 = 3$ the constraint $\{A1, C1\} = \{1, 4\}$ can not be satisfied. Thus, we delete 2 and 3 from D_{A1}. Since $D_{C1} = \{1, 2, 3, 4\}$ the arc now is arc consistent. Next inspect arc **b**. For $A1 = 4$ there are no values of $A2$ and $A4$ that could make the constraint satisfied, so 4 also is removed from D_{A1}. With $D_{A2} = D_{A4} = \{1, 2, 3, 4\}$ arc **b** now is arc consistent. At this point we have $D_{A1} = \{1\}$, i.e. we know that in any possible solution of the sudoku $A1$ must have value 1. We can now continue. For example, making arc **c** arc consistent immediately leads to the reduction $D_{C1} = \{4\}$.

Exercise 4 * Solve exercise 4.12 in PM.

Solution:

- r_1 and r_2 are removed. $r_{11}(B, C)$ is added.
- R_3, r_4, and r_{11} are removed. $R_{12}(C, D, E)$ is added.

Exercise 5 *

Use Variable Elimination to solve the following CSP given by extensional constraints on Boolean variables A, B, C:

A	B
t	f
t	t
f	t

A	C
t	f
f	t

B	C
t	f
f	t

Solution:

We eliminate B first. Joining the two tables containing B gives

$$\begin{array}{cc} A & B \\ \hline \mathrm{t} & \mathrm{f} \\ \mathrm{t} & \mathrm{t} \\ \mathrm{f} & \mathrm{t} \end{array} \bowtie \begin{array}{cc} B & C \\ \hline \mathrm{t} & \mathrm{f} \\ \mathrm{f} & \mathrm{t} \end{array} = \begin{array}{ccc} A & B & C \\ \hline \mathrm{t} & \mathrm{f} & \mathrm{t} \\ \mathrm{t} & \mathrm{t} & \mathrm{f} \\ \mathrm{f} & \mathrm{t} & \mathrm{f} \end{array}$$

Projecting on A, C then gives

A	C
t	t
t	f
f	f

We eliminate C next (the order in which we perform the elimination is not very important in this example). We now have two tables containing C, which we join:

$$\begin{array}{cc} A & C \\ \hline \mathrm{t} & \mathrm{f} \\ \mathrm{f} & \mathrm{t} \end{array} \bowtie \begin{array}{cc} A & C \\ \hline \mathrm{t} & \mathrm{t} \\ \mathrm{t} & \mathrm{f} \\ \mathrm{f} & \mathrm{f} \end{array} = \begin{array}{cc} A & C \\ \hline \mathrm{t} & \mathrm{f} \end{array}$$

The result of this join now is the only constraint left. Therefore, the algorithm terminates with the result that the solutions of the constraint satisfaction problem consist of tuples with $A = t$ and $C = f$. To also determine the possible values for B, we have to join this result table with the previous constraint we computed for A, B, C:

$$\begin{array}{cc} A & C \\ \hline \mathrm{t} & \mathrm{f} \end{array} \bowtie \begin{array}{ccc} A & B & C \\ \hline \mathrm{t} & \mathrm{f} & \mathrm{t} \\ \mathrm{t} & \mathrm{t} & \mathrm{f} \\ \mathrm{f} & \mathrm{t} & \mathrm{t} \end{array} = \begin{array}{ccc} A & B & C \\ \hline \mathrm{t} & \mathrm{t} & \mathrm{f} \end{array}$$

This means that there exists exactly one solution to the problem: $A = t$, $B = t$, $C = f$.

Exercise 6 Solve exercise 4.3 (except d) in PM and using only the network displayed in Figure 4.15(b). For exercise (a), (b), (c) you may use the CSP-implementation found at http://aispace.org/constraint/; observe that the CSP-implementation has a special constraint for the word relations appearing in a cross-word.

Solution:

- See http://cs.ubc.ca/~poole/aibook/figures/ch04/cross2cn.xml for a AISpace.org representation.

  See http://cs.ubc.ca/~poole/aibook/figures/ch04/cross2ac.xml for a AISpace.org representation of the arc consistent network.

- Let's first eliminate 1a: We get the constraint on ⟨1d,2d⟩ with the elements {⟨haste,eta⟩, ⟨sound, one⟩, ⟨think, her⟩}. Eliminate 3a, gives the relation on ⟨1d, 2d⟩ with elements: {⟨haste, eta⟩, ⟨sound, one⟩, ⟨think, her⟩}. These combine to the same relation on ⟨1d, 2d⟩. We can now eliminate 2d, which creates a relation on ⟨1d,4a⟩ with domain {⟨haste,usage⟩, ⟨sound,fuels⟩, ⟨think,first⟩}. Again this provides no more constraints than the previous relation on ⟨1d, 4a⟩. We can now eliminate 1d and create a relation on ⟨6a, 4a⟩ with domain: {⟨easy, usage⟩ , ⟨else, usage⟩ , ⟨ desk, fu If we eliminate 5d, we create a relation on ⟨6a, 4a⟩ with domain: {⟨desk, fuels⟩ , ⟨easy, fuels⟩ , ⟨else, fuels⟩ , ⟨kin These last two can be combined giving their intersection: {⟨desk, fuels⟩ , ⟨kind, first⟩} This is then only relation left and there are two solutions on ⟨6a, 4a⟩. Each of these gives rise to a unique solution.

- If you eliminate 1d (or 4a) first, you create a relation on 3 variables which is much more complicated and less efficient. So that elimination ordering does affect efficiency.

Exercise 7 Solve exercise 4.5 in PM.

Solution:

- It does not work very well. Within a hundred steps, it can solve the problem about 4% of the time.

- Hill climbing (greedy descent) can solve about 60% of the problems with a median of about 10 steps.
- It does better! I got an nearly 90% success rate but with a median around 14 steps using choosing the best node 50% of the time, any red node 50% of the time and the best value 100% of the time.
- The best setting, in terms of number of steps, I found is hill climbing with random restart every 10 steps. This had 100% success rate with a median of around 10 steps.

Exercise 8 *

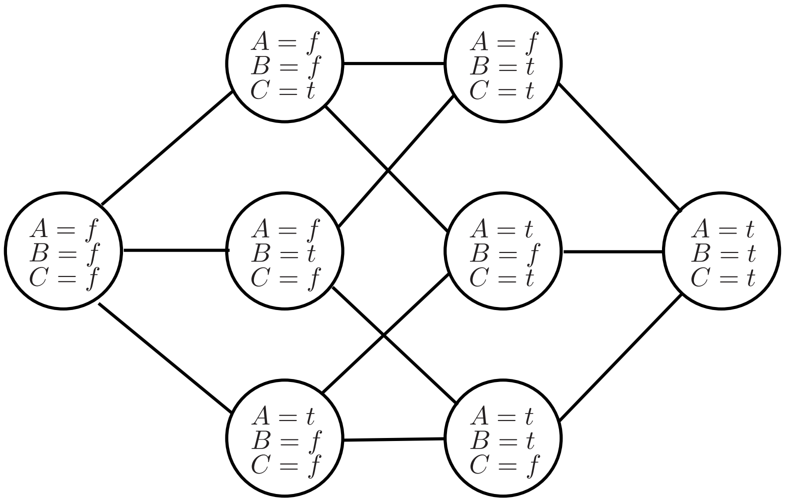
Consider the following soft constraints on three binary variables A, B, C:

A	B	Cost
t	t	2
t	f	5
f	t	1
f	f	5

A	C	Cost
t	t	8
t	f	0
f	t	1
f	f	8

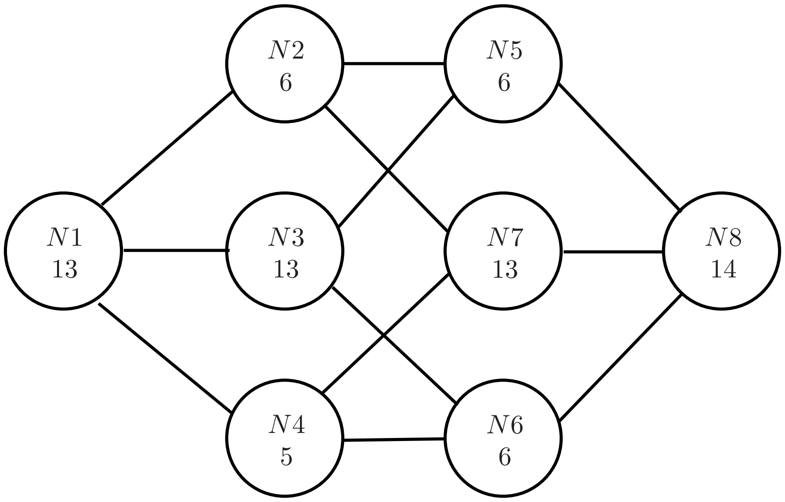
B	Cost
t	4
f	0

- Label the nodes in the state space graph below with the resulting (additive) cost function.
- What will be the solution found if local hill climbing search is started at the possible world $A = f, B = t, C = f$?
- Where would you have to start local hill climbing search in order to find the globally optimal solution?



Solution:

The state space graph with nodes numbered $N1, \ldots, N8$, and labeled with cost function:



- When starting at $N3$, hill climbing will either go to $N5$ or $N6$ (based on some tie-breaking mechanism, e.g. random choice). If the next node is $N5$ the search may either stay there or go to $N2$ (both of which are local minima). If the node following $N3$ is $N6$, then the next node is $N4$, which is a global minimum.
- The following table gives for all possible start nodes the solution returned:

Start	Solution
$N1$	$N4$
$N2$	$N2$ or $N5$
$N3$	$(N5/N2)$ or $N4$
$N4$	$N4$
$N5$	$N2$ or $N5$
$N6$	$N4$
$N7$	$N4$
$N8$	$(N5/N2)$ or $N4$

Thus, only $N2$ and $N5$ cannot lead to the globally optimal solution.

Exercise 9 Consider the cryptarithmetic puzzle

$$\begin{array}{cccc} & \mathrm{T} & \mathrm{W} & \mathrm{O} \\ + & \mathrm{T} & \mathrm{W} & \mathrm{O} \\ \hline \mathrm{F} & \mathrm{O} & \mathrm{U} & \mathrm{R} \end{array}$$

Each letter in a cryptarithmetic problem represents a digit; note that F cannot be 0.

- Construct a constraint network representation for the puzzle. *Hint:* For each of the four columns in the table representation above we have a constraint. E.g. for the first column we have the constraint

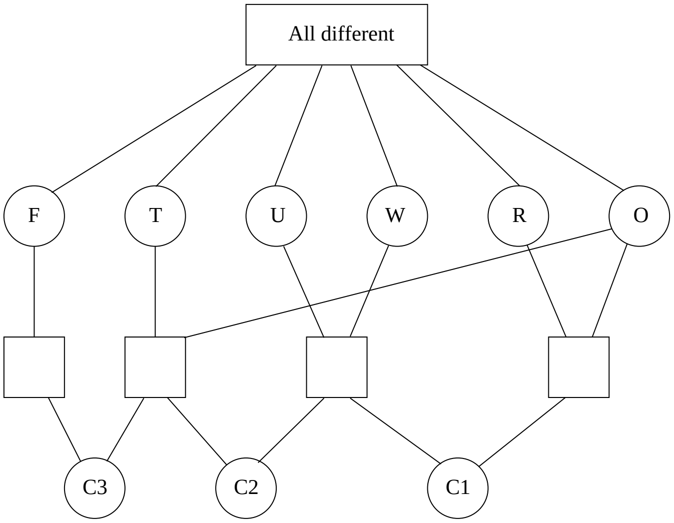
  $$O + O = R + 10 \cdot C_1,$$

  where C_1 is an auxiliary variable representing what is carried over in the 10 column. Use similar auxiliary variables, say C_2 and C_3, for encoding what is carried over to the 100 and 1000 column, respectively.

- Perform a forward-backward search to find a solution to the puzzle. *Hint:* The ordering of the variables has a big impact on the solution size, so a good strategy would be to chose the variable with the smallest state space. E.g. consider starting with $C_3 = 1$.

Solution:

- Here is a constraint network for the cryptarithmetic problem



  The constraint connecting R, O, and C_1 encodes that $O + O = R + 10/cdotC_1$. Similar for the other constraints connected to the auxiliary variables.

- There are many possible solutions to the problem. One possibility is to

  1. Set $C_3 = 1$.

2. Set $F = 1$ (the only possible value).
3. Set (arbitrarily) $C_2 = 0$.
4. Set (arbitrarily) $C_1 = 0$.
5. O must be an even number (it is equal to $T + T$) and less that 5 since $C_1 = 0$. Set (arbitrarily) $O = 4$.
6. R now only has one value, namely 8.
7. T only has one value, namely 7.
8. U must be an even number less than 9. Set $U = 6$.
9. Set $W = 3$.

Exercise 10

a. Let π be the interpretation that assigns the following truth values:

$$\pi(a) = \mathit{true}, \pi(b) = \mathit{false}, \pi(c) = \mathit{false}, \pi(d) = \mathit{true}$$

Determine the truth values for the following propositions:

$$\begin{array}{l} \neg a \rightarrow b \\ (\neg b \vee c) \wedge (d \rightarrow a) \\ (a \rightarrow c) \rightarrow c \end{array}$$

b. For the following propositions, find an interpretation in which they are true:

$$\begin{array}{l} (a \vee (a \rightarrow c)) \rightarrow b \\ (a \wedge (\neg b \vee c)) \wedge (a \rightarrow (c \rightarrow b)) \end{array}$$

Solution:

a. All three propositions are true.

b. The first proposition is true in any interpretation with $\pi(b) = \mathit{true}$.

The second proposition is true for $\pi(a) = \mathit{true}$, $\pi(b) = \mathit{true}$, $\pi(c) = \mathit{true}$. It is also true for $\pi(a) = \mathit{true}$, $\pi(b) = \mathit{false}$, $\pi(c) = \mathit{false}$. These are the only two solutions.

Exercise 11

Show that if a knowledge base KB contains the two propositions a and $a \rightarrow b$, then

$$KB \models b$$

(This means that the Modus Ponens inference rules is *sound*).

Solution:

One has to show that every interpretation in which all propositions in KB are true also makes b true. An interpretation π can only make both a and $a \rightarrow b$ true if $\pi(a) = \mathit{true}$ and $\pi(b) = \mathit{true}$.

Exercises for MI

Exercise sheet 4

Thomas Dyhre Nielsen

When you have finished with the exercises, you should continue with the exam sheet from previous years, which can be found at the course's home page.

