ECE 479 ICC: IoT and Cognitive Computing

Spring 2023, Homework 2

Please use LaTeX or Word compiled pdf to submit your answers.

Question 1: k-Nearest Neighbor (20 pts)

1. **kNN Concepts**

   For each of the statements below, tell whether it is True or False. Use one sentence to briefly explain your choice.

   (a) kNN can only be used for regression but not classification.

   (b) kNN is a parametric model because it uses a hyperparameter k.

   (c) It is always better to choose a larger k value.

   (d) Training a kNN takes almost no computation; most of the computation is done at test time.

   (e) kNN will not suffer from the curse of dimensionality.

   (f) Smaller k value will lead to underfitting, rather than overfitting.

2. **kNN Calculation**

   As shown in Figure 1, we have seven data points in the training dataset. Class 0 is marked with blue dots, and class 1 is marked with orange dots. The corresponding data points are:

   - Class 0: [2, 4.5], [2.5, 3.5], [3, 6]
   - Class 1: [3, 5], [3.5, 3], [3.5, 5.5], [4, 4]

   We have two samples to classify. The sample points are marked with green crosses. They are:

   - Samples: [2.75, 4.5], [2.75, 5.5]

   Please answer the following questions.

   (a) Use the Pythagorean theorem to compute two-dimensional Euclidean distance ($dist(p, q) = \sqrt{(p - q)^2}$) between each sample point and the other data points. You only need to keep three decimal places. Fill the pair-wise distances values in Table 1.

   (b) Using the information in Table 1, classify the two sample points with

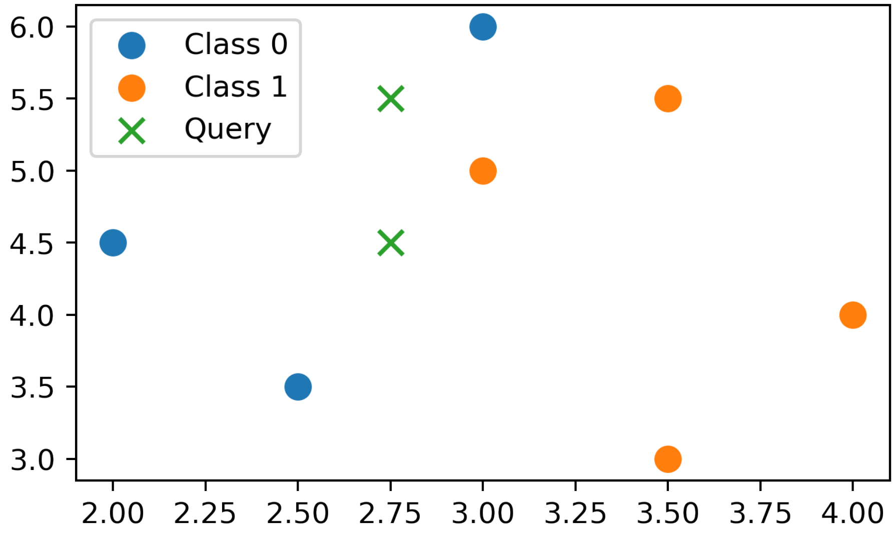


Figure 1: kNN Plot

Label	Data	[2.75, 4.5]	[2.75, 5.5]
0	[2, 4.5]		
0	[2.5, 3.5]		
0	[3, 6]		
1	[3, 5]		
1	[3.5, 3]		
1	[3.5, 5.5]		
1	[4, 4]		

Table 1: kNN Distances

$k = 1, 3, 5$ and fill in Table 2. Since only two classes are in this question, you only need to fill in 0 or 1.

Note: When there is a tie in kNN, you can randomly select one of the tied neighbors.

$k =$	[2.75, 4.5]	[2.75, 5.5]
1		
3		
5		

Table 2: kNN Classification

(c) Did you observe any inconsistencies in the classification results when you use different values of k? Briefly describe your observations and provide a reasonable explanation.

