Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Divyaksh Shukla
Roll Number: 231110603
Date: November 15, 2023

QUESTION 1

- This is solved by taking the cluster which has its mean closest to the test point x_n

$$\arg\min_k ||x_n - \mu_k||^2$$

- If we assume the test point to be closest to a cluster mean, denoted by μ_k then we can get the update equation for μ_k by taking derivative of $\mathcal{L}$ w.r.t. μ_k

$$\frac{\partial \mathcal{L}}{\partial \mu_k} = -2||x_n - \mu_k||$$

  Which can be put into the update equation as

$$\mu_k = \mu_k + \eta||x_n - \mu_k|| \tag{1}$$

- In 1 we have taken all constants to be part of the step-size η. A good choice of η would be a small value that decreases monotonically as the steps progress. By taking a small step size the cluster means will slowly progress towards the expected means and remain unaffected by noisy input datapoints.

Student Name: Divyaksh Shukla
Roll Number: 231110603
Date: November 15, 2023

QUESTION 2

As $z_n \in \mathbb{R}$ is a linear transformation of $\mathbf{x}_n$ and $\mathbf{w}$ we can write:

$$z_n = \mathbf{w}^T \mathbf{x}_n$$

We need to minimise distance between each projected point z_n and its cluster mean, say $\{\mu_-, \mu_+\}$ and maximize distance between the projected means.

$$\max |\mu_- - \mu_+| \tag{2}$$

$$\min \sum_{z_n : y_n = +1} |z_n - \mu_+| + \sum_{z_n : y_n = -1} |z_n - \mu_-| \tag{3}$$

Thus the objective function is:

$$J = \max \left[|\mu_- - \mu_+| - \sum_{z_n : y_n = +1} |z_n - \mu_+| - \sum_{z_n : y_n = -1} |z_n - \mu_-| \right] \tag{4}$$

as all values are in $\mathbb{R}$ taking only simple absolute distance suffices.

Student Name: Divyaksh Shukla
Roll Number: 231110603
Date: November 15, 2023

QUESTION 3

Let us take $\mathbf{S}' = \frac{1}{N}\mathbf{X}\mathbf{X}^T$. 5 represents the equation to calculate eigenvalue λ' and eigenvector $\mathbf{v}$ of $\mathbf{S}'$

$$\mathbf{S}'\mathbf{v} = \lambda'\mathbf{v} \tag{5}$$

Now if we take the value of $\mathbf{S}'$ and pre-multiply with $\mathbf{X}^T$ and readjust the values we get:

$$\frac{1}{N}\mathbf{X}\mathbf{X}^T\mathbf{v} = \lambda'\mathbf{v} \tag{6}$$

$$\frac{1}{N}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{v} = \lambda'\mathbf{X}^T\mathbf{v} \tag{7}$$

$$\mathbf{S}\mathbf{u} = \lambda'\mathbf{u} \tag{8}$$

Thus the eigenvalue remains the same in both forms, only the eigenvectors change, which can be see in blue from Equation 4 and 5. By computing eigenvectors this way we can reduce the complexity of calculating eigenvalues for a $D \times D$ matrix to $N \times N$ matrix, which is feasible in this case as $D < N$.

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Divyaksh Shukla
Roll Number: 231110603
Date: November 15, 2023

QUESTION 4

My solution to problem 4

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Divyaksh Shukla
Roll Number: 231110603
Date: November 15, 2023

QUESTION

5

1 Kernel Ridge Regression

1.1 Regularized Kernel Ridge Regression

I calculated the output for the test-point by using Equation 9.

$$\mathbf{Y}^* = \mathbf{K}^*(\mathbf{K} - \lambda \mathbf{I}_N)^{-1}\mathbf{Y} \tag{9}$$

where $\mathbf{K}*$ and $\mathbf{K}$ are the kernel matrices: $\mathbf{K}^* = k(x*, x_n)$, $\mathbf{K} = k(x_m, x_n)$ and $k(x_m, x_n) = exp\left(-\gamma \, ||x_n - x_m||^2\right)$ is the rbf kernel function. The RMSE value is calculated using:

$$\sqrt{\frac{1}{N}||y_{true} - y_{pred}||^2}$$

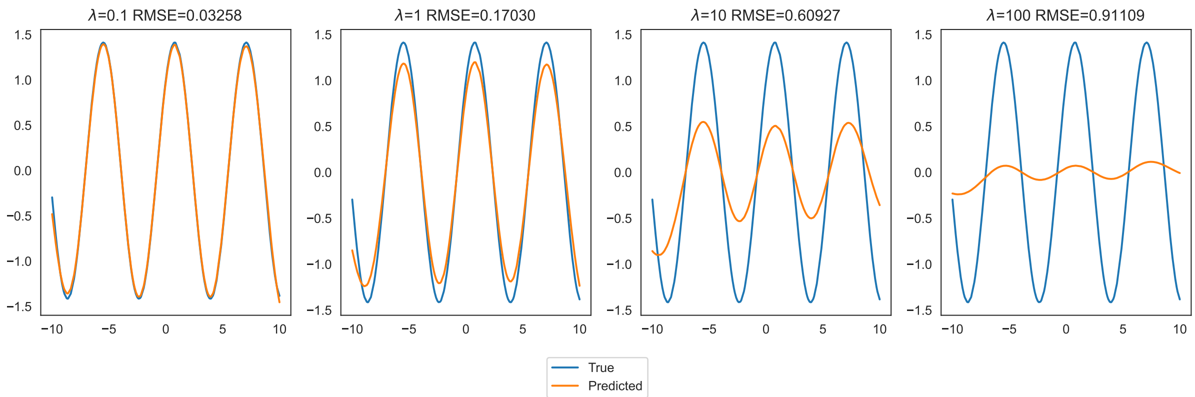

Figure 1: Kernel ridge regression with different regularizations

From Figure 1 we can see that as we increase the regularization hyperparameter λ from 0.1 to 100 we see that the predictions are not able to accurately predict the noiseless y-values. Adding regularization makes the model resistant to noise and improves test-accuray, but in this case the y-values in test data are noiseless so it is better to overfit the model to reduce RMSE value.

1.2 Landmark Kernel Ridge Regression

Here I randomly chose 2, 5, 20, 50 and 100 landmark points from train data and ran the same kernel ridge regression code on the test data. From Figure 2 we can see that as the number of Landmark points increases the regression model starts to predict the model in a better way, the RMSE reduces.

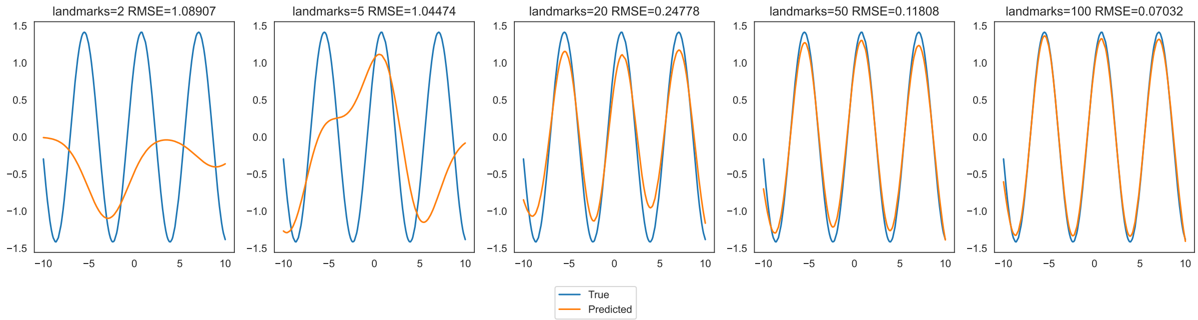

Figure 2: Landmark kernel ridge regression with different number of landmark points

2 K-Means Clustering

In Figure 3 we see that the data is circular and there are 2 intuitive clusters that are possible. So I calculated the radius of each point by applying 10 and 11 and then projecting all the points on 1-dimension to get Figure 4 and applying k-means on the projected data to obtain Figure 5

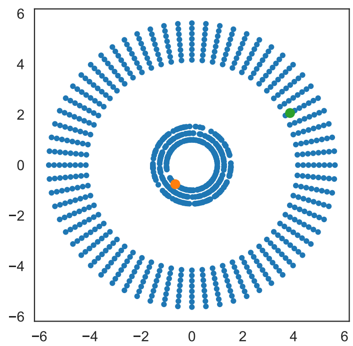

Figure 3: Scatter plot of data with initial centers highlighted

$$\theta = tan^{-1}\left(\frac{y}{x}\right) \tag{10}$$

$$r = \frac{x}{cos(\theta)} \tag{11}$$

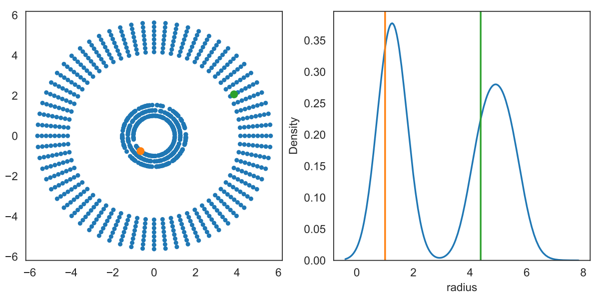

Figure 4: Projecting data points into 1-dimension by taking radius of points from origin. Also highlighting the inital centers