Exercise 1 Consider the experiment of flipping a fair coin, and if it lands heads, rolling a fair four-sided die, and if it lands tails, rolling a fair six-sided die. Suppose that we are interested only in the number rolled by the die, and the possible worlds $\mathcal{S}_A$ for the experiment could thus be the numbers from 1 to 6. Another set of possible worlds could be $\mathcal{S}_B = \{t1, \ldots, t6, h1, \ldots, h4\}$, with for example $t2$ meaning "tails and a roll of 2" and $h4$ meaning "heads and a roll of 4." Choose either $\mathcal{S}_A$ or $\mathcal{S}_B$ and associate probabilities with it. According to your chosen set of possible worlds and probability distribution, what is the probability of rolling either 3 or 5.

Solution:

Probabilities for $\mathcal{S}_{\mathcal{A}}$: $P_A(1) = \cdots = P_A(4) = \frac{5}{24}$ and $P_A(5) = P_A(6) = \frac{1}{12}$.

Probabilities for $\mathcal{S}_{\mathcal{B}}$: $P_B(t1) = \cdots = P_B(t6) = \frac{1}{12}$ and $P_B(h1) = \cdots = P_B(h4) = \frac{1}{8}$.

$P_A(3) + P_A(5) = \frac{7}{24}$.

$P_B(t3) + P_B(t5) + P_B(h3) = \frac{7}{24}$.

Exercise 2 Let $\mathcal{S}_B$ be defined as in the Exercise above, but with a loaded coin and loaded dice. A probability distribution is given in Table 1. What is the probability that the loaded coin lands "tails"? What is the conditional probability of rolling a 4, given that the coin lands tails? Which of the loaded dice has the highest chance of rolling 4 or more?

$t1$	$\frac{5}{18}$	$t6$	$\frac{1}{18}$
$t2$	$\frac{1}{9}$	$h1$	$\frac{1}{24}$
$t3$	$\frac{1}{9}$	$h2$	$\frac{1}{24}$
$t4$	$\frac{1}{18}$	$h3$	$\frac{1}{8}$
$t5$	$\frac{1}{18}$	$h4$	$\frac{1}{8}$

Table 1: Probabilities for $\mathcal{S}_B$ in Exercise 2.

Solution: $P(t) = \frac{5}{18} + \frac{1}{9} + \frac{1}{9} + \frac{1}{18} + \frac{1}{18} + \frac{1}{18} = \frac{2}{3}$.

$P(4 \,|\, t) = P(t4)/P(t) = \frac{1}{12}$.

$P(4 \,|\, t) + P(5 \,|\, t) + P(6 \,|\, t) = \frac{1}{4}$.

$P(4 \,|\, h) = P(h4)/P(h) = 1/8/(\frac{1}{24} + \frac{1}{24} + \frac{1}{8} + \frac{1}{8}) = \frac{3}{8}$.

The four-sided die thus has a higher probability of rolling 4 or more, than the six-sided die.

Exercise 3 * Calculate $P(A)$, $P(B)$, $P(A|B)$, and $P(B|A)$ from the joint probability distribution $P(A, B)$ given in Table 2.

	b_1	b_2	b_3
a_1	0.05	0.10	0.05
a_2	0.15	0.00	0.25
a_3	0.10	0.20	0.10

Table 2: The joint probability distribution for $P(A, B)$.

Solution: In order to find $P(A)$ we need to marginalize (sum) out the variable B. This is done for each state of A. Thus, for $A = a_1$ we get:

$$P(A = a_1) = \sum_B P(A = a_1, B) = 0.05 + 0.10 + 0.05 = 0.20$$

In total we end up with $P(A) = (0.2, 0.4, 0.4)$. Using a similar procedure to find $P(B)$ (now summing out A) we get $P(B) = (0.3, 0.3, 0.4)$.

In order to find $P(A|B)$ we use

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

hence for $B = b_1$ we get

$$P(A|b_1) = \frac{P(A, B = b_1)}{P(B = b_1)} = \frac{(0.05, 0.15, 0.10)}{0.3} = (0.167, 0.5, 0.333).$$

The operations are similar for the other states of B, which gives $P(A|b_2) = (0.333, 0, 0.667)$ and $P(A|b_3) = 0.125, 0.625, 0.25)$.

Last we need $P(B|A)$. We can calculate these probabilities following the same steps as for $P(A|B)$, resulting in

$$\begin{aligned} P(B|a_1) &= (0.25, 0.5, 0.25) \\ P(B|a_2) &= (0.375, 0, 0.625) \\ P(B|a_3) &= (0.25, 0.5, 0.25). \end{aligned}$$

Exercise 4 *

Consider the binary variable A and the ternary variable B. Assume that B has the probability distribution $P(B) = (0.1, 0.5, 0.4)$ and that A has the conditional probability distribution given in Table 3.

	b_1	b_2	b_3
a_1	0.1	0.7	0.6
a_2	0.9	0.3	0.4

Table 3: The conditional probability distribution $P(A|B)$.

Questions:

1. Verify that Table 3 specifies a valid conditional probability distribution.
2. Calculate $P(B|A)$. *Hint:* Consider Bayes rule illustrated by the temperature-sensor example that we discussed in the lecture

Solution:

1. We need to check that $\sum_A P(A|B = b_i) = 1$ for each state b_i of B. This is the case in Table 3, hence the we have a valid conditional probability distribution.
2. According to Bayes rule we have

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

Based on the probabilities given in the exercise, we have for the numerator that

$$P(A|B)P(B) = \begin{array}{|c|c|c|c|}\hline & b_1 & b_2 & b_3 \\ \hline a_1 & 0.1 & 0.7 & 0.6 \\ a_2 & 0.9 & 0.3 & 0.4 \\ \hline \end{array} \cdot (0.1, 0.5, 0.4) = \begin{array}{|c|c|c|c|}\hline & b_1 & b_2 & b_3 \\ \hline a_1 & 0.01 & 0.35 & 0.24 \\ a_2 & 0.09 & 0.15 & 0.16 \\ \hline \end{array}$$

Note that

- $P(A|B)P(B)$ is also equal to $P(A, B)$ (an instance of the *chain rule*)!
- the multiplications are done component-wise so that we only multiply entries that match wrt. the state labels.

For the denominator, we can use the intermediate result ($P(A, B)$) that we just calculated above:

$$P(A) = \sum_B P(A, B) = \sum_B \left(\begin{array}{|c|c|c|c|}\hline & b_1 & b_2 & b_3 \\ \hline a_1 & 0.01 & 0.35 & 0.24 \\ a_2 & 0.09 & 0.15 & 0.16 \\ \hline \end{array} \right) = (0.6, 0.4).$$

	$A = \mathit{yes}$	$A = \mathit{no}$
$T = \mathit{yes}$	0.99	0.001
$T = \mathit{no}$	0.01	0.999

Table 4: Table for Exercise 5. Conditional probabilities $P(T \mid A)$ characterizing test T for A.

Plugging these results into Bayes rule we get:

$$
\begin{aligned}
P(B|A) &= \begin{array}{|c|c|c|c|} \hline & b_1 & b_2 & b_3 \\ \hline a_1 & 0.01 & 0.35 & 0.24 \\ a_2 & 0.09 & 0.15 & 0.16 \\ \hline \end{array} \Big/ (0.6, 0.4) \\
&= \begin{array}{|c|c|c|c|} \hline & b_1 & b_2 & b_3 \\ \hline a_1 & 0.01/0.6 & 0.35/0.6 & 0.24/0.6 \\ a_2 & 0.09/0.4 & 0.15/0.4 & 0.16/0.4 \\ \hline \end{array} \\
&= \begin{array}{|c|c|c|c|} \hline & b_1 & b_2 & b_3 \\ \hline a_1 & 0.0167 & 0.5833 & 0.4000 \\ a_2 & 0.2250 & 0.3750 & 0.4000 \\ \hline \end{array}
\end{aligned}
$$

Exercise 5 * Table 4 describes a test T for an event A. The number 0.01 is the frequency of *false negatives*, and the number 0.001 is the frequency of *false positives*.

(*i*) The police can order a blood test on drivers under the suspicion of having consumed too much alcohol. The test has the above characteristics. Experience says that 20% of the drivers under suspicion do in fact drive with too much alcohol in their blood. A suspicious driver has a positive blood test. What is the probability that the driver is guilty of driving under the influence of alcohol?

(*ii*) The police block a road, take blood samples of all drivers, and use the same test. It is estimated that one out of 1,000 drivers have too much alcohol in their blood. A driver has a positive test result. What is the probability that the driver is guilty of driving under the influence of alcohol?

Hint: Structure-wise this exercise is closely connected to the temperature-sensor example that we discussed in the lecture.

Solution: The solution procedure is the same as for the previous exercise, i.e., use Bayes rule to calculate the probabilities.

$$
\begin{aligned}
P(A = y \mid T = p) &= \frac{P(A = y, T = p)}{P(T = p)} \\
&= \frac{P(T = p \mid A = y) P(A = y)}{P(T = p)}
\end{aligned}
$$

From the problem specification, we have the numbers to put into the numerator, but we still need to calculate the denominator:

$$\begin{aligned} P(T=p) &= P(T=p, A=t) + P(T=p, A=f) \\ &= P(T=p \,|\, A=t)P(A=t) + P(T=p \,|\, A=f)P(A=f) \end{aligned}$$

By plugging this into the previous expression we get:

$$\begin{aligned} P(A=y \,|\, T=p) &= \frac{P(T=p \,|\, A=y)P(A=y)}{P(T=p \,|\, A=t)P(A=t) + P(T=p \,|\, A=f)P(A=f)} \\ &= \frac{0.99 \cdot 0.2}{0.99 \cdot 0.2 + 0.001 \cdot 0.8} \\ &= 0.996 \end{aligned}$$

In the second part of the exercise we change the prior probability $P(A)$ for A and update the calculations using the new probabilities. This gives $P(A = t) = 0.498$. Notice how the change in prior distribution affected the posterior distribution for A; the accuracy of a test should always be considered relative to the frequency of the event which you are trying to predict.

Exercise 6 * A routine DNA test is performed on a person (this exercise is set in the not too distant future!). The test T gives a positive result for a rare genetic mutation M linked to Alzheimer's disease. The mutation is present in only 1 in a million people. The test is 99.99% accurate, i.e. it will give a wrong result in 1 out of 10000 tests performed. Should the person be worried, i.e., what is the probability that the person has the mutation given that the test showed a positive result?

Hint: This is partly a modeling exercises and partly a calculation exercise. First you need to formalize the problem:

- What are the relevant variables and what states do they have?
- Based on the description above, what probability distributions can you infer for the variables?

Based on this formalization, you need to find the rules required to answer the question about the probability of a mutation given a positive test result.

Solution: We analyze the problem using the two random variables $\mathit{Test} \in \{\mathit{positive}, \mathit{negative}\}$, $\mathit{Mutation} \in \{\mathit{present}, \mathit{absent}\}$. What our patient should be interested in is the probability of having the mutation, given everything he/she knows, i.e. the fact that there was a positive test result. Thus, we need to compute

$$P(\mathit{Mutation} = \mathit{present} \mid \mathit{Test} = \mathit{positive}).$$

According to Bayes rule, this is equal to

$$P(\mathit{Test} = \mathit{positive} \mid \mathit{Mutation} = \mathit{present}) \frac{P(\mathit{Mutation} = \mathit{present})}{P(\mathit{Test} = \mathit{positive})}$$

	b_1	b_2
a_1	$(0.006, 0.054)$	$(0.048, 0.432)$
a_2	$(0.014, 0.126)$	$(0.032, 0.288)$

Table 5: $P(A, B, C)$ for Exercise 7.

Of the probabilities on the right, we have $P(\mathit{Test} = \mathit{positive} \mid \mathit{Mutation} = \mathit{present}) = 0.9999$, and $P(\mathit{Mutation} = \mathit{present}) = 0.000001$. We still need $P(\mathit{Test} = \mathit{positive})$. This can be computed as

$$\begin{aligned}
P(\mathit{Test} = \mathit{positive}) &= P(\mathit{Test} = \mathit{positive}, \mathit{Mutation} = \mathit{present}) \\
&\quad + P(\mathit{Test} = \mathit{positive}, \mathit{Mutation} = \mathit{absent}) \\
&= P(\mathit{Test} = \mathit{positive} \mid \mathit{Mutation} = \mathit{present}) P(\mathit{Mutation} = \mathit{present}) \\
&\quad + P(\mathit{Test} = \mathit{positive} \mid \mathit{Mutation} = \mathit{absent}) P(\mathit{Mutation} = \mathit{absent}) \\
&= 0.9999 \cdot 0.000001 + 0.0001 \cdot 0.999999 \sim 0.0001
\end{aligned}$$

We now get

$$P(\mathit{Mutation} = \mathit{present} \mid \mathit{Test} = \mathit{positive}) = 0.9999 \cdot \frac{0.000001}{0.0001} = 0.009999.$$

Thus, there is only about a 1% chance that the patient has the mutation.

Exercise 7 * In Table 5, a joint probability table for the binary variables A, B, and C is given.

- Calculate $P(B, C)$ and $P(B)$.
- Are A and C independent given B?

Solution:

For the first part of the exercise, consider calculating $P(B, c_1)$. To do that we marginalize out A, hence

$$P(B, c_1) = (0.006 + 0.014, 0.048 + 0.032) = (0.02, 0.08).$$

The remaining probabilities are calculated similarly, and we get:

(i) $P(B, c_1) = (0.02, 0.08), P(B, c_2) = (0.18, 0.72), P(B) = (0.2, 0.8)$

For the second part of the exercise we can check that

$$P(A|C = c_1, B) = P(A|C = c_2, B)$$

to verify that A and C are independent given B. These probabilities can be calculated as

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

where the denominator was calculated in the first part of the exercise. When doing the calculations we get

(ii) $P(A|b_1, c_1) = (0.3, 0.7) = P(A|b_1, c_2), P(A|b_2, c_1) = (0.6, 0.4) = P(A|b_2, c_2),$

hence A and C are independent given B.

Exercise 8 Solve Exercise 1(a-d) in PM.

Solution:

$$\begin{aligned} P(A = t \,|\, G = m) &= \frac{37}{69} = 0.54 \\ P(A = t \,|\, G = f) &= \frac{14}{31} = 0.45 \\ P(A = t \,|\, G = m, D = 1) &= \frac{32}{50} = 0.64 \\ P(A = t \,|\, G = f, D = 1) &= \frac{7}{10} = 0.7 \\ P(A = t \,|\, G = m, D = 2) &= \frac{5}{19} = 0.26 \\ P(A = t \,|\, G = f, D = 2) &= \frac{7}{21} = 0.3 \end{aligned}$$

Exercise 9 Continue with the exercises from last time.

Exercises for MI

Exercise sheet 5

Thomas Dyhre Nielsen

When you have finished with the exercises, you should continue with the exam sheet from the previous years, which can be found at the course's home page.

Exercise 1* In the graphs in Figures 1 and 2, determine which variables are d-separated from A. Note that it is sufficient to find a single open path along which evidence can be transmitted; if such a path exists then the variables are *not* d-separated (instead they are said to be d-connected).

