2020-2021 Intro to ML Term 2 Exam solution

This is not an official answer and not all correct. If you spot anything wrong, just contact xz1919@ic.ac.uk :)

1. a. Using 10-fold cross validation, compute the mean accuracy for a 3-nearest neighbour classifier that uses the given distances.

   This question might seem baffling at the first glance and might involve a lot of calculation but it's actually fine

   We first ***shuffle*** the data set and then choose to divide the dataset into training and test set by the ratio of 8:2 in each fold. Since KNN is a lazy learner, it doesn't learn anything until an "explicit request is made at test time", according to the slide from this course.

   So each time we can just pick up two data points in the test set and just classify it based on the classification of the 3 nearest neighbours

Folder No. & Test data points	3 Nearest Neighbours	Classification	Accuracy
Fold 1 Test data: 1, 7	1: 4, 5, 6 7: 8, 9, 10	1: + 7: -	100%
Fold 2 Test data: 4, 9	4: 2, 3, 5 9: 7, 8, 10	4: + 9: -	100%
Fold 3 Test data: 6, 10	6: 1, 5, 8 10: 7, 9, 3	6: + 10: -	50%
Fold 4 Test data: 2, 8	2: 3, 4, 5 8: 9, 6, 1	2: + 8: -	100%
Fold 5 Test data: 3, 6	3: 2, 4, 5 6: 1, 5, 8	3: + 6: +	50%
Fold 6 Test data: 5, 10	5: 6, 1, 4 10: 7, 9, 3	5: + 10: -	100%
Fold 7 Test data: 1, 3	1: 4, 5, 6 3: 2, 4, 5	1: + 3: +	100%
Fold 8 Test data: 4, 7	4: 2, 3, 5 7: 8, 9, 10	4: + 7: -	100%
Fold 9 Test data: 3, 8	3: 2, 4, 5 8: 9, 6, 1	3: + 8: -	100%
Fold 10 Test data: 6, 7	6: 1, 5, 8 7: 8, 9, 10	6: + 7: -	50%

   Hence the total averaged accuracy would be 85%

   You can also split the dataset by the ratio of 9:1 to save calculation, but 8:2 would result in better accuracy

b. You would like to predict whether a person is happy or unhappy based on the person's level of income and stress.

i) Compute the initial entropy for the whole dataset. Show your workings (or justify your answer in one sentence). Please use log2 for your calculations.

The initial entropy would just be one since there are equal number of happy and unhappy instances. A quick heuristic is that if dataset contains equal amount of two/multiple classes then the entropy is one; if the dataset only contains one class then entropy is 0

Calculation: $-\frac{100}{200}\log_2\frac{100}{200}-\frac{100}{200}\log_2\frac{100}{200}=1$

ii) Compute the information gains for selecting the income attribute and for selecting the stress attribute, each with respect to the initial entropy of the dataset. Please show all intermediate calculations. Please use log2 for all calculations.

For selecting the income attribute, we have

income	happy	unhappy	Grand Total
low	60	40	100
high	40	60	100
Grand Total	100	100	200

The information gain is thus

$$\begin{aligned} IG &= 1-\frac{100}{200}(-\frac{60}{100}\log_2\frac{60}{100}-\frac{40}{100}\log_2\frac{40}{100})-\frac{100}{200}(-\frac{40}{100}\log_2\frac{40}{100}-\frac{60}{100}\log_2\frac{60}{100}) \\ &= 1-0.97095 \\ &= 0.02905 \end{aligned}$$

Similarly, for selecting the stress attribute, we have

stress	happy	unhappy	Grand Total
low	70	20	90
high	30	80	110
Grand Total	100	100	200

The information gain is thus

$$\begin{aligned} IG &= 1-\frac{90}{200}(-\frac{70}{90}\log_2\frac{70}{90}-\frac{20}{90}\log_2\frac{20}{90})-\frac{110}{200}(-\frac{30}{110}\log_2\frac{30}{110}-\frac{80}{110}\log_2\frac{80}{110}) \\ &= 1-0.80883 \\ &= 0.19117 \end{aligned}$$

iii) Based on your calculations in (ii), which attribute should be selected as the root node of a decision tree classifier? Justify your answer in one sentence.

