CS5691 Pattern Recognition and Machine Learning

Assignment 1

Submitted by

CS22M025 Ayana Satheesh B

Indian Institute of Technology Madras

Contents

List of Figures

List of Tables

Chapter 1

Question 1

1.1 Principle Component Analysis

1.1.1 PCA

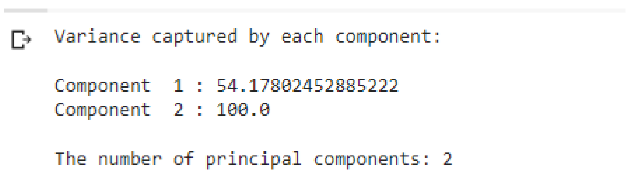
PCA is a method to compress the data and represent the data in a lower dimensional space thereby saving a lot of space especially when the dataset is very large.

1.1.2 Implementation

The source code was implemented in Google Colab Jupyter Notebook where at first it was mounted with google drive and then uploading the dataset to google drive and then accessing it.

1.1.3 Result

The variance captured by each component is shown in figure 1.1



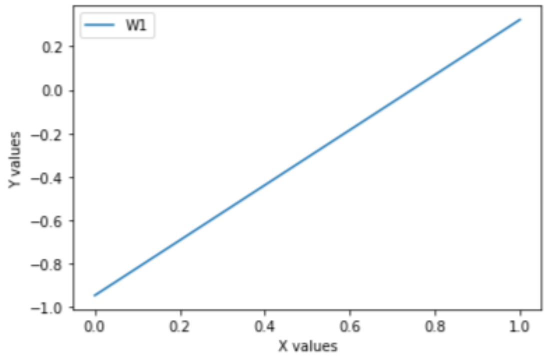
```
Variance captured by each component:

Component  1 : 54.17802452885222
Component  2 : 100.0

The number of principal components: 2
```

Figure 1.1: Variance captured by each Component

Since more than 95 percent of the data is captured by the first top 2 eigen values , no. of Principal Components is 2. Figure 1.2 and figure 1.3 shows w1 and w2 respectively. As we can see w1 is perpendicular to w2. The dataset after PCA is shown in figure1.4



1

Figure 1.2: Principal Component 1

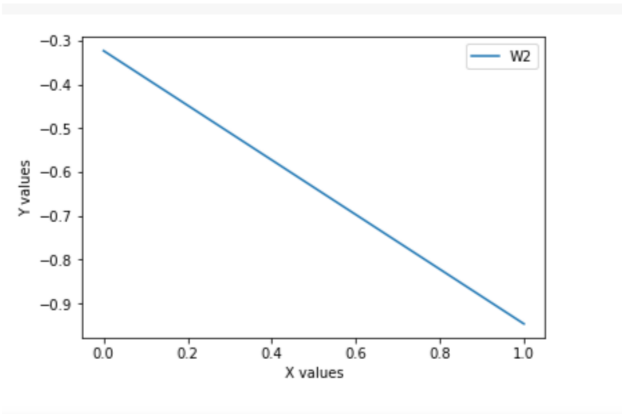


Figure 1.3: Principal Component 2

1.2 PCA without centering

PCA without centering did not produce any difference . This is because the mean that we obtained for centering is very close to (0,0) and so the dataset after centering does not changes much, so the covariance matrix obtained is the almost same and so are the eigen values and eigen vectors there by producing the same result.

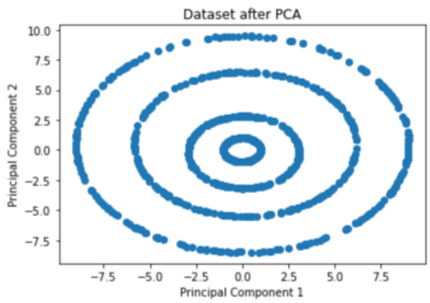


Figure 1.4: Dataset after PCA

1.3 Kernal PCA

1.3.1 Polynomial kernel

The result after kernelising the dataset and performing PCA is shown in figure 1.5 for d=2 and figure 1.6 for d=3 with the new axes being the principal components.

