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

Student Name: Soham Amit Bharambe
Roll Number: 210264
Date: November 17, 2023

QUESTION
1

SGD for K-means Objective

Step 1: Assigning x_n "greedily" to the "best" cluster

To solve step 1 in the online K-means algorithm, we need to find the closest cluster center for the current data point x_n. This is done by updating the assignment variable z_{nk} for the data point x_n. The assignment is based on the distance between x_n and each cluster center μ_k, and x_n is assigned to the cluster with the closest center.

$$z_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_i ||x_n - \mu_i||^2 \\ 0 & \text{otherwise} \end{cases}$$

This means that the assignment variable z_{nk} for the chosen cluster k is set to 1, indicating that x_n belongs to cluster k, and all other assignment variables for x_n are set to 0.

Step 2: Updating the cluster means using SGD on the objective $\mathcal{L}$

Let's derive the update equation for the cluster mean μ_k using SGD on the K-means objective function $\mathcal{L}$.

The K-means objective function is given:

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \|x_n - \mu_k\|^2$$

To perform SGD, we update μ_k in the opposite direction of the gradient:

$$\mu_k \leftarrow \mu_k - \alpha \frac{\partial \mathcal{L}}{\partial \mu_k}$$

Taking the gradient of $\mathcal{L}$ with respect to μ_k:

$$\frac{\partial \mathcal{L}}{\partial \mu_k} = \sum_{n=1}^{N} 2 z_{nk} (x_n - \mu_k)$$

Substituting the gradient expression and for a single z_{nk}, we get:

$$\mu_k \leftarrow \mu_k + \alpha z_{nk} (x_n - \mu_k)$$

This is the update equation for μ_k in the online version of the K-means algorithm using SGD. The intuition behind this update is that we move μ_k toward the current data point x_n if x_n is assigned to cluster k ($z_{nk} = 1$). The learning rate α controls the step size of this update.

Choice of the step size:

A step size of $\frac{1}{t}$, where t is the iteration number, can be a good choice. This ensures that the step size decreases over time, allowing the algorithm to converge more smoothly as it gets closer to the optimal solution.

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

Student Name: Soham Amit Bharambe
Roll Number: 210264
Date: November 17, 2023

QUESTION

2

An Ideal Projection

To achieve the goal of projecting inputs into one dimension with a vector $\boldsymbol{w}$ such that the distance between the means of the two classes is maximized, and the variances of the projected classes are minimized, we can formulate the objective function as follows (which can be made negative or inverted for min):

$$\max_{\boldsymbol{w}:\|\boldsymbol{w}\|=1} \frac{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^2}{s_1^2 + s_2^2}$$

Here:

$\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the means of the projected classes.
s_1^2 and s_2^2 are the variances of the projected classes.
$\boldsymbol{w}$ is the direction of projection vector.

Let:

$\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ be the means of the two classes before projection.

Distance between projected means:

$$(\mu_1 - \mu_2)^2 = (\boldsymbol{w}^\top \mathbf{m}_1 - \boldsymbol{w}^\top \mathbf{m}_2)^2$$

$$= (\boldsymbol{w}^\top (\mathbf{m}_1 - \mathbf{m}_2))^2$$

$$= \boldsymbol{w}^\top (\mathbf{m}_1 - \mathbf{m}_2) \cdot (\mathbf{m}_1 - \mathbf{m}_2)^\top \boldsymbol{w}$$

Variance of the projection for class j:

$$s_j^2 = \sum_{\boldsymbol{x}_i \in C_j} (\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \boldsymbol{m}_j)^2$$

$$= \sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{w}^\top (\boldsymbol{x}_i - \boldsymbol{m}_j) \cdot (\boldsymbol{x}_i - \boldsymbol{m}_j)^\top \boldsymbol{w}$$