We should select the stress attribute because it results in a higher information gain. The higher the information gain, the more accurate we could get when splitting the dataset.

2. a. We have a neuron with 2 inputs and a sigmoid (logistic) activation function, which we use for binary classification. The weights of the neuron corresponding to the inputs are [−0.6, 0.8] and the bias is −0.9.

   i) You have one datapoint with features [0.9, −0.7] and it has the true label 1. We decide to use mean squared error as the loss function and 0.8 as the learning rate. Calculate the updated parameter values for this neuron after one step of stochastic gradient descent using this datapoint. Show the path of your calculations.

   We perform one step of stochastic gradient descent with the given data point.

$$
\begin{aligned}
W_{new} &= W_{old} - \alpha \frac{\partial Loss}{\partial W_{old}} \\
&= [-0.6 \quad 0.8] - 0.8 \cdot \frac{\partial Loss}{\partial \hat{Y}} \cdot \frac{\partial \hat{Y}}{\partial Z} \cdot \frac{\partial Z}{\partial W_{old}} \\
&= [-0.6 \quad 0.8] - 0.8 \cdot (\hat{Y} - Y) \cdot X^T \circ (g(z)(1 - g(z))) \\
&= [-0.6 \quad 0.8] - 0.8 \cdot (0.11920 - 1) \cdot [0.9 \quad -0.7] \circ 0.10499 \\
&= [-0.53342 \quad 0.74821]
\end{aligned}
$$

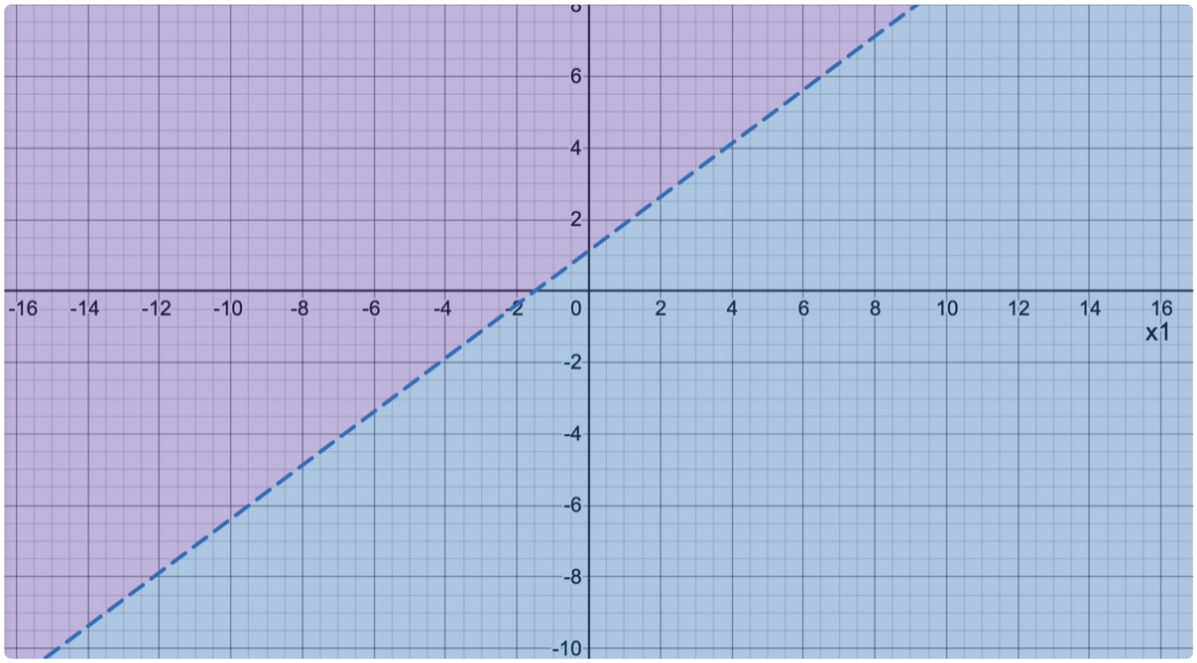
   where $\hat{Y} = 0.9 \cdot -0.6 - 0.7 \cdot 0.8 - 0.9 = -2$ and $g(z) = g(-2) = \frac{1}{1+e^2} = 0.11920$