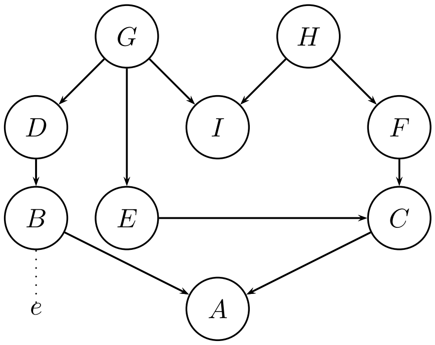


Figure 1: Figure for Exercise 1.

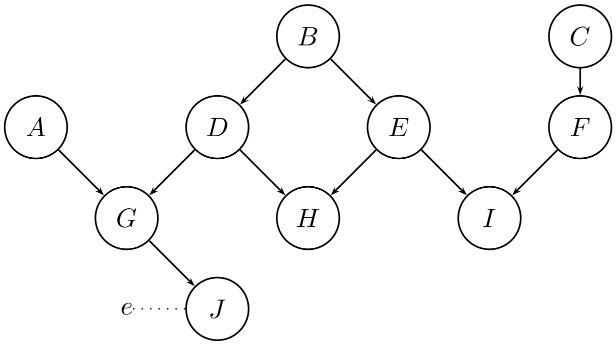


Figure 2: Figure for Exercise 1.

Solution:

In (a), all variables are d-connected to A. We can see that by checking whether there *exists* an open path between each pair of variables. Take A and D for instance. The path $A \leftarrow C \leftarrow F \leftarrow H \rightarrow I \leftarrow G \rightarrow D$ is closed, since it contains the connection $(H \rightarrow I \leftarrow G)$, which is converging and there is no evidence on I (nor on any of its descendants for which it has none). However, the path $A \leftarrow C \leftarrow E \leftarrow G \rightarrow D$ is open since each triplet of nodes along the path (i.e., $(A \leftarrow C \leftarrow E)$, $(C \leftarrow E \leftarrow G)$, $(E \leftarrow G \rightarrow D)$ defines an open connection according to the d-separation rules. With this path we have established the existence of an open path, and we therefore have that A and D are no d-separated given the evidence provided.

In (b), all variables except C and F are d-connected to A.

Exercise 2* Consider the network in Figure 3.

- What are the minimal set(s) of variables that we should have evidence on in order to d-separate C and E (that is, sets of variables for which no proper subset d-separates C and E)?
- What are the minimal set(s) of variables we should have evidence on in order to d-separate A and B?
- What are the maximal set(s) of variables that we can have evidence on and still d-separate C and E (that is, sets of variables for which no proper superset d-separates C and E)?
- What are the maximal set(s) of variables that we can have evidence on and still d-separate A and B?

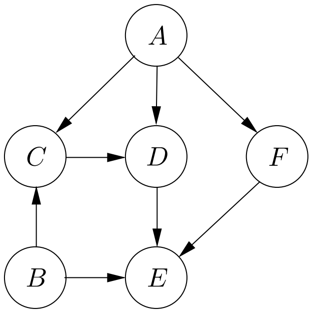


Figure 3: A causal network for Exercise 3.

Solution:

Minimal sets d-separating C and E: $\{B, D, F\}$ and $\{A, B, D\}$.

Minimal sets d-separating A and B: $\emptyset$.

Maximal set d-separating C and E: $\{A, B, D, F\}$.

Maximal set d-separating A and B: $\{F\}$.

Exercise 3 Consider the network in Figure 3. Which conditional probability tables must be specified to turn the graph into a Bayesian network?

Solution: The needed tables are $P(A)$, $P(B)$, $P(C \mid A, B)$, $P(D \mid A, C)$, $P(E \mid B, D, F)$, and $P(F \mid A)$.

Exercise 4 Construct a Bayesian network for Exercise 5 in the last exercise sheet.

Solution: See the .

Exercise 5* Peter is currently taking three courses on the topics of probability theory, linguistics, and algorithmics. At the end of the term he has to take an exam in two of the courses, but he has yet to be told which ones. Previously he has passed a mathematics and an English course, with good grades in the mathematics course and outstanding grades in the English course. At the moment, the workload from all three courses combined is getting too big, so Peter is considering dropping one of the courses, but he is unsure how this will affect his chances of getting good grades in the remaining ones. What are reasonable variables of interest in assessing Peter's situation? How do they group into information, hypothesis, and mediating variables?

Hint: The grades that Peter has already received constitute evidence about certain variables (i.e., variables for which we observe the states they are in).

Solution:

There are two information variables, *English Grade* and *Math Grade* and three hypothesis variables, *Prob. Grade*, *Ling. Grade*, and *Alg. Grade*. Mediating variables could be *Mathematical Talent* and *Linguistic Talent*.

Exercise 6* Construct a Bayesian network (ignoring the probabilities) and follow the reasoning in the following story based on how information is transmitted in the networks (according the d-separation rules). Mr. Holmes is working in his office when he receives a phone call from his neighbor, who tells him that Holmes' burglar alarm has gone off. Convinced that a burglar has broken into his house, Holmes rushes to his car and heads for home. On his way, he listens to the radio, and in the news it is reported that there has been a small earthquake in the area. Knowing that earthquakes have a tendency to turn on burglar alarms, he returns to work.

Solution: See the Hugin network.

Exercise 7 We want to construct a Bayesian network for the following random variables (all with domain *true,false*):

Mexico	Person X has recently travelled to Mexico
Svine_flu_infection	Person X has been infected with the svine flu virus
Vaccination	Person X has been vaccinated against svine flu
Svine_flu_sick	Person X is sick from svine flu
Fever	Person X has fever

a. Use the above ordering of the variables to determine a Baysian network structure based on the chain rule and conditional independence relations.

b. Repeat the construction with the alternative variable ordering:

Vaccination,Mexico,Svine_flu_sick,Fever,Svine_flu_infection

Solution:

a. We write the joint distribution of the variables using the chain rule:

$$\begin{aligned}&P(M, S\_f\_i, V, S\_f\_s, F) =\\ &\quad P(M)\cdot\\ &\quad P(S\_f\_i \mid M)\cdot\\ &\quad P(V \mid M, S\_f\_i)\cdot\\ &\quad P(S\_f\_s \mid M, S\_f\_i, V)\cdot\\ &\quad P(F \mid M, S\_f\_i, V, S\_f\_s)\cdot)\end{aligned}$$

We now simplify the conditional distributions by making some conditional independence assumptions (these are reasonable assumptions based on our knowledge of the domain – but not necessarily provably correct).

$$P(V \mid M, S\_f\_i) = P(V \mid M)$$

Justification: whether or not a person gets a vaccination may depend on whether he plans to travel to mexico. His decision to get a vaccination (generally) takes place before he the potential time of infection, so the decision is independent of whether he will catch the virus.

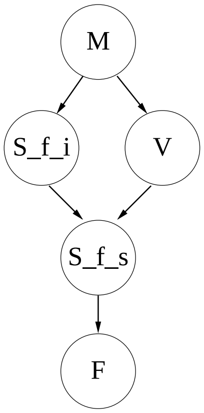
$$P(S\_f\_s \mid M, S\_f\_i, V) = P(S\_f\_s \mid S\_f\_i, V)$$

Justification: Given that we know whether a person has been infected with the virus, and whether or not he has a vaccination, it is no longer relevant to know where he may have contracted the virus, especially whether or not he traveled to Mexico. Note that this does not mean that $P(S\_f\_s \mid M) = P(S\_f\_s)$: as long as we don't know about S_f_s and V, the information of whether the person traveled to Mexico is still very relevant for the probability of S_f_s.

$$P(F \mid M, S\_f\_i, V, S\_f\_s) = P(F \mid S\_f\_s)$$

Justification: the fever is a direct consequence of the sickness. Once we know whether a person is sick, it gives us no additional useful information (regarding the probability of fever) to know whether the person was in Mexico, or got a vaccination.

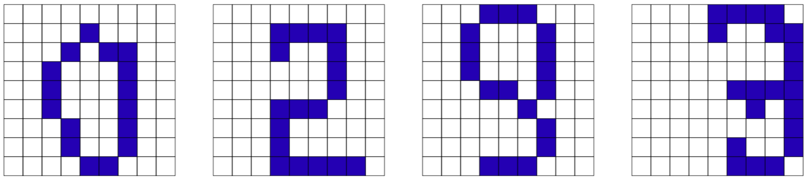
The representation of the joint distribution as the product of the simplified conditional distributions corresponds to the Bayesian network structure



b.

Going through the same procedure for the alternative ordering leads to fewer simplifications, and a more complicated network structure. The basic reason for this is that the second ordering does not respect the cause-effect relationships between the variables (i.e., taking cause variables before their effects).

Exercise 8* Design a Bayesian network that can be used to recognize handwritten digits 0,1,2,. . . ,9 from scanned, pixelated images like these:



- What are hypothesis and information variables?
- Could there be any useful mediating variables (consider e.g. the last image above)?
- How could you design a network structure
  - so that the conditional independencies are (approximately) reasonable
  - so that specification and inference complexity remain feasible

- How do you fill in the conditional probability tables?

Solution:

- The hypothesis variable(s) must enable us to make the intended inferences by querying these variables. Here we are interested in predicting which digit is actually represented by the image. Thus, we should have a variable

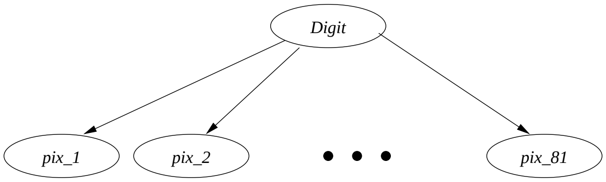
  $$\mathit{digit} \in \{0, 1, \ldots, 9\}$$

  The information variables must allow us to enter into the Bayesian networks the relevant observations which we make. Here we observe $9 \cdot 9 = 81$ pixels, each of which can be black or white:

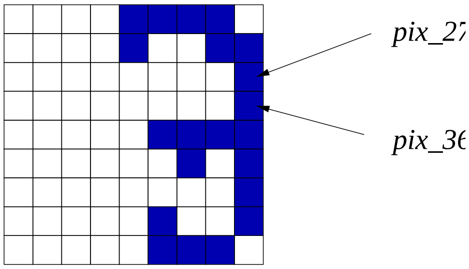
  $$\mathit{pixel\_i} \in \{b, w\} \qquad i = 1, \ldots, 81$$

- Mediating variables: see below

- Writing a given digit can be seen as a cause for the pixels to become black or white. Just inserting edges for these direct cause-effect relationships gives us the structure



According to this structure, any two pixels are conditionally independent given the digit (this type of structure is also called a *Naive Bayes Model*). This is not entirely realistic: for example consider pixels 27 and 36:



According to the network structure:

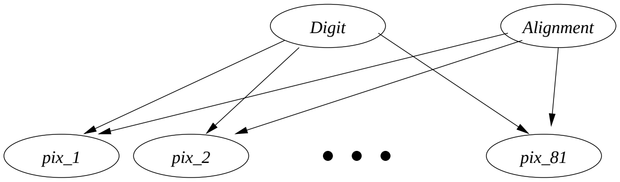
$$P(\mathit{pixel\_27} = b \mid \mathit{digit} = 3, \mathit{pixel\_36}{=}b) = P(\mathit{pixel\_27} = b \mid \mathit{digit} = 3),$$

i.e. knowing that the actual digit is a 3, the information on pixel 36 carries no relevant information for pixel 27. However, if in addition to *digit*=3 we know that *pixel_36*=b, this will indicate that the digit is written flush right in the box, and should thereby increase the probability of *pixel_27*=b.

We can improve the model by introducing a mediating variable

$$alignment \in \{l, r, c\}$$

(for left, right, and center alignment) and add it to the network like this:



Now two pixels are only independent given both *digit* and *alignment* – which still will not make this a perfectly accurate model, but already better than the first one.

Exercise 9 You are confronted with three doors, A, B, and C. Behind exactly one of the doors there is \$10,000. When you have pointed at a door, an official will open another door with nothing behind it. After he has done so, you are allowed to alter your choice. Should you do that (i.e., will altering your choice improve your chances of winning the prize)?

Solution: See the Hugin network.

Exercise 10 *This exercise is intended to be solved using Hugin. If you would like more experience doing probability updating manually, you can also solve the first part of the exercise by hand. In that case, you should take into account the evidence given in the description when constructing the tables so that you have fewer numbers to deal with, i.e., throw away the parts of the tables inconsistent with the evidence.*

Consider the insemination example from Section 3.1.13 in BNDG (pdf available form the lecture sheet). Let the probabilities be as in Table 1 ($Ho = y$ means that hormonal changes have taken place) $P(Pr) = (0.87, 0.13)$.

(i) What is $P(Pr \mid BT = n, UT = n)$?

(ii) Construct a naive Bayes model. Determine the conditional probabilities for the model by making inference queries in the model above using Hugin. What is $P(Pr \mid BT = n, UT = n)$ in this model and how does it compare to the result you got above? Try to (qualitatively) account for any differences.

Solution: *Part 1:*

$$P(Pr|Bt = n, Ut = n) = \frac{P(Pr, Bt = n, Ut = n)}{P(Bt = n, Ut = n)}$$

	$Pr = y$	$Pr = n$
$Ho = y$	0.9	0.01
$Ho = n$	0.1	0.99

	$Ho = y$	$Ho = n$
$BT = y$	0.7	0.1
$BT = n$	0.3	0.9

	$Ho = y$	$Ho = n$
$UT = y$	0.8	0.1
$UT = n$	0.2	0.9

Table 1: Tables for Exercise 10.

Next, we should calculate $P(Pr, Bt = n, Ut = n)$ and $P(Bt = n, Ut = n)$:

$$P(Pr, Bt = n, Ut = n) = \sum_{Ho} P(Pr, Bt = n, Ut = n, Ho)$$
$$P(Bt = n, Ut = n) = \sum_{Pr} P(Pr, Bt = n, Ut = n)$$

Thus, we only need to calculate $P(Pr, Bt = n, Ut = n, Ho)$ and this can be done using the chain rule:

$$P(Pr, Bt = n, Ut = n, Ho) = P(Ut = n|Ho)P(Bt = n|Ho)P(Ho|Pr)P(Pr)$$

The final value is $P(Pr|Bt = n, Ut = n) = (0.53, 0.47)$. See also the Hugin network.

Part 2:

The following probabilities can be calculated from the original network by inserting the evidence $Pr = y$ and $Pr = n$, respectively:

$$\begin{aligned} P(BT|Pr = y) &= (0.64, 0.36) \\ P(BT|Pr = n) &= (0.106, 0.894) \\ P(UT|Pr = y) &= (0.73, 0.27) \\ P(UT|Pr = n) &= (0.107, 0.893) \end{aligned}$$

Using the calculated probabilities in a nave Bayes structure, we get

$$\begin{aligned} P(Pr|Bt = n, Ut = n) &= \frac{P(Bt = n, Ut = n|Pr)P(Pr)}{P(Bt = n, Ut = n)} \\ &= \frac{P(Bt = n|Pr)P(Ut = n|Pr)P(Pr)}{P(Bt = n, Ut = n)} \\ &= (0.449, 0.551) \end{aligned}$$

Note:

- In the second step we exploit that BT and UT are conditionally independent given Pr in the naive Bayes model.

