Machine Learning Assignment 3
Calculations

Alireza Dastmalchi Saei

Student ID: *993613026*

University: *University of Isfahan*

Course: *Machine Learning Course*

January 5, 2024

1 Question 1

1.1 a

Our given data point is (25, 5, 15) that are length, weight, height of an animal respectively. Using K nearest neighbors (in this example K=5) we can determine which class this data point belongs to.

Sorting the nearest neighbors to given data point by their Euclidean distances, we will have:

1. $X_1 = (20, 2, 10) \rightarrow$ Euclidean Distance: $\sqrt{59} = 7.6811$
2. $X_2 = (30, 3, 15) \rightarrow$ Euclidean Distance: $\sqrt{29} = 5.3852$
3. $X_3 = (60, 5, 15) \rightarrow$ Euclidean Distance: $\sqrt{1225} = 35$
4. $X_4 = (55, 12, 21) \rightarrow$ Euclidean Distance: $\sqrt{985} = 31.3847$
5. $X_5 = (54, 24, 30) \rightarrow$ Euclidean Distance: $\sqrt{1427} = 37.7757$
6. $X_6 = (60, 25, 50) \rightarrow$ Euclidean Distance: $\sqrt{2850} = 53.3854$
7. $X_7 = (23, 2.75, 12) \rightarrow$ Euclidean Distance: $\sqrt{18.0625} = 4.25$
8. $X_8 = (33, 5.75, 10) \rightarrow$ Euclidean Distance: $\sqrt{89.5625} = 9.4637$
9. $X_9 = (34, 4, 16) \rightarrow$ Euclidean Distance: $\sqrt{83} = 9.1104$
10. $X_{10} = (35, 13, 10.5) \rightarrow$ Euclidean Distance: $\sqrt{184.25} = 13.5739$

Nearest K=5 points to the given data point (25, 5, 15) are X_7 (Cat), X_2 (Cat), X_1 (Cat), X_9 (Cat), X_8 (Dog). So our data point belongs to **Cats** class.

1.2 b

Let's change values of K from 1 to 10 to see what happens to see how does it affect our prediction.

K = 1 → Predicted Class: **Cat** (1 Cat)

K = 2 → Predicted Class: **Cat** (2 Cats)

K = 3 → Predicted Class: **Cat** (3 Cats)

K = 4 → Predicted Class: **Cat** (4 Cats)

K = 5 → Predicted Class: **Cat** (4 Cats, 1 Dog)

K = 6 → Predicted Class: **Cat** (5 Cats, 1 Dog)

K = 7 → Predicted Class: **Cat** (6 Cats, 1 Dog)

K = 8 → Predicted Class: **Cat** (6 Cats, 2 Dog)

K = 9 → Predicted Class: **Cat** (6 Cats, 3 Dog)

K = 10 → Predicted Class: **Cat** (6 Cats, 4 Dog)

2 Question 2

2.1 a

$2^{(k-1)}$ By the definition of decision trees, Each feature can only be used once in each path from root to leaf. The maximum depth is O(k).

2.2 b

By the definition of decision trees, Continuous values can be used multiple times, so the maximum number of leaf nodes can be the same as the number of samples, N and the maximal depth can also be N.

3 Question 3

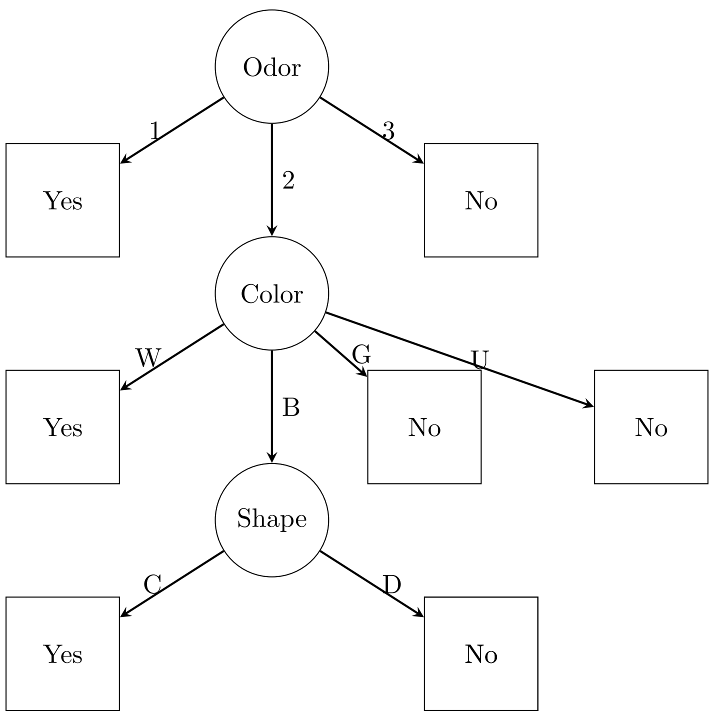
3.1 a

$$H(Edible|Order = 1orOrder = 3) = -\frac{3}{6}\log\frac{3}{6} - \frac{3}{6}\log\frac{3}{6} = 1$$

