Department of Computer Science and Engineering,
Indian Institute Of Technology Madras

CS5691 : Pattern Recognition and Machine Learning
Assignment 1

Pranit Zope
AE20B046

March 10, 2024

Contents

1 Problem 1 : Principle Component Analysis

Dataset used : A subset of the MNIST Dataset such that it contains 100 random images of each label (0 to 9)

1.1 Part (i) : PCA and Variance

For PCA, we take the dataset, and compute the covariance matrix for it,

$$X = \{x_1, x_2, \dots x_n\}; \ x_i \in \mathbb{R}^d$$

$$C = \frac{1}{n} X X^T$$

Then to find the principle components and the explained variances, we eigencompute the covariance to get the eigenvalues (λ_i) and eigenvectors($\mathbf{v}_i$).

$$C\mathbf{v} = \lambda \mathbf{v}$$

Sorting the eigenvalue-eigenvector pair according to descending eigenvalues gives us the principle components and their explained variance.

On doing the PCA on this particular dataset, we obtain the following principle components depicted in the plot below. First 10 principle components are shown below in decreasing order of he explained variance (magnitude of eigenvalues).

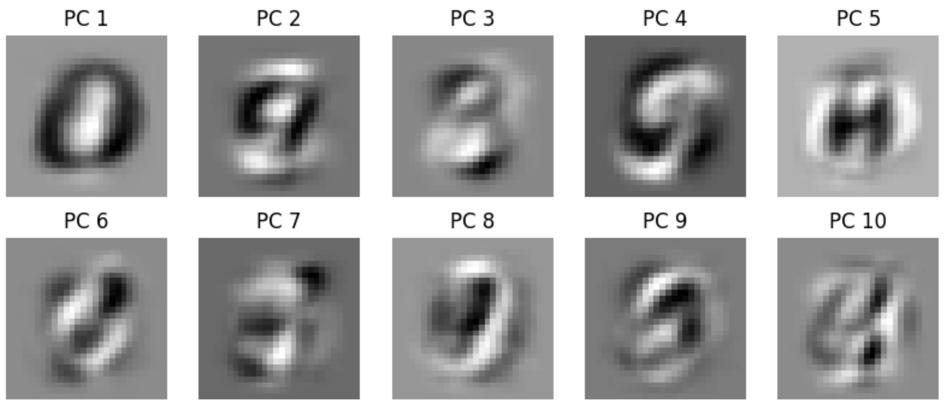


Figure 1: Top 10 Principle components obtained from PCA

The variance explained by any principal component is the eigenvalue obtained after eigencomputing the covariance matrix. Hence, We obtain the covariance matrix after centering the data.

$$X_c = X - \mu$$

Eigenvaules obtained by eigencomputing the matrix $X_c X_c^T$ give us the corresponding variance for each principal component. The percantage of explained variance is simply calculated by using

$$r = \frac{\lambda_i}{\sum_{i=1}^{d} \lambda_i} \times 100\%$$

Principle Component	Var (%)
1	10.038 %
2	7.239 %
3	6.260 %
4	5.250 %
5	4.667 %
6	4.508 %
7	3.545 %
8	2.846 %
9	2.767 %
10	2.452 %

Table 1: Explained Variance

1.2 Part (ii) : Reconstruction

Let's say we need K dimensions (or principle components) to recreate the images and use them for a downstream tasks. The threshold of variance for such a case is 95%. i.e.

$$\frac{\sum_{j=1}^{K} \lambda_j}{\sum_{j=1}^{d} \lambda_j} \geq 0.95$$

```
>> (d, req_var) = (132, 95.00855069238898)
```

Thus we can conclude that the $d = 132$ (number of dimensions to carry out a downstream task).

To reconstruct the dataset, we use K of the obtain principle componets and obtain the image by :

$$x_i = \mu + \sum_{k=1}^{K} x_i^T \omega_k$$

where x_i is the data point and ω_k is the k-th most significant principal component.
We can plot the reconstructed images for different values of K. This plot shows the reconstruction for several different values of K on a random data point.

