UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR10069 AND INFR11182 INTRODUCTORY APPLIED MACHINE LEARNING

Monday 13 th December 2021

13:00 to 15:00

INSTRUCTIONS TO CANDIDATES

1. **Note that ALL QUESTIONS ARE COMPULSORY.**
2. **DIFFERENT QUESTIONS MAY HAVE DIFFERENT NUMBERS OF TOTAL MARKS. Take note of this in allocating time to questions.**
3. **This is an OPEN BOOK examination.**

Year 3 Courses

Convener: D.Armstrong
External Examiners: J.Bowles, A.Pocklington, H.Vandierendonck

MSc Courses

Convener: A.Pieris
External Examiners: A. Cali, V. Gutierrez Basulto

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. **Decision Trees**

You have begun a collaboration with an environmental scientist who asks if you can develop a machine learning classifier that can determine if a sample of water is dangerous or safe to drink. As an initial pilot study, you are given a small training dataset which contains information about eight different water samples. Each sample contains a set of values (x_1, x_2, y), where x_1 and x_2 are measurements of two different chemical properties of water and $y \in \{0, 1\}$ indicates if the sample is dangerous to drink (i.e. $y = 1$) or safe to drink (i.e. $y = 0$).

The training set is presented below, where each column is a different water sample:

	1	2	3	4	5	6	7	8
x_1	1.5	2.7	1.7	3.5	1.9	4.0	2.5	0.0
x_2	0.0	4.0	3.0	2.0	6.0	5.0	2.0	3.0
y	0	1	1	0	0	0	1	0

(a) You have the option to use a Gaussian Naive Bayes, Logistic Regression, or Decision Tree classifier for addressing this problem. For each of the choices, briefly discuss if the classifier, once trained, would result in a training error of 0% for the specific data above. You should justify your answer with reference to the data above, but you do not need to perform the steps required to train the classifiers, nor report the training error for each. [*3 marks*]

(b) For your first attempt, you decide to use a Decision Tree classifier. Compute the information gain associated with splitting the x_1 attribute using a threshold of 2.0 for only the first node in the tree. Compare this to splitting the same attribute using a threshold of 3.0, again only for the first node. Report the information gain for both cases and explain which is better. You should use entropy as your measure of impurity. In each case, show your workings. You can use the following values in your calculations: [*4 marks*]

a:	1/8	2/8	3/8	4/8	5/8	6/8	7/8	8/8
$\log_2(a)$:	−3.0	−2.0	−1.4	−1.0	−0.7	−0.4	−0.2	0.0

(c) You take a look at the original data and notice that there is one additional water sample $(2.7, 4.0, 0)$ that you previously missed. What issues, if any, would arise if you added this to the data and how would you address them? [*2 marks*]

(d) Your collaborator performed some additional data collection and was able to add hundreds of new samples of safe water (i.e. $y = 0$) to the training set. Assuming you have trained the classifier again by adding this new set of data to your original training set, describe a suitable performance measure for evaluating how good the training performance is. Discuss any considerations that should be taken into account when performing this evaluation. [*2 marks*]

(e) Someone suggests that you could try training a Random Forest classifier on the new larger training set. Do you think this a good suggestion for this dataset? Explain why. [*1 mark*]

2. (a) **Performance Evaluation**

You have been asked to evaluate the performance of two different probabilistic binary classifiers, g_1 and g_2. Each classifier takes a vector of attributes $\boldsymbol{x}_i$ as input and estimates the probability that $\boldsymbol{x}_i$ is from the positive class i.e. $p(y_i = 1 \mid \boldsymbol{x}_i) = g_k(\boldsymbol{x}_i),\ k = 1, 2$. You do not have direct access to the classifiers, but instead have their predictions for a set of validation examples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_6\}$, along with the ground truth class labels $\{y_1, y_2, ..., y_6\}$.

i	1	2	3	4	5	6
y_i	0	1	1	0	0	1
$g_1(\boldsymbol{x}_i)$	0.1	0.9	0.7	0.3	0.2	0.2
$g_2(\boldsymbol{x}_i)$	0.2	0.2	0.6	0.7	0.6	0.3

