Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN3150 – Pattern Recognition

Kernel Methods

(Assignment 04 - Report)

Pattern-Recognition/Kernel Methods at main · kavindukalinga/Pattern-Recognition (github.com)

200087A Chandrasiri Y.U.K.K.

1. Kernel Methods

Input space to feature space mapping (projection) is given by the following function.

$$\Phi(x) = (1, \sqrt{2}x, x^2).$$

For a one-dimensional input space, the corresponding kernel function is as below.

Inner product of feature space gives the Kernal function.

$$k(x,z) = <\Phi(x), \Phi(z)>$$

$$k(x,z) = 1 \cdot 1 + (\sqrt{2}x \cdot \sqrt{2}z) + (x^2 \cdot z^2)$$

$$k(x,z) = 1 + 2xz + x^2z^2$$

$$k(x,z) = (1 + xz)^2$$

We can express the kernel function provided above for a two-dimensional input space as below, where $x = (x_1, x_2)$ and $z = (z_1, z_2)$.

$$k(x,z) = (1 + x.z)^2$$

$$k(x,z) = \left(1 + (x_1, x_2).(z_1, z_2)\right)^2$$

$$k(x,z) = \left(1 + (x_1 \cdot z_1 + x_2 \cdot z_2)\right)^2$$

$$k(x,z) = 1 + 2(x_1 \cdot z_1 + x_2 \cdot z_2) + (x_1 \cdot z_1 + x_2 \cdot z_2)^2$$

$$k(x,z) = 1 + 2\,x_1 \cdot z_1 + 2\,x_2 \cdot z_2 + (x_1 \cdot z_1)^2 + 2\,x_1 \cdot z_1 \cdot x_2 \cdot z_2 + (x_2 \cdot z_2)^2$$

The mapping function $\Phi(x)$ for the kernel function provided above is:

$$\Phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

Consider the kernel $k = (1 + x^T z)^2$

For the given dataset, determined kernel matrix (gram matrix) is as below.

$$\boldsymbol{G} = \begin{bmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & k(\boldsymbol{x}_1,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_1,\boldsymbol{x}_N) \\ k(\boldsymbol{x}_2,\boldsymbol{x}_1) & k(\boldsymbol{x}_2,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_2,\boldsymbol{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N,\boldsymbol{x}_1) & k(\boldsymbol{x}_N,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_N,\boldsymbol{x}_N) \end{bmatrix}$$

```python
import numpy as np
dataset = np.array (((1,5),(3,4),(4,2),(10,12)))

def kernalfuntion(x,z):
    return (1+np.dot(x,z))**2

gm = np.zeros((4,4))
for i in range(4):
    for j in range(4):
        gm[i,j] = kernalfuntion(dataset[i] , dataset[j])

print("The gram matrix obtained\n",gm)
```

[4] ✓ 0.0s

```
The gram matrix obtained
 [[  729.   576.   225.  5041.]
 [  576.   676.   441.  6241.]
 [  225.   441.   441.  4225.]
 [ 5041.  6241.  4225. 60025.]]
```

Is this a valid kernel for this data set? To answer this, the eigen values of the gram matrix are obtained.

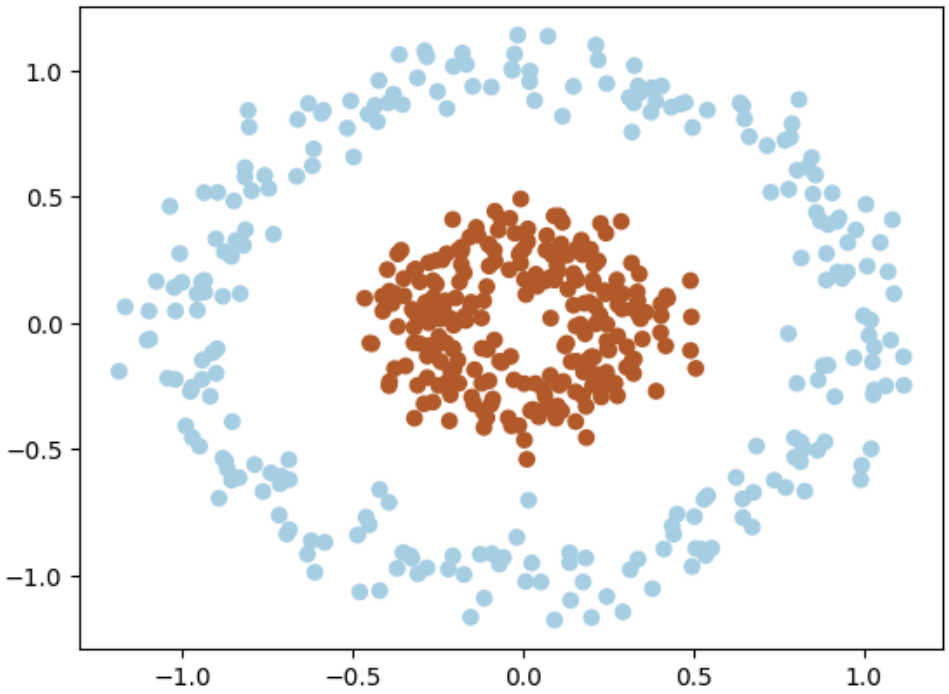
```
Eigen values:
 [6.13971435e+04 3.82206067e+02 8.15317855e+01 1.01186495e+01]
```

We can observe that the gram matrix has non-negative eigen values. Therefore, the gram matrix should be positive semi definite. From that we can say that the gram matrix is symmetric and therefore, the kernel $k = (1 + x^T z)^2$ is a valid kernel for the given dataset.