- Dividing with $P(Bt = n, Ut = n)$ simply normalizes the results so that we get a proper conditional probability distribution; we can easily calculate this value based on the numerator in the expression above $P(Bt = n, Ut = n) = P(Bt = n|Pr = y)P(Ut = n|Pr = y)P(Pr = y) + P(Bt = n|Pr = n)P(Ut = n|Pr = n)P(Pr = n)$.

Notice the difference between this result and the result you got from the original network.

Exercise 11 *Use Hugin to solve this exercise* The following relations hold for the Boolean variables
A, B, C, D, E, and F:

$(A \vee \neg B \vee C) \wedge (B \vee C \vee \neg D) \wedge (\neg C \vee E \vee \neg F) \wedge (\neg A \vee D \vee F) \wedge$
$(A \vee B \vee \neg C) \wedge (\neg B \vee \neg C \vee D) \wedge (C \vee \neg E \vee \neg F) \wedge (A \vee \neg D \vee F)$.

(i) Is there a truth value assignment to the variables making the expression true? (Hint: Represent the expression as a Bayesian network.)

(ii) We receive the evidence that A is false and B is true. Is there a truth value assignment to the other variables making the expression true?

Solution:

(1) See the Hugin network here. As the probability of $Result = y$ is positive, there are assignments of truth values making the expression true.

(ii) Insert $A = n$ and $B = n$ as evidence and propagate. As $P(Result = y) > 0$, there are assignments of the remaining variables making the expression true. If you insert "$Result = y$" and propagate, you see that the assignments must be $C = y, D = y, E = y, F = y$.

Exercise 12 (* only part of it)

For 10000 emails in your inbox you determine the values of the following three boolean variables:

Spam	the email is spam
Caps	the subject line is in all capital letters
Pills	Body of the message contains the word "pills"

You obtain the following counts:

	Caps			
	yes		*no*	
	Pills		*Pills*	
Spam	*yes*	*no*	*yes*	*no*
yes	150	850	600	3400
no	1	99	49	4851

Are

- *Spam* and *Caps* independent?
- *Pills* and *Caps* independent?
- *Pills* and *Caps* independent given *Spam*?
- *Spam* and *Caps* independent given *Pills*?

Solution:

To determine whether *Spam* and *Caps* are independent, we consider the joint and marginal distribution of these two variables:

	Caps		
Spam	*yes*	*no*	
yes	0.1	0.4	0.5
no	0.01	0.49	0.5
	0.11	0.89	

The entries are obtained by summing over the *Pills* variable, and normalizing by dividing by 10000. E.g. $0.4 = \frac{600+3400}{10000}$. Since, for example $P(\mathit{Spam} = \mathit{no}, \mathit{Caps} = \mathit{yes}) = 0.01 \neq P(\mathit{Spam} = \mathit{no})P(\mathit{Caps} = \mathit{yes}) = 0.5 \cdot 0.11 = 0.055$, we see that the two variables are not independent.

To see whether *Spam* and *Caps* are independent given *Pills*, we first determine the conditional distribution of *Spam* and *Caps* given *Pills*=*yes*:

	Caps		
Spam	*yes*	*no*	
yes	0.1875	0.75	0.9375
no	0.00125	0.06125	0.0625
	0.18875	0.81125	

Here, e.g. $0.1875 = \frac{150}{800}$ (800 is the total number of cases with *Pills*=*yes*).

Again, we find that the joint distribution is not the product of the marginals, e.g. $0.0625 \cdot 0.18875 = 0.0118 \neq 0.00125$. Thus, *Spam* and *Caps* are not independent given *Pills*.

For the (conditional) independence of *Pills* and *Caps* one proceeds in the same manner. Now the findings should be: *Pills* and *Caps* are not independent, but *Pills* and *Caps* are independent given *Spam*. For the last result one has to check both the conditional distribution given *Spam*=*yes* and given *Spam*=*no* (above it was enough to consider the conditional distribution of *Spam* and *Caps* given *Pills*=*yes*, because there we already found that *Spam* and *Caps* are not conditionally independent).

Exercise 13 Continue with the exercises from last time.

Exercises for MI

Exercise sheet 6

Thomas Dyhre Nielsen

When you have finished with the exercises, you should continue with the exam sheet from the previous years, which can be found at the course's home page.

Exercise 1*

For the Bayesian network on slide 6.16:

- define (somewhat) reasonable conditional probability tables for the five nodes of the network (use only probability values 0,0.1,0.2,. . . ,0.9,1 in order to facilitate the subsequent computations)

- perform the variable elimination computations of slide 6.17 to determine the conditional probability $P(MC \mid B = t)$ according to the numbers you specified.

Solution:

Assume that we have the following probability tables:

	Burglary	
	t	f
	.1	.9

	Earthquake	
	t	f
	.1	.9

		Alarm	
Burglary	Earthquake	t	f
t	t	.9	.1
t	f	.8	.2
f	t	.5	.5
f	f	.1	.9

	JohnCalls	
Alarm	t	f
t	.8	.2
f	.1	.9

	MaryCalls	
Alarm	t	f
t	.7	.3
f	.1	.9

The abstract computation on slide 6.16 are given as:

$$\sum_{a\in\{t,f\}}\sum_{eq\in\{t,f\}}\sum_{jc\in\{t,f\}} P(B=t)P(EQ=eq)P(A=a \mid B=t, EQ=eq)P(JC=jc \mid A=a)P(MC \mid A=a) =$$

$$\sum_{a\in\{t,f\}}\sum_{eq\in\{t,f\}} P(B=t)P(EQ=eq)P(A=a \mid B=t, EQ=eq)P(MC \mid A=a)F_1(a) =$$

$$\sum_{a\in\{t,f\}} P(B=t)P(MC \mid A=a)F_1(a)F_2(a) =$$

$$P(B=t)F_3(MC)$$

With the concrete tables we have specified for the conditional distributions, we can now compute: to eliminate the variable JohnCalls, we multiply all tables that contain the variable JohnCalls, and sum out the JohnCalls variable. The result is the table, or factor, F_1. Since there is only one table containing JohnCalls, the result is very simple:

	$F_1(Alarm)$	
	t	f
	1	1

Next we eliminate *Earthquake*. There are two tables containing Earthquake: the table of the Earthquake node, and the conditional distribution of Alarm. The latter table is restricted to the cases that are consistent with the observed evidence $B = t$, which gives a table only containing Earthquake and Alarm:

	Alarm	
Earthquake	t	f
t	.9	.1
f	.8	.2

Multiplying this with the Earthquake table gives

	Alarm	
Earthquake	t	f
t	.09	.01
f	.72	.18

Summing out the Earthquake variable then gives the factor F_2:

	$F_2(Alarm)$	
	t	f
	0.81	0.19

Finally, we eliminate *Alarm*. Multiplying F_1, F_2, and the table for MaryCalls gives

	MaryCalls	
Alarm	t	f
t	$.7\cdot$ 0.81=0.567	$.3 \cdot$ 0.81=0.243
f	$.1\cdot$ 0.19 = 0.019	$.9 \cdot$ 0.19= 0.171

Summing out Alarm gives

	$F_3(MaryCalls)$	
	t	f
	0.586	0.414

This multiplied with $0.1 = P(B = t)$ gives

	$P(B = t)F_3(MaryCalls)$	
	t	f
	0.0586	0.0414

which is now the table containing the function $P(B = t, MC)$. To obtain the conditional distribution $P(MC \mid B = t)$ this has only to be normalized by dividing with 0.0586+0.0414, which gives

	$P(MaryCalls \mid B = t)$	
	t	f
	0.586	0.414

Note that by inspecting the semantics of the potentials calculated above, we could have stopped after summing out Alarm, since the resulting potential is the conditional distribution $P(MC \mid B = t)$.

Exercise 2* Complete Exercise 8.10 in PM.

Solution: To calculate $P(E)$ you can start by removing D and F, since they are both barren, i.e., the result of marginalizing out these two variables will simply be unity factors. Thus, we end up with $P(A)$, $P(B)$, $P(C|A, B)$, and $P(E|C)$.

Start by eliminating A:

This will create a factor $F_1(C, B) = \sum_A P(A)P(C|A, B)$, with the intermediate factor $P(A)P(C|A, B)$

		C	
A	B	t	f
t	t	0.09	0.81
t	f	0.72	0.18
f	t	0.07	0.03
f	f	0.04	0.06

and the final factor

	C	
B	t	f
t	0.16	0.84
f	0.76	0.24

Eliminate B:

This will create a factor $F_2(C) = \sum_B P(B)F_1(C|B) = (0.64, 0.36)$.

Eliminate C:

This will create a factor $F_3(E) = \sum_C P(E|C)F_2(C) = (0.52, 0.48)$.

Exercise 3 Consider the network defined by the two binary variables A and B, where A is the parent of B. Assume that the conditional probability tables are given as $P(A) = (0.1, 0.9)$ and

	A	
	a_1	a_2
b_1	0.05	0.2
b_2	0.95	0.8

- Assume that you want to estimate $P(b_1)$ using sampling. How many samples would be required if you only accept an error larger than 0.15 in 10% of the cases?
- Implement the network above in Hugin and use Hugin to sample the number of cases that you have just calculated; use the function 'Simulate cases' under 'File'.
- Use the sampled cases to estimate $P(b_1)$ and compare the result with Hugin. Feel free to use a spreadsheet for the counting.

Solution: Hoeffding's inequality gives us

$$P(|s - p| > 0.15) \leq 2e^{-2n0.15^2} < 0.1,$$

which we can rewrite to find that $n > 66.57$.

Exercise 4* Consider again the network in the exercise above, and assume that you want to use rejection sampling to estimate $P(A|B = b_1)$. How many samples do you expect you would have to generate in order to end up (after rejection) with a sample set of 1000 cases for estimating the probability.

Solution:

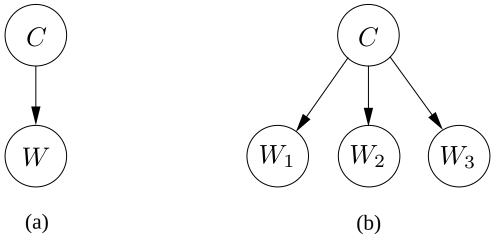
The probability of $B = b_1$ is $P(B = b_1) = 0.185$, hence only 18.5% of all the generated samples would not be rejected. We therefore need to generate approximately 5400 cases to end up with a sample set of 1000 cases.

Exercise 5

Complete Exercise 8.6(a-b) in PM.

Solution:

We have two variables in the network: *Company* (C) with states *green* (g) and *blue* (b) and *Witness* (W) with states *green* (g) and *blue* (b). The structure of the network incorporating one witness is illustrated in Figure 1(a).



(a)

(b)

Figure 1: The witness model: Model (a) includes one witness and model (b) includes three witnesses.

The conditional probability tables for the model are given by:

$P(C=g)$	$P(C=b)$
0.85	0.15

	$C=g$	$C=b$
$W=g$	0.8	0.2
$W=b$	0.2	0.8

$P(W \mid C)$

For calculating $P(C = b \mid W = b)$ we use Bayes rule:

$$\begin{aligned}
P(C=b \mid W=b) &= \frac{P(W=b \mid C=b)P(C=b)}{P(W=b)} \\
&= \frac{P(W=b \mid C=b)P(C=b)}{P(C=b, W=b) + P(C=g, W=b)} \\
&= \frac{0.8 \cdot 0.15}{0.8 \cdot 0.15 + 0.2 \cdot 0.15} \\
&= \frac{0.12}{0.12 + 0.03} = 0.8
\end{aligned}$$

With three independent witnesses we have the network in Figure 1(b), where we assign the same conditional probability distribution to each of the three witnesses. Thus, we have a naive Bayes model with three information variables.

We can perform probability updating in this model as shown on the slides in the previous lecture. E.g.

$$\begin{aligned} P(C = b \,|\, W_1 = W_2 = W_3 = b) &= \frac{P(W_1 = b \,|\, C = b) P(W_2 = b \,|\, C = b) P(W_3 = b \,|\, C = b) P(C = b)}{P(W = b)} \\ &= \frac{0.8 \cdot 0.8 \cdot 0.8 \cdot 0.15}{0.8 \cdot 0.8 \cdot 0.8 \cdot 0.15 + 0.2 \cdot 0.2 \cdot 0.2 \cdot 0.85} = 0.92 \end{aligned}$$

Exercise 6

Continue with the exercises from last time.

Exercises for MI

Exercise sheet 7

Thomas Dyhre Nielsen

Note: Some of the exercises below asks you to solve the exercises using Weka. If you feel adventurous (or perhaps would like to get some hands-on programming experience) you are also most welcome to solve these exercises using other (programming) tools such as scikit-learn, which support decision tree learning.

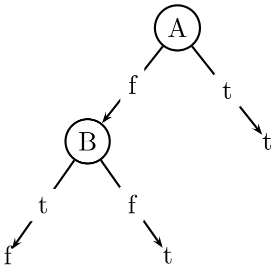
When you have completed the exercises below, continue with the remaining exercises from the last session (if any) or the decision tree related questions from the last exams.

Exercise 1* Give decision trees to represent the following Boolean functions:

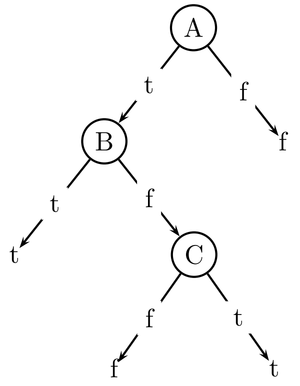
- $A \vee \neg B$
- $A \wedge (B \vee C)$
- $A\ XOR\ B$
- $(A \vee B) \wedge (B \vee C)$

Solution:

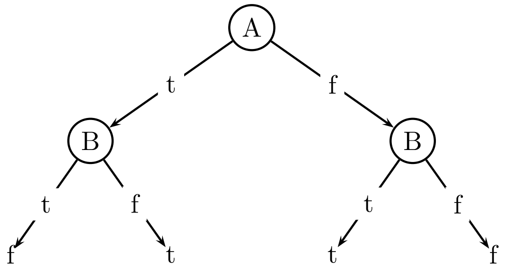
Part (a):



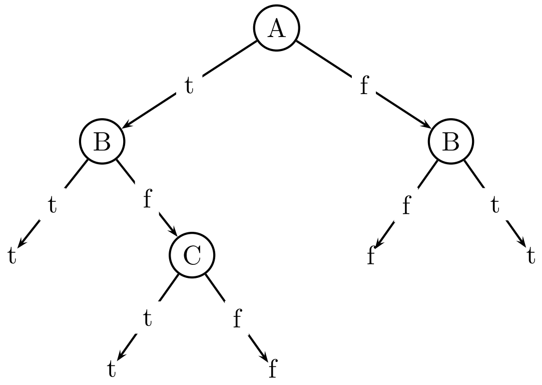
Part (b):



Part (c):



Part (d):



Exercise 2 Download and install the WEKA data-mining toolbox:

http://www.cs.waikato.ac.nz/ml/weka/

WEKA provides several user-interfaces. Select the 'Explorer' interface from the 'Applications' menu, and try the following:

- Load the 'Iris' dataset. This dataset contains measurements from 150 individual plants of the genus Iris, belonging to 3 different species 'Iris setosa', 'Iris versicolor', and 'Iris virginica'. The machine learning task associated with this dataset is: predict the species from the four measurement values.
- Use the 'Visualize' tab to get an overview of the attribute values and their relation to the class label. Sketch by hand a small decision tree for predicting the class label.
- Use WEKA's decision tree construction methods to build a decison tree (under the 'Classify' tab select e.g. J48 or the SimpleCart classifier). Compare with your own proposed decision tree.