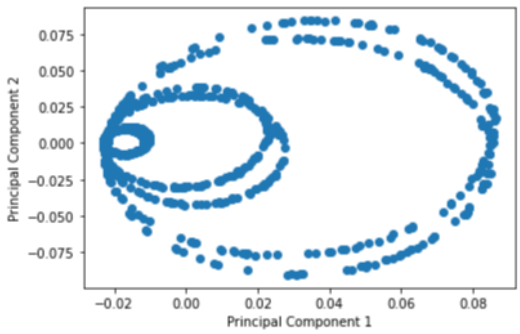


Figure 1.5: d=2

1.3.2 Gaussian kernel

The result after kernelising the dataset using gaussian kernel and performing PCA is shown in following figures from 1.7 to 1.16 for different values of sigma.

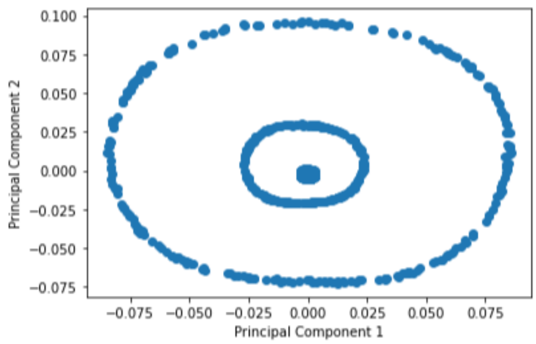


Figure 1.6: d=3

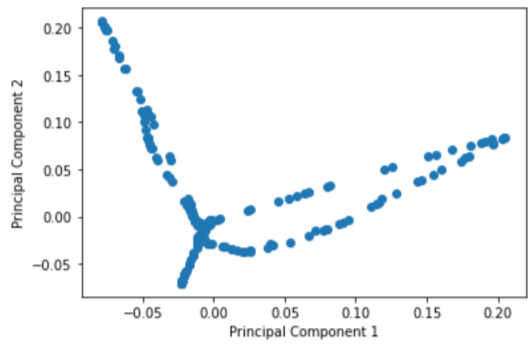


Figure 1.7: sigma=0.1

1.4 Best Kernel

In situations, when data is non-linear traditional PCA is not applicable since PCA finds only linear relationship. For those cases, Kernel PCA is used. The main aim of kernel PCA is to map the dataset to a higher dimension such that in that high dimension the dataset is linear and we can perform PCA in that high dimension. As it is clearly visible from the above figures for the gaussian kernel is best suited for the given dataset since the data we get after kernelising is almost linear in it and reduces the error also and also makes the data linearly seperable.