Objective would be:

$$\frac{\boldsymbol{w}^\top (\mathbf{m}_1 - \mathbf{m}_2) \cdot (\mathbf{m}_1 - \mathbf{m}_2)^\top \boldsymbol{w}}{\boldsymbol{w}^\top \left(\sum_{\boldsymbol{x}_i \in C_1} (\boldsymbol{x}_i - \boldsymbol{m}_1) \cdot (\boldsymbol{x}_i - \boldsymbol{m}_1)^\top + \sum_{\boldsymbol{x}_i \in C_2} (\boldsymbol{x}_i - \boldsymbol{m}_2) \cdot (\boldsymbol{x}_i - \boldsymbol{m}_2)^\top \right) \boldsymbol{w}}$$

This ensures that the means are as far apart as possible, and the variances within each class are as small as possible for the clusters C_1 and C_2.

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

Student Name: Soham Amit Bharambe
Roll Number: 210264
Date: November 17, 2023

QUESTION

3

Eigenchangers!

Given $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top$ with eigenvalue λ, we have:

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{v} = \lambda\boldsymbol{v}$$

Now, let's multiply both sides by $\boldsymbol{X}^\top$:

$$\frac{1}{N}(\boldsymbol{X}^\top\boldsymbol{X})(\boldsymbol{X}^\top\boldsymbol{v}) = \lambda(\boldsymbol{X}^\top\boldsymbol{v})$$

Now, if we define $\boldsymbol{u} = \boldsymbol{X}^\top\boldsymbol{v}$, we get:

$$\frac{1}{N}(\boldsymbol{X}^\top\boldsymbol{X})\boldsymbol{u} = \lambda\boldsymbol{u}$$

This shows that $\boldsymbol{u}$ is an eigenvector of the covariance matrix $S = \frac{1}{N}\boldsymbol{X}^\top\boldsymbol{X}$ with the same eigenvalue λ. The advantage of this approach is that the dimensionality of the matrix involved is smaller (ND instead of D^2), which is computationally more efficient when $D > N$.

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

Student Name: Soham Amit Bharambe
Roll Number: 210264
Date: November 17, 2023

QUESTION

4

Latent Variable Models for Supervised Learning

Part 1

This approach allows the model to capture different patterns in the data by assigning different clusters to different subsets of the input space, providing a more flexible representation compared to the standard probabilistic linear model.

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

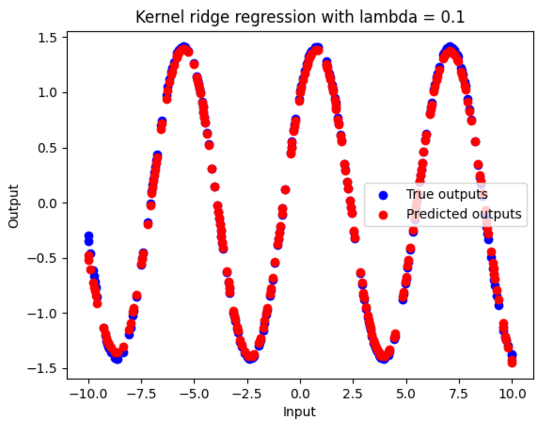
Student Name: Soham Amit Bharambe
Roll Number: 210264
Date: November 17, 2023