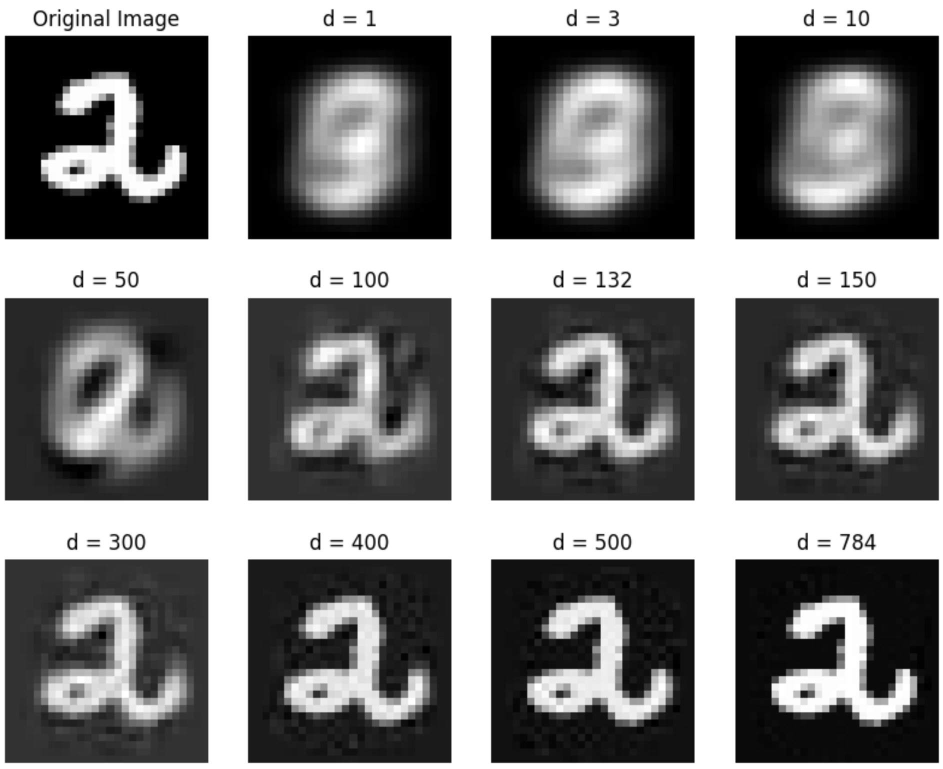


Figure 2: Reconstruction of a random image using different values of K

Moreover we can have a look at how certain other images with different labels look like when reconstructed with out computed K.

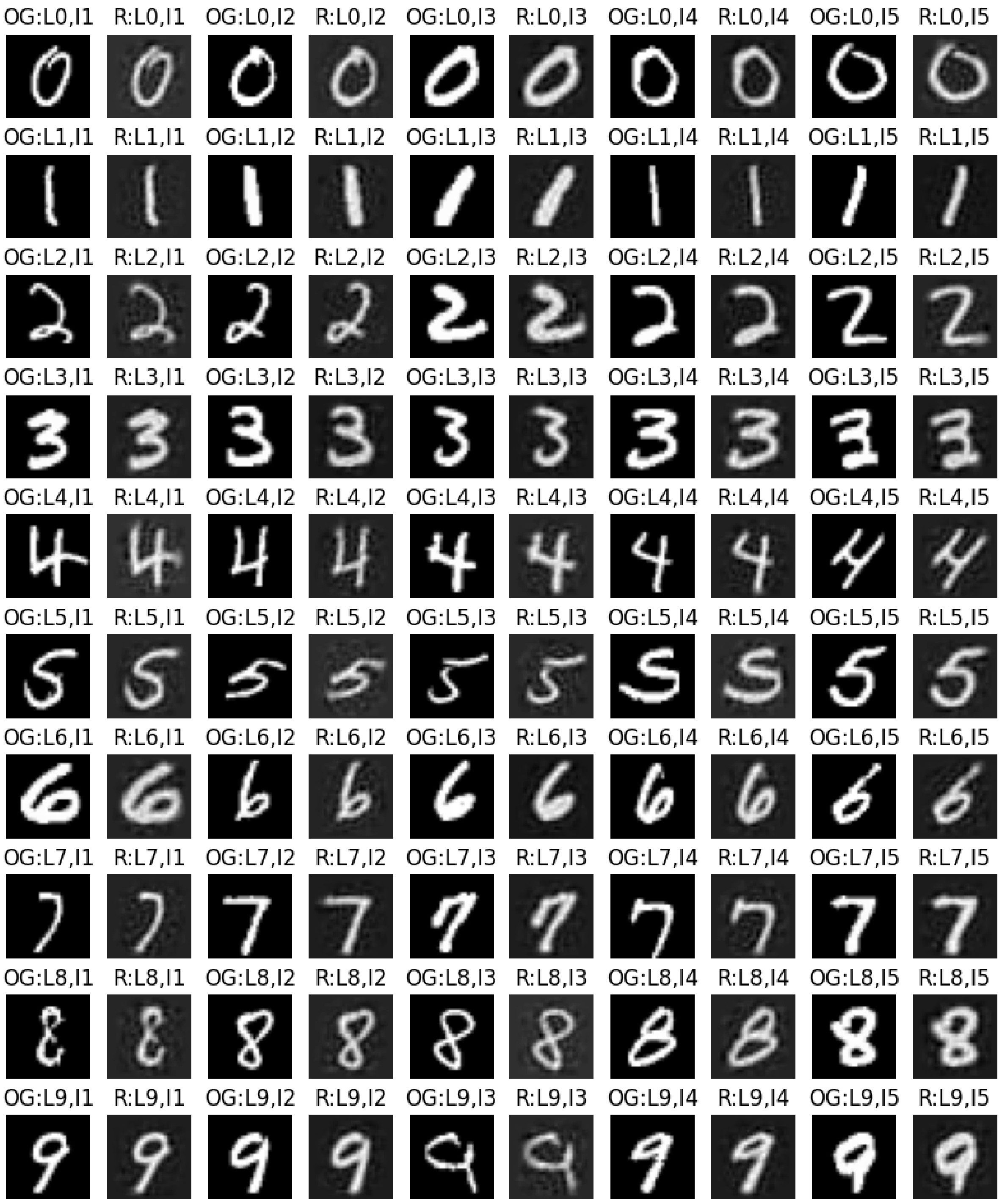


Figure 3: Reconstruction of several images for $K = 132$

Hence to conclude, we can set d to be 132 as first 132 dimensions contribute to the 95% variance and hence, can be used to carry out any downstream task.