Exercise 3

- Download the Pregnancy dataset. Note that the format of this file does not follow the standard file-format used by Weka. When trying to load the file you will therefore have to use the 'converter' suggested by Weka.
- Construct a decision tree for classification. Try to reason about the structure of the tree. Hint: have a look at the underlying Bayesian network model (which can be found here) that we have previously looked at in the course.

Exercise 4* Consider a database of cars represented by the five training examples below. The target attribute *Acceptable*, which can have values `yes` and `no`, is to be predicted based on the other attributes of the car in question. These attributes indicate a) the age of the car (*Age* having values < 5 `years` and ≥ 5 `years`), b) the make of the car (*Make* having states `Toyota` and `Mazda`), c) the number of previous owners (*#Owners* having values `1`, `2` and `3`), d) the number of kilometers (*#Kilometers* having values $> 150k$ and $\leq 150k$) and e) the number of doors (*#Doors* having values 3 and 5).

	Attributes					Target
	Age	*Make*	*#Owners*	*#Kilometers*	*#Doors*	*Acceptable*
1	< 5	`Mazda`	`1`	$> 150k$	`3`	`yes`
2	≥ 5	`Mazda`	`3`	$> 150k$	`3`	`no`
3	≥ 5	`Toyota`	`1`	$\leq 150k$	`3`	`no`
4	≥ 5	`Mazda`	`3`	$> 150k$	`5`	`yes`
5	≥ 5	`Toyota`	`2`	$\leq 150k$	`5`	`yes`

a) Calculate the entropy for the attribute *#Owners*.[1]

b) Show the decision/classification tree that would be learned by the learning algorithm assuming that it is given the training examples in the database.

[1] Note that $\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$.

c) Show the value of the information gain for each candidate attribute at each step in the construction of the tree.

Solution: For question (a):

$$ENT(\#Owners) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = 1.521 \quad (1)$$

For question (b) we need to calculate the information gain for each of the features. That is, for the generic feature X we should calculate

$$\mathit{Gain}(X) = \mathit{Ent}(\text{Accept}) - \mathit{ExpectedEntropy}(\text{Accept}|X),$$

where $\mathit{ExpectedEntropy}(\text{Accept}|X)$ is the expected entropy of *Accept* wrt. X.

For $\mathit{Ent}(\text{Accept})$ we get

$$\mathit{Ent}(\text{Accept}) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = 0.971$$

Considering now the features in sequence from left to right, we have that the expected entropy of *Age* is

$$\begin{aligned}&\mathit{ExpectedEntropy}(\text{Accept}|\text{Age})\\ &= P(\text{Age} < 5)\mathit{Ent}(\text{Accept}|\text{Age} < 5) + P(\text{Age} \geq 5)\mathit{Ent}(\text{Accept}|\text{Age} \geq 5)\\ &= \frac{1}{5}\mathit{Ent}(\text{Accept}|\text{Age} < 5) + \frac{4}{5}\mathit{Ent}(\text{Accept}|\text{Age} \geq 5).\end{aligned}$$

Here $\mathit{Ent}(\text{Accept}|\text{Age} < 5)$ and $\mathit{Ent}(\text{Accept}|\text{Age} \geq 5)$ denote the entropy of *Accept* when only considering the instances restricted to Age < 5 and Age ≥ 5, respectively. Thus we get

$$\begin{aligned}\mathit{Ent}(\text{Accept}|\text{Age} < 5) &= -\frac{1}{1}\log_2\left(\frac{1}{1}\right) - \frac{0}{1}\log_2\left(\frac{0}{1}\right) = 0\\ \mathit{Ent}(\text{Accept}|\text{Age} \geq 5) &= -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1\end{aligned}$$

Plugging these values into the expression for the expected entropy above, we get

$$\mathit{ExpectedEntropy}(\text{Accept}|\text{Age}) = \frac{1}{5}\cdot 0 + \frac{4}{5}\cdot 1 = \frac{4}{5}$$

and the information gain therefore becomes

$$\mathit{Gain}(\text{Age}) = 0.971 - \frac{4}{5} = 0.171.$$

Doing the same calculations for the remaining attributes we end up with

$$\begin{aligned}
Gain(\text{Make}) &= 0.0202\\
Gain(\#\text{Owners}) &= 0.171\\
Gain(\#\text{Kilo}) &= 0.0202\\
Gain(\#\text{Doors}) &= 0.4202
\end{aligned}$$

Since #Doors has the highest information gain we put that feature at the top of the tree.

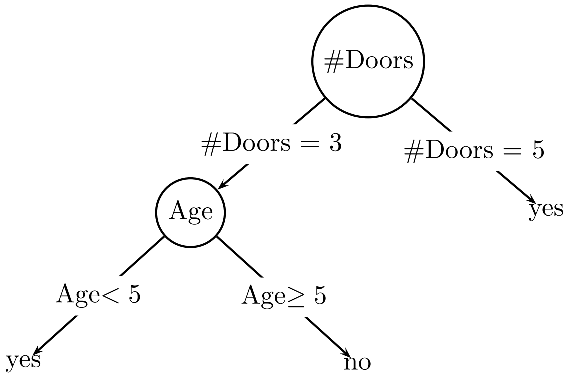
For the branch of the tree with #Doors = 5 there is nothing more to do, since both of the training examples with this feature value has the same value for *Accept*, namely *yes*. For #Doors = 3, we have two instances with *Accept* being *yes* and one instance with *Accept* being *no*. Thus, we need to make another feature test for instances where #Doors = 3. To do that we proceed along the same lines as above, expect that we now only consider the instances consistent with #Doors = 3. For this restricted data set we find the entropy for *Accept*:

$$Ent(\text{Accept}|\#\text{Doors} = 3) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) = 0.918$$

Based on this value we can calculate the information gain for each of the remaining features (this involves calculating the expected entropy for these features using the same calculation scheme as above):

$$\begin{aligned}
Gain(\text{Age}) &= 0.918 - 0\\
Gain(\text{Make}) &= 0.918 - 2/3\\
Gain(\#\text{Owners}) &= 0.918 - 2/3\\
Gain(\#\text{Kilo}) &= 0.918 - 2/3
\end{aligned}$$

Hence, for #Doors= 3 we pick *Age* as the next node. The classification tree can now directly be constructed based on the results above:



Exercise 5 (part of it: *) Solve Exercise 7.3 (except sub-question f) in PM.

Solution:

1. (a) The optimal decision tree with one node predicts Likes = false. It has 5 errors.

2. (b) The optimal prediction is to predict likes with probability 5/12. It has sum-of- squares error $5 \cdot (7/12)^2 + 7 \cdot (5/12)^2 = 2.92$.

3. (c) The optimal (with respect to sum of absolute errors) decision tree of depth 2 is: *if lawyers then Likes=true else Likes=false*. It has 3 errors. At the root are all of the examples $(e_1, \ldots, e_{13})$. Filtered to the *lawyers = true* node are $e_2, e_3, e_4, e_8, e_9, e_{10}$. Filtered to the *lawyers = false* node are $e_1, e_5, e_6, e_7, e_{11}, e_{12}$.

4. (d) The optimal (with respect to sum-of-squares error) decision tree of depth 2 is: *if lawyers then likes with probability 2/3 else likes with probability 1/6* The error is $2 \cdot (1/3)^2 + 4 \cdot (2/3)^2 + 5 \cdot (1/6)^2 + 1 \cdot (5/6)^2 = 2.83$.

5. (e) The smallest tree that correctly classifies all training examples is: *if guns then lawyers else comedy* The information gain split gives a more complicated tree that represents the same function. To construct the tree follow the same procedure as for the preceding exercise.

6. (g) It is not linearly separable. The examples e_3 and e_{10} must be on the same side of a hyperplane (as they are both true on Likes). Therefore any linear interpolation also must be on the same side, but e_4 is between these in the input categories, but has a different classification.

Exercises for MI

Exercise sheet 8

Thomas Dyhre Nielsen

Note: Some of the exercises below asks you to solve the exercises using Weka. If you feel adventurous (or perhaps would like to get some hands-on programming experience) you are also most welcome to solve these exercises using other (programming) tools such as scikit-learn, which support decision tree learning.

When you have finished with the exercises, you should continue with the exam sheet from the previous years, which can be found at the course's home page.

Exercise 1*

Suppose you want to use a neural network for predicting user preferences based on the data set shown on Slide 7.13. What neural network structure could you use, especially: what would be the input and output units?

Solution:

This is a problem of finding a numerical encoding for discrete input and output attributes. There are several possibilities:

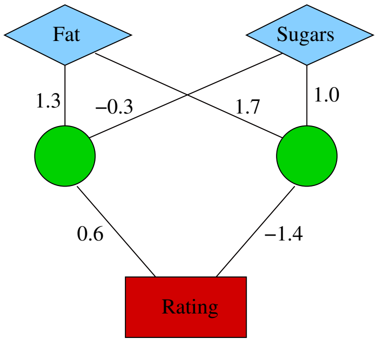
Assuming that all attributes are binary (i.e., only can have the two values that appear in the table on slide 9.05, and, e.g., 'Length' can not also have the value 'medium'), one can use a neural network with one input node for each input attribute, and, for each attribute, encode one of the possible values as 0, and the other as 1. For example, assuming that 'unknown', 'follow Up', 'short' and 'home' are the values encoded as 0, one would have that the input in the first example is given as (1,1,1,0). In the same way, the output attribute 'User Action' would be encoded.

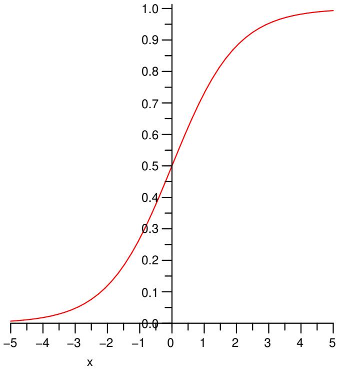
Alternatively (and this also works for attributes with more than 2 values), one can use one input node for each attribute-value combination. Here we would have the 8 different attributes Author_is_known, Author_is_unknown, . . . , WhereRead_is_home, WhereRead_is_work. Then we encode the actual inputs as 0/1 values of these new attributes. In the first example, the inputs then are set to Author_is_known=1, Author_is_unknown=0, . . . , WhereRead_is_home=1, WhereRead_is_work=0.

Exercise 2* Compute for the neural network below the *Rating* output computed for the two inputs

Fat	Sugars
1	5
0	14

The two hidden units have the sigmoid activation function. Values for this function can be either computed precisely according to the definition $\sigma(x) = 1/(1+e^{-x})$, or you can read approximate function values off the plot on the right below. The output unit has the identity activation function, i.e. the output is just the weighted sum of the inputs.





Solution:

The computation in the neural network proceeds top-to-bottom, where each node computes its output from the input it receives from the nodes in the preceding layer.

The input nodes don't perform any computations. Their output is just the input, i.e. in the first case, the *Fat* input node outputs a 1, and the *Sugars* input node outputs a 5.

Next, the two nodes in the hidden layer perform their computations. Each node first computes the weighted sum of its input, and then applies the activation function to compute the final output.

The weighted sum of inputs is:

left hidden node: $1.3 \cdot 1 + (-0.3) \cdot 5 = -0.2$
right hidden node: $1.7 \cdot 1 + 1.0 \cdot 5 = 6.7$

Now the activation function is applied to these numbers. The function value can be approximately read off the plot, or computed precisely:

output left hidden node: $1/(1+e^{0.2}) = 0.45$ *output left hidden node*: $1/(1+e^{-6.7}) = 0.998$

Next, the 'Rating' output node can compute its output. The weighted input is

$0.6 \cdot 0.45 + (-1.4) \cdot 0.998 = -1.1272.$

Since the output node has the identity activation function, this is also already the output of the 'Rating' node.

Exercise 3* Assume that we have the following training examples:

X_1	X_2	T
1	1	1
-1	1	-1
1	-1	1
-1	-1	-1

That is, with input $X_1 = 1$ and $X_2 = -1$ we want the output 1.

Consider a perceptron with threshold input 1 and with initial weights $w_0 = 0$,$w_1 = 0$ and $w_2 = 0$.

- Show the first two iterations (as on Slide 8.24) when learning a perceptron (having the sign function as activation function) using learning rate $\alpha = 0.25$ and error function $E = t - o$; t is the desired output and o is the actual output.

Solution:

First iteration:

Cases:	$(1,1,1)$	$(1,-1,1)$	$(1,1,-1)$	$(1,-1,-1)$
t	1	-1	1	-1
o	-1	-1	-1	-1
E	$\underline{2}$	0	2	0

$$\bar{w} := (0,0,0) + \frac{1}{4} \cdot 2 \cdot (1,1,1) = (0.5, 0.5, 0.5)$$

Second iteration:

Cases:	$(1,1,1)$	$(1,-1,1)$	$(1,1,-1)$	$(1,-1,-1)$
t	1	-1	1	-1
o	1	1	1	-1
E	0	$\underline{-2}$	2	0

$$\bar{w} := (0.5, 0.5, 0.5) - \frac{1}{4} \cdot 2 \cdot (1,-1,1) = (0,1,0)$$

Exercise 4 Load the Iris dataset in WEKA (link provided in the exercise description for the previous lecture) and choose the MultilayerPerceptron classifier model. Use Test options: "Use training set". Observe how the performance of the learned model (and the time needed for learning) changes when you modify the following parameters of the learning procedure:

- hidden Layers: this controls the structure of the network (use GUI:true to check).
- trainingTime: controls how many iterations are performed in the weight learning.
- learning rate: controls the stepsize in the gradient descent.

Exercise 5* Complete one more iteration of the back propagation algorithm for the example on slide 08.34.