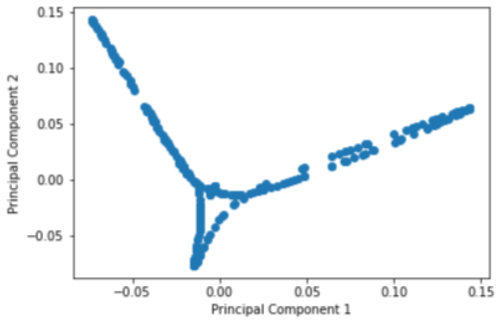


Figure 1.8: sigma=0.2

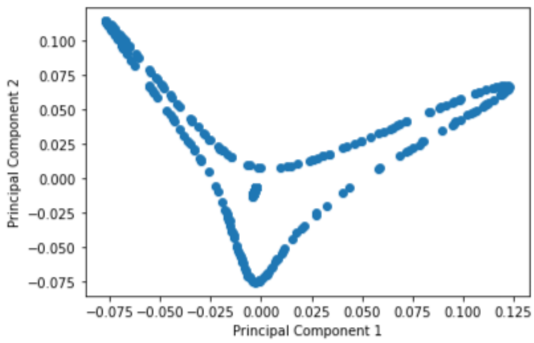


Figure 1.9: sigma=0.3

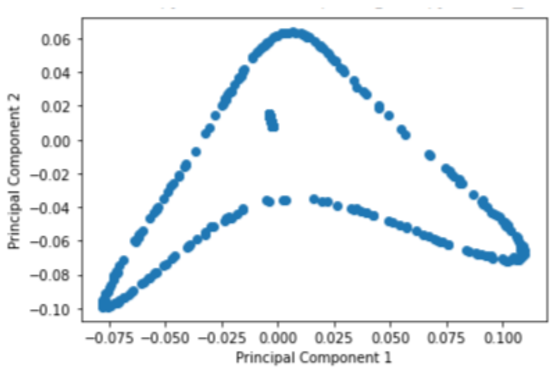


Figure 1.10: sigma=0.4

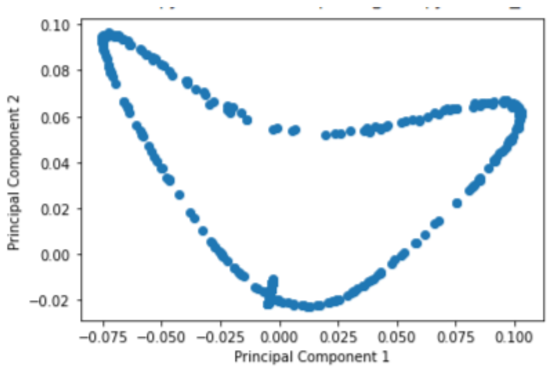


Figure 1.11: sigma=0.5

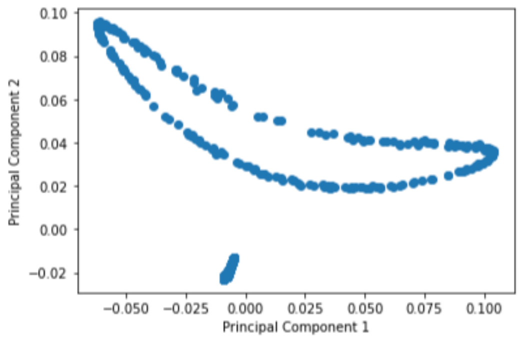


Figure 1.12: sigma=0.6

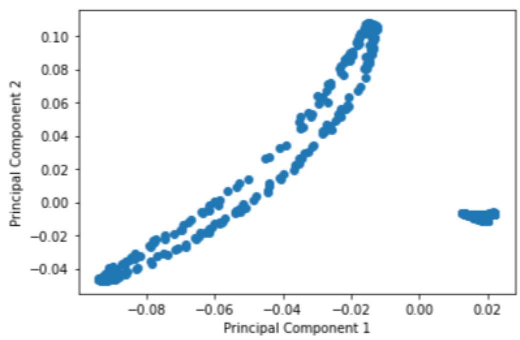


Figure 1.13: sigma=0.7

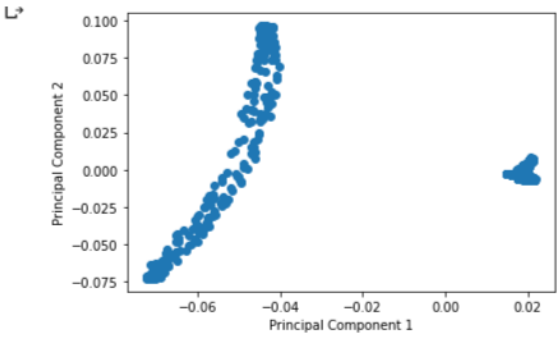


Figure 1.14: sigma=0.8

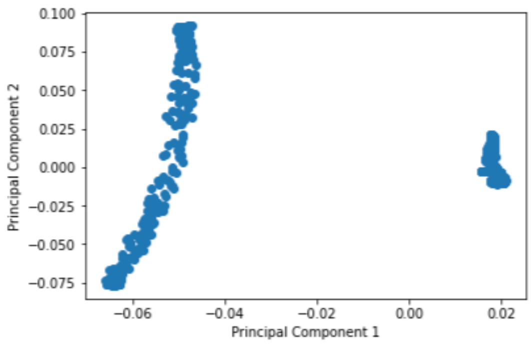


Figure 1.15: sigma=0.9

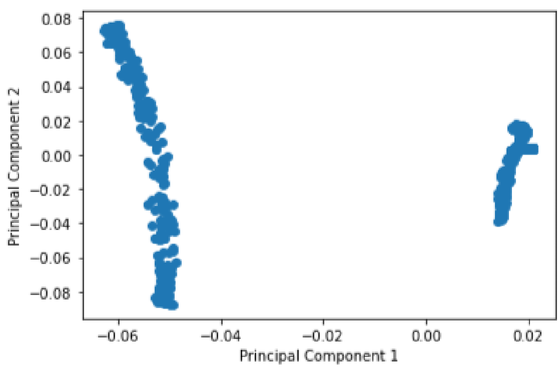


Figure 1.16: sigma=1.0

Chapter 2

Question 2

2.1 K means clustering for k=4