1.3 Part (iii) : Kernel PCA

Kernel PCA is a special type of PCA where instead of carrying out tradional PCA which takes $O(d^3)$ time, we can do it in $O(n^3)$ time using a function called kernel function. The steps are as follows :

- Calculate the Kernel Matrix : $K_{ij} = \kappa(x_i, x_j)$
- Center the Kernel Matrix
- Eigencompute K and extraxt the top $k = 2$ components.

$$K\mathbf{v} = \lambda\mathbf{v}$$

- Project the datapoints.

A. We take kernel functions as

$$\kappa(x, y) = (1 + x^T y)^d$$

We use the Kernel PCA algorithm on this with $d = \{2, 3, 4\}$. Then we lot the projection of each point in the dataset onto the top-2 components this kernel.

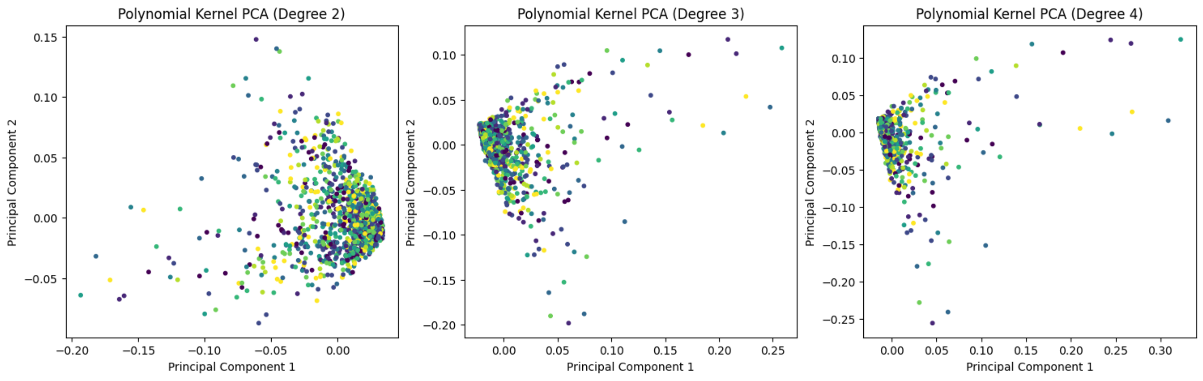


Figure 4: Top 2 components obtained from Kernel PCA with polynomial kernel

B. We now use the kernel function as

$$\kappa(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

We use the Kernel PCA algorithm on this with $\sigma = \{0.5, 1, 5\}$. Then we lot the projection of each point in the dataset onto the top-2 components this kernel.

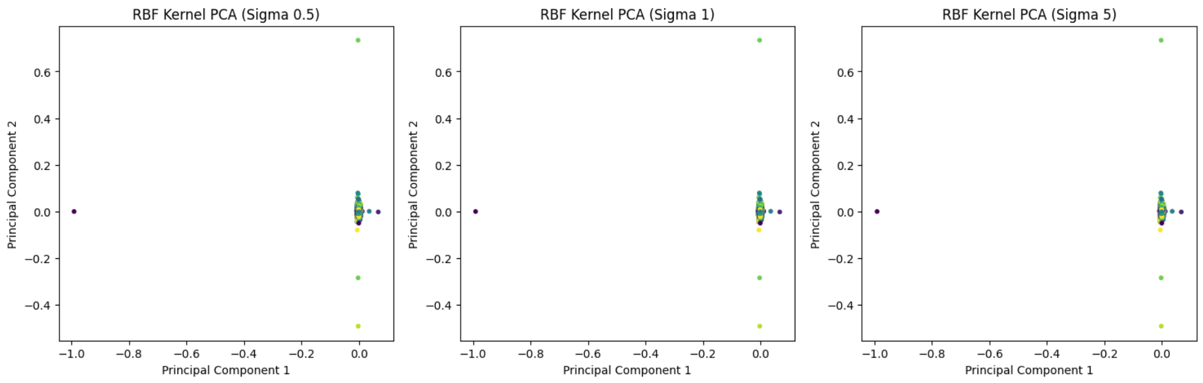


Figure 5: Top 2 components obtained from Kernel PCA with polynomial kernel

1.4 Part (iv) : Ananlysis on kernel used

The polynomial kernel with degree $d = 2$ is the best suited for this particular dataset. This is because the different labels can be observed distinctly with ease, that is it has the maximum variance amongst different labels present.

Thus the best suited kernel for this dataset according to the plot would be

$$\kappa(x, y) = (1 + x^T y)^2$$

.

2 Problem 2 : Clustering

We are given a dataset of 1000 points and we will perform certains tasks on clustering.

2.1 Part (i) : K-Means and Error Function

Dataset consists of points in two dimensional real plane.

$$X = \{x_1, x_2, \dots x_n\};\ x_i \in \mathbb{R}^2$$

We will apply Lloyd's algorithm which operates as follows:

$$\underset{t=\{1,2,\dots,K\}}{Z_i^{t+1}} = \arg\min_k \|x_i - \mu_k^t\|$$

where Z is the cluster labelling and $\mu_k^t = \frac{\sum_{i=1}^n x_i \mathbb{1}(Z_i^t = k)}{\sum_{i=1}^n \mathbb{1}(Z_i^t = k)}$.

