

S228/419C

DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET, DUBLIN 8

BSc. (Hons) in Computer Science

Stage 4

SEMESTER 2 EXAMINATIONS 2012/2013

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

ARTIFICIAL INTELLIGENCE II

Dr. John Kelleher
Dr. Deirdre Lillis
Mr. D. Tracey

Monday
13th May 2013
4:00 p.m to 6:00 p.m

Question 1 is **compulsory**

Answer Question 1 (40 marks) **and**
any 2 Other Questions (30 marks each).

***** SOLUTIONS *****

***** SOLUTIONS *****

SOLUTIONS

1. (a) Explain what is meant by **inductive learning**.

(5 marks)

Inductive Learning involves the process of learning by example where a system tries to induce a general rule from a set of observed instances.

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

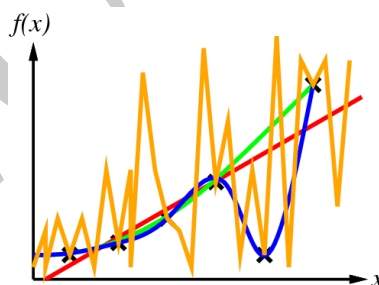
(5 marks)

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.

- (c) Inductive machine learning is often referred to as an **ill-posed problem**. What is meant by this description?

(15 marks)

Inductive machine learning algorithms essentially search through a hypothesis space to find a the best hypothesis that is consistent with the training data used. It is possible to find multiple hypotheses that are consistent with a given training set (i.e. agrees with all training examples). It is for this reason that inductive machine learning is referred to as an ill-posed problem as there is typically not enough information in the training data used to build a model to choose a single best hypothesis. Inductive machine learning algorithms must somehow choose one of the available hypotheses as the *best*. An example like that shown in the figure below would be useful at this point



- (d) In the context of machine learning, explain what is meant by the term **inductive bias** and illustrate your explanation using examples of inductive biases used by machine learning algorithms.

(15 marks)

- The inductive bias of a learning algorithm:
 - (i) is a set of assumption about what the true function we are trying to model looks like.
 - (ii) defines the set of hypotheses that a learning algorithm considers when it is learning.
 - (iii) guides the learning algorithm to prefer one hypothesis (i.e. the hypothesis that best fits with the assumptions) over the others.
 - (iv) is a necessary prerequisite for learning to happen because inductive learning is an ill posed problem.
- An example of the specific inductive bias introduced by particular machine learning algorithms would be good here. E.g.:
 - Maximum margin: when drawing a boundary between two classes, attempt to maximize the width of the boundary. This is the bias used in Support Vector Machines. The assumption is that distinct classes tend to be separated by wide boundaries.
 - Minimum cross-validation error: when trying to choose among hypotheses, select the hypothesis with the lowest cross-validation error.

Table 1: Dataset for the 3-Nearest Neighbor question

ID	Feature1	Feature2	Target
101	4	180000	C1
102	3	120000	C2
103	7	360000	C2
104	5	420000	C1
105	8	480000	C2

Table 2: Query instance for the 3-Nearest Neighbor question.

ID	Feature1	Feature2	Target
250	4	240000	?

2. (a) Table 1 lists a dataset containing examples described by two descriptive features, **Feature1** and **Feature2**, and labelled with a target class **Target**. Table 2 lists the details of a query that we want to predict a the target label for. We have decided to use a **3-Nearest Neighbor** model for this prediction and we will use Euclidean distance as our distance metric:

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^n ((x_1.f_i - x_2.f_i)^2)}$$

- (i) What target class (**C1** or **C2**) will our **3-Nearest Neighbor** model label the query with? Provide an explanation for you answer.

(8 marks)

The first stage is to calculate the Euclidean distance between each of the examples and the query:

ID	Euclidean Distance
101	60000
102	120000
103	120000
104	180000
105	240000

From this table we can see that the three closest examples to the query are examples 101, 102, and 103. Example 101 has a target label of C1 and both 102 and 103 have target labels C2. Consequently C2 is the majority label in local model constructed by the 3-Nearest Neighbor classifier for this query instance and the query will be labelled with class C2.

- (ii) There is a large variation in range between **Feature1** and **Feature2**. To account for this we decide to normalize the data. Compute the normalized data to four decimal points of precision using range normalization

$$x_i.f' = \frac{x_i.f - \min(f)}{\max(f) - \min(f)}$$

(4 marks)

ID	Feature1	Feature2	Target
101	0.2	0.1667	C1
102	0	0.0000	C2
103	0.8	0.6667	C2
104	0.4	0.8333	C1
105	1	1.0000	C2

- (iii) Assuming we use the normalized dataset as input what target class (**C1** or **C2**) will our **3-Nearest Neighbor** model label the query with? Provide an explanation for your answer and remember to normalize the query to four places of precision.

(8 marks)

The normalized query instance is:

ID	Feature1	Feature2	Target
250	0.2	0.3333	?