   Notice that for the derivative of the MSE, you can either use $\frac{1}{2}$ or 1 for the coefficient.

   Similar calculation follows for b

$$
\begin{aligned}
b_{new} &= b - \alpha \frac{\partial Loss}{\partial b_{old}} \\
&= -0.9 - 0.8 \cdot \frac{\partial Loss}{\partial \hat{Y}} \cdot \frac{\partial \hat{Y}}{\partial Z} \cdot \frac{\partial Z}{\partial b_{old}} \\
&= -0.9 - 0.8 \cdot (\hat{Y} - Y) \cdot (g(z)(1 - g(z))) \\
&= -0.9 - 0.8 \cdot (0.11920 - 1) \cdot 0.10499 \\
&= -0.82602
\end{aligned}
$$

   ii) The decision boundary is as follows



The purple region represents classification label of 1, while the blue region represents 0

b. You have a neural network model for predicting the price of a company stock. You train the model for 100 epochs and get good RMSE performance on the training set but unexpectedly poor performance on the validation set, shown on the graph below. Diagnose the problem and offer two possible solutions to try in order to improve your final model performance.

This is clearly an overfit since the model has a very good loss result on the training set but not perform will on the validation set. Possible solutions include reducing the complexity of the model, stop training earlier, dropout, etc.

3. For simplicity, we will use the Manhattan distance (the L1 norm) throughout this question. We consider the following dataset of 2D coordinates:

a. Apply the K-Means algorithm with k = 2 and the data-points of indexes 1 and 6 as initial values for the centroids. Show the details of your calculations.

The K-means algorithm involves two steps: assignment and update.

We first assign each point to one of the two clusters based on the Manhattan distance.

Cluster 1 (index 1): 2, 4, 8

Cluster 2 (index 6): 3, 5, 7

Calculation is omitted. Basically for each point you compare the Manhattan distance from point 1 and point 6 and assign it to the nearer cluster.

We then update the two centroids based on the assignment:

$$C_{1Xnew} = \frac{1+2+3+6}{4} = 3.0$$
$$C_{1Ynew} = \frac{3+5+4+0}{4} = 3.0$$
$$C_{2Xnew} = \frac{8+2+4+6}{4} = 5.0$$
$$C_{2Ynew} = \frac{6+8+10+8}{4} = 8.0$$

The new centroids are therefore point (3, 3) and point (5, 8)

b. We now want to consider three clusters. We executed k-means with k=3 twice, with different initial conditions each time. We obtained the following two sets of clusters. Which of the two executions produced the best result? Show the details of your calculations.

We determine which is the best by looking at the average Manhattan distance of the two sets

For the first set, we first assign each point to one of the three centroids:

Cluster 1 (point [3, 9]): 5, 6

Cluster 2 (point [7, 7]): 3, 7

Cluster 3 (point [3, 3]): 1, 2, 4, 8

then calculate the mean Manhattan distance:

$$Avg.Dist = \frac{2+2+2+2+2+3+1+6}{8} = 2.5$$

For the second set, we first assign each point to one of the three centroids:

Cluster 1 (point [2, 4]): 1, 2, 4

Cluster 2 (point [6, 0]): 8

Cluster 3 (point [5, 8]): 3, 5, 6, 7

then calculate the mean Manhanttan distance:

$$Avg.Dist = \frac{2+1+1+0+5+3+3+1}{8} = 2$$

According to the calculations above, the second execution produced the best result because the average Manhattan distance is smaller for the second set of centroids.

c. Instead of using K-means to find the optimal position of the three centroids, we now want to use an Evolutionary Algorithm to achieve the same result.