We will also plot the error Function wrt number of interations (t) which can be given by :

$$e = \sum_{i=1}^{n} ||\mathbf{x}_i - \boldsymbol{\mu}_c^t||^2$$

. Procedure for clustering :

- We begin with $k = 2$. Then we select k random points as our centroids.
- Change the labels according to the Lloyd's algorithm and keep a track of errors.
- Break the loop when there is no change in the centroids and plot the data using different colours for different labels.
- Repeat the process for different random initializations.

In this case, 5 random initializations were carried out which resulted in the following plots :

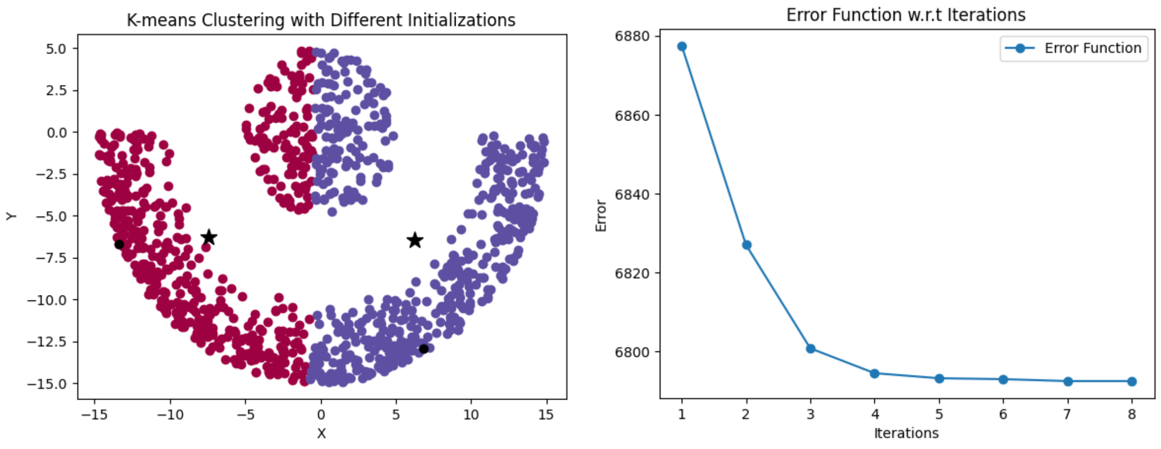


Figure 6: Lloyd's Algorithm with centroids $(-13.379, -6.7052)$ and $(6.818, -12.95)$

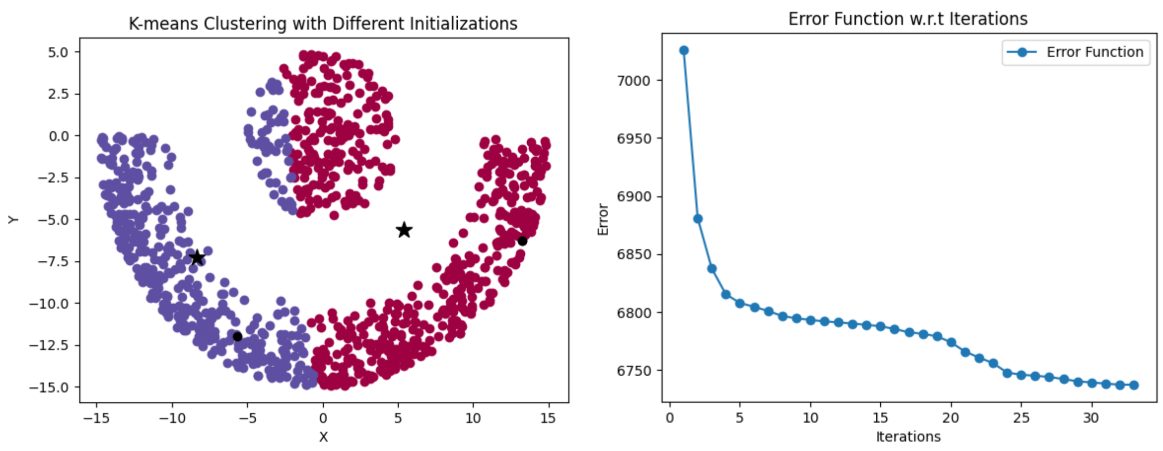


Figure 7: Lloyd's Algorithm with centroids $(13.232, -6.2824)$ and $(-5.6678, -11.988)$

