S228/406

DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET, DUBLIN 8

BSc (Hons) in Computer Science

Stage 4

SEMESTER 2 EXAMINATIONS 2009

*** *SOLUTIONS* ***

ARTIFICIAL INTELLIGENCE 2

Dr. John Kelleher
Dr. D. Lillis
Dr. I. Arena

Duration: 2 Hours

Answer Question 1 (40 marks) **and**

any 2 Other Questions (30 marks each).

***** *SOLUTIONS* *****

***** *SOLUTIONS* *****

1. (a) Given the joint distribution for X and Y listed in Table 1 calculate $P(Y = y_2)$

(5 marks)

0.14 + 0.32 = 0.46

(b) Given the joint distribution for X and Y listed in Table 1 calculate $P(Y = y_2|X = x_1)$

(5 marks)

From the product rule: $P(a|b) = \frac{P(a \wedge b)}{P(b)} \rightarrow$
$P(Y = y_2|X = x_1) = \frac{P(Y=y_2 \wedge X=x_1)}{P(X=x_1)} \rightarrow$
$P(Y = y_2|X = x_1) = \frac{0.14}{0.26}$

(c) In the context of machine learning, explain what is meant by **overfitting** the training data.

(5 marks)

Overfitting occurs when classifiers make decisions based on accidental properties of the training set that will lead to errors on the test set (or new data). As a result, whenever there is a large set of possible hypotheses, one has to be careful not to use the resulting freedom to find meaningless ”regularity” in the data.

(d) In the context of inductive learning explain what is meant by a **consistent hypothesis**.

(5 marks)

A hypothesis is consistent if it agrees with the true function on all examples that we have.

(e) What is the aim of **inductive logic learning**?

(5 marks)

The aim of inductive logic learning is to find an equivalent logical expression for the goal predicate that we can use to classify examples correctly.

(f) In the context of inductive logic learning, what is meant by the **extension** of a hypothesis?

(5 marks)

Each hypothesis predicts that a certain set of examples, namely those that satisfy the hypotheses definition, are examples that satisfy the goal predicate. This set of examples is called the extension of the hypothesis. For example, assuming a standard interpretation, the extension of the predicate $digit(X)$ is $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$

Table 1: Joint Distribution for X and Y

	$X = x_1$	$X = x_2$
$Y = y_1$	0.02	0.30
$Y = y_2$	0.14	0.32
$Y = y_3$	0.10	0.12

(g) In the context of machine learning distinguish between **false negatives** and **false positives**.

(5 marks)

False negative an example can be a false negative for the hypothesis, if the hypothesis says it should be negative but in fact it is positive.

False positive an example can be a false positive for the hypothesis, if the hypothesis says it should be positive but in fact it is negative.

(h) In the context of machine learning, what does it mean if two classes C_1 and C_2 are described as **linearly separable**?

(5 marks)

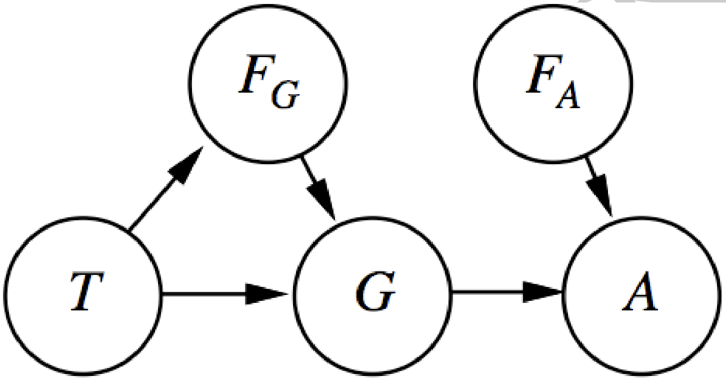
This means that for each class C_i there exists a hyperplane H_i such that on its positive side lie all $x \in C_i$ and on its negative side lie all $x \in C_j, j \neq i$

2. (a) In your local power station, there is an alarm that senses when a temperature gauge exceeds a given threshold. The gauge measures the temperature of the core. Consider the Boolean variables A (alarm sounds), F_A (alarm is faulty), and F_G (gauge is faulty); and multivalued nodes G (gauge reading) and T (actual core temperature).

(i) Draw a Bayesian network for this domain, given that the gauge is more likely to fail when the core temperature gets too high.

(5 marks)

The key aspects are: the failure nodes are parents of the sensor nodes, and the temperature node is a par ent of both the gauge and the gauge failure node. It is exactly this kind of correlation that makes it difficult for humans to understand what is happening in complex systems with unreliable sensors.