The Euclidean distances between the normalized data and normalized query are:

ID	Euclidean Distance
101	0.1667
102	0.3887
103	0.6864
104	0.5385
105	1.0414

From this table we can see that the 3 closest neighbors are: 101, 103 and 104. 101 and 104 are both labelled as class **C1**. So **C1** is the majority class in the neighborhood and the query will be labelled as belonging to it.

- (b) Table 3, on the next page lists a classification dataset. Each instance in the dataset has two descriptive features (attribute 1 and attribute 2) and is classified as either a positive (+) or a negative(-) example. Note that Table 4, also on the next page, lists some equations that you may find useful for this question.

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

(5 marks)

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

- (ii) Calculate the **information gain** for Feature 1 and Feature 2.

(5 marks)

$$\begin{aligned} \text{Entropy for feature 1} &= T - \frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} = 0.811 \\ \text{Entropy for feature 1} &= F - 0 - \frac{1}{1}\log_2\frac{1}{1} = 0 \\ \text{Gain for attribute 1} &= 0.971 - (\frac{4}{5} \times 0.811 + \frac{1}{5} \times 0) = 0.322 \\ \text{Entropy for feature 2} &= T - \frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918 \\ \text{Entropy for feature 2} &= F - \frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1.0 \\ \text{Gain for feature 2} &= 0.971 - (\frac{3}{5} \times 0.918 + \frac{2}{5} \times 1) = 0.02 \end{aligned}$$

Feature 1	Feature 2	Classification
T	T	+
T	F	-
T	F	+
T	T	+
F	T	-

Table 3: Classification dataset for information question.

$$\begin{aligned}
 \text{Entropy}(DS) &= -\sum_{i=1}^k p_i \times \log_2(p_i) \\
 \text{Remainder}(F) &= \sum_{v \in \text{Domain}(F)} \frac{|DS_v|}{|DS|} \text{Entropy}(DS_v) \\
 \text{InformationGain}(F, DS) &= \text{Entropy}(DS) - \text{Remainder}(F)
 \end{aligned}$$

Table 4: Equations from information theory.

Table 5: Movie and Song Title Dataset

Movie Titles	Song Titles
<i>A Perfect World</i>	<i>A Perfect Day</i>
<i>My Perfect Woman</i>	<i>Electric Storm</i>
<i>Pretty World</i>	<i>Another Rainy Day</i>

Table 6: Query Title

Perfect Storm

3. Table 5 lists a dataset of song and movie titles. Table 6 lists the title of a query instance that we would like to classify as being either a movie or a song based on its title.

- (a) Using **Laplacian smoothing**, where

$$p(x = v) = \frac{\text{count}(x = v) + k}{\text{count}(x) + (k \times |\text{Domain}(x)|)}$$

, with **k=1** and a **vocabulary size of 11** calculate the following probabilities:

- (i) $P(\text{Movie}) = ?$

(3 marks)

$$P(\text{Movie}) = \frac{3+1}{6+(1 \times 2)} = \frac{4}{8} = 0.5$$

- (ii) $P(\text{Song}) = ?$

(3 marks)

$$P(\text{Song}) = \frac{3+1}{6+(1 \times 2)} = \frac{4}{8} = 0.5$$

- (iii) $P('Perfect' | \text{Movie}) = ?$

(3 marks)

$$P('Perfect' | \text{Movie}) = \frac{2+1}{8+(1 \times 11)} = \frac{3}{19} = 0.1579$$

- (iv) $P('Perfect' | \text{Song}) = ?$

(3 marks)

$$P('Perfect' | \text{Song}) = \frac{1+1}{8+(1 \times 11)} = \frac{2}{19} = 0.1053$$

- (v) $P('Storm' | \text{Movie}) = ?$

(3 marks)

$$P('Storm' | \text{Movie}) = \frac{0+1}{8+(1 \times 11)} = \frac{1}{19} = 0.0526$$

- (vi) $P('Storm' | \text{Song}) = ?$

(3 marks)

$$P('Storm' | \text{Song}) = \frac{1+1}{8+(1 \times 11)} = \frac{2}{19} = 0.1053$$

- (b) Calculate the probability of the query title in Table 6 belonging to the Movie class under the **Naive Bayes assumption** and using the **smoothed probabilities** you calculated in Part 1:

$$P(\text{Movie} | \text{'Perfect Storm'}) = ?$$

(8 marks)

$$\begin{aligned} & P(\text{Movie} | \text{'Perfect Storm'}) \\ &= \frac{P(\text{'Perfect'} | \text{Movie})P(\text{'Storm'} | \text{Movie})P(\text{Movie})}{(P(\text{'Perfect'} | \text{Movie})P(\text{'Storm'} | \text{Movie})P(\text{Movie})) + (P(\text{'Perfect'} | \text{Song})P(\text{'Storm'} | \text{Song})P(\text{Song}))} \\ &= \frac{0.1579 \times 0.0526 \times 0.5}{(0.1579 \times 0.0526 \times 0.5) + (0.1053 \times 0.1053 \times 0.5)} \\ &= 0.4286 \end{aligned}$$