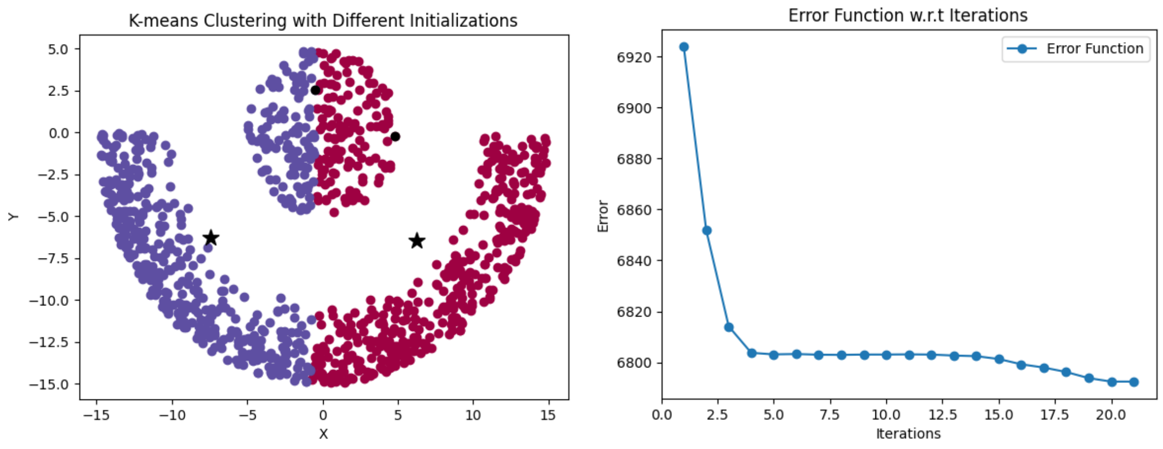


Figure 8: Lloyd's Algorithm with centroids $(4.8108, -0.20383)$ and $(-0.49906, 2.511)$

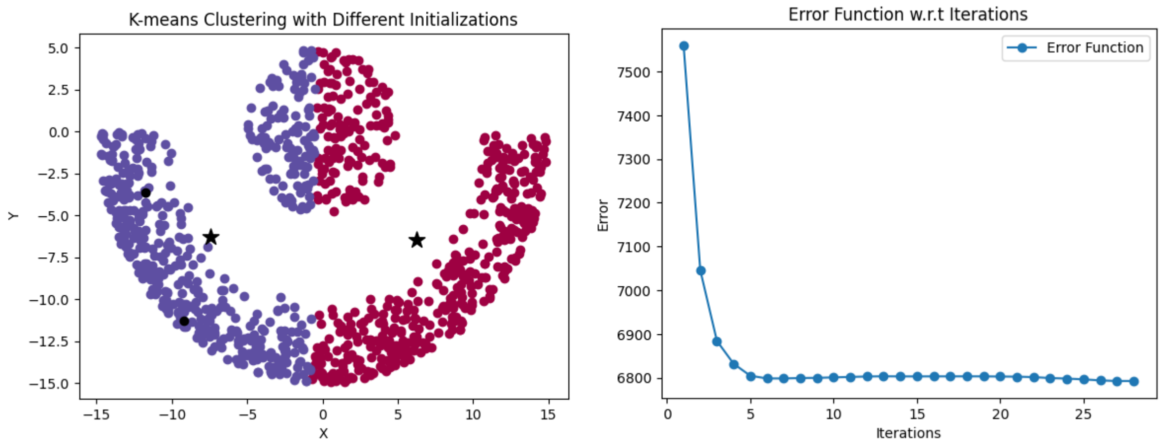
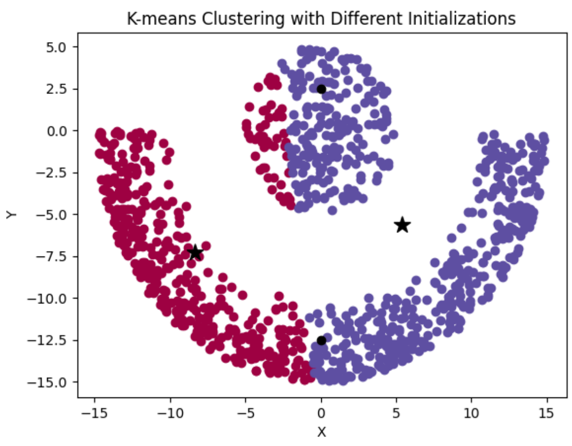


Figure 9: Lloyd's Algorithm with centroids $(-9.1809, -11.264)$ and $(-11.738, -3.6524)$



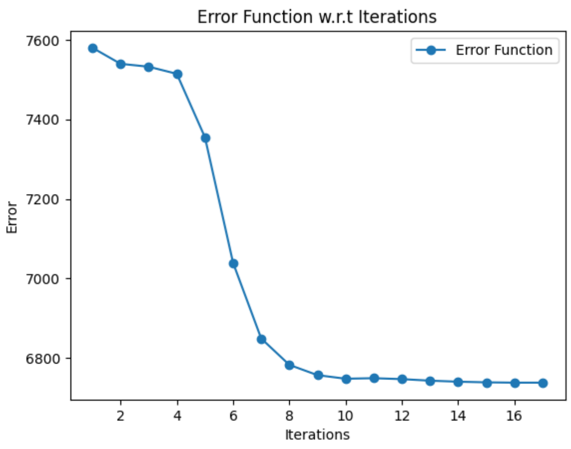


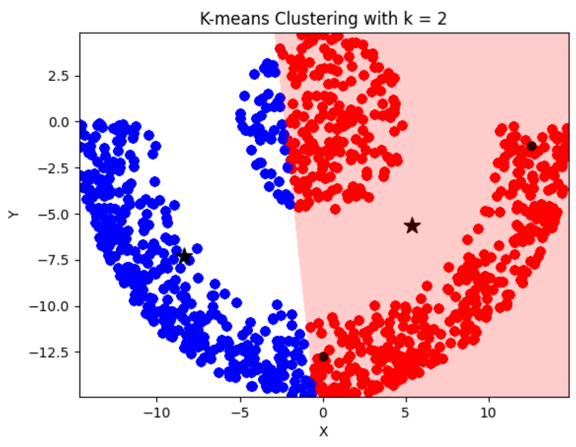
Figure 10: Lloyd's Algorithm with centroids $(0, -12.5)$ and $(0, 2.5)$