(ii) Suppose there are just two possible actual and measured temperatures: normal and high. The probability that the gauge gives the correct temperature is x when it is working, but y when it is faulty. Give the conditional probability table associated with node G.

(5 marks)

Note the semantics of F_G , which is true when the gauge is faulty, i.e., not working.

	$T = Normal$		$T = High$	
	F_G	$\neg F_G$	F_G	$\neg F_G$
$G = Normal$	y	x	$1 - y$	$1 - x$
$G = High$	$1 - y$	$1 - x$	y	x

(b) Suppose you are a security guard at some secret underground installation. You want to know whether it's raining today, but your only access to the outside world occurs each morning when you see the director coming in with, or without, an umbrella. For each day t, the set $\mathbf{E}_t$ contains a single evidence variables U_t (whether the umbrella appears), and the set $\mathbf{X}_t$ contains a single state variable

R_t (whether it is raining). Figure 1 provides the Bayesian network structure and conditional distributions that describe this scenario.

(i) Assuming that you have a prior belief about whether it rained on day 0, just before the observation sequence begins of: $\mathbf{P}(R_0) =< 0.5, 0.5 >$ and that the umbrella appears on day 1, so $U_1 = true$, **compute the probability that it rained on day 1**, (i.e., compute $\mathbf{P}(R_1|u_1)$).

(10 marks)

We want to compute $\mathbf{P}(R_1|u_1)$. This can be computed by rewriting the general filtering model:

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = \alpha \underbrace{\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})}_{Sensor\ Model.} \sum_{x_t} \underbrace{\mathbf{P}(\mathbf{X}_{t+1}|x_t)}_{Transition\ Model.} \underbrace{P(x_t,|\mathbf{e}_{1:t})}_{Current\ State\ Distribution.}$$

in terms of the variables in our scenario (i.e., the hidden boolean variables $R_0, \ldots, R_t$ each denoting whether or not it is raining for a particular timeslice $t = 0, \ldots, t$ and the evidence boolean variables $U_t, \ldots U_t$ each denoting whether or not the umbrella appeared at a particular timeslice $t = 0, \ldots, t$). Rewriting in the filtering model in these terms give us:

$$\mathbf{P}(R_1|u_1) = \alpha \quad \underbrace{\mathbf{P}(u_1|R_1)}_{Sensor\ Model.} \quad \sum_{r_0} \quad \underbrace{\mathbf{P}(R_1|r_0)}_{Transition\ Model.} \quad \underbrace{P(r_0)}_{Current\ State\ Distribution.}$$

Plugging in the appropriate values from the conditional probability tables give us:

$$\mathbf{P}(R_1|u_1) = \alpha \quad \underbrace{\langle 0.9, 0.2 \rangle}_{Sensor\ Model.} \quad \sum_{r_0} \underbrace{\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}}_{Transition\ Model.} \quad \underbrace{\langle 0.5, 0.5 \rangle}_{Current\ State\ Distribution.}$$

$$\begin{aligned}
&= \alpha \quad \underbrace{\langle 0.9, 0.2 \rangle}_{Sensor\ Model.} \quad \underbrace{\langle 0.7, 0.3 \rangle \times 0.5 + \langle 0.3, 0.7 \rangle \times 0.5}_{Prediction\ From\ t=0\ to\ t=1.} \\
&= \alpha \quad \underbrace{\langle 0.9, 0.2 \rangle}_{Sensor\ Model.} \quad \underbrace{\langle 0.5, 0.5 \rangle}_{Prediction\ From\ t=0\ to\ t=1.} \\
&= \alpha \langle 0.45, 0.1 \rangle \approx \langle 0.818, 0.182 \rangle
\end{aligned}$$

(ii) Using the value for $\mathbf{P}(R_1|u_1)$ that you computed in part (i) of this question, and assuming that the umbrella appears on day 2 (i.e. $U_2 = true$), **compute the probability that it rained on day 2** (i.e., compute $\mathbf{P}(R_2|u_1, u_2)$).