QUESTION

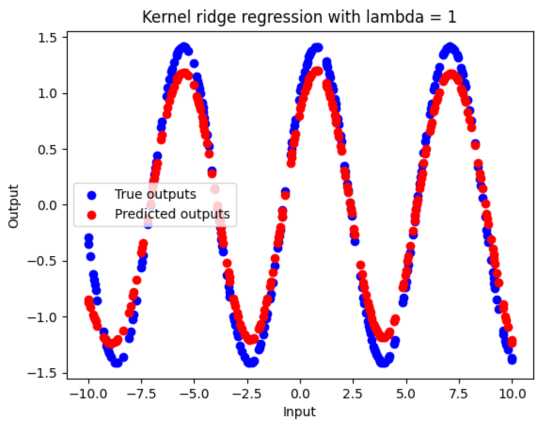
5

Programming Problems

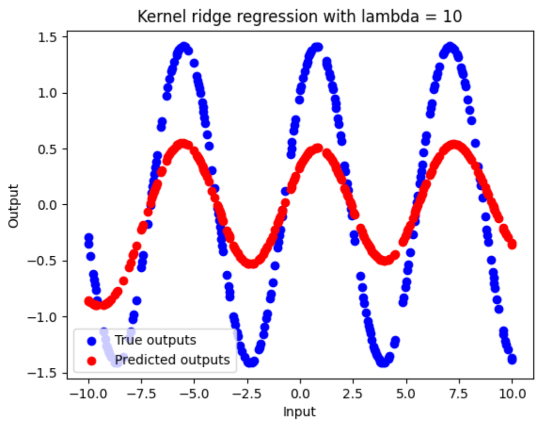
Part 1.1: Kernel Ridge Regression



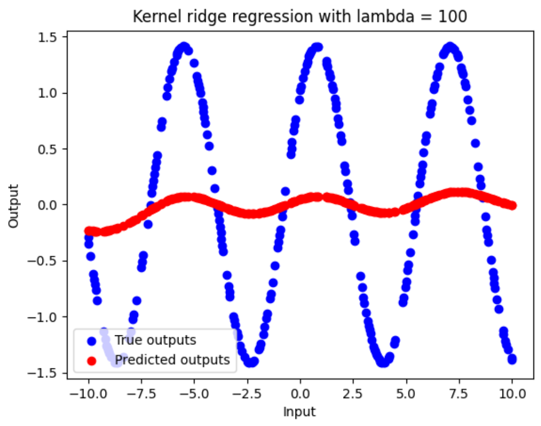
(a) RMSE with lambda = 0.1: 0.0326



(b) RMSE with lambda = 1: 0.1703



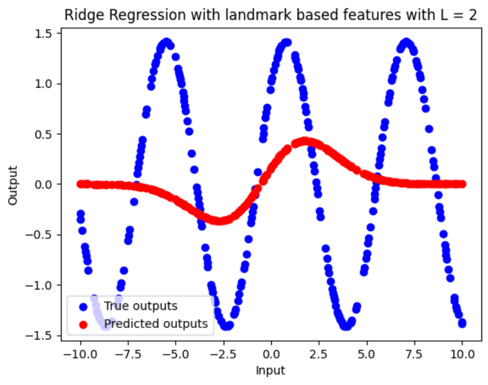
(a) RMSE with lambda = 10: 0.6093



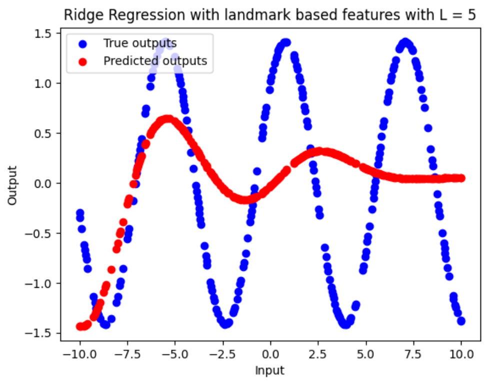
(b) RMSE with lambda = 100: 0.9111

For kernel ridge regression, as the value of lambda increases, the model becomes less sensitive to the training data and more biased towards the prior. As a result, the model tends to underfit the data and produce a smoother curve that does not capture the variations in the data well as can be seen by the increasing RMSE.

