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
13^{th} 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 ***

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.

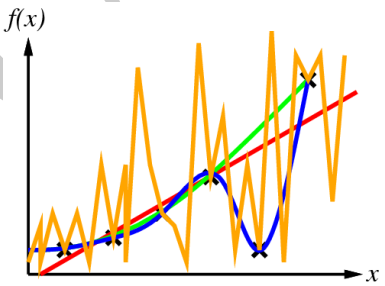
(5 marks)

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?

(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	Feature 1	Feature 2	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	Feature 1	Feature 2	Target
250	4	240000	?

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

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

(i) With which target class (**C1** or **C2**) will our **3-Nearest Neighbor** model label the query? Provide an explanation for your 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 **Feature 1** and **Feature 2**. To account for this we decide to normalize the data. Compute the normalized versions of Feature 1 and Feature 2 to four decimal places of precision using range normalization

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

(4 marks)

ID	Feature 1	Feature 2	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, with which target class (**C1** or **C2**) will our **3-Nearest Neighbor** model label the query? Provide an explanation for your answer.

(8 marks)

The normalize query instance is:

ID	Feature 1	Feature 2	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 (Feature 1 and Feature 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)

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)

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

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

Table 3: Classification dataset for information question.

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

Table 4: Equations from information theory.

Table 5: Movie and Song Title Dataset

Movie Titles	**Song Titles**
A Perfect World	*A Perfect Day*
My Perfect Woman	*Electric Storm*
Pretty World	*Another Rainy Day*

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

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

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

   (i) $P(Movie) = ?$

   (3 marks)

   $P(Movie) = \frac{3+1}{6+(1\times2)} = \frac{4}{8} = 0.5$

   (ii) $P(Song) = ?$

   (3 marks)

   $P(Song) = \frac{3+1}{6+(1\times2)} = \frac{4}{8} = 0.5$

   (iii) $P('Perfect'|Movie) = ?$

   (3 marks)

   $P('Perfect'|Movie) = \frac{2+1}{8+(1\times11)} = \frac{3}{19} = 0.1579$

   (iv) $P('Perfect'|Song) = ?$

   (3 marks)

   $P('Perfect'|Song) = \frac{1+1}{8+(1\times11)} = \frac{2}{19} = 0.1053$

   (v) $P('Storm'|Movie) = ?$

   (3 marks)

   $P('Storm'|Movie) = \frac{0+1}{8+(1\times11)} = \frac{1}{19} = 0.0526$

   (vi) $P('Storm'|Song) = ?$

   (3 marks)

   $P('Storm'|Song) = \frac{1+1}{8+(1\times11)} = \frac{2}{19} = 0.1053$

   (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 (a):

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

(8 marks)

$P(Movie|'PerfectStorm')$

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

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

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

(4 marks)

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

$P(Movie|'PerfectStorm')$

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

x	0	1	2	3	4
y	3	6	7	8	11

Table 7: Example Dataset for Linear Regression Question

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

$$f(x) = w_1 x + w0$$

where $w1$ and $w0$ are computed using the following formulae, where M is number of data points in the dataset:

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

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

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 = 5
- $\sum_{i=1}^{M} x_i y_i = 0 + 6 + 14 + 24 + 44 = 88$
- $\sum_{i=1}^{M} x_i = 10$
- $\sum_{i=1}^{M} y_i = 35$
- $\sum_{i=1}^{M} x_i^2 = 0 + 1 + 4 + 9 + 16 = 30$
- $(\sum_{i=1}^{M} x_i)^2 = 10^2 = 100$

Given these values, w_1:

$$w_1 = \frac{(5*88)-(10*35)}{(5*30)-100} = \frac{90}{50} = 1.8$$

And, w_0:

$$w_0 = (\frac{1}{5} * 35) - (\frac{1.8}{5} * 10) = 7 - 3.6 = 3.4$$

(b) Figure 1, on the next pages, shows a backprogation network 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, carry out the calculations listed below. Note that Table 8, also on the next page, lists some equations that you may find useful when doing this question.