2.2 Part (ii) : K-Means and Voronoi Regions

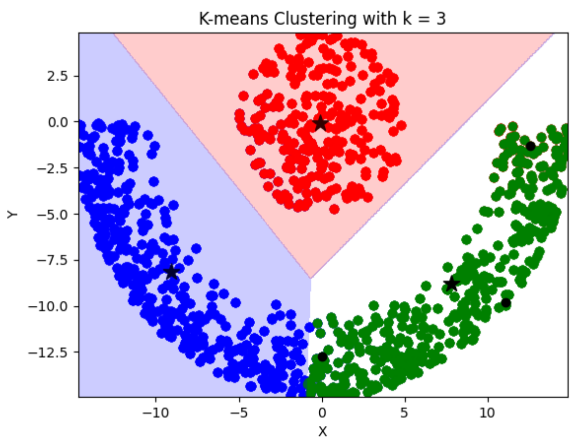
Here we take $K = \{2, 3, 4, 5\}$ and fix a random initialization, that is fix 5 randomly generated centriods. The initialisation fixed is as follows :

```
c = [[ 12.605      -1.3078 ]
     [  0.03279    -12.74   ]
     [ 11.11       -9.8368 ]
     [  5.4916     -11.178 ]
     [-10.55       -8.5064 ]]
```

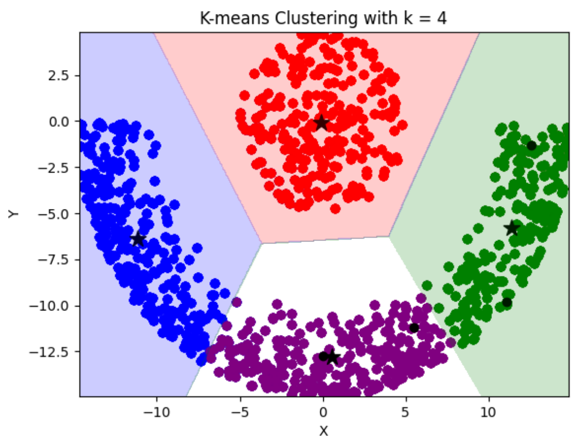
We then use the Lloyd's formula and calculate the final cluster labelling for each K. Then the voronoi regions are also plotted.



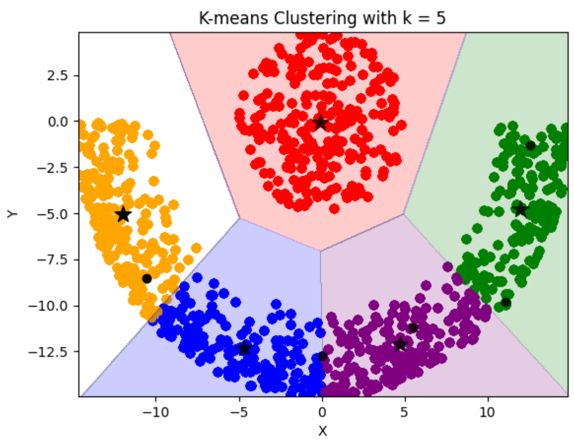
(a) K-Means Clustering when $K = 2$



(b) K-Means Clustering when $K = 3$



(c) K-Means Clustering when $K = 4$



(d) K-Means Clustering when $K = 5$

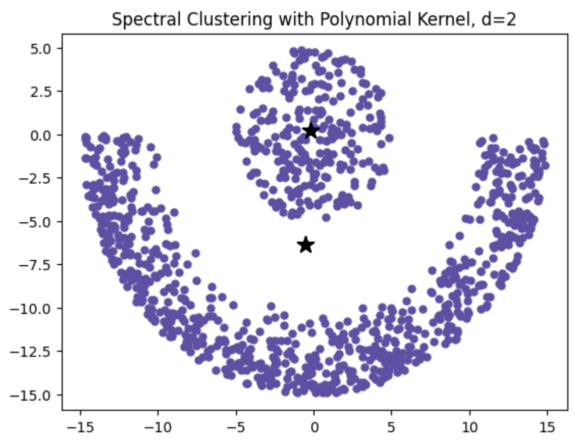
Figure 11: K-Means Clustering with varying k

2.3 Part (iii) : Spectral Clustering

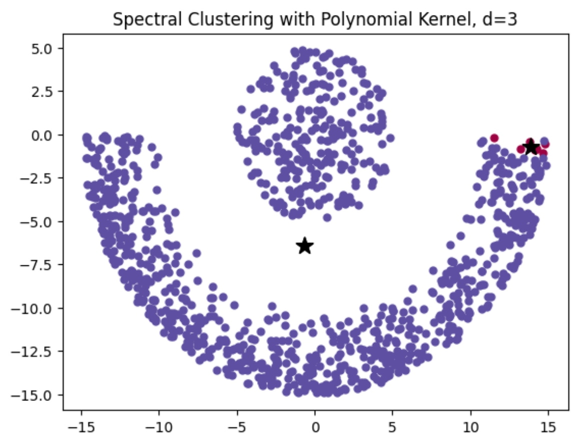
For spectral clustering, we use the following algorithm. REF_URL:

- Calculate the Kernel Matrix : $S_{ij} = \kappa(x_i, x_j)$
- Compute Laplacian $L = D - S$ where D is the degree matrix.
- Normalise L and transform it to orthogonal matrix H.
- Use the top k signifiant points of H as datapoints and do standard K-Means on them.