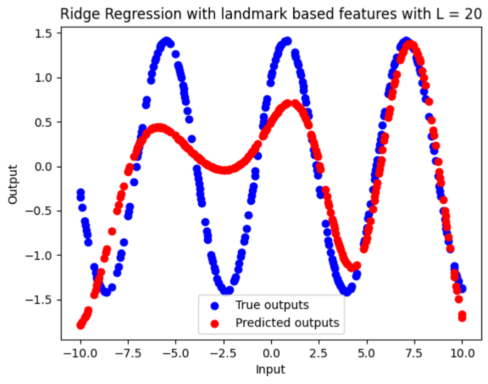
Part 1.2: Landmark Ridge Regression



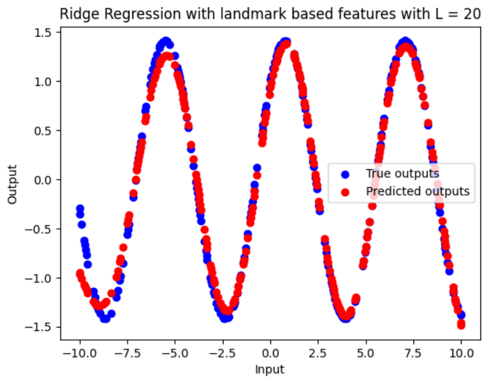
(a) RMSE with L = 2: 0.9558



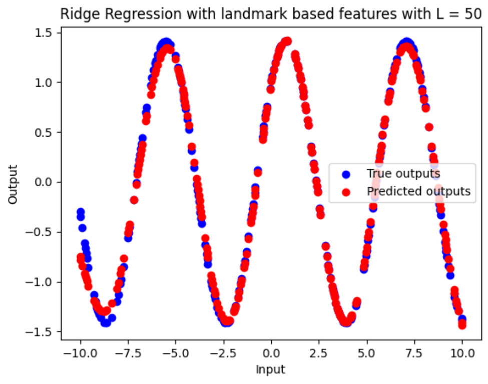
(b) RMSE with L = 5: 0.8997



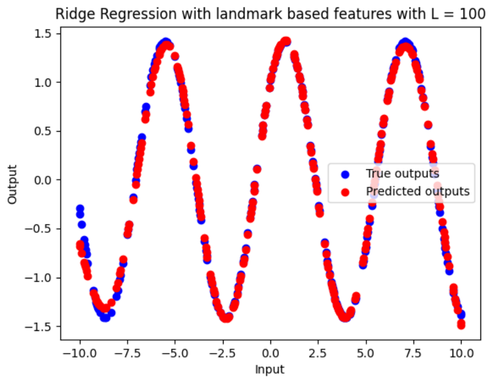
(c) RMSE with L = 20: 0.5802



(d) RMSE with L = 20: 0.1035



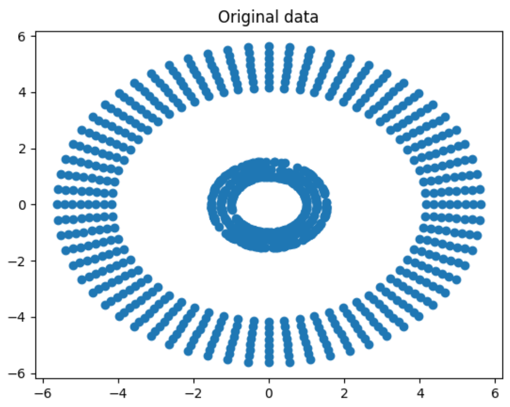
(e) RMSE with L = 50: 0.0735



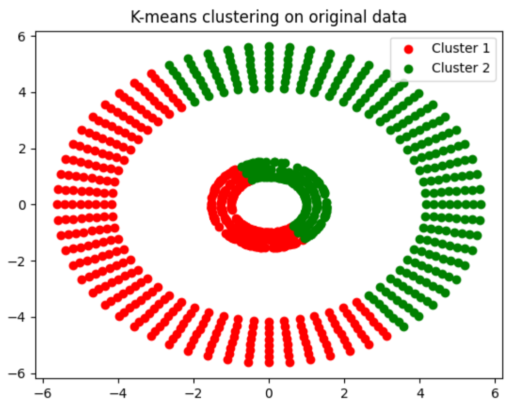
(f) RMSE with L = 100: 0.0572

For ridge regression with landmark based features, as the value of L increases, the model becomes complex, and hence, can capture more information from the data. As a result, the model produces a curve that follows the variations in the data too closely. This can be seen by the decreasing RMSE. However, after a certain point, increasing L does not improve the model performance significantly as can be observed for L = 50 and L = 100. It is seen that the model has fitted perfectly for L = 50, but also at the graph (d) with L = 20 with good initial landmarks.