- (c) Calculate the probability of the query title in Table 6 belonging to the Movie class under the **Naïve Bayes assumption** and using **maximum likelihood** probabilities (i.e. the probabilities we could get if we did not use Laplacian smoothing):

$$P(\text{Movie} | \text{'Perfect Storm'}) = ?$$

(4 marks)

Because the word 'Storm' does not appear in any of the Movie titles the maximum likelihood (i.e., unsmoothed) probability of $P(\text{'Storm'} | \text{Movie}) = 0$. As a result the maximum likelihood probability of $P(\text{Movie} | \text{'Perfect Storm'}) = 0$. Showing the complete calculation:

$$\begin{aligned} & P(\text{Movie} | \text{'Perfect Storm'}) \\ &= \frac{P(\text{'Perfect'} | \text{Movie})P(\text{'Storm'} | \text{Movie})P(\text{Movie})}{(P(\text{'Perfect'} | \text{Movie})P(\text{'Storm'} | \text{Movie})P(\text{Movie})) + (P(\text{'Perfect'} | \text{Song})P(\text{'Storm'} | \text{Song})P(\text{Song}))} \\ &= \frac{\frac{2}{8} \times \frac{0}{8} \times \frac{3}{6}}{(\frac{2}{8} \times \frac{0}{8} \times \frac{3}{6}) + (\frac{1}{8} \times \frac{1}{8} \times \frac{3}{6})} \\ &= 0.0 \end{aligned}$$

x	2	4	6	8
y	2	5	5	8

Table 7: Example Dataset for Linear Regression Question

4. (a) Assuming a domain with one explanatory variable x and one dependent variable y linear regression uses the following formula to model the relationship between the explanatory and dependent variable:

$$f(x) = w_1x + w_0$$

where w_1 and w_0 are computed using the following formulae (where M is number of data points in the dataset):

$$w_1 = \frac{(M \sum_{i=1}^M x_i y_i) - (\sum_{i=1}^M x_i \sum_{i=1}^M y_i)}{(M \sum_{i=1}^M x_i^2) - (\sum_{i=1}^M x_i)^2}$$

$$w_0 = \left(\frac{1}{M} \sum_{i=1}^M y_i\right) - \left(\frac{w_1}{M} \sum_{i=1}^M x_i\right)$$

Using the data in Table 7 compute the values of w_0 and w_1 that provide the best linear fit to the data.

(10 marks)

First we need to compute the values of the equation components:

- $M = 4$
- $\sum_{i=1}^M x_i y_i = 4 + 20 + 30 + 64 = 118$
- $\sum_{i=1}^M x_i = 2 + 4 + 6 + 8 = 20$
- $\sum_{i=1}^M y_i = 2 + 5 + 5 + 8 = 20$
- $\sum_{i=1}^M x_i^2 = 4 + 16 + 36 + 64 = 120$
- $(\sum_{i=1}^M x_i)^2 = 20^2 = 400$

Given these values, w_1 :

$$w_1 = \frac{(4 \cdot 118) - (20 \cdot 20)}{(4 \cdot 120) - 400} = \frac{72}{80} = 0.9$$

And, w_0 :

$$w_0 = \left(\frac{1}{4} \cdot 20\right) - \left(\frac{0.9}{4} \cdot 20\right) = 5 - 4.5 = 0.5$$

- (b) Figure 1 shows a backpropagation network that is currently processing the training vector $[1.0, 0.9, 0.9]$ that 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 = $-0.3 \times 0.6 + 0.9 \times 0.8 = 0.54 \rightarrow f(0.54) = 0.632$
 Second output unit input = $-0.6 \times 0.6 + -0.1 \times 0.8 = -0.44 \rightarrow f(-0.44) = 0.392$
 Third output unit input = $0.4 \times 0.6 + 1.2 \times 0.8 = 1.2 \rightarrow f(1.2) = 0.769$

- (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 = $(0.1 - 0.769) = -0.696$

- (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: $Err_j = \sum_{i=1}^n W_{ji} \times Err_i$:
 $Err_B = (-0.3 \times -0.532) + (-0.6 \times 0.508) + (0.4 \times -0.696) = 0.1596 + -0.3048 + -0.2784 = -0.4236$
 $Err_C = (0.9 \times -0.532) + (-0.1 \times 0.508) + (1.2 \times -0.696) = -0.4788 + -0.0508 + -0.8352 = -1.3648$

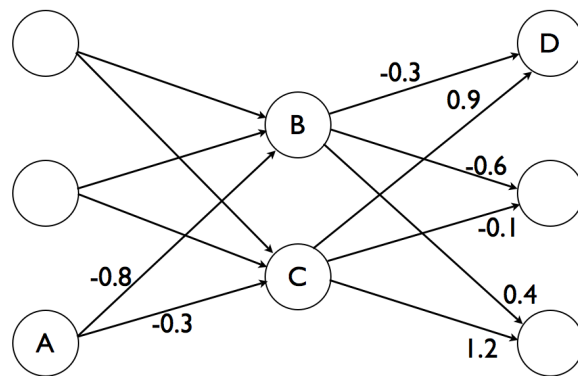


Figure 1: Example Neural Net