(10 marks)

We want to compute $\mathbf{P}(R_2|u_1, u_2)$. This can be computed by rewriting the general filtering model:

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = \alpha \underbrace{\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})}_{Sensor\ Model.} \sum\nolimits_{x_t} \underbrace{\mathbf{P}(\mathbf{X}_{t+1}|x_t)}_{Transition\ Model.} \underbrace{P(x_t, |\mathbf{e}_{1:t})}_{Current\ State\ Distribution.}$$

in terms of the variables in our scenario (i.e., the hidden boolean variables $R_0, \ldots, R_t$ each denoting whether or not it is raining for a particular timeslice $t = 0, \ldots, t$ and the evidence boolean variables $U_t, \ldots U_t$ each denoting whether or not the umbrella appeared at a particular timeslice $t = 0, \ldots, t$). Rewriting in the filtering model in these terms give us:

$$\mathbf{P}(R_2|u_1, u_2) = \alpha \underbrace{\mathbf{P}(u_2|R_2)}_{Sensor\ Model.} \sum\nolimits_{r_1} \underbrace{\mathbf{P}(R_2|r_1)}_{Transition\ Model.} \underbrace{P(r_1|u_1)}_{Current\ State\ Distribution.}$$

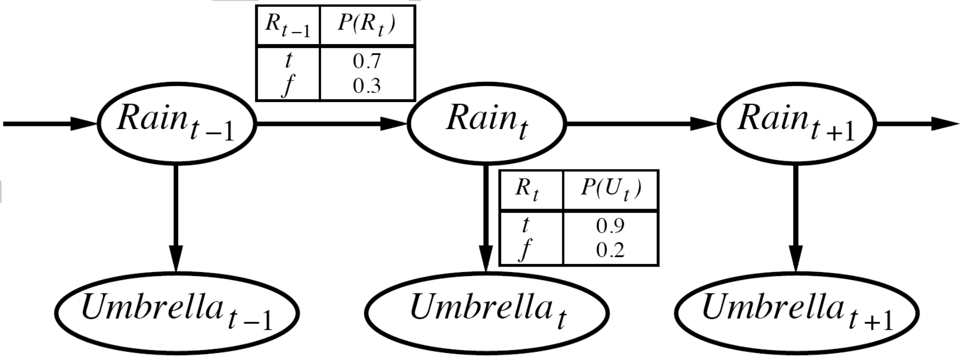
Plugging in the appropriate values from the conditional probability tables give us:

$$\mathbf{P}(R_2|u_1, u_2) = \alpha \underbrace{\langle 0.9, 0.2 \rangle}_{Sensor\ Model.} \sum\nolimits_{r_1} \underbrace{\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}}_{Transition\ Model.} \underbrace{\langle 0.818, 0.182 \rangle}_{Current\ State\ Distribution.}$$

$$= \alpha \underbrace{\langle 0.9, 0.2 \rangle}_{Sensor\ Model.} \underbrace{\langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182}_{Prediction\ From\ t=1\ to\ t=2.}$$

$$= \alpha \underbrace{\langle 0.9, 0.2 \rangle}_{Sensor\ Model.} \underbrace{\langle 0.627, 0.373 \rangle}_{Prediction\ From\ t=1\ to\ t=2.}$$

$$= \alpha \langle 0.565, 0.075 \rangle \approx \langle 0.883, 0.117 \rangle$$



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

R_t	$P(U_t)$
t	0.9
f	0.2

Figure 1: Bayesian network structure and conditional distribution describing the umbrella world. The transition model is $P(Rain_t|Rain_{t-1})$ and the sensor model is $P(Umbrella_t|Rain_t)$

X	Y	Class
T	T	+
T	F	-
T	F	+
T	T	+
F	T	-

Table 2: X and Y Classification Data

3. (a) In the context of machine learning, distinguish between **supervised** and **unsupervised** learning.

(5 marks)

The distinction is that with **supervised learning** we know the actual label or category for each piece of data on which we train, whereas with **unsupervised learning** we do not know the classification of the data in the training sample. Unsupervised learning can thus often be viewed as a **clustering** task, while supervised learning can usually be seen as a **classification** task, or equivalently as a function-fitting task where one extrapolates the shape of a function based on some data points.

(b) Discuss the advantages and disadvantages of k-**Nearest Neighbour** classification.

(10 marks)

Strengths

(i) No training involved lazy learning

(ii) New data can be added on the fly

(iii) Some explanation capabilities

(iv) Robust to noisy data by averaging k-nearest neighbors

Weaknesses

(i) Not the most powerful classification (generally its accuracy will be lower than an ANN or SVM model)

(ii) Slow classification

(iii) Curse of dimensionality (as you increase the number of features you need more and more examples to cover the problem space - kNN are particularly susceptible to this issue as they do not do any feature selection).

(c) Table 2 provides a classification for a data set of X Y pairs.

(i) Calculate the **entropy** for this classification.

(5 marks)

Entropy is $-\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.971$

(ii) Calculate the **information gain** for X and Y.

