

Department of Electronic and Telecommunication
Engineering
University of Moratuwa



BM3150 – Pattern recognition
Kernel Methods

200003P
T.L Abeygunathilaka

1. Kernel Methods

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

Kernel function can be represented as,

$$\begin{aligned}k(x, z) &= \Phi(x) \cdot \Phi(z) \\k(x, z) &= 1 * 1 + \sqrt{2}x * \sqrt{2}z + x^2 * z^2 \\k(x, z) &= 1 + 2xz + (xz)^2 \\k(x, z) &= (1 + xz)^2\end{aligned}$$

Therefore, the kernel function for 1-D input space can be written as $\mathbf{k(x, z)} = (\mathbf{1 + xz})^2$

1.2. Let $x = (x1, x2)$ and $z = (z1, z2)$

For 2-D input space $k(x, z)$ can be written as below;

$$\begin{aligned}\mathbf{k(x, z)} &= (\mathbf{1 + x1 * z1 + x2 * z2})^2 \\&\text{or} \\k(x, z) &= (1 + x \cdot z)^2\end{aligned}$$

1.3. Mapping function for the 2-D input space kernel.

$$\begin{aligned}k(x, z) &= (1 + x1 * z1 + x2 * z2)^2 \\k(x, z) &= 1 + 2x_1z_1 + 2x_2z_2 + (x_1z_1)^2 + (x_2z_2)^2 \\k(x, z) &= (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2) \cdot (1, \sqrt{2}z_1, \sqrt{2}z_2, \sqrt{2}z_1z_2, z_1^2, z_2^2) \\k(x, z) &= \Phi(x) \cdot \Phi(z) \\\Phi(x) &= (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2)\end{aligned}$$

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

Data set;

Sample Index	Data sample	Feature 1	Feature 2
1	X ₁	1	5
2	X ₂	3	4
3	X ₃	4	2
4	X ₄	10	12

Gram matrix calculation

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

```
def kernel(x,z):
    return (1+ x[0]*z[0]+ x[1]*z[1])**2

def gram_matrix(data):

    gram = np.zeros((data.shape[0], data.shape[0]))
    for i in range(data.shape[0]):
        for j in range(data.shape[0]):
            gram[i][j] = kernel(data[i],data[j])
    return gram
```

Output gram matrix: -

```
[ [ 729.   576.   225.  5041.]
  [ 576.   676.   441.  6241.]
  [ 225.   441.   441.  4225.]
  [ 5041.  6241.  4225. 60025.]]
```

Gram matrix should be positive semi definite to make the kernel valid.

Therefore calculated eigen values are as follows,

Eigenvalues:

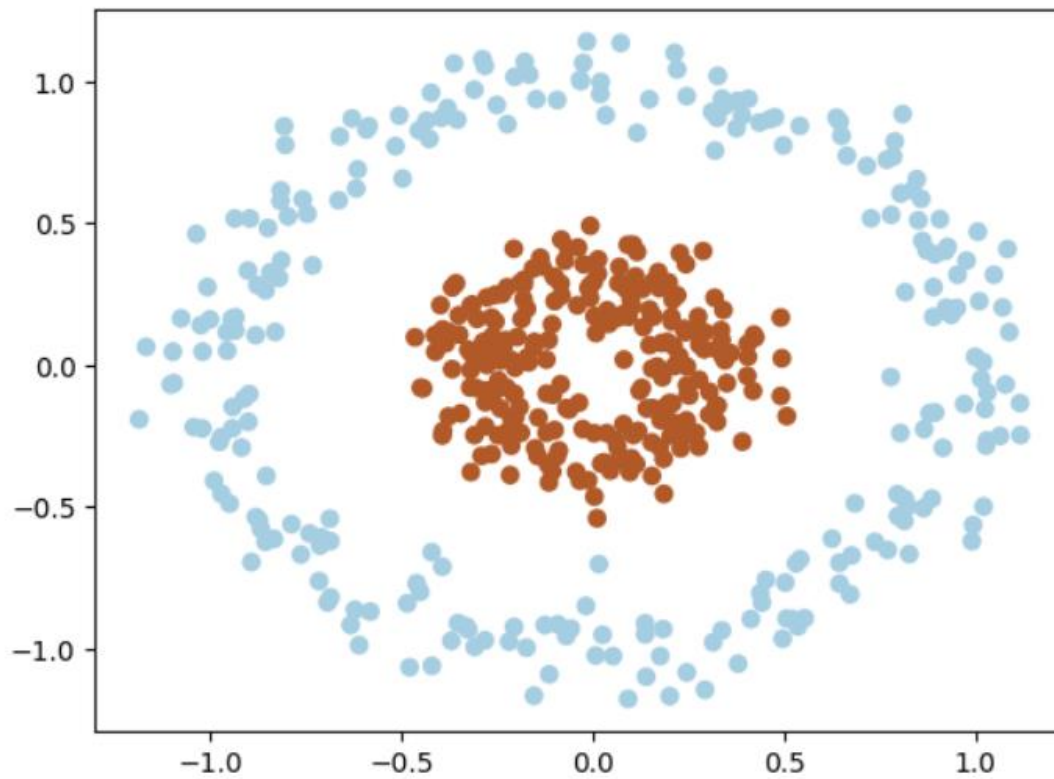
```
[6.13971435e+04  3.82206067e+02  8.15317855e+01  1.01186495e+01]
```

All the eigen values are non-negative.

