S228/419C

DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET, DUBLIN 8

BSc. (Hons) in Computer Science

Stage 4

SEMESTER 2 EXAMINATIONS 2013/2014

*** *SOLUTIONS* ***

ARTIFICIAL INTELLIGENCE II

Dr. John Kelleher
Dr. Deirdre. Lillis
Mr. P. Collins

Monday
12^{th} May 2014
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**.

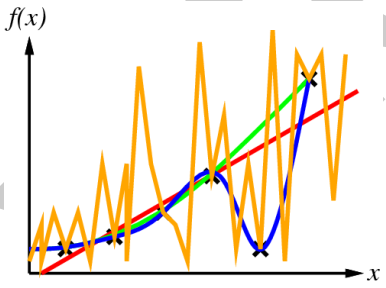
(5 marks)

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) 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



(c) 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 assumptions 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.
- Examples of the specific inductive bias introduced by particular machine learning algorithms, include:
  - 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.

(d) Explain what can go wrong when a machine learning classifier uses the wrong inductive bias.

(5 marks)

- If the inductive bias of the learning algorithm constrains the search to only consider simple hypotheses we may have excluded the real function from the hypothesis space. In other words, the true function is **unrealizable** in the chosen hypothesis space, (i.e., we are **underfitting**).
- If the inductive bias of the learning algorithm allows the search to consider complex hypotheses, the model may hone in on irrelevant factors in the training set. In other words the model with **overfit** the training data.

2. (a) A data analyst building a k-nearest neighbour model for a continuous prediction problem is considering appropriate values to use for k.

   (i) Initially the analyst uses a simple average of the target variables for the k nearest neighbours in order to make a new prediction. After experimenting with values for k in the range $0 - 10$ it occurs to the analyst that they might get very good results if they set k to the total number of instances in the training set. Do you think the analyst is likely to get good results using this value for k?

   (5 marks)

   In answering this question students should realise that if the analyst set k to the number of training examples all predictions would essentially be the average target value across the whole dataset. To score very well students should realise that this is an example of massive underfitting.

   (ii) If the analyst was using a distance weighted averaging function rather than a simple average for their predictions would this have made their idea any more useful?

   (5 marks)

   Students should realise that yes, if distance weighted voting is used (particularly if a $\frac{1}{d^2}$ type distance weight is used) then examples that are far away from the query will have very little impact on the result. Again to score well students should mention that when distance weighted voting is used the value of k in k-NN classifiers is much less important.

   (b) A dataset showing the decisions made by an individual about whether to wait for a table at a restaurant is listed in Table 1 on the next page. (Note that Table 2, also on the next page, lists some equations that you may find useful for this question.)

   (i) Given that the $WillWait$ column lists the values of the target variable, compute the entropy for this dataset.

   (5 marks)

   There are 6 positive and 6 negative examples in this dataset. This means that the entropy for the dataset is:

$$\begin{aligned} I(\frac{6}{12},\frac{6}{12}) &= -\frac{6}{12}\log_2\frac{6}{12} + -\frac{6}{12}\log_2\frac{6}{12} \\ &= (-\frac{1}{2}\log_2\frac{1}{2}) + (-\frac{1}{2}\log_2\frac{1}{2}) \\ &= -\frac{1}{2}(-1) + -\frac{1}{2}(-1) \\ &= 1bit \end{aligned}$$

   (ii) What is the information gain for the $Patrons$ feature?

   (5 marks)

   $Gain(Patrons) = 1 - (\frac{2}{12}I(0,1) + \frac{4}{12}I(1,0) + \frac{6}{12}I(\frac{2}{6},\frac{4}{6})) \approx 0.541$ bits

   (iii) What is the information gain for the $Type$ feature?

(5 marks)

$Gain(Type) = 1 - (\frac{2}{12}I(\frac{1}{2},\frac{1}{2}) + \frac{1}{2}I(\frac{1}{2},\frac{1}{2}) + \frac{4}{12}I(\frac{2}{4},\frac{2}{4}) + \frac{4}{12}I(\frac{2}{4},\frac{2}{4})) = 0$ bits

(iv) Given a choice between the $Patrons$ and $Type$ feature, which feature would the ID3 algorithm choose as the root node for a decision tree?

(5 marks)

The ID3 algorithm would choose the Patrons feature as the root node for the decision tree because it has the higher information gain.

ID	Bar	$Patrons$	$Price$	$Rain$	$Type$	WillWait
1	F	Some	$$$	F	French	T
2	F	Full	$	F	Thai	F
3	T	Some	$	F	Burger	T
4	F	Full	$	F	Thai	T
5	F	Full	$$$	F	French	F
6	T	Some	$$	T	Italian	T
7	T	None	$	T	Burger	F
8	F	Some	$$	T	Thai	T
9	T	Full	$	T	Burger	F
10	T	Full	$$$	F	Italian	F
11	F	None	$	F	Thai	F
12	T	Full	$	F	Burger	T

Table 1: A dataset describing the previous decisions made by an individual about whether to wait for a table at a restaurant.