Fitness function: the negative of the mean Manhattan distance of samples in every cluster. Negating it is for the purpose of making objective towards maximizing fitness function rather than minimizing.

Genotype: 2 * k real numbers where k is the number of clusters. Alternatively we can use 2 * k bit string where each string represent the X/Y coordinate of the centroids.

Phenotype: The location of k centroids along with the assignments for each point

Function used to develop a genotype into a phenotype: Mapping each pair of 2 * k real numbers onto each of the k centroids. If you use other representation such as bit string, then you need to map it differently.

Mutation operator: For real number representation, we can add Gaussian noices onto each number. For bitstring and other representation, mutating certain bits would work.

Notice in the exam you only need to cover one type of representation.

Q1

1a. This question tests your understanding of applying a k-NN classifier using cross-validation. Many did not answer this question. For those who did, many have managed to score full marks. Note that a k-NN classifier has to make a firm classification decision (even only with 2/3 confidence). Since each fold only has one test instance, the accuracy for each fold can only either be 0 or 1 (the classifier either gets the prediction correct or not). Full marks are awarded if you computed each accuracy correctly, averaged the accuracies, and reported the correct average accuracy. Those whose reported individual accuracies that are not 0 or 1 end up with about half of the allotted marks on average based on their workings. Partial marks (often 1/6 of the allotted marks) were given to those who only showed the k-NN classifier output without computing any accuracies.

1b. Everyone who answered received full marks for (i) and (iii). For (ii), you should understand that entropy/IG is a measure of 'purity' of a dataset to be able to solve this. Many of you received full marks. Marks are awarded for both your workings and the final answer for computing the individual entropies per attribute value and the IG per attribute. Minor calculation errors are often not penalised too strongly as long as your workings are correct. Those who struggled are those who could not figure out the correct numbers to plug into the equations as provided in the lectures.

Q2

2a.i) Mostly solved correctly but with various small errors. For example, combining the input features with the incorrect weights when performing the forward pass, or forgetting about the bias term entirely. In some cases the update rule was used with the incorrect sign or omitted. Basic arithmetic errors also occurred. Whenever a mistake was made, points were deducted for that particular error but points were still awarded for the following operations if they were correct.

2a.ii) The core idea to understand was that the decision boundary for a single neuron is a straight line. That holds true even when the neuron has a nonlinear sigmoid activation - the sigmoid only scales the output value but doesn't change the decision boundary. This could also be worked out by feeding some points into the neuron and seeing what the predicted label is. The task then becomes solving a linear equation, to find the line where the neuron predictions are exactly between two classes.

2b) This was solved correctly by the majority of students. Many possible methods were acceptable suggestions to stop the model from overfitting.

Q3

3a) The objective of this question is to apply K-Means (with k=2) to the provided dataset. This question was mostly solved correctly. The main errors were mostly some mistakes when computing the distance and making the assignments to the different clusters. Some students also lost points because they did not finish the question. The majority of the students seems to understand the overall principles of the algorithm.

3b) For the second question, the goal was to evaluate the quality of two independent outputs from the K-means algorithm. Most of the students found the appropriate way to solve this question. The main error was again some computation mistake and lack of time.

3c) For the final question, the objective was to use an evolutionary algorithm to find the optimal centroids. The main source of error in this question was the selection of an inappropriate evolutionary algorithm. Given that the centroid positions are floats, evolutionary strategies should be used. When the appropriate algorithm was picked, the other elements (genotype, phenotype, mutations) were correctly defined by the students. The other most common mistake was the lack of a negative sign in front of the fitness function. Evolutionary algorithms always maximise, so to minimise a distance "d", the fitness should be "-d".