Question 2: K-Means Clustering (20 pts)

1. **Clustering concepts**

   For each of the statements below, tell whether it is True or False. Use one sentence to briefly explain your choice.

   (a) Clustering does not require prior labeling; thus, it is a kind of unsupervised learning.

   (b) The goal of all clustering algorithms is to minimize total SSE (sum of squared error) within each cluster.

   (c) It is impossible to define a similarity metric for categorical data in order to perform clustering.

   (d) "K-means algorithm" is the same concept as the "clustering algorithm."

   (e) "K" is usually given or predefined for a "K-means" clustering problem.

2. **Clustering applications**

   Please list at least **three real applications** of the clustering algorithm. Each of them should include

   (a) a brief description of the application

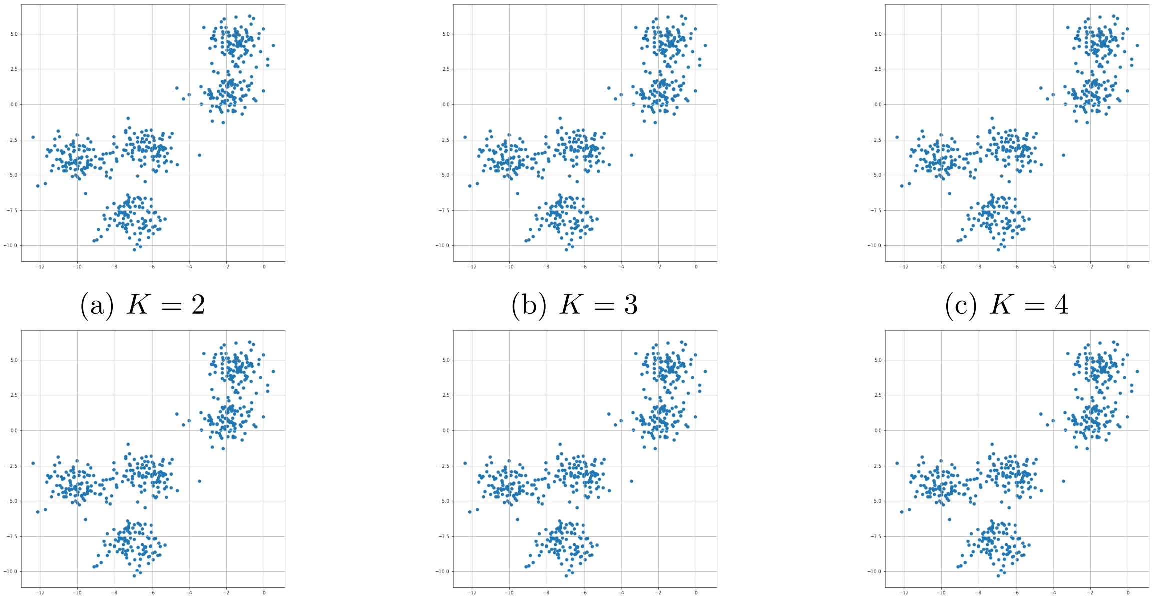
   (b) the type of clustering algorithm it uses

   (c) a reference to support your statement (URL link or title of the articles, etc.)

3. **Visualizing a K-means problem**

   This part of the question aims to help you build an intuitive sense of how the "K-means algorithm" works. The data points in this question are randomly generated with a pre-defined number of centroids.

   In Fig. 2, try to circle out the clusters according to the given K for each of them, and then comment on the results. Give your best guess on what should be the optimal K, and explain the reason using the "SSE vs K" diagram. (For example: if "K=1", you should circle all data points to be the same cluster since there is only 1 cluster.)



(a) $K = 2$ (b) $K = 3$ (c) $K = 4$

(d) $K = 5$ (e) $K = 6$ (f) $K = 7$

Figure 2: Potential clustering result for various K values.