5)

The code given in “Listing 1: Data generation” is used to generate data.

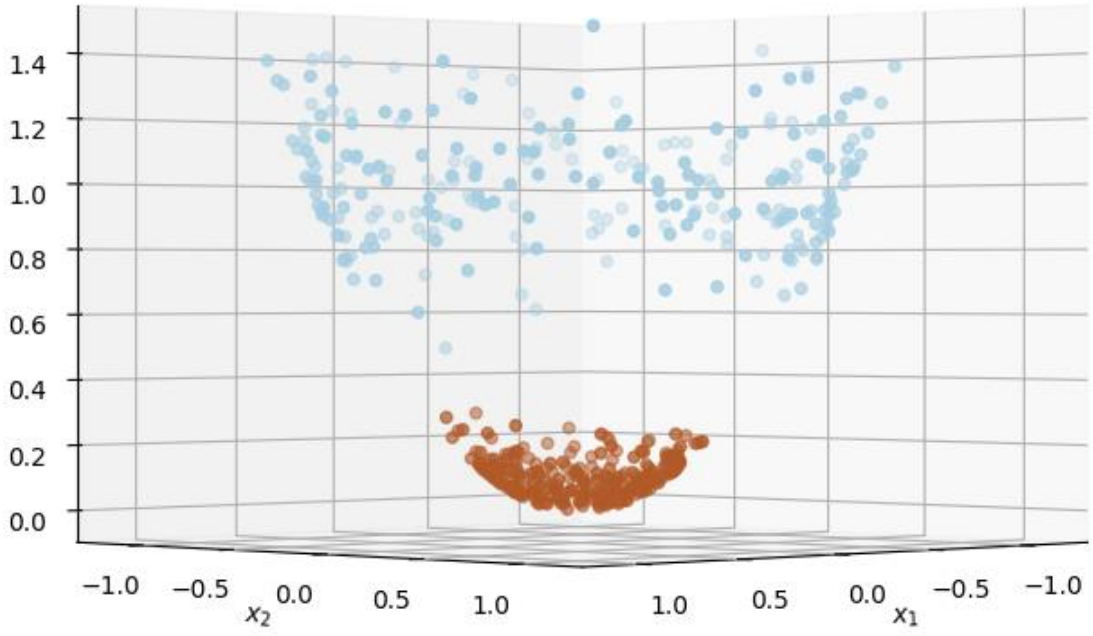
Figure: The scatter plot of the data



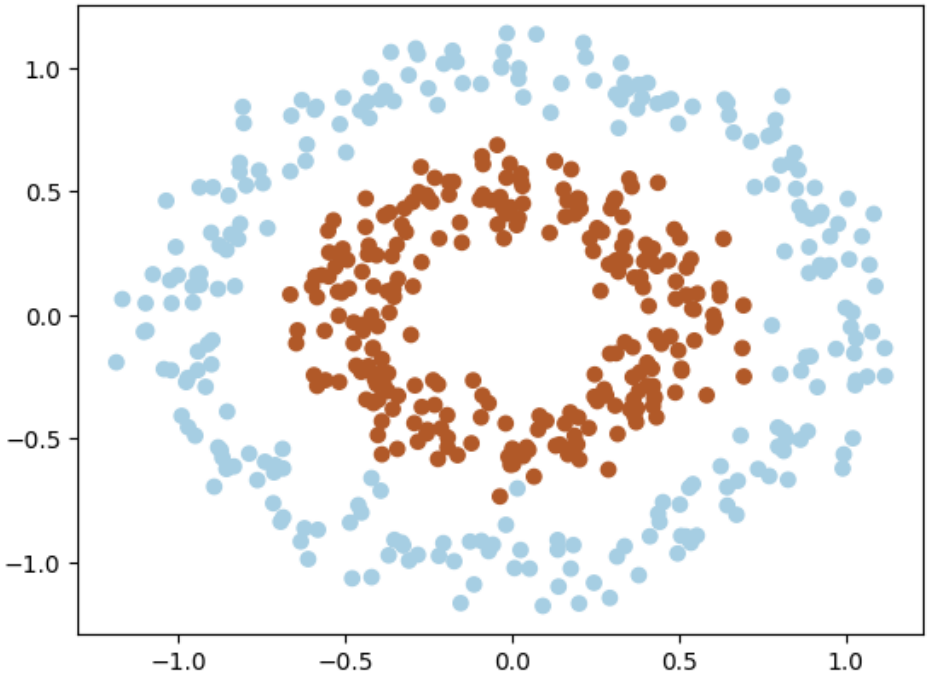
The following mapping is used to map two-dimensional space to three-dimensional space (feature space). This feature space set is known as projected set.

$$\Phi: x = (x_1, x_2) \rightarrow \Phi(x) = (x_1, x_2, x_1^2 + x_2^2) \in R^3$$

Figure: Projected Dataset



Now changed the "factor=0.5" and observed the feature space in 3D.