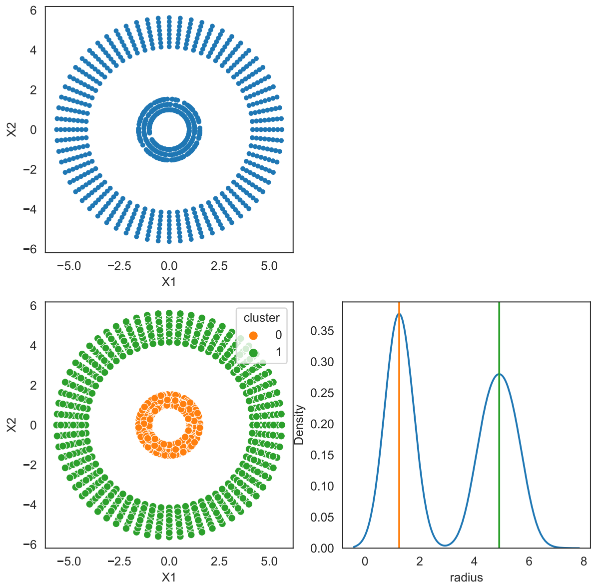

Figure 5: Clustering the data points based on radius and showing the position of the cluster centers on the 1-dimensional plot.

Next I chose 1 landmark point randomly and updated the cluster means accordingly, while using RBF kernel function. Then I ran a k-means prediction algorithm which led to the good and bad clustering. Figure 6 shows an example of the bad clustering that was obtained.

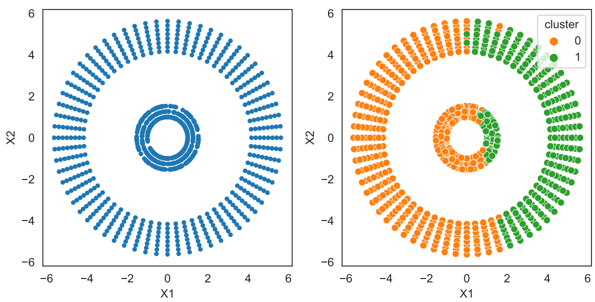

Figure 6: Applying k-means with 1 training landmark point. *This is just one of the obtained clusters*

3 PCA and TSNE

TSNE is able to make clusters which are naturally observable on 2 dimensions in a better way compared to PCA. In PCA, we can see significant overlap and mixing of points from different clusters. Thus, TSNE is a better projection technique to visualize high-dimensional data.

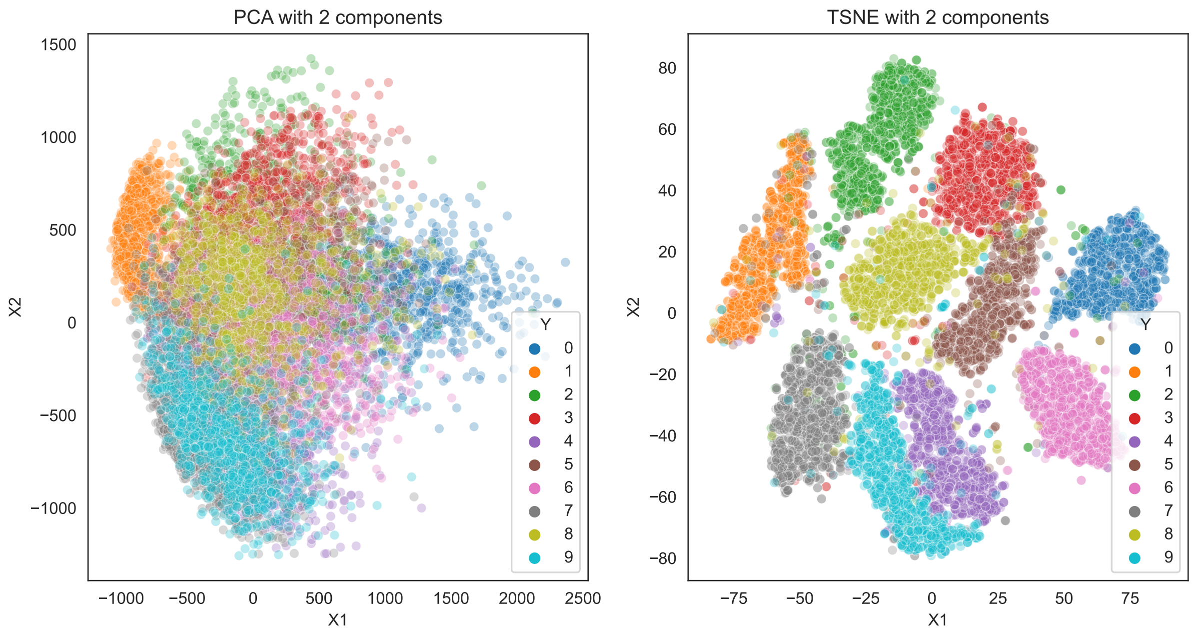

Figure 7: 2-Dimensional PCA and TSNE plots of MNIST