Exercise 6 Complete the exercises for the last lecture.

Exercises for MI

Exercise sheet 9

Thomas Dyhre Nielsen

When you have finished with the exercises, you should continue with the exam sheet from the previous years, which can be found at the course's home page, and any unfinished earlier exercises.

Exercise 1 Solve Exercise 10.1 in PM.

Solution: The posterior distribution for C given $A = a$ and $B = b$ is

$$P(C \mid a, b) = \frac{P(C, a, b)}{P(C = 0)P(a|C = 0)P(b \mid C = 0) + P(C = 1)P(a|C = 1)P(b \mid C = 1)}.$$

Both terms in the denominator are zero due to $P(A \mid C = 0)$ and $P(B \mid C = 1)$, respectively.

Exercise 2*

Consider a poker game consisting of two rounds, and where each player is initially dealt three cards. During the first round all three cards can be changed (FC), but during the second round at most two cards can be changed (SC). When deciding on whether to call or fold you can taken into account the number of cards changed by your opponent as well as your current hand (MH). After playing 20 games we have the results in Table 1, where BH shows who has the best hand.

- Construct a naive Bayes classifier for the poker domain.
- Use the data cases to learn the parameters in the model; if you feel comfortable with the estimation procedure, you only need to estimate the probabilities required for solving the exercise below.
- What class does your classifier assign to a case with *MH=1a*, *FC=1*, and *SC=1*?

Solution:

See the Hugin network poker_model.net. The probabilities in the model have been calculated using simple frequency counting. For example, for $P(FC =$

Case number:	*BH*	*MH*	*FC*	*SC*
1	op	no	3	1
2	op	1a	2	1
3	draw	2 v	1	1
4	me	2 a	1	1
5	draw	fl	1	1
6	me	st	3	2
7	me	3 v	1	1
8	me	sfl	1	0
9	op	no	0	0
10	op	1 a	3	2
11	draw	2 v	2	1
12	me	2 v	3	2
13	op	2 v	1	1
14	op	2 v	3	0
15	me	2 v	3	2
16	draw	no	3	2
17	draw	2 v	1	1
18	op	fl	1	1
19	op	no	3	2
20	me	1 a	3	2

Table 1: Training data for constructing a poker classifier.

$1|BH = op)$ we get

$$P(FC = 1|BH = op) = \frac{N(FC = 1, BH = op)}{N(BH = op)} = \frac{2}{8} = \frac{1}{4}.$$

By estimating the remaining entries in the conditional probability tables and inserting the evidence $MH = 1a, FC = 1, SC = 1$, we get the posterior probability $P(BH|MH = 1a, FC = 1, SC = 1) = (0.671_{op}, 0.329_{me}, 0_{dr})$.

Observe that if you are *only* interested in calculating the probability $P(BH|MH = 1a, FC = 1, SC = 1)$, then you need not estimate all the probabilities required to get a fully specified naive Bayesian network (this would require estimating $P(BH), P(MH|BH), P(FC|BH), P(SC|BH)$). Instead you only need to estimate probabilities for the configurations that are consistent with the configuration that you are conditioning on. Specifically,

$$\begin{aligned} &P(BH|MH = 1a, FC = 1, SC = 1) \\ &\quad = \frac{P(BH)P(MH = 1a|BH)P(FC = 1|BH)P(SC = 1|BH)}{\sum_{BH} P(BH)P(MH = 1a|BH)P(FC = 1|BH)P(SC = 1|BH)}, \end{aligned}$$

so you need not estimate probabilities that are not used in the calculations above (e.g. $P(MH = 2v|BH)$).

Exercise 3*

You want to predict whether a person will pay back a loan based on the features *Income*, *Houseowner* and *Marital Status* of the person. Domains and distance functions on the domains of these features are defined as follows:

Income	*low*	*medium*	*high*
low	0	1	2
medium	1	0	1
high	2	1	0

Houseowner	*yes*	*no*
yes	0	1
no	1	0

Marital Status	*unmarried*	*married*	*divorced*
unmarried	0	1	1
married	1	0	1
divorced	1	1	0

Define the distance between two examples by the sum of the distances for the three features.

Your training examples are:

	Income	Houseowner	Marital Status	Pay back
1	high	yes	married	yes
2	high	yes	unmarried	yes
3	medium	no	divorced	no
4	low	yes	married	no
5	low	no	unmarried	no

Now you want to predict 'Pay back' for a new case with Income = high, Houseowner = no, Marital Status = divorced.

- What is the prediction obtained by the 1-nearest-neighbor rule?
- What is the prediction obtained by the 3-nearest-neighbor rule?
- What could be a sensible modification of the distance function such that you would get a different result from the 1-nearest-neighbor rule?

Solution:

The distance of the new case to the 5 training examples is:

$$\begin{array}{ll}
1 & 0+1+1=2 \\
2 & 0+1+1=2 \\
3 & 1+0+0=1 \\
4 & 2+1+1=4 \\
5 & 2+0+1=3
\end{array}$$

Thus:

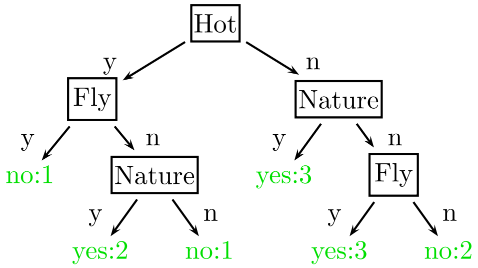
- The prediction from the 1-nearest-neighbor rule is Pay back = no, because training example 3 has smallest distance to the new case.
- The prediction from the 3-nearest-neighbor rule is Pay back = yes, because 2 out of the 3 nearest neighbors (training cases 1,2,3) have Pay back = yes.
- A sensible modification of the distance function could be to give lower weight to the distance contributed by the marital status attribute, as by the income attribute. For example, if we give a weight of 10 to the income attribute, a weight of 5 to the houseowner attribute, and a weight of 1 to the marital status attribute, then the distances are

  $$\begin{array}{lc}
  1 & 10\cdot 0+5\cdot 1+1=6 \\
  2 & 10\cdot 0+5\cdot 1+1=6 \\
  3 & 10\cdot 1+5\cdot 0+0=10 \\
  4 & 10\cdot 2+5\cdot 1+1=26 \\
  5 & 10\cdot 2+5\cdot 0+1=21
  \end{array}$$

  Now there is a tie between training cases 1 and 2 for being the nearest neighbor, but both have Pay back = yes, so now the 1-nearest neighbor rule would predict Pay back = yes.

Exercise 4*

Based on 12 training examples the following decision tree was learned:



Here, for example, the label "yes:2" at the end of the branch Hot=y, Fly=n, Nature=y means that there were two examples in the training set with Hot=y, Fly=n, Nature=y, and both examples had class label Likes=yes. Thus, the decision tree has 100% accuracy on the training data (all leaves are class pure).

Now suppose you have the following 5 examples (not used in the construction of the tree), which you want to use as a validation set:

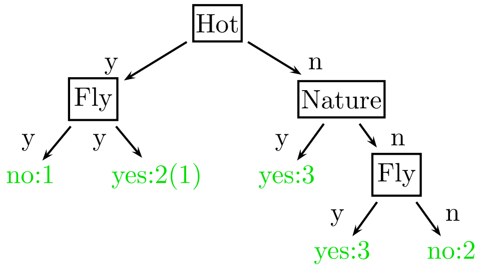
Culture	Fly	Hot	Music	Nature	Likes
no	no	yes	yes	yes	no
no	no	yes	no	no	yes
yes	yes	no	no	no	yes
yes	no	yes	no	yes	yes
no	no	no	no	no	no

Based on these validation examples, we perform post-pruning of the decision tree:

- First check whether the 'Nature' node reached by Hot=y, Fly=n should be pruned (eliminated)
- If yes, what does the new tree look like after pruning?
- Continue the pruning process by checking for other nodes whether they should be pruned (in a bottom-up order; the next candidate for pruning could be the 'Fly' node reached by Hot=n, Nature=n).

Solution:

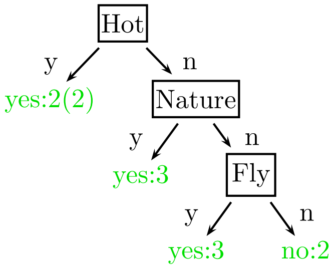
3 of the validation examples (1,2,and 4) go into the Hot=y, Fly=n branch. 2 of these (1,2) will be misclassified based on the 'Nature' feature. If the 'Nature' node is pruned, then it will be replaced by a leaf labeled with 'yes' (because the majority of the *original training examples* are 'yes' cases). With the resulting tree, then only validation example 1 will be misclassified. Thus, the performance on the validation set improves, we would perform the pruning to obtain:



The ' yes:2(1)' label means: cases ending in this leaf are predicted 'yes', and this is correct for 2 training examples, but there is also one 'no' training example.

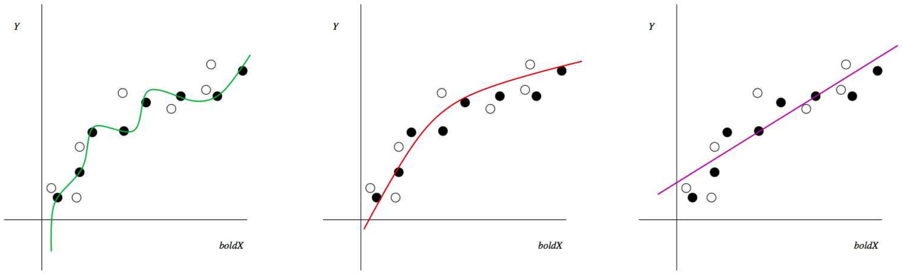
Next we check whether the rightmost 'Fly' node should also be pruned. Two validation examples (3,5) reach this node. Both of them are correctly classified based on the 'Fly' feature. If that node was eliminated, then one of the two would have to be mis-classified. Therefore, the accuracy on the validation set is higher when the 'Fly' node is kept.

Finally, we can check whether pruning the 'Fly' node reached by Hot=y should be pruned. This node is still only reached by the three examples 1,2,4, two out of which are correctly classified. If the node was pruned, then the resulting leaf could be labeled either 'yes' or 'no' (because there are then 2 training examples for each of these labels). Assuming that we would label the resulting leaf 'yes', we would still obtain 2 out of 3 correct classifications for these three validation examples. Thus, by pruning, we obtain a tree with the same accuracy, but one that is a bit smaller than the previous. This is a borderline situation, and depending on how the pruning algorithm exactly resolve this, one might decide not to prune 'Fly', or prune this node also and end up with the tree:



Exercise 5

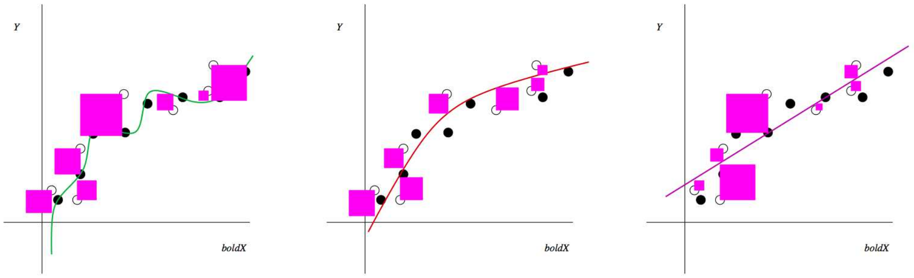
The following graphs show three regression models learned from training examples (filled dots):



The open dots represent a set of future observations (or a validation set). Which of the three models has the smallest SSE on these future observations? No exact computations required – try to read it (approximately) off the graphs!

Solution:

Indicating the squared errors of each validation example by a colored square (cf. slide 09.22) gives:



From this it appears (subject to exact verification by measurement!) that the SSE of the leftmost model is much higher than that of the other two, and that the one of the middle model is a little lower than that of the right (linear) model.

Exercise 6 Complete the exercises from last time.

Exercises for MI

Exercise sheet 10

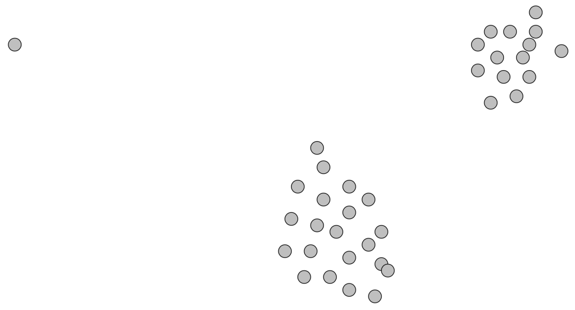
Thomas Dyhre Nielsen

When you have finished with the exercises, you should continue with the exam sheets from the previous years, which can be found at the course's home page.

Exercise 1 Continue with the exercises from last time.

Exercise 2

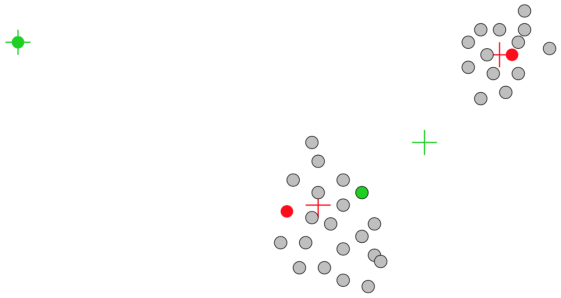
For the following data set:



- identify two initial cluster centers such that 2-means clustering initialized with these cluster centers will terminate with the single outlier forming one cluster, and all other points the second cluster
- identify two initial cluster centers such that 2-means clustering initialized with these cluster centers will terminate with the two big groups of points forming different clusters (and the outlier belonging to one of those)

In both cases indicate the (approximate) position of the final cluster centers.

Solution:



- The green filled instances are possible initial cluster centers; the green crosses are the (approximate) final cluster centers.
- Same with red ...

Exercise 3*

Consider the data points plotted in Figure 1.

- Perform two iterations of the k-means algorithm using
  - the data points $(2, 6)$ and $(3, 5)$ as the initial cluster centers.
  - the Euclidean distance as distance metric
- Calculate the sum of squared errors using the initial cluster centers and the cluster centers that you found above.

Solution:

The result of the first two iterations are illustrated in Figure 2. The cluster centers are located at:

- 1. iteration: $(2.5, 6.5)$ and $(4.4, 5)$.
- 2. iteration: $(2.33, 5.67)$ and $(5, 5.25)$.

The sum of squared errors are 31.0 and 12.91, respectively.

Exercise 4 Use WEKA to perform clustering experiments on the datasets clustering clusters.arff and clustering random.arff.