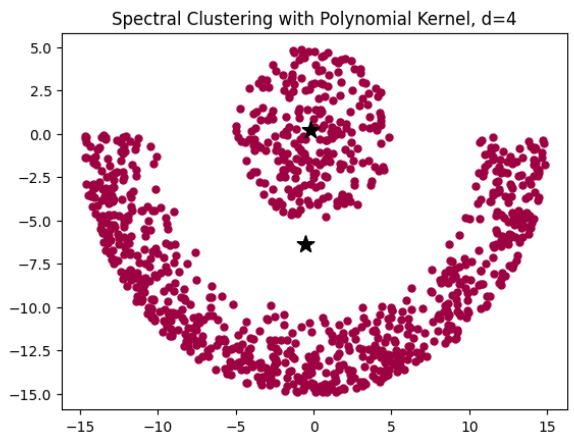
We obtain the following results. The kernels used were Radial Basis Function and the Polynomial Functions.



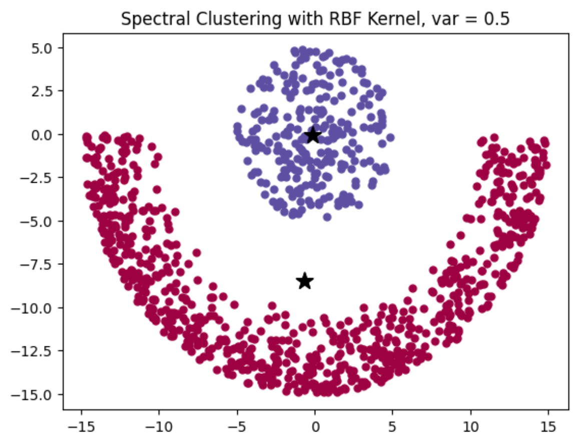
(a) Polynomial Kernel, $d = 2$



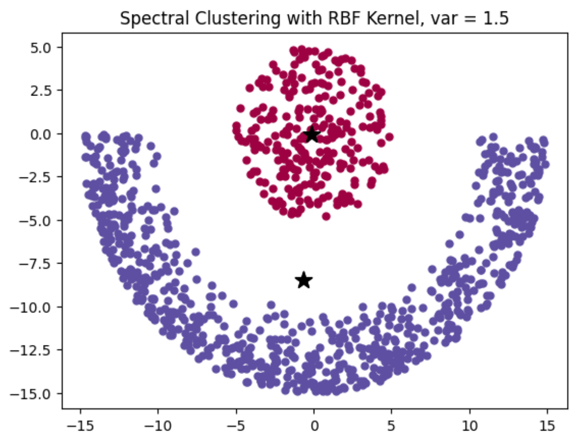
(b) Polynomial Kernel, $d = 3$



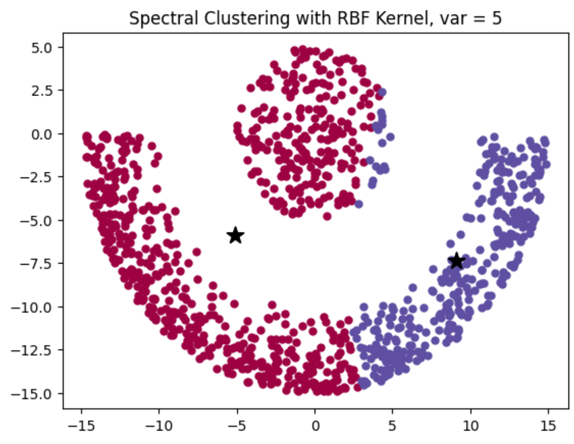
(c) Polynomial Kernel, $d = 4$



(d) Radial Basis Kernel, $\sigma = 0.5$



(e) Radial Basis Kernel, $\sigma = 1.5$



(f) Radial Basis Kernel, $\sigma = 5$

Figure 12: Overall Caption for the Figure

Here we can conclude that an rbf kernel, i.e

$$\kappa(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

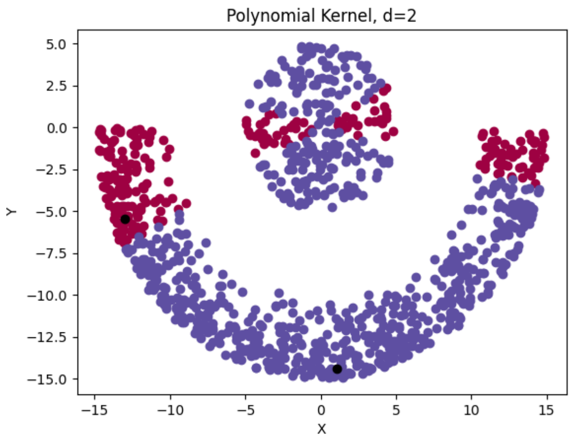
with $\sigma = 1.5$ is ideal for this dataset. The clustering done is accurate and the two regions which are apparent clusters are distinguished properly.