(10 marks)

Entropy for X = T $-\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4} = 0.811$
Entropy for X = F $0 - \frac{1}{1}log_2\frac{1}{1} = 0$
Gain for X $0.971 - (\frac{4}{5} \times 0.811 + \frac{1}{5} \times 0) = 0.322$
Entropy for Y = T $-\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.918$
Entropy for Y = F $-\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1.0$
Gain for Y $0.971 - (\frac{3}{5} \times 0.918 + \frac{2}{5} \times 1) = 0.02$

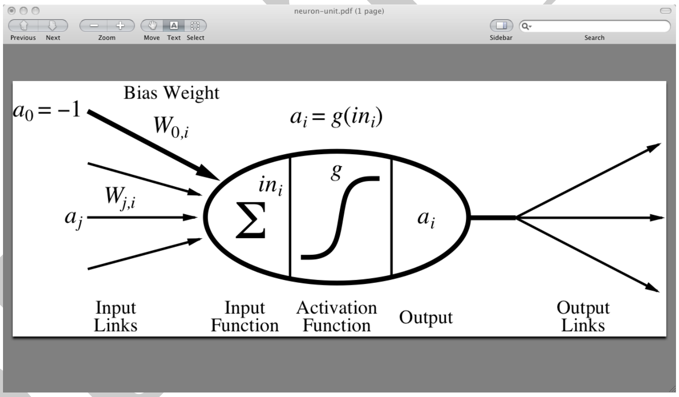
4. (a) Describe the processing stages of a McCulloch-Pits ”unit”.

(10 marks)

The processing stages of a unit are:

(i) Each unit i first compute a weighted sum of its inputs: $in_i \leftarrow \sum_j W_{j,i}a_j$

(ii) Then it applies an **activation function** g to this sum to derive the output (activation) a_i: $a_i \leftarrow g(in_i) = g\left(\sum_j W_{j,i}a_j\right)$



(b) Figure 2 shows a backprogation network, with weights as shown and all biases set to 0, that is currently processing the training vector $[1.0, 0.9, 0.9]$ which has an associated target vector $[0.1, 0.9, 0.1]$. Given that the output from unit B is 0.6 and from C is 0.8, and assuming that the activation function used at all nodes in the network is the logistic function (i.e., $f(x) = \frac{1}{1+\exp^{-x}}$):

(i) Calculate the actual output vector (to 3 decimal places).

(5 marks)

Output of unit $i = f(\sum_{j=1}^{n} W_{j,i} \times activation_j)$
First output unit input = -0.3 x 0.6 + 0.9 x 0.8 = 0.54 → f(0.54) = 0.632
Second output unit input = -0.6 x 0.6 + -0.1 x 0.8 = -0.44 → f(-0.44) = 0.392
Third output unit input = 0.4 x 0.6 + 1.2 x 0.8 = 1.2 → f(1.2)= 0.769

(ii) Calculate the error for each output unit.

(5 marks)

Error = target - output
First output unit = (0.1 - 0.632) = - 0.532
Second output unit = (0.9 - 0.392) = 0.508
Third output unit = (0.1 - 0.769) = - 0.669

(iii) Calculate the error for each hidden unit B and C.

(10 marks)

Each hidden node j is responsible for some fraction of the error Err_i of each of the output units i to which it connects. Thus the Err_i values are divided according to the strengths of the connection between the hidden node and the output nodes and are propagated back to the hidden nodes. Where a hidden node feeds-forward into more than 1 output node the errors propagated back to it are summed: $Err_j = \sum_{i=1}^{n} W_{ji} \times Err_i$:
$Err_B = (-0.3 \times -0.532) + (-0.6 \times 0.508) + (0.4 \times -0.669) = 0.1596 + -0.3048 + -0.2676 = -0.4128$
$Err_C = (0.9 \times -0.532) + (-0.1 \times 0.508) + (1.2 \times -0.669) = -0.4788 + -0.0508 + -0.8028 = -1.3324$

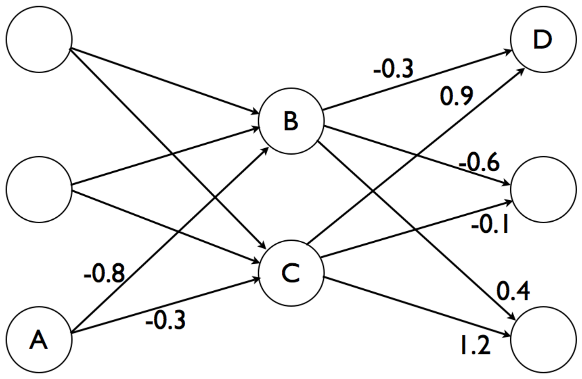


Figure 2: Example Neural Net