1. Perform k-means clustering for $k = 1, 2, 3, 4, 5, 6, 7, 8$ on the two data sets. For each clustering, WEKA outputs the "Within cluster sum of squared

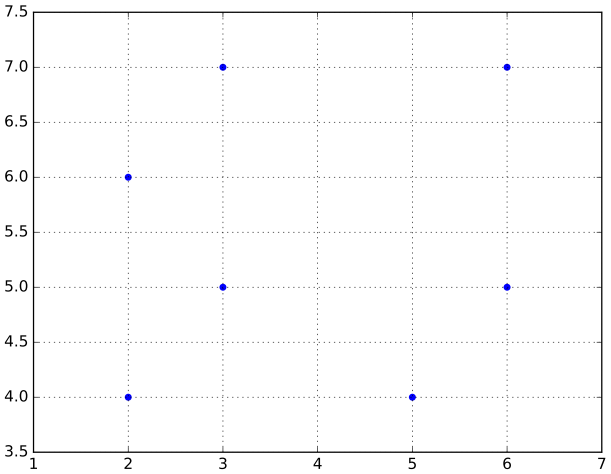


Figure 1: Data to be clustered in Exercise 3.

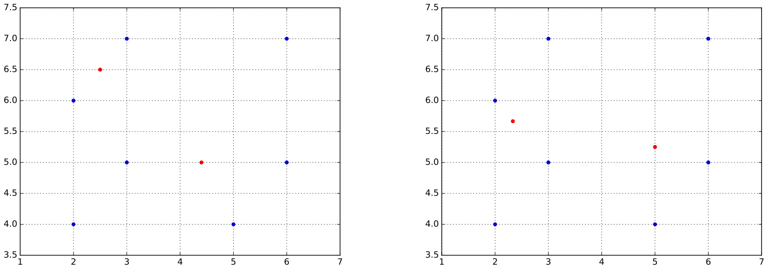


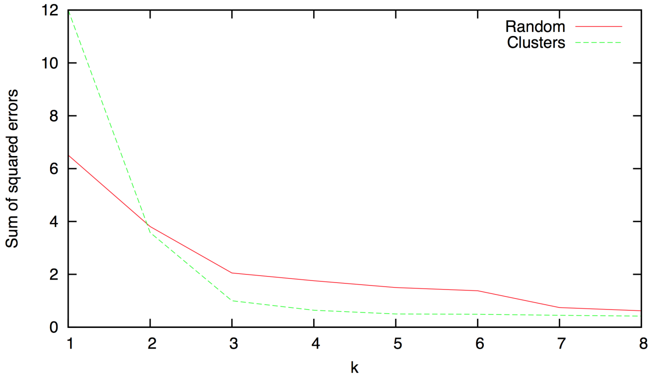
Figure 2: The cluster centers (red dots) after the first and second iterations, respectively.

errors” (which corresponds to the sum of squared errors). Make a plot of this error as a function of k for both datasets. How can plots like these be used to determine the “right” number of clusters?

2. For $k = 3$ perform 4 runs each of k-means clustering using different setting of the random seed (left click in the ’Clusterer’ text field to set the random seed). Compare the results obtained in the different runs using the “Visualize cluster assignment” function (accessible via the Result list panel). How does this help you to decide which of the two datasets has 3 “real” clusters?

Solution:

1. A plot of the within cluster sum of squared errors:



   The plot indicates that for clustering clusters.arff a particularly sharp drop in the error value (compared to the drop for a random dataset) up to $k = 3$, where the curve then levels off. The random data shows a more uniform decrease in error value over the whole k-range. “Knees” in the SSE error function are a (heuristic) indicator for the “right” number of clusters.

2. The comparison shows that for k = 3 the clusters in clustering clusters.arff are stable, i.e. the same clusters are returned independent of the random seed (which determines the random initial cluster centers of the algorithm). For clustering random.arff the final result is different for different settings of the seed, indicating that the computed clusters are not well-defined clusters in the data.

Exercise 5* Perform one more iteration of the EM algorithm for the example on Slide 11.20. Note that you will first need to complete the last two maximization calculations for the 2nd iteration, which is left unfinished on the slides.

Solution:

First we start by completing the second iteration of the EM-algorithm. This consists of updating the distributions for $P_2(F_2|C)$ and $P_2(F_3|C)$; note that the subscript 2 refers to the iteration number.

$$P_2(F_2|C) = \frac{\sum_{F_1,F_3} A(F_1,F_2,F_3,C)}{\sum_{F_1,F_2,F_3} A(F_1,F_2,F_3,C)}$$

$$= \frac{\begin{array}{c|cc} & \text{C=1} & \text{C= 2} \\ \hline F_2 = t & 0.88 + 0 + 0 + 0 & 0.12\text{+}0\text{+}0\text{+}0 \\ F_2 = f & 0\text{+}0.66\text{+}0.48\text{+}0.47 & 0\text{+}0.34\text{+}0.52\text{+}0.53 \end{array}}{(0.88 + 0.66 + 0.48 + 0.47, 0.12 + 0.34 + 0.52 + 0.53)}$$

$$= \frac{\begin{array}{c|cc} & \text{C=1} & \text{C= 2} \\ \hline F_2 = t & 0.88 & 0.12 \\ F_2 = f & 1.61 & 1.39 \end{array}}{(2.49, 1.51)}$$

$$= \begin{array}{c|cc} & \text{C=1} & \text{C= 2} \\ \hline F_2 = t & 0.35 & 0.08 \\ F_2 = f & 0.65 & 0.92 \end{array}$$

The tables above are structured in the same way as on the slides. Thus, the columns correspond to the two states of C and the rows correspond to $F_2 = t$ and $F_2 = f$, respectively. For $P_2(F_3|C)$ we end up with (the intermediate calculations follow the same steps as above):

$$P_2(F_3|C) = \frac{\begin{array}{c|cc} & C = 1 & C = 2 \\ \hline F_3 = t & 2.01 & 0.99 \\ F_3 = f & 0.48 & 0.52 \end{array}}{(2.49, 1.51)}$$

$$= \begin{array}{c|cc} & C = 1 & C = 2 \\ \hline F_3 = t & 0.81 & 0.66 \\ F_3 = f & 0.19 & 0.34 \end{array}$$

Continuing with the third iteration of the EM-algorithm, we start by calculating the expected counts that should go into our count table $A(F_1,F_2,F_3,C)$. This should be done using the conditional probability tables estimated during the last iteration (i.e., $P_2(C)$, $P_2(F_1|C)$, $P_2(F_2|C)$, and $P_2(F_3|C)$) and reduces to calculating $P_2(C|F_1,F_2,F_3)$ for the four different configurations of F_1, F_2, and F_3 observed in the data. In total we end up with:

F_1	F_2	F_3	$P(C\|F_1,F_2,F_3)$
t	t	t	(0.92,0.08)
t	f	t	(0.64,0.36)
t	f	f	(0.44,0.56)
f	f	t	(0.43,0.57)

Based on the updated count table, we recalculate the conditional probabilities of the model, i.e., $P_3(C)$, $P_3(F_1|C)$, $P_3(F_2|C)$, and $P_3(F_3|C)$. The calculation procedures are the same as for the previous two iterations and results in the following tables:

$$P_3(C) = (0.61, 0.39)$$

$P_3(F_1|C) =$

	$C = 1$	$C = 2$
$F_1 = t$	0.82	0.64
$F_1 = f$	0.18	0.36

$P_3(F_2|C) =$

	$C = 1$	$C = 2$
$F_1 = t$	0.39	0.05
$F_1 = f$	0.61	0.95

$P_3(F_3|C) =$

	$C = 1$	$C = 2$
$F_3 = t$	0.82	0.64
$F_3 = f$	0.18	0.36

Exercises for MI

Exercise sheet 11

Thomas Dyhre Nielsen

When you have finished with the exercises, you should continue with the exam sheet from the previous years, which can be found at the course's home page.

Exercise 1*

A contestant on the show "who wants to be a millionaire" might be faced with the following situation: she is asked a question, which she can choose to answer or not to answer. If she chooses not to answer, then the game is over and she wins 10.000. If she answers, and the answer is incorrect, then she wins 5.000. If she answers and the answer is correct she wins 20.000. Assume the contestant is about 60% sure she knows the answer, and decides not to answer.

- Represent the scenario as a choice between two lotteries.
- Given the decision of the contestant, what can you say about the utility of money function for this contestant? (draw a partial graph of this function, indicating the relative position of some points on this graph).
- How would the utility of money function look for a person who decides to try to answer the question even if she had only a 40% confidence in knowing the right answer?

Solution:

We can define the decision problem as a choice between the following two lotteries:

- $A : [1 : \$10000]$ and $B : [0.4 : \$5000, 0.6 : \$20000]$

A partial utility function consistent with the decision $A \succ B$ could look like the one shown in Figure 1:

A possible utility function consistent with $A \prec B$ could have a convex shape, which would promote a risk-seeking behavior.

Exercise 2* In your computer science studies you attend two courses, *Graph Algorithms* and *Machine Intelligence.* In the middle of the term, you realize

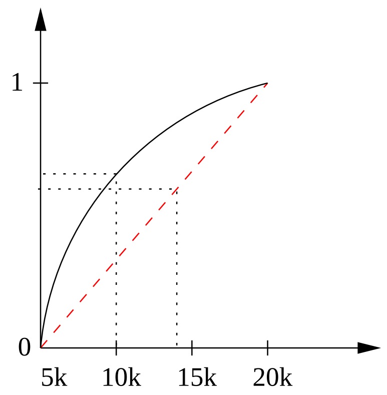


Figure 1: A partial concave utility function for the adverse contestant in Exercise 1.

that you cannot keep pace. You can either reduce your effort in both courses slightly or you can decide to attend one of the courses superficially. What is the best decision?

You have three possible actions:

Gm: Keep pace in Graph Algorithms and follow Machine Intelligence superficially.

SB: Slow down in both courses.

Mg: Keep pace in Machine Intelligence and follow Graph Algorithms superficially.

The results of the actions are your final marks for the courses (excluding -03). You have certain expectations for the marks given your effort in the rest of the term. They are shown in Table 1.

- Assuming that you wish to maximize the sum of the expected marks, what is your best course of action?
- Specify a reasonable utility function for the marks and determine your best course of action according to the utility function.

Solution: For maximizing the sum of the expected marks, the calculations are

$$EM(Gm) = \sum_{m \in GA} P(m \mid kp)m + \sum_{m \in MI} P(m \mid fs)m = 8.4 + 5.3 = 13.7,$$

$$EM(SB) = \sum_{m \in GA} P(m \mid sd)m + \sum_{m \in MI} P(m \mid sd)m = 6.8 + 7.6 = 14.4,$$

	kp	*sd*	*fs*
0	0	0	0.1
2	0.1	0.2	0.1
4	0.1	0.1	0.4
7	0.2	0.4	0.2
10	0.4	0.2	0.2
12	0.2	0.1	0

$P(GA \mid \mathit{effort})$

	kp	*sd*	*fs*
0	0	0	0.1
2	0	0.1	0.2
4	0.1	0.2	0.2
7	0.2	0.2	0.3
10	0.4	0.4	0.2
12	0.3	0.1	0

$P(MI \mid \mathit{effort})$

Table 1: The conditional probabilities of the final marks in Graph Algorithms (GA) and Machine Intelligence (MI) given the efforts *keep pace* (*kp*), *slow down* (*sd*), and *follow superficially* (*fs*).

$$EM(Mg) = \sum_{m \in GA} P(m \mid \mathit{fs})m + \sum_{m \in MI} P(m \mid \mathit{kp})m = 5.2 + 9.4 = 14.6.$$

From this, you would conclude that you should follow Graph Algorithms superficially but keep pace in Machine Intelligence.

For the second part of the exercise, the structure of the calculations is the same as above except that you should use the utility of the individual marks $U(m)$ instead of the actual marks m.

Exercise 3* Solve Exercise 9.6 in PM.

Solution: The table in the exercise represents the result of eliminating all variables succeeding *Run* in the decision network. Since *Run* is a decision variable it should be eliminated by maximization (and not summation as we do when eliminating chance variables). This produces the following table:

Look	*See*	*Value*
true	false	56
true	true	23
false	true	28
false	false	22

The optimal decision function for run is a conditional function that for each configuration of *Look* and *See* specifies the state of *Run* that maximizes the utility as defined by the table in the exercise description:

Look	*See*	*Run*
true	false	no
true	true	yes
false	true	yes
false	false	no

Exercise 4*

(a)

Construct a decision network for the following version of the exam preparation problem: you have to decide whether you *prepare some* or *prepare all* of the questions. At the exam, you get one of 0,7,10 as a grade (to simplify matters, we consider only three grades). You take into consideration whether this is your 1., 2. or 3. attempt at this exam (include in your network a chance node *Attempt* which you observe before you make your decision).

(b) Make a table containing all possible worlds with *Attempt*=1. For each possible world ω in the table compute $P(\omega \mid \mathit{Prepare} = p) \cdot U(\omega)$, where p is the value of *Prepare* in ω. From the table, determine the optimal decision for the case *Attempt*=1.

(c)

Solve the decision network to obtain the optimal decision rule for the *Prepare* decision node (i.e. the optimal decision for all possible states of *Attempt*).

Solution: The network structure is shown in Figure 2

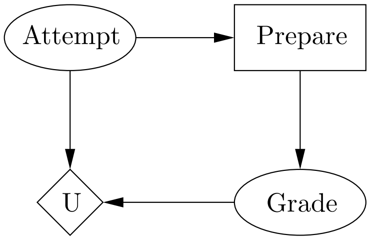


Figure 2: A one-shot decision network for the grade problem in Exercise 4. The arc from Attempt to Prepare indicates that Attempt is observed before we decide on Prepare.

For finding an optimal decision rule for Prepare we calculate

$$\delta(\mathit{Attempt}) = \arg\max_{\mathit{Prepare}} \sum_{\mathit{Grade}} P(\mathit{Grade}|\mathit{Prepare}) U(\mathit{Grade}, \mathit{Attempt}),$$

which prescribes a decision for each state of Attempt.

The actual calculations requested in the exercise will depend on the numbers that you have chosen to quantify the decision network.

Exercise 5

Use Hugin to construct a sequential decision model for the preparation of (at most) two exams: the model should contain the two decision nodes *Prepare_exam*, *Prepare_reexam*, both with the two possible values *prepare some* and *prepare all*. Consider only 0,7, and 10 as possible outcomes of the exam.

Note that there is a somewhat subtle peculiarity with the problem: if you pass the course the first time, then you are *not* allowed to take the exam again.

Model the problem as a decision network and solve it using Hugin. What are the optimal decision functions?

Solution: See the Hugin model here. Note that in this model we use a utility function U_3 (containing large negative values) to ensure that non-admissible decision options will never be part of an optimal strategy. This also includes extending the state space of *Reexam* with an artificial state labeled *NA*, which the utility function ensures will be optimal if you have already passed the regular exam.

Exercise 6 Consider again the Monty hall problem from one of the previous exercise sessions: You are confronted with three doors, A, B, and C. Behind exactly one of the doors there is $10,000. When you have pointed at a door, an official will open another door with nothing behind it. After he has done so, you are allowed to alter your choice. Should you do that?