Part 2.1: Using Hand-crafted Features

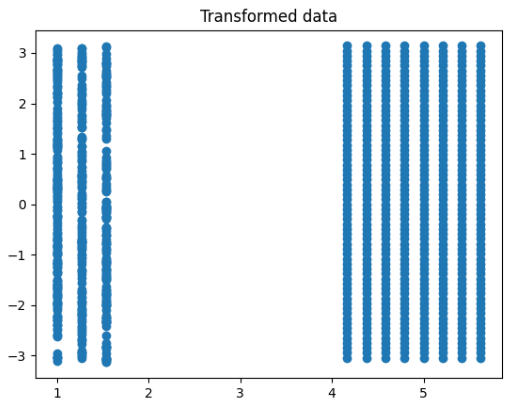


(a) Original Data

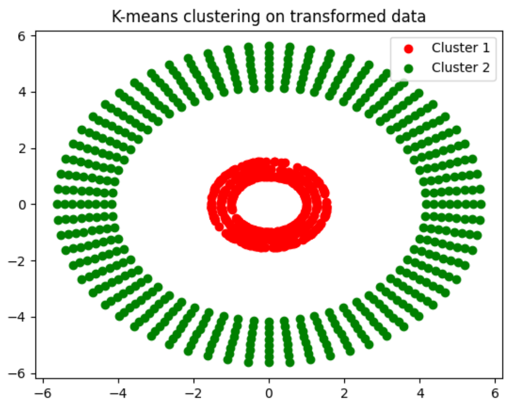


(b) K-means on Original Data

Figure 4: Linear K-means on Original Data



(a) Transformed Data

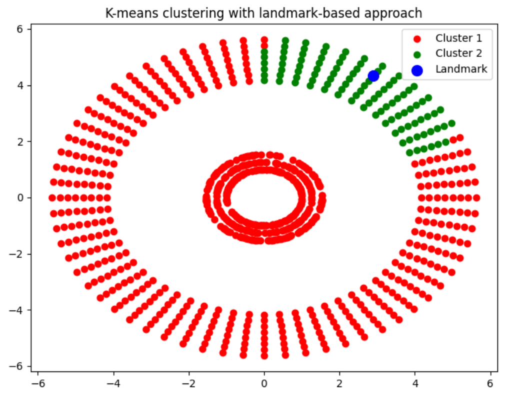


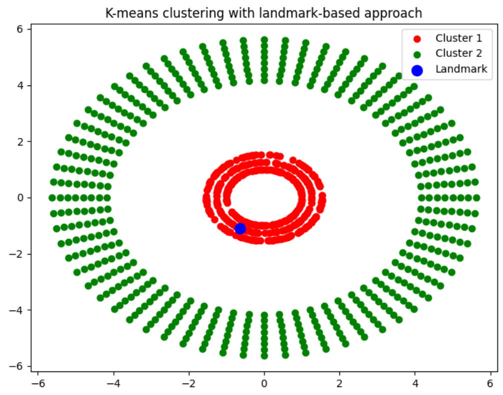
(b) K-means on Transformed Data

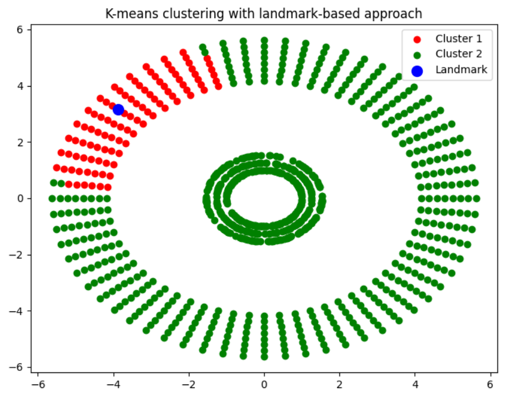
Figure 5: Linear K-means on Transformed Data

