S249/419C

DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET, DUBLIN 8

BSc. (Hons) in Computer Science

Stage 4

SEMESTER 2 EXAMINATIONS 2012/2013

ARTIFICIAL INTELLIGENCE II

Dr. John Kelleher
Dr. Deirdre. Lillis
Mr. R. Walshe

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).

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

(5 marks)

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

(5 marks)

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

(15 marks)

(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)

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)

(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)

(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)

(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)

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

(5 marks)

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)

   (ii) $P(Song) =?$

   (3 marks)

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

   (3 marks)

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

   (3 marks)

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

   (3 marks)

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

   (3 marks)

   (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)

   (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)

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:

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

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

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)

(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)

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

(6 marks)

(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)

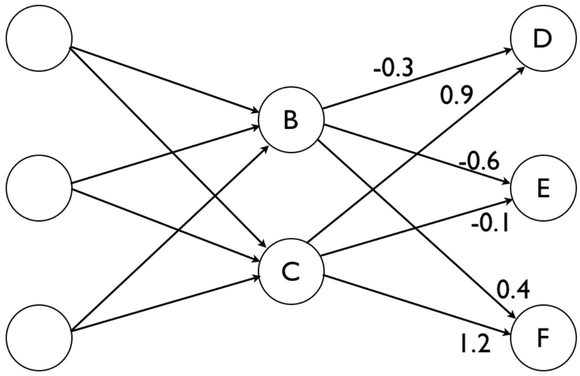

Figure 1: Example Neural Net

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.