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

Table 2: Equations from information theory.

3. Table 3 (on the next page) lists a dataset of the subject lines from emails. Table 4 (also on the next page) shows the subject line for an email that we would like to classify as Spam or Ham.

   (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 12**, calculate the following probabilities:

   (i) $P(Spam) = ?$

   (2 marks)

   $P(Spam) = \frac{3+1}{8+(1\times2)} = \frac{4}{10} = 0.4$

   (ii) $P(Ham) = ?$

   (2 marks)

   $P(Ham) = \frac{5+1}{8+(1\times2)} = \frac{6}{10} = 0.6$

   (iii) $P('Fun'|Spam) = ?$

   (2 marks)

   $P('Fun'|Spam) = \frac{0+1}{9+(1\times12)} = \frac{1}{21} = 0.0476$

   (iv) $P('Fun'|Ham) = ?$

   (2 marks)

   $P('Fun'|Ham) = \frac{2+1}{15+(1\times12)} = \frac{3}{27} = 0.1111$

   (v) $P('is'|Spam) = ?$

   (2 marks)

   $P('is'|Spam) = \frac{1+1}{9+(1\times12)} = \frac{2}{21} = 0.0952$

   (vi) $P('is'|Ham) = ?$

   (2 marks)

   $P('is'|Ham) = \frac{1+1}{15+(1\times12)} = \frac{2}{27} = 0.0741$

   (vii) $P('Free'|Spam) = ?$

   (2 marks)

   $P('Free'|Spam) = \frac{3+1}{9+(1\times12)} = \frac{4}{21} = 0.1905$

   (viii) $P('Free'|Ham) = ?$

   (2 marks)

   $P('Free'|Ham) = \frac{1+1}{15+(1\times12)} = \frac{2}{27} = 0.0741$

   (b) Calculate the probability of the query title in Table 4 belonging to the Spam class under the **Naive Bayes assumption** and using the **smoothed probabilities** you calculated in Part (a):

   $$P(Spam|'Fun\ is\ Free') = ?$$

   (7 marks)

$P(Spam|'FunisFree')$

$$= \frac{P('Fun'|Spam)P('is'|Spam)P('Free'|Spam)P(Spam)}{(P('Fun'|Spam)P('is'|Spam)P('Free'|Spam)P(Spam)) + (P('Fun'|Ham)P('is'|Ham)P('Free'|Ham)P(Ham))}$$

$$= \frac{0.0476 \times 0.0952 \times 0.1905 \times 0.4}{(0.0476 \times 0.0952 \times 0.1905 \times 0.4) + (0.1111 \times 0.0741 \times 0.0741 \times 0.6)}$$

$$= 0.48543863$$

(c) Calculate the probability of the query title in Table 4 belonging to the Spam 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):

$$P(Spam|'Fun\ is\ Free') = ?$$

(7 marks)

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

P(Spam—'Fun is Free')

$$= \frac{P('Fun'|Spam)P('is'|Spam)P('Free'|Spam)P(Spam)}{(P('Fun'|Spam)P('is'|Spam)P('Free'|Spam)P(Spam)) + (P('Fun'|Ham)P('is'|Ham)P('Free'|Ham)P(Ham))}$$

$$= \frac{\frac{0}{9} \times \frac{1}{9} \times \frac{3}{9} \times \frac{3}{8}}{(\frac{0}{9} \times \frac{1}{9} \times \frac{3}{9} \times \frac{3}{8}) + (\frac{2}{15} \times \frac{1}{15} \times \frac{1}{15} \times \frac{5}{8})}$$

$$= 0.0$$

Table 3: Spam and Ham Dataset

Spam
Offer is Free
Free Learning Link
Cick Free Link

Ham
Great Learning Fun
Great Machine Learning
Free Learning Event
Learning is Fun
Learning Costs Money

Table 4: Query Title

Fun is Free

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

Table 5: 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 descriptive and target features:

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

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 5 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 6 (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)

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

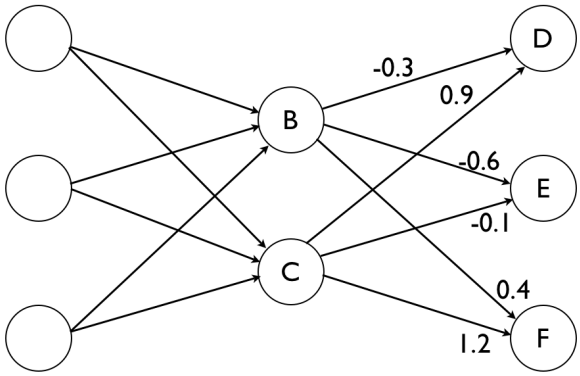


Figure 1: Example Neural Net

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	$\Delta_j = \left(\sum_{i=1}^{n} W_{ji} \times \Delta_i\right) \times a_j(in_j) \times (1 - a_j(in_j))$
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 6: Equations used in Perceptron and Neural Network training.