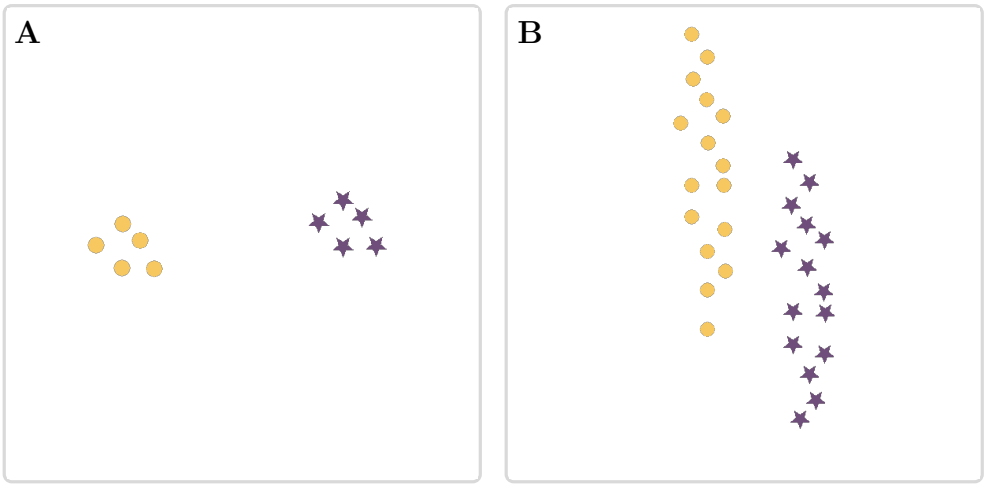
i. Report the F-measure for the two classifiers using their predictions above, using a threshold of 0.5. Which of the two classifiers performs better on this validation set? Show your workings. *[2 marks]*

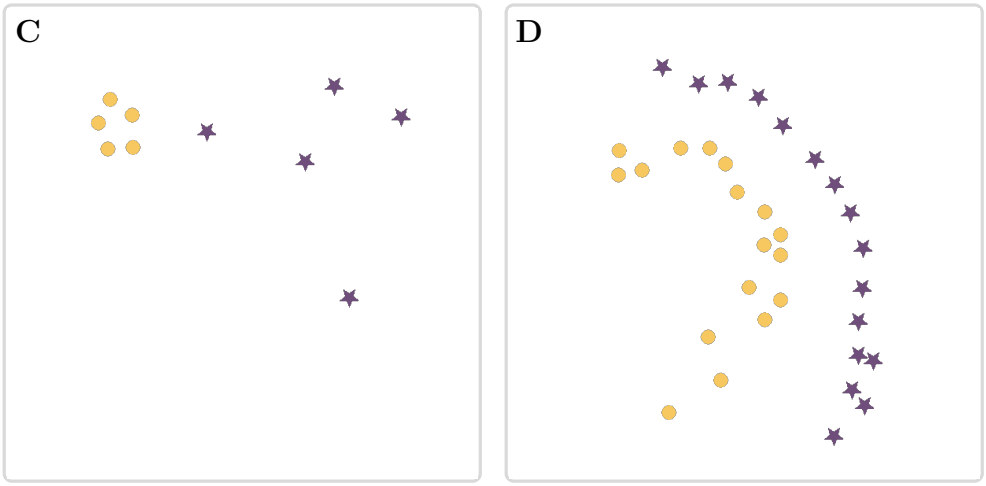
ii. For the second classifier only (i.e. g_2), report the true positive rate and false positive rate corresponding to the thresholds 0.0, 0.33, 0.66, and 1.0. Show your workings. *[2 marks]*

iii. Sketch the resulting ROC plot for g_2. You only need to show the points on the curve corresponding to the thresholds from the previous question. Comment on any observations from your plot. *[3 marks]*

(b) **Clustering**

Consider the following four datasets (A-D) containing 2-dimensional points, each showing a K=2 clustering. Cluster assignments are denoted by yellow circles for cluster 0 and purple stars for cluster 1.