The result for first random initialisation and error function is shown in figure 2.1 and figure 2.2 respectively. In the first iteration error is large but in subsequent iterations it decreases and approaches to zero.

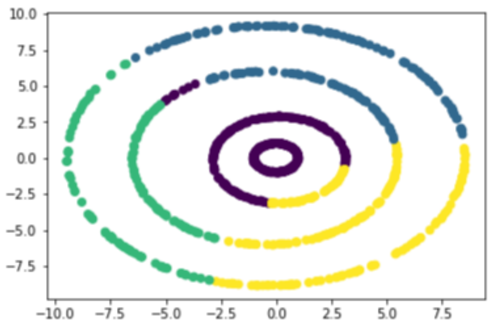


Figure 2.1: K-means clustering for a random partition

2.2 K means clustering for fixed partition

The result after kmeans clustering for different values of k are shown in figure 2.3 to figure 2.6

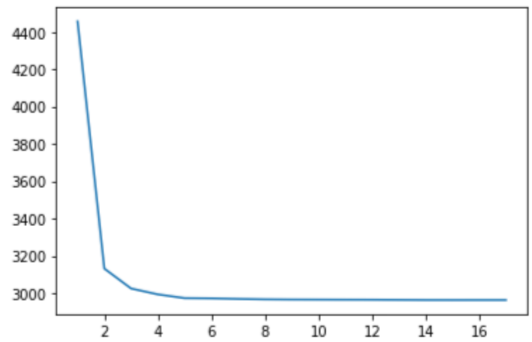


Figure 2.2: Error function

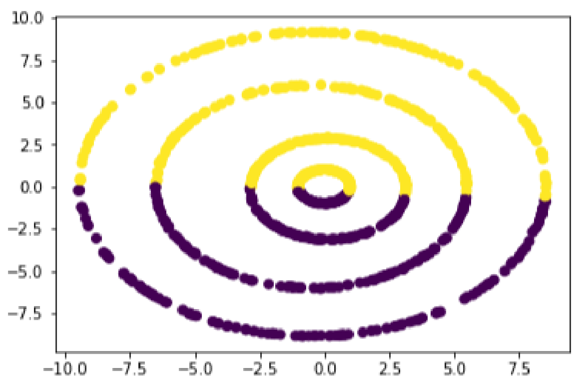


Figure 2.3: k=2

2.3 Spectral clustering

The result after spectral clustering using polynomial kernel with d=2 is shown in figure 2.7. The polynomial kernel with d=2 is chosen since it treis to best cluster the data approximately compared to other kernel for this dataset.