3.2 b

Odor $\rightarrow$ Has Highest Information Gain (IG)

3.3 c



3.4 d

3.4.1 Training Set Accuracy

For the given Training Set (Excluding Validation Set), our prediction with the decision tree above, will be:

Shape	**Color**	**Odor**	**Edible**	**Prediction**
C	B	1	Yes	Yes
D	B	1	Yes	Yes
D	W	1	Yes	Yes
D	W	2	Yes	Yes
C	B	2	Yes	Yes
D	B	2	No	No
D	G	2	No	No
C	U	2	No	No
C	B	3	No	No
C	W	3	No	No
D	W	3	No	No

As a result, the accuracy for the training set, will be:

$$Accuracy = \frac{\#TruePredictions}{\#AllPredictions} = \frac{11}{11} = 100\%$$

$$MisclassifiedPredictions = \frac{\#FalsePredictions}{\#AllPredictions} = \frac{0}{11} = 0\%$$

3.4.2 Validation Set Accuracy

Our validation set has following data points:

Shape	**Color**	**Odor**	**Edible**	**Prediction**
C	B	2	No	Yes
D	B	2	No	No
C	W	2	Yes	Yes

As a result, the accuracy for this validation set, will be:

$$Accuracy = \frac{\#TruePredictions}{\#AllPredictions} = \frac{2}{3} \approx 67\%$$

$$MisclassifiedPredictions = \frac{\#FalsePredictions}{\#AllPredictions} = \frac{1}{3} = 33.33\%$$

4 Question 4

4.1 a

As given the output of our single neuron is a scalar (y), so the formula of Error function will be:

$$\text{Error Function} \rightarrow (y - \mathbf{W}^T x)^2$$

Update Rule based on Gradient Descent will be the derivative of Error Function in Backward Propagation Step:

$$\text{Update Rule} \rightarrow W_i \leftarrow W_i + 2\lambda x_i (y - \mathbf{W}^T x)$$

4.2 b

Using the Neural Network Functionality:

$$\begin{aligned} y &\approx \sum_j W_j \sum_k w_{k,j} x_k \\ &= \sum_k \left(\sum_j W_j w_{k,j} \right) x_k \\ &= \sum_k \beta_k x_k \end{aligned}$$

Or we can write the above equation in matrix multiplication form:

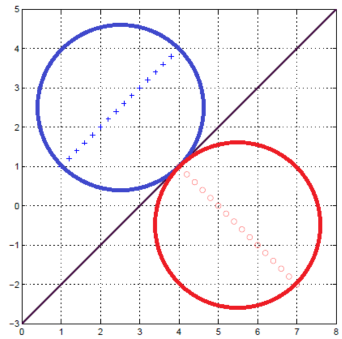
$$\mathbf{W}^T \mathbf{w}^T x = (\mathbf{W}^{'})^T x \quad \textbf{where } \mathbf{W}^{'} = \mathbf{W}^T \mathbf{w}^T$$

These equation prove that a network of linear neurons with one hidden layer of m units, n input units, and one output unit, can act the same as a single-layer linear network with no hidden units.

5 Question 5

5.1 toydata1

In **toydata1**, GNB learns diagonal covariance matrices yielding axis-aligned Gaussians. In our figure, two circles are learned by GNB.

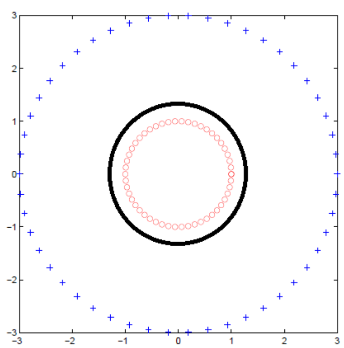


A

Figure 1: toydata 1

5.2 toydata2

In **toydata2**, GNB learns two Gaussians, one for the smaller circle (low-variance data) and one for the larger circle (high-variance data). The decision boundary is shown in the figure below.



B

Figure 2: toydata 2