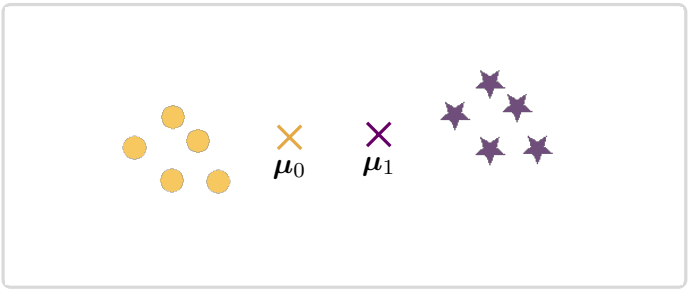


i. For each dataset, select which amongst {K-Means clustering, GMMs, single-link clustering} could have led to the cluster assignments shown. Choose all that could apply, or report `None` if none apply. [*2 marks*]

ii. Discuss and briefly explain if any of the three clustering methods mentioned in part (i) *would not* have resulted in the clustering shown in image D. [*2 marks*]

iii. Consider applying K-Means clustering and GMM (trained with EM) to the data shown in image A. Will these two clustering algorithms result in the *same* cluster centers? Briefly explain why, or why not? [*3 marks*]

iv. For a GMM, consider the given estimates of mean vectors $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ of the two mixture components after the k^{th} iteration of EM shown in the image below. Which direction, if any, will each of $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ move in during the next M-step? (*Hint*: Providing general directions like 'left' or 'none' is sufficient.) Will the marginal likelihood of the data increase or decrease in the next EM iteration? Explain your reasoning briefly. [*2 marks*]



3. (a) Given a training set comprised of the following data points and class labels:

x_1	0	-1	-2	0	1	1
x_2	2	2	3	0	0	1
class	$+1$	$+1$	$+1$	-1	-1	-1

i. Plot these six data points. Are the classes $\{-1, +1\}$ linearly separable? *[1 mark]*

ii. Determine all *support vectors* for a linear SVM and, by observation, report the weight vector that defines the *maximum margin hyperplane*. *[2 marks]*

iii. Does the width of the margin for the maximum-margin hyperplane change if one of the support vectors is removed? Discuss your answer. *[3 marks]*

(b) Consider a second training set with three data points and their class labels as in the table below:

x	0	$-\sqrt{2}$	$+\sqrt{2}$
class	$+1$	-1	-1

i. Are the two classes in this second training set linearly separable? *[1 mark]*

ii. Apply the transformation $\boldsymbol{\phi}(x) = [1, x, x^2]^T$ on the second training set. Are the two classes in this new 3-dimensional space linearly separable? If yes, determine the maximum-margin hyperplane. If no, justify your answer. *[2 marks]*

iii. Again using the same training set from (b), let us define a vector $\boldsymbol{w} = [w_1, w_2, w_3]^T$ and a class variable $y_i \in \{-1, +1\}$, which is associated with data point x_i. We know that the max-margin support vector machine classifier solves the following optimisation problem (where $\|\cdot\|_2^2$ is the squared Euclidean norm):

$$\min_{\boldsymbol{w}, w_0} \frac{1}{2} \|\boldsymbol{w}\|_2^2 \quad \text{s.t.} \quad y_i(\boldsymbol{w}^T \boldsymbol{\phi}(x_i) + w_0) \geq 1, \; i = 1, 2, 3.$$

We also know that for the support vectors, the constraint in the optimisation problem above becomes $y_i(\boldsymbol{w}^T \boldsymbol{\phi}(x_i) + w_0) = 1$. Using Lagrange Multipliers, we can show that the Lagrangian function $L(\boldsymbol{w}, \boldsymbol{\alpha}, w_0)$ will be:

$$L(\boldsymbol{w}, \boldsymbol{\alpha}, w_0) = \frac{1}{2} \|\boldsymbol{w}\|_2^2 - \sum_{i=1}^{3} \alpha_i \left(y_i(\boldsymbol{w}^T \boldsymbol{\phi}(x_i) + w_0) - 1 \right),$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]$. The partial derivatives of $L(\boldsymbol{w}, \boldsymbol{\alpha}, w_0)$ with

respect to $\boldsymbol{w}$ and w_0 are:

$$\frac{\partial L(\boldsymbol{w}, \boldsymbol{\alpha}, w_0)}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{3} \alpha_i y_i \boldsymbol{\phi}(x_i)$$

$$\frac{\partial L(\boldsymbol{w}, \boldsymbol{\alpha}, w_0)}{\partial w_0} = -\sum_{i=1}^{3} \alpha_i y_i$$