2.4 Different approach for clustering

This mapping also produces similar result as that of kmeans clustering with k=4 except that more data points are more data points are mapped to yellow cluster. This is because eigen vectors in H* are in sorted order and we are picking that index in which there is maximum value. (figure 2.8)

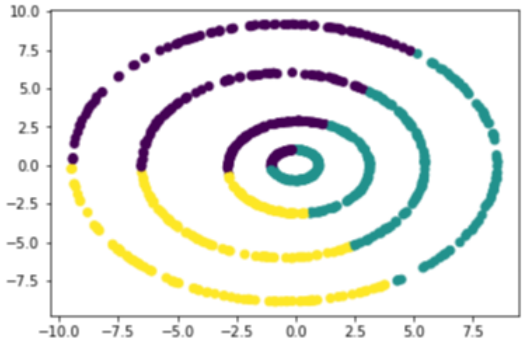


Figure 2.4: k=3

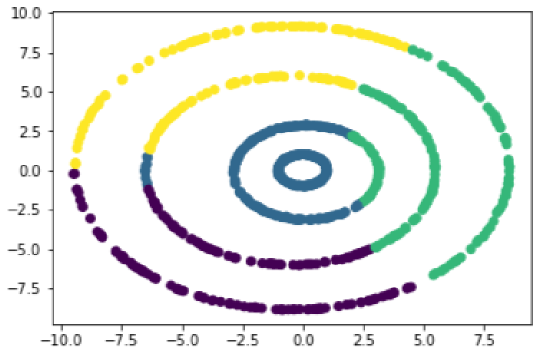


Figure 2.5: k=4

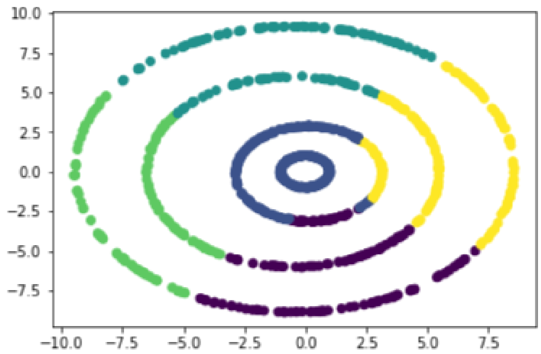


Figure 2.6: k=5

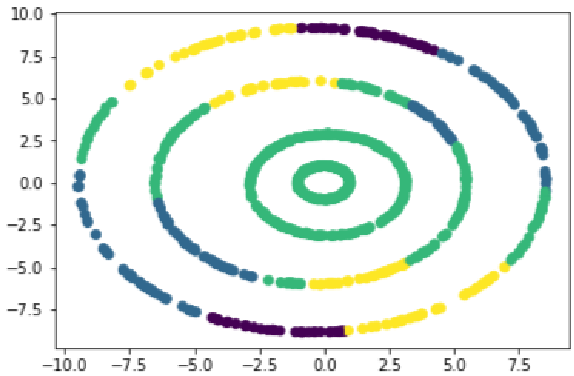


Figure 2.7: spectral clustering

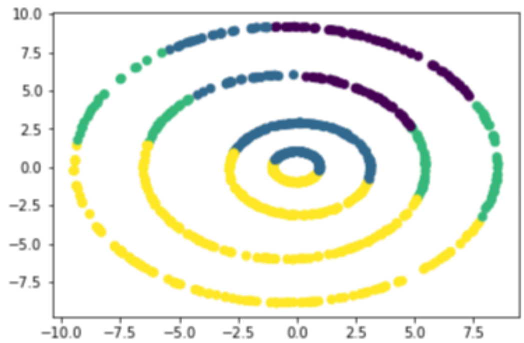


Figure 2.8: different approach for clustering