VANCENCE NO
1007279
50.007 - ML HW #1
C02

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

50.007 Machine Learning, Summer 2024
Homework 1

Due on 9th June 2024, 11:59 pm

1. Classification Theory [15 points]

Consider data points from a 2-d space where each point is of the form $x = (x_1, x_2)$. You are given a dataset with five positive examples: (0, 0), (-1, 1), (1, -1), (-2, 2) and (2, -2) and three negative examples (2, -3), (-3, -2) and (-3, 2).

(a) [5 points] Suppose that the positive examples are above and on the line, and the negative examples are below the line, define a linear classify by assuming a hypothesis space with a line through the origin with normal vector $\theta = (\theta_1, \theta_2)$ (or $[\theta_1, \theta_2]^T$).

(b) Assume a hypothesis space with an origin-centered circle with radius r (r is the parameter), solve the following:

i. [5 points] Suppose that the positive examples reside inside the circle, and negative examples resdie outside the circle, define a non-linear classifier that can correctly classify all the examples in the dataset.

ii. [5 points] Suppose that two additional data points are added to the dataset: (r, 0) as a positive example and (-r, 0) as a negative example. Can the non-linear classifier you defined above classify all the examples correctly? Why?

2. Classification Implementation [30 points]

Automatic handwritten digit recognition is an important machine learning task. The US Postal Service Zip Code Database (http://www.unitedstateszipcodes.org/zip-code-database/) provides 16 $\times$ 16 pixel images preprocessed from scanned handwritten zip codes (US zip codes are the analogues of Singapore postal codes). The task is to recognize the digit in each image. We shall consider the simpler goal of recognizing only two digits: 1 and 5. To simplify our task even further, let's consider only two features: intensity and symmetry. Digit 5 generally occupies more black pixels and thus have higher average pixel intensity than digit 1. Digit 1 is usually symmetric but digit 5 is not. By defining asymmetry as the average difference between an image and its flipped versions, and symmetry as the negation of asymmetry, we can get higher symmetry values for digit 1.

Write an implementation of the perceptron algorithm. Train it on the training set (`train_1_5.csv`), and evaluate its accuracy on the test set (`test_1_5.csv`). The dataset is provided in "HW1_data/2" folder. The training and test sets are posted on eDimension. csv stands for comma-separated values. In the files, each row is an example. The first value is the symmetry, the second is the average intensity, and the third is the label.

Note: please do NOT shuffle the data. Visit the instances sequentially in the training set when running the perceptron algorithm.

(a) [10 points] Run the perceptron algorithm with offset on the training data for 1 epoch (i.e., traversing the training set 1 time), report the θ, offset and accuracy on the test set.

(b) [10 points] Run the perceptron algorithm with offset on the training data for 5 epochs, report the θ, offset and accuracy on the test set.

(c) [5 points] Plot the test data on a 2-D scatter plot, clearly label the positive samples and negative samples in the legend. Then, overlay the 2 decision boundaries (i.e., the lines) you obtained from (a) and (b) on the same scatter plot. Label the first line as "Line-1-epoch" and the second line as "Line-5-epochs" also in the legend.

(d) [5 points] Is it possible to classify all samples correctly? That is, is it possible to achieve 100% accuracy? Why?

3. Regression Theory [20 points]

Given the following dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$, where $\mathbf{x} \in \mathbb{R}^2$ is a 2-D input data, and y is the corresponding output in the continuous space. Consider a regression model with parameter $\Theta = (\theta_1^2, \theta_2^2)$ that will be used to predict the output y.

(a) [1 point] What is the size of the dataset? (i.e., how many data in this dataset $\mathcal{D}$)

(b) [2 points] Write down the linear equation of the above regression model which has model parameter $\Theta = (\theta_1^2, \theta_2^2)$.

(c) [2 points] Write down the objective (error) function used for optimization of the above linear regression model.