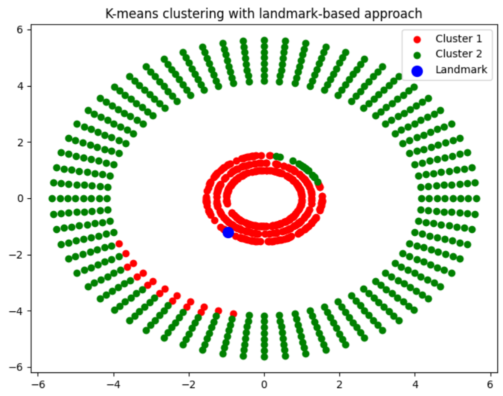
Transformation that helped in this case was to convert the Cartesian co-ordinates of each point to polar coordinates, which were more suitable for capturing the circular shape of the clusters. After applying this feature transformation, the K-means algorithm can successfully cluster the data into two groups, as shown in the last plot.

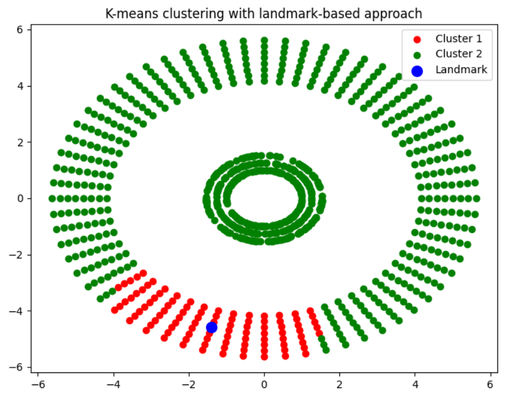
Part 2.2: Using Kernels

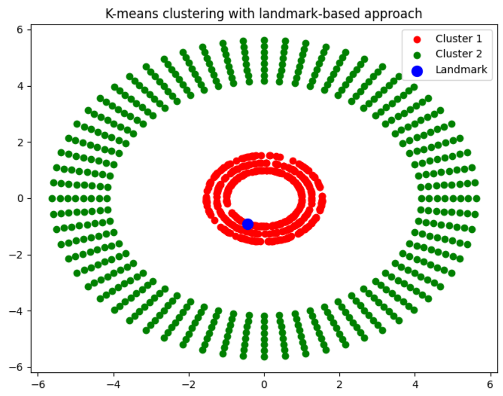


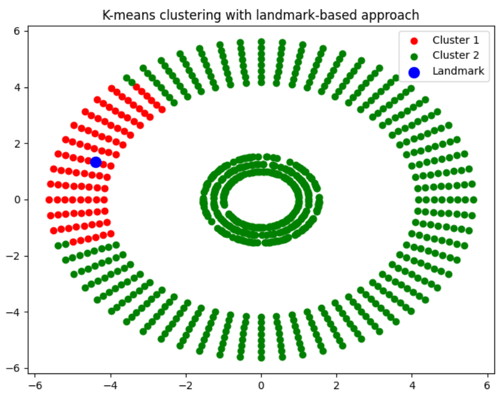


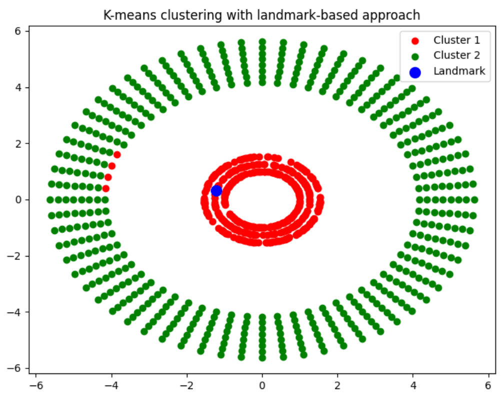


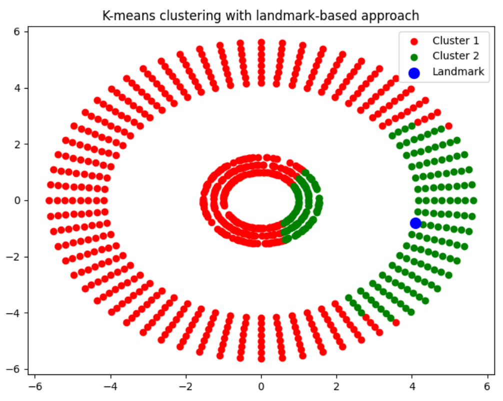


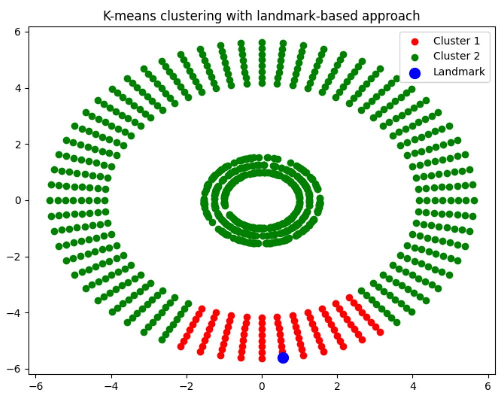






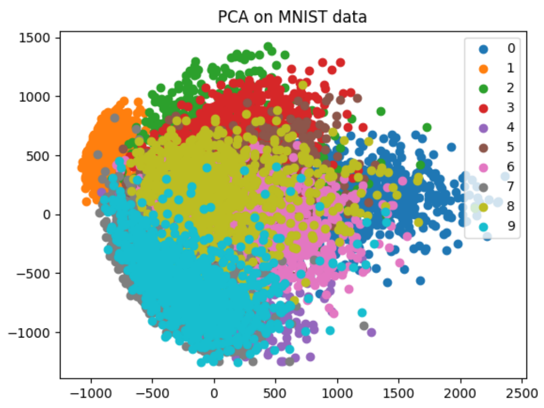




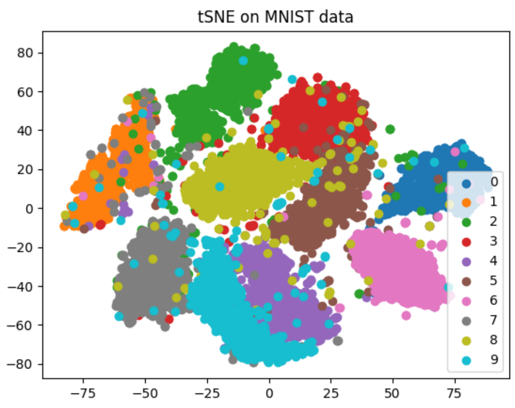


As one can see, the landmark based approach can produce different results depending on the choice of the landmark. The reason for this is that the landmark based approach transforms the data using the RBF kernel, which measures the similarity between each point and the landmark. The similarity is high when the points are close to the landmark, and low when the points are far from the landmark. This makes the clustering easy, as the K-means algorithm can separate the data based on the values. However, if the landmark is in bad position, the clustering will be difficult, as the K-means algorithm will be unable to separate the data based on the values.

Part 3: PCA and tSNE



(a) PCA



(b) tSNE

Figure 8: Dimensionality reduction using PCA and tSNE

PCA tries to capture the directions of maximum variance in the data, while tSNE tries to capture the similarities between nearby points in the data. PCA preserves the global structure of the data, while tSNE preserves the local structure of the data. As a result, PCA tends to spread out the data points more evenly, while tSNE tends to cluster the data points more tightly. This means that PCA can sometimes fail to separate the classes that are not linearly separable.