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

(10 marks)

$$
\begin{aligned}
a_i(in_i) &= \frac{1}{1+\exp^{-((W_{BD}\times a_B(in_B))+(W_{CD}\times a_C(in_C)))}} \\
&= \frac{1}{1+\exp^{-((-0.3\times 0.6)+(0.9\times 0.8))}} \\
&= \frac{1}{1+\exp^{-0.54}} \\
&= 0.632
\end{aligned}
$$

$$
\begin{aligned}
a_j(in_j) &= \frac{1}{1+\exp^{-((W_{BE}\times a_B(in_B))+(W_{CE}\times a_C(in_C)))}} \\
&= \frac{1}{1+\exp^{-((-0.6\times 0.6)+(0.1\times 0.8))}} \\
&= \frac{1}{1+\exp^{-(-0.44)}} \\
&= 0.392
\end{aligned}
$$

$$
\begin{aligned}
a_k(in_k) &= \frac{1}{1+\exp^{-((W_{BF}\times a_B(in_B))+(W_{CF}\times a_C(in_C)))}} \\
&= \frac{1}{1+\exp^{-((0.4\times 0.6)+(1.2\times 0.8))}} \\
&= \frac{1}{1+\exp^{-1.2}} \\
&= 0.769
\end{aligned}
$$

(ii) Calculate the Δ error for each output unit (to 3 decimal places).

(6 marks)

$$
\begin{aligned}
\Delta_D &= (target_D - a_D(in_D)) \times a_D(in_D) \times (1 - a_D(in_D)) \\
&= (0.1 - 0.632) \times 0.632 \times (1 - 0.632) \\
&= -0.124
\end{aligned}
$$

$$
\begin{aligned}
\Delta_E &= (target_E - a_E(in_E)) \times a_E(in_E) \times (1 - a_E(in_E)) \\
&= (0.9 - 0.392) \times 0.392 \times (1 - 0.392) \\
&= 0.121
\end{aligned}
$$

$$
\begin{aligned}
\Delta_F &= (target_F - a_F(in_F)) \times a_F(in_F) \times (1 - a_F(in_F)) \\
&= (0.1 - 0.769) \times 0.769 \times (1 - 0.769) \\
&= -0.119
\end{aligned}
$$

(iii) Calculate the new weight W_{BD} for the connection from unit B to the output unit D after the training example has been processed. Use a learning rate of

$\eta = 0.25$.

(4 marks)

$$\begin{aligned} W_{B,D} &= W_{B,D} + (\eta \times a_B(in_B) \times \Delta_D) \\ &= -0.3 + (0.25 \times 0.6 \times -0.124) \\ &= -0.319 \end{aligned}$$

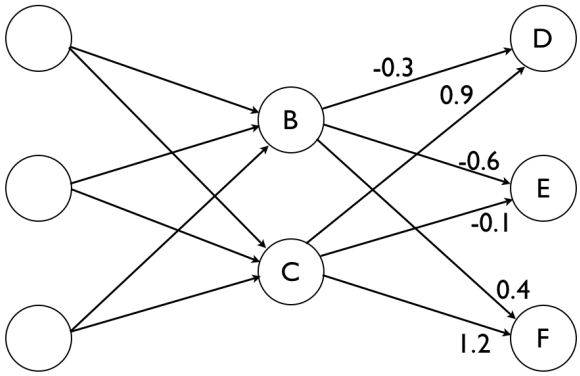


Figure 1: Example Neural Net

Weighted sum of inputs for unit i with j inputs:	$in_i = \sum_j W_{ji} a_j(in_j)$
Activation Function (Logistic) for unit i:	$a_i(in_i) = \frac{1}{1+\exp^{-in_i}}$
Perceptron weight update rule for link $j \rightarrow i$	$w_{ji} = w_{ji} + \eta\,(t_i - a_i(in_i)) \times a_j(in_j)$
Hebbian Weight Update Rule for link $j \rightarrow i$	$w_{ji} = \eta \times a_j(in_j) \times a_i(in_i)$
Partial Derivative for Logistic Activation Function	$\frac{\delta a_i(in_i)}{\delta in_i} = a_i(in_i) \times (1 - a_i(in_i))$
Error for an output unit i	$error_i = target_i - a_i(in_i)$
Delta Error for an output unit i	$\Delta_i = error_i \times a_i(in_i) \times (1 - a_i(in_i))$
Delta Error for a hidden unit j feeding into n units	$\Delta_j = (\sum_{i=1}^{n} W_{ji} \times \Delta_i) \times a_j(in_j) \times (1 - a_j(in_j))$
Delta Weight Update Rule for link $x \rightarrow k$	$W_{x,k} = W_{x,k} + (\eta \times a_x(in_x) \times \Delta_k)$

Table 8: Equations used in Perceptron and Neural Network training.