(d) [10 points] Derive the optimal value for θ_1 using the least squares loss for the above regression model. [Hint: θ_1 is a function of x_1, x_2, y and θ_2].

(e) [2 points] Suppose that you want to modify the regression model with Ridge penalty. Write down the objective (error) function of the ridge regression model.

(f) [3 points] Based on your answer in (b) and (e), please explain the different between linear regression and ridge regression.

4. Ridge regression Implementation [15 points]

In this problem, we will explore the effects of ridge regression on generalization. We will use `hw1_ridge_x.dat` as the inputs and `hw1_ridge_y.dat` as the desired output. Please note that a column vector of 1s is already added to the inputs. Recall from Lecture Notes 4, the optimal weight for ridge regression is given by

$$\hat{\theta} = (n\lambda I + X^T X)^{-1} X^T Y \tag{1}$$

To find a suitable value for λ, we will set aside a small subset of the provided data set for estimating the test loss. This subset is called *validation set*, which we use to compute *validation loss*. The remainder of the data will be called the *training set*. Let the first 40 entries of the data set be the training set, and the last 10 entries be the validation set. Concatenate their features into matrices vX and tX, and their responses into vectors vY and tY.

(a) [10 points] Write a function *ridge_regression*(tX, tY, l) that takes the training features, training responses and regularizing parameter λ, and outputs the exact solution θ for ridge regression. Report the resulting value of θ for $\lambda = 0.15$.

(b) [5 points] Use the sample code snippet in `hw1q4_plot_samplecode.py` to plot graphs of the validation loss and training loss as λ varies on logarithmic scale from $\lambda = 10^{-5}$ to $\lambda = 10^{0}$. Write down the value of λ that minimizes the validation loss.

5. Clustering Theory [20 points]

In clustering, Euclidean distance is not the only way to measure the distance between two points/vectors. l_p norms is a family of distance measures that are parameterized by $p \geq 1$. The l_p norm of a vector is:

$$\|x\|_p = \left(\sum_j {x_j}^p \right)^{\frac{1}{p}}.$$

Euclidean distance is the l_2 norm of the vector difference between two points, i.e.,

$$\|x - y\|_2 = \left(\sum_j x_j - {y_j}^2 \right)^{\frac{1}{2}}.$$

The Manhattan distance is the l_1 norm of the vector difference between two points, i.e.,

$$\|x - y\|_1 = \sum_j x_j - y_j.$$

The l_∞ distance is the maximum absolute element in the vector difference between two points, i.e.,

$$\|x - y\|_\infty = \max_j x_j - y_j.$$

Consider a set of points $X = (0.6, 0.8), (0.8, 0.6), (-0.8, 0.6)$. Compute the value of z that minimizes $\sum_{x \in X} d(x, z)$ when $d(x, z)$ is defined as follows respectively:

(a) [5 points] the Euclidean distance between x and z,

(b) [5 points] the squared Euclidean distance between x and z,

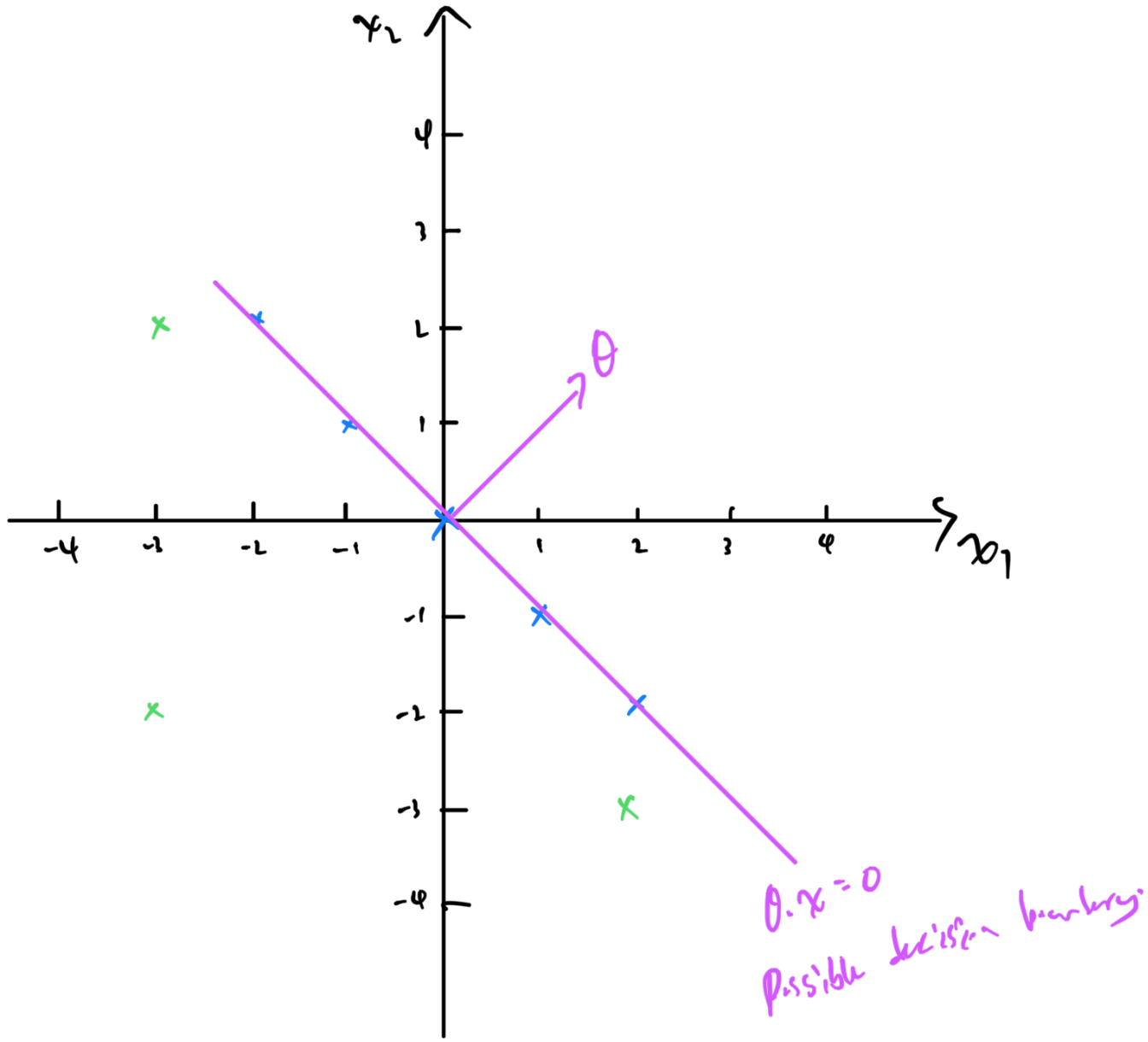
(c) [5 points] the Manhattan distance between x and z.

(d) [5 points] K-Means clustering creates cluster centroids that do not correspond to any real data points, whereas k-Medoids selects real data points as cluster centers. What are the advantages and disadvantages of K-Medoids compare to K-Means?

Question 1: Classification Theory

Given a dataset with 5 positive examples: $(0,0)$, $(-1,1)$, $(1,-1)$, $(-2,2)$, $(2,-2)$.
and 3 negative examples: $(2,-3)$, $(-3,-2)$, $(-3,2)$

(a) Define a linear classifier by assuming a hypothesis space with a line that passes through the origin with normal vector $\theta = (\theta_1, \theta_2)$ or $[\theta_1, \theta_2]^T$



Linear classifier through origin:

$$h(x;\theta) = \text{sign}(\theta_1 x_1 + \theta_2 x_2) = \text{sign}(\theta \cdot x) = \begin{cases} +1, & \theta \cdot x \geq 0 \\ -1, & \theta \cdot x < 0 \end{cases}$$