Therefore, the kernel is valid for the dataset.

1.5.

1.5.1. Scatter plot of the generated data.

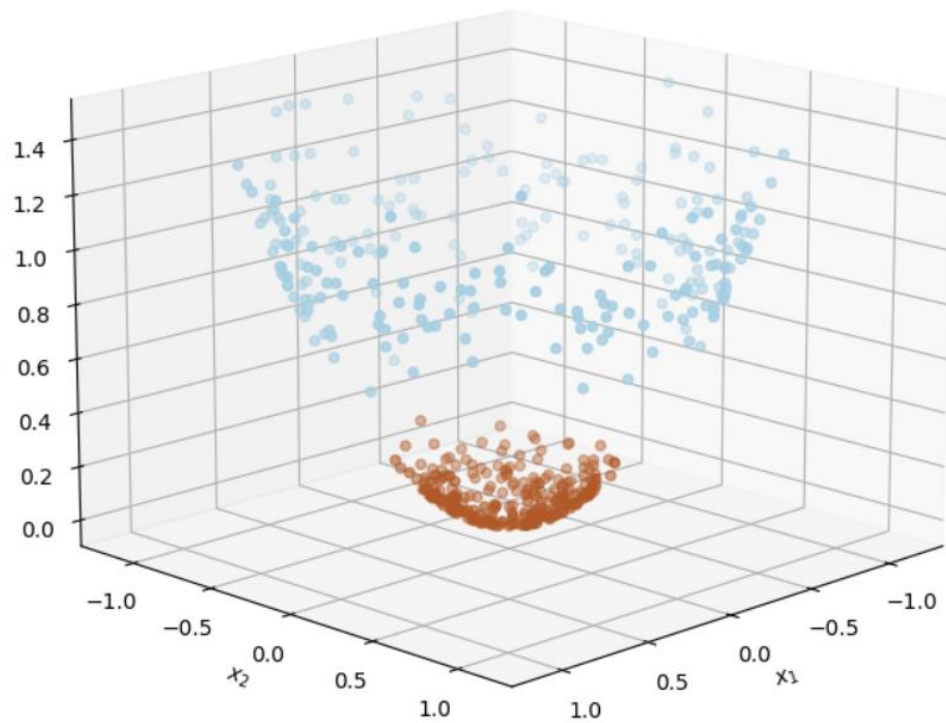


1.5.2. 2-D to 3-D mapping was done using following feature space

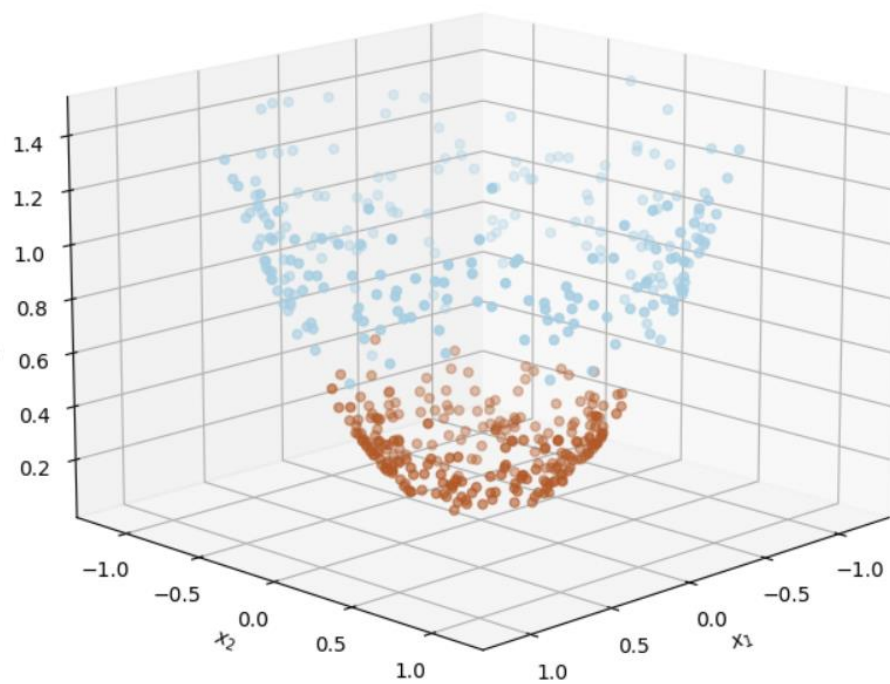
$$\Phi : \mathbf{x} = (x_1, x_2) \rightarrow \Phi(\mathbf{x}) = (x_1, x_2, x_1^2 + x_2^2) \in \mathbb{R}^3$$

```
def map_X1(data):  
    a = np.zeros((data.shape[0], 3))  
    a[:, :2] = data  
    a[:, 2] = data[:, 0]**2 + data[:, 1]**2  
    return a
```