What changes to observe?

The class inside has spread than before making the boundary between 2 classes narrow. It made hard to classify now compared to previous one.

For the same dataset, the mapping $\Phi: x = (x_1, x_2) \rightarrow \Phi(x) = (x_1^2, x_2^2, x_1 x_2)$ is used.

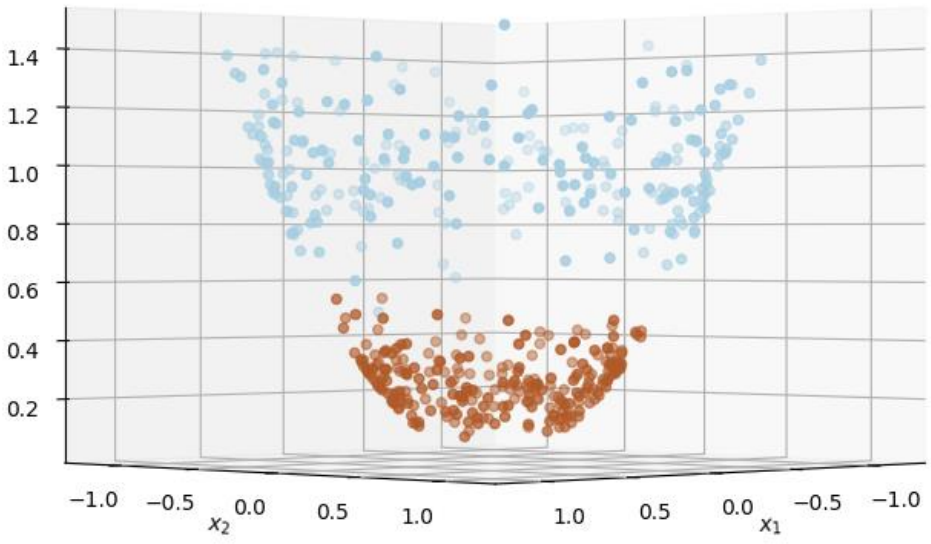


Figure: Projected dataset using previous mapping

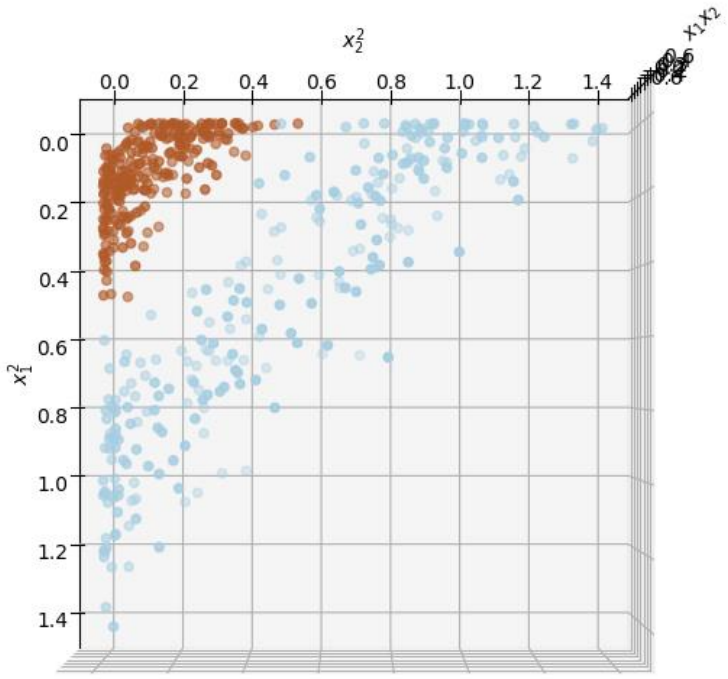


Figure: Projected dataset using new mapping

Is this mapping better than the previous mapping? When observing the two plots, we can not see a significant difference between these two when considering the classifying the two classes. Both mappings can be used to classify the dataset with almost same accuracy.

Linear SVC is run on the original dataset which is generated based on "Listing 1: Data generation" and the projected set using the mapping given in (5b).

The classification accuracies for both cases as follows.

```
Accuracy for original dataset: 0.668
Accuracy for projected dataset: 1.0
```

Using the feature map, we have got maximum accuracy of 1.0 significantly better than the accuracy of original dataset of 0.668.