Training error:

$$\varepsilon(\theta) = \frac{1}{n} \sum_{t=1}^{n} \left[y^{(t)} \neq h(x^{(t)};\theta) \right] = \frac{1}{n} \sum_{t=1}^{n} \left[y^{(t)} (\theta \cdot x^{(t)}) \leq 0 \right]$$

Using the perceptron update rule to find an appropriate θ value:

$x^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $y^{(1)} = +1$

$x^{(2)} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, $y^{(2)} = +1$

$x^{(3)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $y^{(3)} = +1$

$x^{(4)} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$, $y^{(4)} = +1$

$x^{(5)} = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$, $y^{(5)} = +1$

$x^{(6)} = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$, $y^{(6)} = -1$

$x^{(7)} = \begin{bmatrix} -3 \\ -2 \end{bmatrix}$, $y^{(7)} = -1$

$x^{(8)} = \begin{bmatrix} -3 \\ 2 \end{bmatrix}$, $y^{(8)} = -1$

If $y^{(t)}(\theta^{(t)} \cdot x^{(t)}) \leq 0$ then

$$\theta^{(k+1)} = \theta^{(k)} + y^{(t)} x^{(t)}$$

Starting with $\theta^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$:

$$y^{(1)}(\theta^{(0)} \cdot x^{(1)}) = (+1)\left[(0 \times 0) + (0 \times 0)\right] = 0 \leq 0$$

update $\theta^{(0)}$:

$$\theta^{(1)} = \theta^{(0)} + y^{(1)} x^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0$$

⋮

the θ value will always be 0 no matter how much we update it since the only possible decision boundary is the line that pass through all positive points & the origin to satisfy a linear classifier.

$\therefore$ checking by plucking positive & negative pts when $\theta = 0$:

checking with positive point $x^{(5)}$:

$$h(x;\theta) = \text{sign}(x_1 + x_2) = \text{sign}(x)$$
$$= \text{sign}(2 + (-2)) = \text{sign}(0) \geq 0$$
$$= +1 \quad \checkmark$$

checking with negative point $x^{(8)}$:

$$h(\theta;x) = \text{sign}(x_1 + x_2) = \text{sign}(x)$$
$$= \text{sign}(-3 + 2) = \text{sign}(-1) < 0$$
$$= -1 \quad \checkmark$$

$\therefore$ Possible Linear classify would be $\theta_1 = 1$ & $\theta_2 = 1$:

$$h(x) = \text{sign}(x_1 + x_2) = \text{sign}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

(b) Assume a hypothesis space with an origin-centered circle with radius r.

(c) Suppose that positive examples reside inside circle & negative examples reside outside circle, define a non-linear classifier that can correctly classify all examples in dataset

using the equation of a circle, we can apply it to this scenario:

$$x_1^2 + x_2^2 = r^2$$

For a point to be classified positive it has to be within the circle, $\therefore x_1^2 + x_2^2 \leq r^2$ & for it to be negative it falls outside the circle, $\therefore x_1^2 + x_2^2 > r^2$.

Applying it to positive data points:

$x^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0^2 + 0^2 = 0$

$x^{(2)} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} = (-1)^2 + (1)^2 = 1$

$x^{(3)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} = (1^2 + (-1)^2) = 1$

$x^{(4)} = \begin{bmatrix} -2 \\ 2 \end{bmatrix} = ((-2)^2 + 2^2) = 8$

$x^{(5)} = \begin{bmatrix} 2 \\ -2 \end{bmatrix} = (2^2 + (-2)^2) = 8$

Applying to negative data points:

$x^{(6)} = \begin{bmatrix} 2 \\ -3 \end{bmatrix} = (2^2 + (-3)^2) = 13$

$x^{(7)} = \begin{bmatrix} -3 \\ -2 \end{bmatrix} = ((-3)^2 + (-2)^2) = 13$

$x^{(8)} = \begin{bmatrix} -3 \\ 2 \end{bmatrix} = ((-3)^2 + 2^2) = 13$

Maximum squared distance = 8

Minimum squared distance = 13

$\therefore 8 < r^2 < 13$

$\sqrt{8} < r < \sqrt{13}$

$2.828 < r < 3.606$ (3.s.f)

r can fall in this range for t. satisfy the classification,

letting $r = 3$ to satisfy the equation:

$x_1^2 + x_2^2 \leq r^2$ to be positive 4

$x_1^2 + x_2^2 > r^2$ to be negative

$\therefore$ possible non-linear classifier:

$$h(x) = x_1^2 + x_2^2 = \begin{cases} +1, & h(x) \leq 9 \\ -1, & h(x) > 9 \end{cases}$$

(ii) Suppose that 2 additional data points are added to dataset:

(r, 0) as positive 4 (-r, 0) as negative, can the above classify correctly classify? why?

Checking with positive data point:

$$x^{(9)} = \begin{bmatrix} r \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$h(x^{(9)}) = x_1^2 + x_2^2$$
$$= 3^2 + 0^2$$
$$= 9 \leq 1$$
$$= +1$$

Checking with negative data point:

$$x^{(10)} = \begin{bmatrix} -r \\ 0 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$$

$$h(x^{(10)}) = x_1^2 + x_2^2$$
$$= (-3)^2 + 0^2$$
$$= 9 \leq 9$$
$$= +1$$

No. The non-linear classify would not be able to classify all examples correctly because it lies on the decision boundary of the circle, however we can adjust the decision boundary of the classifier to fit the the requirements since the given range is $2.828 < r < 3.46$, by lowering the value of r, we would be able to classify all examples.

Question 3: Regression Theory

$D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$, where $x \in \mathbb{R}^2$ is a 2D input data, and y is the corresponding output in the continuous space. Consider a regression model with a parameter $\theta = (\theta_1^2, \theta_2^2)$ that will be used to predict output y.

(a) Size of dataset = m

(b) Linear Regression Equation of regression model which has model parameter $\theta = (\theta_1^2, \theta_2^2)$.

$$\text{Linear Equation} = \theta_0 x = \theta_1 x_1 + \dots + \theta_1 x_1 + \theta_0 = \theta^T x + \theta_0$$

$$= \theta_1^2 x_1 + \theta_2^2 x_2 + \theta_0$$

(c) The objective function is derived from the Least Square loss.

Let objective function be $J_m(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(y^{(i)} - (\theta_1^2 \cdot x_1^{(i)} + \theta_2^2 \cdot x_2^{(i)}) \right)^2$

$$= \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - (\theta_1^2 \cdot x_1^{(i)} + \theta_2^2 \cdot x_2^{(i)}) \right)^2$$

(d) Using the least squared loss for the regression model, derive the optimal value of θ_1.

To find optimal value of θ_1:

$$\frac{\partial J(\theta)}{\partial \theta_1} = \frac{\partial}{\partial \theta_1} \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - (\theta_1^2 \cdot x_1^{(i)} + \theta_2^2 \cdot x_2^{(i)}) \right)^2$$

$$= \frac{2}{2m} \sum_{i=1}^{m} \left(y^{(i)} - (\theta_1^2 \cdot x_1^{(i)} + \theta_2^2 \cdot x_2^{(i)}) \right) \cdot \left(-2\theta_1 \cdot x_1^{(i)} \right)$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} - (\theta_1^2 \cdot x_1^{(i)} + \theta_2^2 \cdot x_2^{(i)}) \right) \left(-2\theta_1 x_1^{(i)} \right)$$

[Cont. next pg.]