Show that the weights of this support vector machine classifier are $\widehat{\boldsymbol{w}} = [0, 0, -1]^T$ and $\widehat{w}_0 = 1$. *Hint*: You do not need to calculate $\boldsymbol{\alpha}$. [*3 marks*]

4. **Neural Networks**

You are given a trained neural network being used for a specific binary classification problem. It takes a 2-dimensional attribute vector $\boldsymbol{x} = [x_0, x_1]^T$ as input and predicts a single continuous output $y \in [0, 1]$, which can be interpreted as the probability the input is "positive" i.e. $p(y = 1 \mid \boldsymbol{x})$. The network performs the following operations:

$$\boldsymbol{h} = \tanh(W\boldsymbol{x} + \boldsymbol{b})$$
$$y = \sigma(V\boldsymbol{h} + b^\star)$$

with parameters $W = \begin{bmatrix} w_{00} & w_{01} \\ w_{10} & w_{11} \end{bmatrix}$, $\boldsymbol{b} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$, $V = \begin{bmatrix} v_0 & v_1 \end{bmatrix}$, and $b^\star$, and activation functions (that apply independently over each dimension) given as

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}},\ \sigma(a) = \frac{1}{1 + e^{-a}}.$$

(a) You notice that the network uses *two* different non-linear activation functions. Why was a tanh function also not used for the output? [*1 mark*]

(b) Suppose the network has the following weights and biases:

$$W = \begin{bmatrix} 50 & -50 \\ 50 & -50 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} -50 \\ 50 \end{bmatrix}, \quad V = \begin{bmatrix} -50, -50 \end{bmatrix}, \quad \text{and } b^\star = -50.$$

Specify a range of values for $\boldsymbol{x}$ such that the network always classifies an input as "positive" (i.e., $p(y = 1 \mid \boldsymbol{x}) = 1$). Justify your answer. [*2 marks*]

Hint: You may find simplifying the activations to 'hard' versions helpful for reasoning, i.e., tanh $\approx$ sign, and $\sigma \approx$ ReLU, where:

$$\text{sign}(a) = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{if } a = 0 \\ -1 & \text{if } a < 0 \end{cases}, \quad \text{ReLU}(a) = \begin{cases} a & \text{if } a > 0 \\ 0 & \text{otherwise} \end{cases}$$

(c) You have been asked to change the network so that it now classifies all inputs $\boldsymbol{x}$ such that $x_0 - 1 < x_1 < x_0 + 1$ as "positive" and all other inputs as "negative". However, you must do this by changing *only one* weight or bias of the network. Which weight or bias should be changed and to what value? Justify your answer. [*2 marks*]

(d) A colleague with an interest in Boolean logic asks if your neural network can emulate the simplicity of logical gates. Taking only the latter part of the network above, i.e., $y = \sigma(V\boldsymbol{h} + b^\star)$, she stipulates that the given inputs $\boldsymbol{h} = [h_0, h_1]^T$ and output y are binary, i.e., $h_0, h_1, y \in \{0, 1\}$. She also provides you with the "truth tables" for some common Boolean logic functions and asks if your (reduced) network can replicate them.

	AND				OR				NOT		
h_0	0	0	1	1	h_0	0	0	1	1	h_0	0 1
h_1	0	1	0	1	h_1	0	1	0	1	y	1 0
y	0	0	0	1	y	0	1	1	1		

Find parameter values, i.e., V and $b^\star$ for each of these functions such that the neural network produces the corresponding outputs for the given inputs. For `NOT`, you may assume that h_1 can be ignored. You can assume that the parameters are all simple integers, and that the activation function in this instance is the ReLU function shown above (i.e. $\sigma = \text{ReLU}$).

Hint: You should report three sets of parameters, one for each logical gate. [*3 marks*]

(e) She issues one final challenge, asking you to emulate an `XOR` function using the same neural network model.

h_0	0	0	1	1
h_1	0	1	0	1
y	0	1	1	0

Is this feasible? If so, what parameter values $(V, b^\star)$ would satisfy `XOR`? If not, explain why and describe a remedy to satisfy the requirements. [*2 marks*]