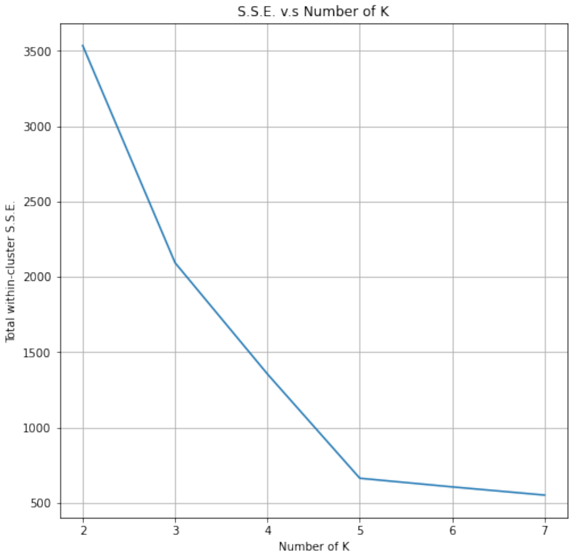


Figure 3: SSE v.s Number of K

4. **Calculating and minimizing SSE**

Convergence is a common challenge in clustering problems. One of the advantages of K-means clustering is that it is guaranteed to converge after a certain number of iterations since the centroid for minimizing the within-cluster SSE is deterministic. A brief proof can be found here, credit to IITB. In this question, calculate the

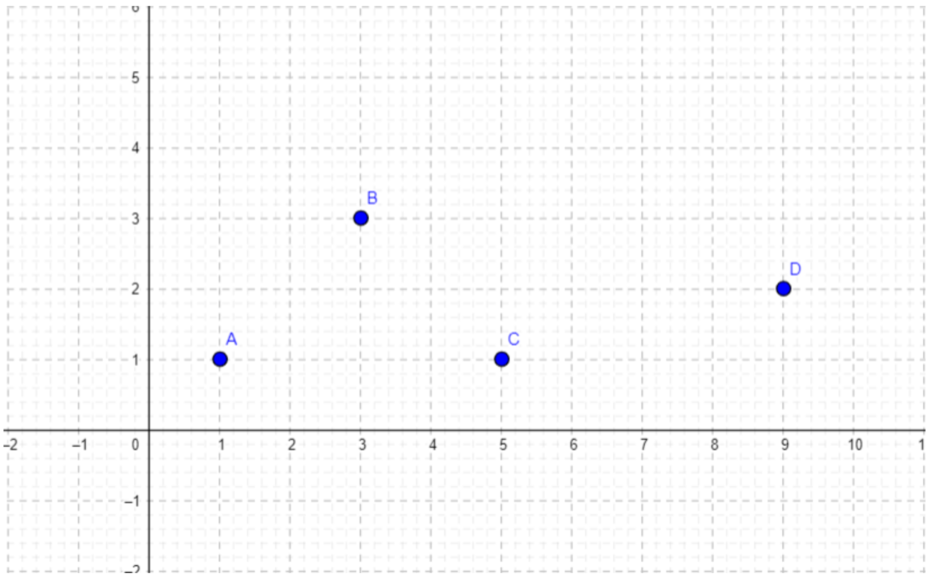


Figure 4: Points Within a Cluster

within-cluster SSE with the given information. You should first find out the centroid that minimizes the SSE (We are using Euclidean distance).

$$A(1,1)$$
$$B(3,3)$$
$$C(5,1)$$
$$D(9,2)$$

The detection of outliers (points that should be put into another cluster) is another challenge in clustering problems, and a naive way to identify the outlier is to remove one of the points and then observe the change of within-cluster SSE. Repeat this step until the outlier that leads to the largest drop in SSE can be confirmed.

Assume that we know there is an outlier in the given cluster, identify it, and calculate the new minimum within-cluster SSE. Then, explain why you choose this point as the outlier.

Question 3: Linear Classifiers, SVM and Mapping Trick (20 pts)

1. **Linear Classifiers**
   Consider two classification problems:

   (a) Data points belonging to label `‘0’`: {(0,0), (0,1)}.
   Data points belonging to label `‘1’`: {(1,0), (1,1)}.

   (b) Data points belonging to label `‘0’`: {(0,0), (1,1)}.
   Data points belonging to label `‘1’`: {(1,0), (0,1)}.