$$= -\frac{2}{m}\sum_{i=1}^{m}\theta_1 x_1^{(i)}\left(y^{(i)} - \theta_1^2 x_1^{(i)} - \theta_2^2 x_2^{(i)}\right)$$

$$= -\frac{2}{m}\sum_{i=1}^{m}\theta_1\left[y^{(i)}x_1^{(i)} - \theta_1^2\left(x_1^{(i)}\right)^2 - \theta_2^2 x_1^{(i)}x_2^{(i)}\right]$$

Set $\frac{\partial J(\theta)}{\partial \theta_1} = 0$ & solve for θ_1:

$$-\frac{2}{m}\sum_{i=1}^{m}\theta_1\left[y^{(i)}x_1^{(i)} - \theta_1^2\left(x_1^{(i)}\right)^2 - \theta_2^2 x_1^{(i)}x_2^{(i)}\right] = 0$$

$$-\frac{2}{m}\theta_1\sum_{i=1}^{m}\left[y^{(i)}x_1^{(i)} - \theta_1^2\left(x_1^{(i)}\right)^2 - \theta_2^2 x_1^{(i)}x_2^{(i)}\right] = 0$$

$$-\theta_1^2\sum_{i=1}^{m}\left(x_1^{(i)}\right)^2 + \sum_{i=1}^{m}\left[y^{(i)}x_1^{(i)} - \theta_2^2 x_1^{(i)}x_2^{(i)}\right] = 0$$

$$\theta_1^2\sum_{i=1}^{m}\left(x_1^{(i)}\right)^2 = \sum_{i=1}^{m}\left(y^{(i)}x_1^{(i)} - \theta_2^2 x_1^{(i)}x_2^{(i)}\right)$$

$$\theta_1^2 = \frac{\sum_{i=1}^{m}\left(y^{(i)}x_1^{(i)} - \theta_2^2 x_1^{(i)}x_2^{(i)}\right)}{\sum_{i=1}^{m}\left(x_1^{(i)}\right)^2} = \frac{\sum_{i=1}^{m}x_1^{(i)}\left(y^{(i)} - \theta_2^2 x_2^{(i)}\right)}{\sum_{i=1}^{m}\left(x_1^{(i)}\right)^2}$$

$$\therefore \theta_1 = \sqrt{\frac{\sum_{i=1}^{m}\left(y^{(i)} - \theta_2^2 x_2^{(i)}\right)}{\sum_{i=1}^{m}x_1^{(i)}}}$$

(e) Suppose we change the regression model with Ridge penalty.

Let objective function of ridge regression model defined $J_{m,\lambda}(\theta)$, s.t. $\lambda \geq 0$ is the regularization parameter & $\theta = (\theta_1^2, \theta_2^2)$:

$$J_{m,\lambda}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - \theta_1^2 x_1^{(i)} - \theta_2^2 x_2^{(i)} \right)^2 + \frac{\lambda}{2} \| (\theta_1^2, \theta_2^2) \|^2$$

(f) Linear Regression minimises error based on the difference between the calculated predictor and its actual labels using the least square loss function. It penalises more heavily if the difference between actual values and predictor is huge and lesser if the difference is small.

Ridge Regression, however, has an additional component labelled as the regularization function $\left(\frac{\lambda}{2} \|\theta\|^2 \right)$ to control the complexity of the model. It ensures that the model focus is to fit the data when $\lambda = 0$ and to simplify the model when $\lambda > 0$, preventing overfitting and underfitting, therefore leading to potentially better generalisation on new data.

Question 5: Clustering Theory

Consider set of points $X = (0.6, 0.8), (0.8, 0.6), (-0.8, 0.6)$.

Compute the value of z that minimises $\sum_{x \in X} d(x, z)$ when $d(z, x)$ is defined as followed respectively:

(a) the Euclidean distance between x & z:

Let the set of points in X be x_1, x_2 & x_3 respectively.

Euclidean Distance is the L_2 norm of the vector difference between 2 points, defined as:

$$\|x - z\|_2 = \sqrt{\sum_j |x_j - z_j|^2}$$

Using Python code in folder, I obtained the z-value to be $(0.6, 0.8)$.