Visualization of the above data points after feature mapping is shown below.

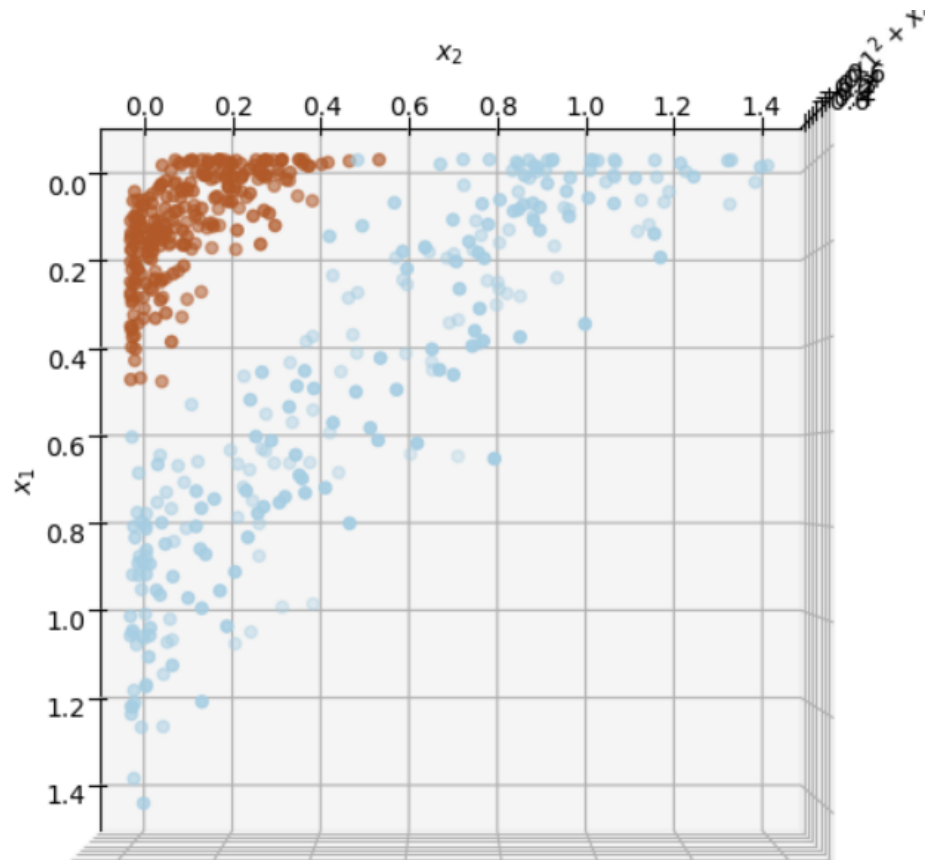


1.5.3. Then changed the factor to 0.5 and visualized the feature space in 3D.



In the feature space, there is no clear boundary between two data classes. We can clearly observe some overlapping areas between 2 data classes. Therefore, the feature map we used earlier is not suitable for this dataset.

Then we used $\mathbf{x} = (x_1, x_2) \rightarrow (x_1^2, x_2^2, x_1x_2)$ feature map for the new dataset.



As we can see in the above image after using the new feature mapping, we can see a clear boundary between 2 classes more than above method. Therefore, this mapping is better than previous one.

1.5.4. SVC models were fitted for the dataset. For both 2-D original data set and for the 3-D mapped dataset.

```
pred1=svm.SVC(kernel='linear').fit(X,y).predict(X)
print("Accuracy = {}".format(accuracy_score(y,pred1)))
```

```
pred2=svm.SVC(kernel='linear').fit(X_new,y).predict(X_new)
print("Accuracy = {}".format(accuracy_score(y,pred2)))
```

```
Accuracy for 2D dataset = 0.668
Accuracy for mapped dataset = 1.0
```

therefore we can state that even though there cannot be established a perfect linear model to separate both classes in the original dataset, after mapping into higher dimensional space a perfectly fitted linear model could be obtained.