   Can linear classifiers learn each pattern in these two classification problems? Justify your answer.

2. **1-D Linear Classification**
   Sometimes, when a set of data is not separable in lower dimensions, we can transform and map the data to a higher dimension so that it becomes separable. To demonstrate "the mapping trick", we first consider a 1D classification problem. Suppose that we have the following points to be classified:

   Class A: (-2, -1.8, 0.3, 0.6, 1) Label: -1
   Class B: (-1, -0.7, -0.6, -0.3, 2) Label: 1

   Plot the points on a number axis, and comment on whether they can be perfectly classified with a 1D linear classifier.
   Hint: a linear classifier in 1D can be seen as assigning different labels to $x > b$ and $x < b$ where b is the division (hyperplane).

3. **The Mapping Trick**
   Now, transform the points into a 2D space using the following mapping function:

   $$\psi(x) = (x, x^3 - 2x)$$

   Again, plot the points in the 2D space and comment on whether they can be classified with a simple linear classifier and explain why or why not.

4. **SVM and hyperplane**
   Unlike the general case where we need to apply convex programming techniques to find the optimal solution, in this question, since we can easily identify the points that correspond to the support vectors in your plot, we can directly find the hyperplane using geometric interpretation and observation.

In the plot, circle the points that correspond to the support vectors, then find out the w vector and offset b of the hyperplane, such that $||w||_2$ is minimized.

Hint: Notice that when finding w we should always ensure that the constraint $y_i(w^T x^{(i)} + b) \geq 1$ is satisfied for all i, and $y_i(w^T x^{(i)} + b) = 1$ for the support vectors. And remember that a hyperplane can correspond to all the $\gamma ||w||$ where γ can be any real number. That is, scaling w does not change the hyperplane.

5. **Margin and optimality**
   Recall the SVM margin in Lecture 7 Page 32. Now calculate the margin of the hyperplane you found, then show why the line $y = 0$ is not an optimal hyperplane.

Question 4: Neural Networks & Model Compression (20 pts)

1. **Neural network for edge devices**
   For a classification problem with 16 input features and 16 classes, compare the following two networks:

   (a) a deeper network with nine dense (fully-connected) layers with 32 nodes in each layer

   (b) a shallower network that has three dense (fully-connected) layers with 128 nodes in each layer

   If we use `float32` to store the weights, what are the memory requirements of these two configurations? Which one would you prefer to deploy on an edge device? (Note the input and the output layers are counted. The networks have no bias. 32 bits = 4 bytes)

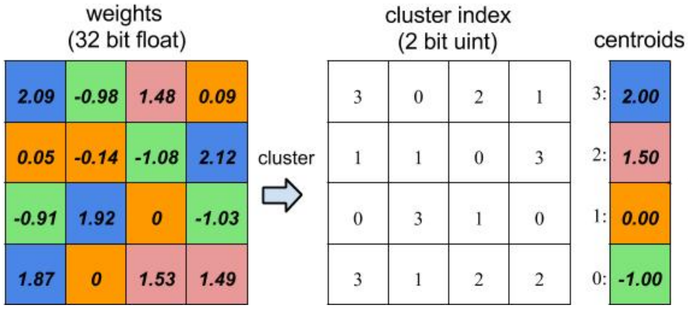
2. **CNN vs MLP**
   You have trained a DNN model for static image classification applications. You deploy the model in the field with a camera. However, you find that the camera's view has shifted in space compared to the camera used for capturing training images. In other words, the area of interest has moved off-center while the objects in your training images are perfectly at the center. What would happen to the classification accuracy if you used (1) a Multi-layer Perceptron or (2) a Convolutional Neural Network as your model? Justify your answer (Hint. Think about Translation Invariance).