2.4 Part (iv) : Clustering by custom mapping function

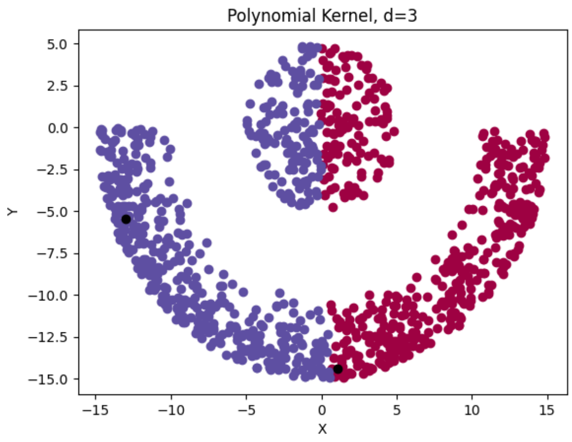
Here, we again use both the kernel functions, polynomial and radial basis, But the difference lies in the cluster assignment portion. Instead of K-Means which is generally used, we use a different approach which is such that each data point x_i is assigned to cluster ℓ if the ℓ^{th} entry of the eigenvector of K associated with the j^{th} largest eigenvalue of K is the maximum among all entries.

$$\ell = \arg \max_{j=1,\ldots,k} v_i^j$$

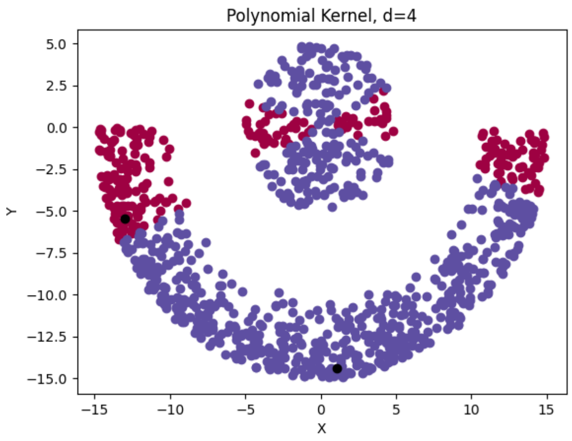
The results of this are as follows :



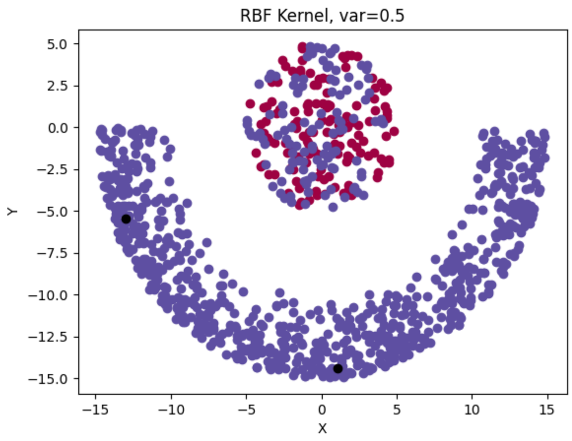
(a) Polynomial Kernel, $d = 2$



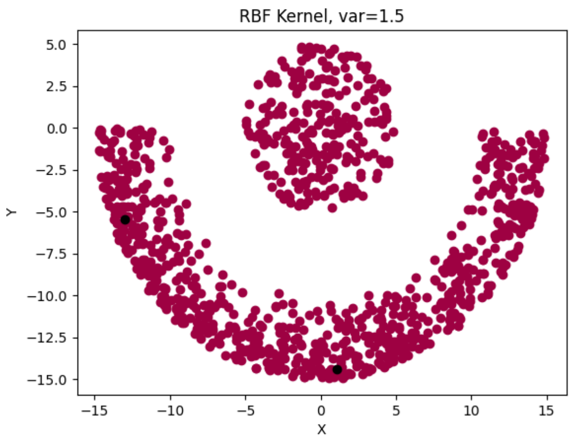
(b) Polynomial Kernel, $d = 3$



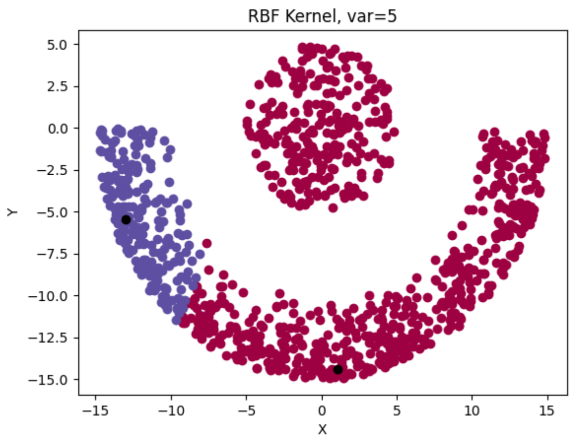
(c) Polynomial Kernel, $d = 4$



(d) Radial Basis Kernel, $\sigma = 0.5$



(e) Radial Basis Kernel, $\sigma = 1.5$



(f) Radial Basis Kernel, $\sigma = 5$

Figure 13: Clustering using the mapping based on eigenvectors

This mapping doesnt perform very well as compared to the spectral clustering wth k-means but we can still get some acceptable clustering from this algorithm.