In the code, it uses an iterative approach to optimise the minimum $\sum_{x \in X} d(x, z)$ Euclidean distance between x & z.

$\therefore$ Minimizing z for Euclidean Distance = $(0.6, 0.8)$

(b) Squared Euclidean Distance

Squared Euclidean Distance can then be defined as,

$$\|x - z\|_2^2 = \sum_j (x_j - z_j)^2$$

Minimising the sum of squared Euclidean distance between x & z:

$$\frac{\partial}{\partial z} \left(\sum_i |x^{(i)} - z|^2 \right) = 0$$

$$\frac{\partial}{\partial z} \sum_i \left[(x_1^{(i)} - z_1)^2 + (x_2^{(i)} - z_2)^2 \right] = 0$$

$$\frac{\partial}{\partial z} = \sum_i \left[\left((x_1^{(i)})^2 - 2x_1^{(i)} z_1 + z_1^2 \right) + \left((x_2^{(i)})^2 - 2x_2^{(i)} z_2 + z_2^2 \right) \right] = 0$$

$$\frac{\partial}{\partial z_1} = \sum_i (-2x_1^{(i)} + 2z_1) = 0 \quad \Big| \quad \frac{\partial}{\partial z_2} = \sum_i (-2x_2^{(i)} + 2z_2) = 0$$

$$= 2 \sum_i (-x_1^{(i)} + z_1) = 0 \quad \Big| \quad = 2 \sum_i (-x_1^{(i)} + z_2) = 0$$

$$= i z_1 - \sum x_1^{(i)} = 0 \quad \Big| \quad = i z_2 - \sum_i x_2^{(i)} = 0$$

$$z_1 = \frac{0.6 + 0.8 - 0.8}{3} \quad \Big| \quad z_2 = \frac{0.8 + 0.6 + 0.6}{3}$$

$$= \frac{1}{5} \quad \Big| \quad = \frac{2}{3}$$

∴ Minimizing z for Squared Euclidean = $\left(\frac{1}{5}, \frac{2}{3} \right)$

(c) Manhattan Distance is defined as the l_1 norm of the vector difference between two points,

$$\|x - z\|_1 = \sum_j |x_j - z_j|$$

The value of z that minimises the sum of Manhattan Distances to the points in X is the component-wise median of the points in X.

For X-coordinates:

Median of $\{0.6, 0.8, -0.8\} = 0.6$

For Y-coordinates:

Median of $\{0.8, 0.6, 0.6\} = 0.6$

$\therefore$ Minimizing z for Manhattan Distance $= (0.6, 0.6)$

(b) Advantages

1. K-Medoids is less sensitive to outliers and noise because it selects actual data points as centers to classify the clusters rather than calculating a mean, making it heavily influenced to extreme values or anomaly.

2. Since K-Medoids use actual data points as medoids, the resulting cluster can be more interpretable and meaningful in certain contexts.

3. K-Medoids are able to work with any distance metrics (non-Euclidean), making it more flexible for different types of data. It is able to use distance metrics such as Manhattan distance or Cosine Similarity, which can be more appropriate to certain data types or applications.

Disadvantages

1. It is generally more computationally expensive to use k-Medroids than k-Means especially for large datasets because the process of finding the medroid involves computing the pairwise dissimilarities between each cluster which can be computionally intensive.
2. Due to point #1, since k-Medoids is computationally expensive, it does not scale as well compared to k-Means for large datasets. An example would be in big data applications where speed and scalability are crucial, k-Means would be more preferred.
3. Optimization process of k-Medoids are less efficient because it requires more iterations for it to converge and it is prone to getting stuck at a local minima, therefore leading back to point #1.

Assignment Reading & Discussion done with:

- David, MLO2
- Jing Huan, MLO2
- Stack Overflow for Q5: Geometric Median, Centroid Mean & Component Wise Median