3. **Batch normalization**
   Batch normalization layers are used in most of today's state-of-the-art convolutional neural networks. During inference, the mini-batch mean and variance are fixed and independent of the input data. We can *fuse* the batch normalization layer into the convolutional layer. Suppose that we have w_i and b_i to denote the weight and bias in the convolutional layer. Write down the new w_i' and b_i' for the fused convolutional layer with the parameters from batch normalization. Hint: The result of convolution for each pixel $x_i' = w_i * x_i + b_i$.

4. **Weight Quantization**
   Assuming your design needs to fetch N weights from the memory and the average memory access bandwidth is S bytes per second, calculate the data transfer latency (in units of second) when using FLOAT32 to represent each weight. Then, calculate the data transfer latency if we apply quantization to convert the model to INT8.

5. **Weight Quantization vs Weight Clustering**
   Besides quantization, we can also compress the model by clustering the weight

values into M groups. As shown in Fig. 5, we can convert the N weights in the last sub-question to N indices in the cluster index matrix and M centroids. In this example, $N = 4 \times 4 = 16$ and $M = 4$. If we keep using FLOAT32 to represent the centroids and $log2(M)$ bits to represent each index, what is the maximum value of M to gain lower data transfer latency compared to the **quantized** model in the last sub-question? (assuming $N = 64$, M must be a power of 2, and the bandwidth is still S bytes per second)



2.09	-0.98	1.48	0.09
0.05	-0.14	-1.08	2.12
-0.91	1.92	0	-1.03
1.87	0	1.53	1.49

3	0	2	1
1	1	0	3
0	3	1	0
3	1	2	2

3:	2.00
2:	1.50
1:	0.00
0:	-1.00

Figure 5: Example of weight clustering, credit to Oreilly book here.

Question 5: Backpropagation (20 pts)

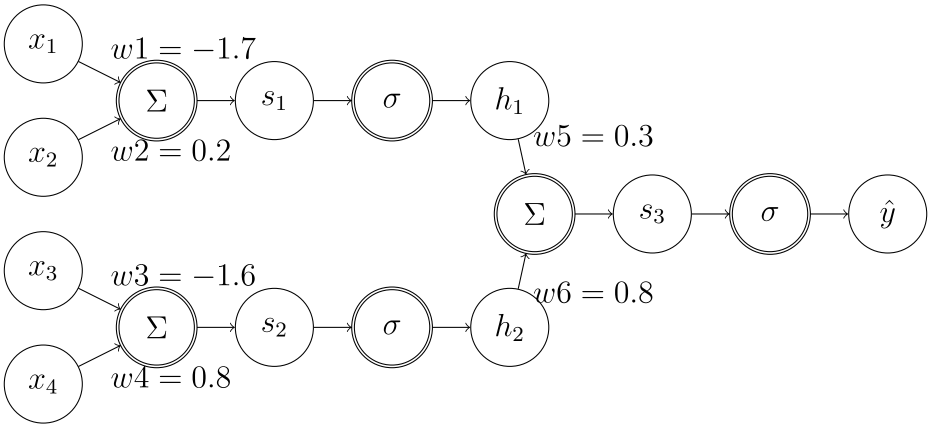


Figure 6: A Simplified Neural Network Example

Consider the neural network in Figure 6. Single-circled nodes denote variables: x_1 is an input variable, h_1 is an intermediate variable, and $\hat{y}$ is an output variable, etc. Double-circled nodes denote functions: Σ takes the sum of its inputs, and σ denotes the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$.

Suppose we have an MSE loss $L(y, \hat{y}) = \|y - \hat{y}\|_2^2$, we are given a data point $(x1, x2, x3, x4) = (0.3, 1.4, 0.9, -0.6)$ with the true label 0.37. The gradient of the MSE loss function is $2\|y - \hat{y}\|$.

1. Use the backpropagation algorithm to compute the partial derivative$\frac{\partial L}{\partial w_1}$, $\frac{\partial L}{\partial w_2}, \frac{\partial L}{\partial w_3}, \frac{\partial L}{\partial w_4}, \frac{\partial L}{\partial w_5}, \frac{\partial L}{\partial w_6}$.
2. Use 3-4 sentences to explain what vanishing gradients are. You can watch this video before answering the question.