Model the problem as a decision network and solve it using Hugin.

Solution: See the Hugin model here. Yes, you should change door. We can see this by simulating the different evidence scenarios using the Hugin model.

Exercise 7 An oil wildcatter must decide whether to drill or not to drill. The cost of drilling is $70,000. If he decides to drill, the hole may be soaking (with a return of $270,000), wet (with a return of $120,000), or dry (with a return of $0). The prior probabilities for soaking, wet, and dry are (0.2, 0.3, 0.5). At the cost of $10,000, the oil wildcatter could decide to take seismic soundings of the geological structure at the site. The specifics of the test are given in Table 2.

$T \setminus S$	dr	wt	so
n	0.6	0.3	0.1
o	0.3	0.4	0.4
c	0.1	0.3	0.5

$P(Test \mid Structure)$

Table 2: Table for Exercise 7. The states n, o, and c are the outcomes of the test.

- Construct a decision network for the problem above.*
- Solve the problem using Hugin.

Solution: See the Hugin model here.

A peculiarity with this decision problem is that a test outcome will be observed only if we initially decide to perform a test. Otherwise it will be unobserved.

From this perspective the problem can be considered *asymmetric*, since which observations will become available depends on previous decisions. In the decision network above, we handle this issue by introducing an artificial chance variable T', which is in the same state as the actual test result (T) if we decide to perform the test; otherwise it is in the artificial state n with probability 1 no matter the state of T. The dependence between T and T' is defined in the conditional probability table associated with T', which is also the variable we observe before deciding on *Drill*.

Exercise 8 One morning, a farmer goes out into his field to inspect the quality of his crops. At this time of the year, the farmer estimates that the prior probability for the crops being in a good condition is 0.9. To get more information the farmer takes a sample of his crops to determine its quality (which can either be good or bad). The farmer expects that if the general quality of the crops is good, then the sample will also have a good quality with probability 0.95 and if the quality of the crops is bad, then the sample will also be bad with probability 0.90. Based on the quality of the sample, the farmer then decides whether to apply fertilization. He knows that if the quality of the crops is bad, then a week after applying fertilization the quality of the crops will become good (with probability 0.8). If the quality is already good, then (with probability 1) the quality will still be good no matter whether fertilization is applied. On the other hand, if the quality is bad, then (without applying fertilization) the quality of the crops may improve to good with probability 0.3. The cost of applying fertilization is $20,000$ Dkr.

A week after having applied the fertilization, the farmer should then decide whether to harvest now or wait two more weeks. The value of this decision is determined by the quality of the crops: if the quality is good, then the farmer expects to earn $100,000$ Dkr if he harvests now, but only $80,000$ if he waits. On the other hand, if the quality is bad, then harvesting now will only give him $20,000$ Dkr, whereas he expects that by waiting the quality of the crops will improve so much that he can get $40,000$ Dkr from harvesting.

- Construct an influence diagram for the farmer from the description above.*
- Solve the influence diagram using Hugin.

Exercise 9 Solve Exercise 9.7 in PM.

Solution: Conditional on *Positive test*, the optimal decision option for *Discard sample* does not depend of the state of *Contaminated specimen.* Thus, the value of information for *Contaminated specimen* is zero.

On the other hand, the value of *Positive test* for the decision *Discard sample* is positive (greater than zero).

Exercise 10* A farmer inseminated a cow five weeks ago, and he should now decide whether to repeat the insemination or wait an additional five weeks before doing the insemination. The probability that the cow is pregnant is 0.8, and the cost of repeating the insemination is 1000 Dkr regardless of whether the cow is pregnant. On the other hand, waiting with the insemination will incur an additional loss of 500 Dkr if the cow is not pregnant (giving a total of 1500 Dkr).

Before making this decision the farmer can decide to perform a scanning test of the cow at the cost of 100 Dkr. The scanning test's frequency of false positives and false negatives is 0.3 and 0.05, respectively.

- Perform a value of information analysis of the decision problem above.

Solution:

An ID representation of the decision problem can be found here.

In order to determine whether the farmer should perform the scanning test, we need to compare the expected value of performing the test with the expected value of not performing the test.

For the first scenario, where a test is not performed, the expected utility is:

$$\begin{aligned} EU_1 &= \max_I \sum_{Pr} P(Pr)U(Pr, I) \\ &= \max_I(-1000 \cdot 0.8 + (-1000) \cdot 0.2, 0.8 \cdot 0 + (-1500) \cdot 0.2 \\ &= -300 \end{aligned}$$

For the second scenario, we calculate the expected utility

$$EU_2 = \sum_S P(S) \max_I \sum_{Pr} P(Pr \mid S)U(Pr, I) \qquad (1)$$

Thus, we first need to find $P(S)$ and $P(Pr \mid S)$. The latter can be found using Bayes rule, which will also provide us with $P(S)$ as an intermediate result:

$$P(Pr \mid S) = \frac{P(Pr)P(S \mid Pr)}{\sum_{Pr} P(Pr)P(S \mid Pr)} = \frac{P(Pr)P(S \mid Pr)}{P(S)}. \qquad (2)$$

For the numerator $P(Pr)P(S \mid Pr)$ we get

$$P(Pr)P(S \mid Pr) = P(S, Pr) =$$

		Pr	
		y	n
S	p	0.76	0.06
	n	0.04	0.14

and from that we have $P(S) = \sum_{Pr} P(Pr, S) = (0.82, 0.18)$. Inserting the results back into Equation 2 we obtain

$$P(Pr\,|\,S) = \begin{array}{cc|cc} & & \text{Pr} & \\ & & \text{y} & \text{n} \\ \hline \text{S} & \text{p} & 0.93 & 0.07 \\ & \text{n} & 0.22 & 0.78 \end{array}.$$

We now have the ingredients for finding the expected utility EU_2. As a first step, if we assume that a scanning test is being performed, an optimal policy for I is

$$\begin{aligned} \delta_I(S) &= \arg\max_I \sum_{Pr} P(Pr\,|\,I)U(Pr, I) \\ &= \arg\max_I \sum_{Pr} P(Pr\,|\,I)U(Pr, I) \\ &= \arg\max_I \begin{array}{cc|cc} & & \text{I} & \\ & & \text{y} & \text{n} \\ \hline \text{S} & \text{p} & \text{-1000} & \text{-105} \\ & \text{n} & \text{-1000} & \text{-1170} \end{array} \\ &= \begin{array}{cc|c} \hline \text{S} & \text{p} & \text{n(-105)} \\ & \text{n} & \text{y(-1000)} \end{array} \end{aligned}$$

From the policy we see that if the test is negative, an insemination should be performed; otherwise not.

In order to find the expected utility of performing the test (at the time of deciding whether to perform the test we do not now the result of the test), we need to weigh the expected utilities of the two decision options with the probabilities of seeing the different results from the scanning test (see Equation 1):

$$EU_2 = 0.82 \cdot (-105) + 0.18 \cdot (-1000) = -266.1.$$

The scanning test cost 100 Dkr, however, so the total expected utility of performing the test is -366.1. This is less than the expected utility of not performing the test, meaning that the farmer should *not* perform the test.

Exercise 11 Using Hugin solve the decision problem in Exercise 7 as a value of information problem.

Solution: See the Hugin model here

Exercise 12 Complete the exercises from last time.

Exercises for MI

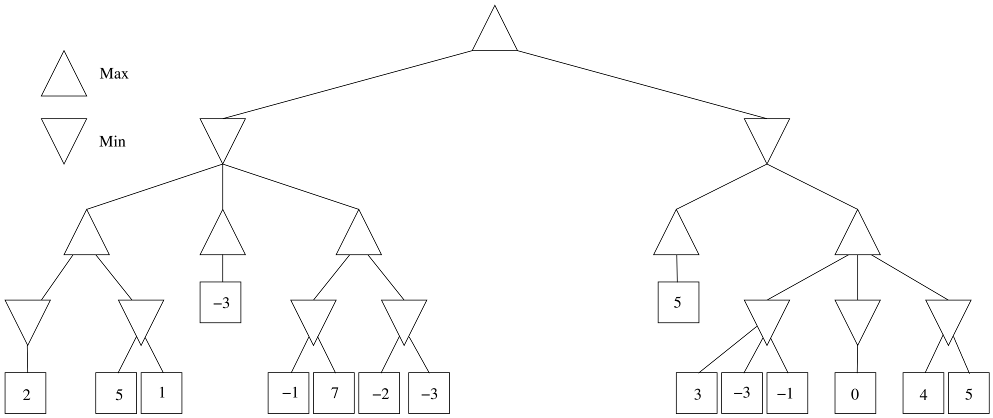
Exercise sheet 12

Thomas Dyhre Nielsen

Exercise 1 Continue with the exercises from last time.

Exercise 2*

Consider the following game tree:



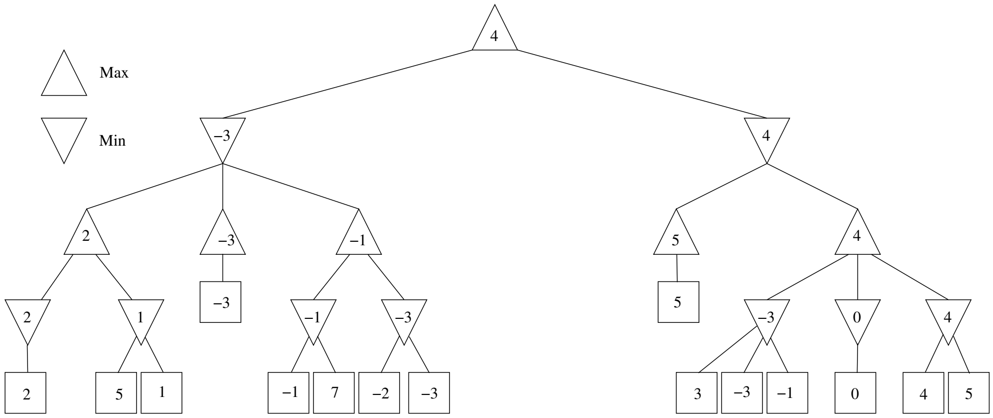
a. Compute the utility values for all nodes.

b. If the utility values are computed in a depth-first order that always considers branches in left-to-right order, which nodes can be pruned, i.e. for which nodes is it not required to compute the utility value in order to determine the optimal strategy for both players?

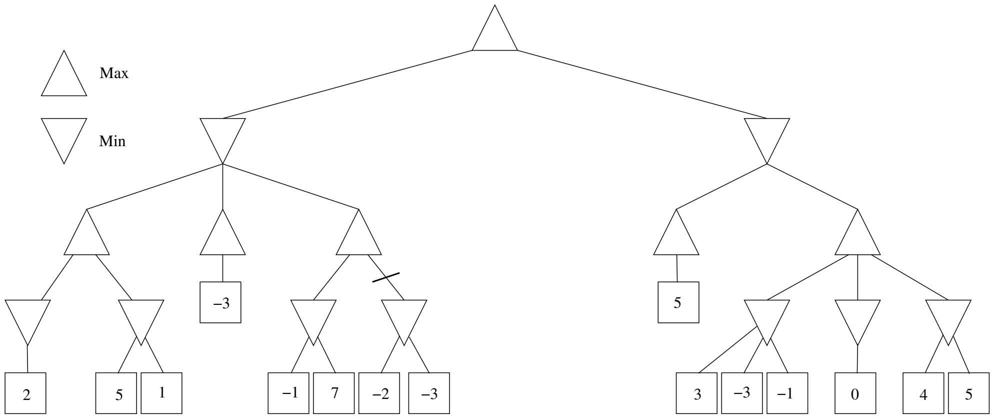
c. For each node in the game tree, determine the ordering of the outgoing branches that is optimal in the following sense: if utility values are computed for nodes in that order, then a maximal number of nodes can be pruned in the utility computation.

Solution:

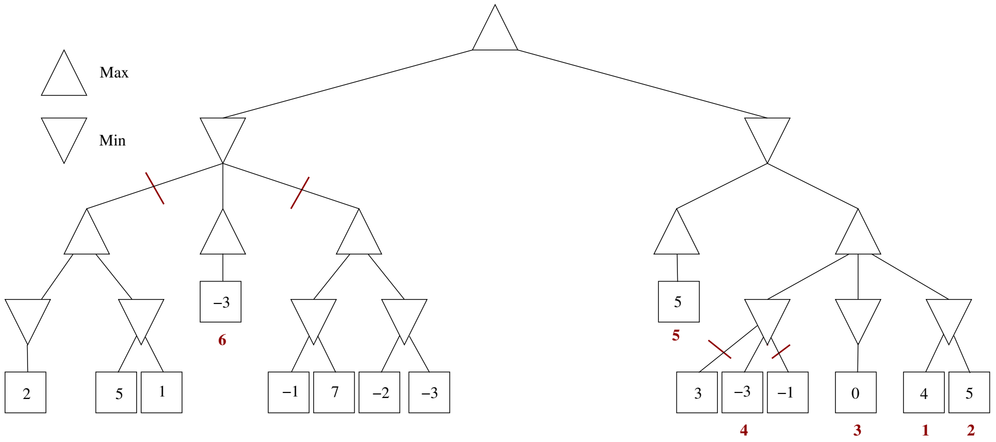
Part a


Max
Min
4
−3
4
2
−3
−1
5
4
2
1
−3
−1
−3
5
−3
0
4
2
5
1
−1
7
−2
−3
3
−3
−1
0
4
5

Part b


Max
Min
−3
5
2
5
1
−1
7
−2
−3
3
−3
−1
0
4
5

Part c



Exercise 3*

Consider the following game representation in normal form:

	Andy		
Barb	a_1	a_2	a_3
b_1	2 0	1 0	2 2
b_2	2 0	1 1	0 0
b_3	2 1	0 0	0 2
b_4	2 0	0 0	0 2
b_5	0 0	1 1	0 2
b_6	0 0	1 1	0 0

The matrix shows the utilities for Andy (red numbers) and Barb (green numbers) for each combinations of strategies they can choose (Andy has 3 strategies to choose from, Barb has 6).

- Determine at least two Nash equilibria consisting of pure strategies for this game.
- Show that there is no Nash equilibrium where Barb plays b_4, and Andy plays any (possibly mixed) strategy.

Solution:

Part a

(a_1, b_3) and (a_2, b_5)

Part b

If Barb plays b_4 and Andy plays (p_1, p_2, p_3), the expected utility for Barb is

$$EU(Barb) = p_1 \cdot 0 + p_2 \cdot 0 + p_3 \cdot 2 = 2 \cdot p_3$$

However, if Barb switches to b_3 her expected utility becomes

$$EU(Barb) = p_1 \cdot 1 + p_2 \cdot 0 + p_3 \cdot 2 = 1 \cdot p_1 + 2 \cdot p_3.$$