Kernel Methods

Rohit Singhatwadia

May, 2018

What are Kernels?

- A kernel is a function that for all x, z ϵ X satisfies

$$k(x, z) = \langle \phi(x), \phi(z) \rangle$$

where,

$$\phi : x \epsilon R^n \mapsto \phi(x) \epsilon F \subset R^N$$

Why do we need Kernels?

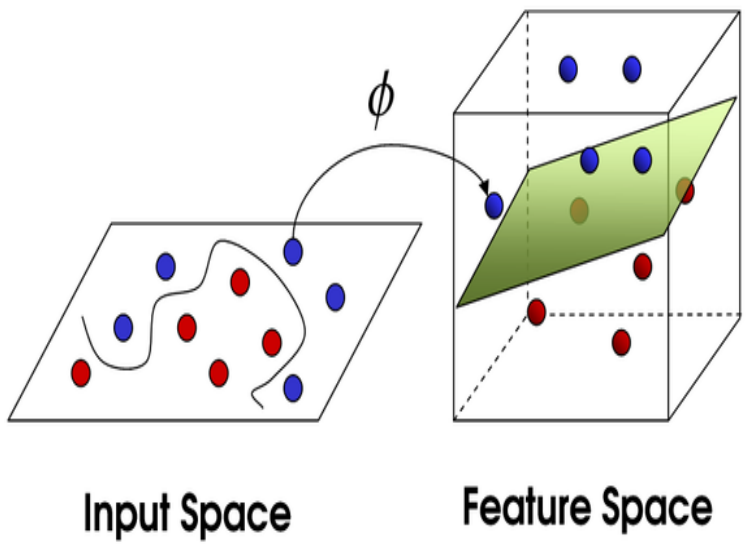
- In Machine Learning most of the theory and algorithms developed are good in handling linearly separable data.
- However, most of the data analysis problem require methods capable of handling nonlinear separable data.
- To handle such data we map the features of the data to a new feature space (high-dimensional).
- In this new feature space our data should be linearly separable.Now we can apply our traditional machine learning algorithm to it.

Transformation into High dimension space

- We will consider an embedding map

$$\phi : x \epsilon R^n \mapsto \phi(x) \epsilon F \subset R^N$$

The choice of the map ϕ aims to convert the nonlinear relations into linear ones.



The Kernel Trick

- After mapping it to the higher dimensions what we really need is the dot product between the features in the higher dimension space.
- However if the dimension of the new feature space i.e. N is large than computing the dot product can be a expensive operation both in terms of time and space(to store the higher dimension features).
- The inner products can, however, sometimes be computed more efficiently as a direct function of the input features, without going to the higher dimension.
- This is where kernels help us. They give us the dot product directly without going to the higher space.We can now directly compute the dot product in the higher dimension by our original features. This is known as the kernel trick.

Example

- Suppose we have 2 vectors $x = (1, 2, 3) y = (4, 5, 6)$.
- We wish to compute there dot product in some higher dimensional space.Let the features in that space be $\phi(x) = (x1x1, x1x2, x1x3, x2x1, x2x2, x2x3, x3x1, x3x2, x3x3)$.
- Our vectors are $\phi(x) = (1, 2, 3, 2, 4, 6, 3, 6, 9)$ and $\phi(y) = (16, 20, 24, 20, 25, 30, 24, 30, 36)$.
- Now finally there dot product is $\langle \phi(x), \phi(y) \rangle = 16 + 40 + 72 + 40 + 100 + 180 + 72 + 180 + 324 = 1024$.
- Now we use kernels to compute these directly $k(x, y) = (\langle x, y \rangle)^2 = (4 + 10 + 18)^2 = 1024$.

Hilbert Space

- A vector space X is an inner product space if there exists a real-valued symmetric bilinear map, that satisfies

$$\langle x, x \rangle \geq 0$$

  The bilinear map is known as the inner, dot or scalar product.
- A Hilbert Space F is a inner product space which follows two additional properties completeness and separability.

Hilbert Space: Completeness and Separability

- Completeness states that every Cauchy sequence(a sequence whose elements become closer to each other as sequence progress) of elements of F converges to a element h ϵ F.
- Separability states that for any $\epsilon > 0$ there is a finite set of elements $h_1 \ldots h_N$ of F such that for all h ϵ F

$$min||h_i - h|| < \epsilon$$

.

Hilbert Space: Completeness and Separability

- The reason for the importance of the properties of completeness and separability is that together they ensure that Hilbert spaces are similar to R^n.
- Intuitively, by completeness we mean that there are no missing points from it. For example the set of rational number is incomplete as $(2)^{1/2}$ is missing from it. And a Cauchy sequence of rational numbers can converge to it.
- Any space which is finite or countably infinite is separable as the whole space is a countable dense subset of itself. Example of an uncountable separable space is the real line, in which the rational numbers form a countable dense subset.A simple example of a space which is not separable is a uncountable discrete space.

Hilbert Space

- Each element in this space is a function. This function is a linear combination of our input features.
- This is similar to our weight vector as it is a linear combination of the feature vectors of the training points.
- Thus finding the weight vector is equivalent to finding a equivalent element of this space.
- A reproducing kernel Hilbert space (RKHS) is a Hilbert space of functions in which point evaluation is a continuous linear function .Thus this guarantee that the function element that we are looking for is linear.

Gram Matrix

- Suppose we have l input vectors $x_1 \dots x_l$ and we are using a kernel k to evaluate the dot product in a feature space with feature map ϕ than we define gram matrix as

$$G_{ij} = \langle \phi(x_i), \phi(x_j) \rangle = k(x_i, x_j)$$

- This matrix is symmetric. And it contain all the information needed to find the pairwise distance in the data set.
- But it looses the information of the original data set w.r.t origin. We loose the information like the angle at which the feature is located w.r.t origin .This signify that gram matrix are rotation invariant.

Positive semi-definite Kernels

- A symmetric matrix K of (n x n) is positive semi-definite, if follows for every column vector of n real numbers.

$$c' K c \geq 0$$

or

$$\sum_{i,j} c_i c_j K_{ij} \geq 0$$

for all $c_i \epsilon R$.

- Let X be a nonempty set. A function k : $X * X \mapsto R$ which give rise to a positive semi-definite gram matrix for all values of $x \epsilon X$ is called positive semi-definite kernel.

Construction of the Reproducing Kernel Hilbert space

- A function

$$k : XxX \mapsto R$$

which is either continuous or has a finite domain, can be decomposed

$$k(x, y) = \langle \phi(x), \phi(y) \rangle$$

into a feature map ϕ into a Hilbert space F applied to both its arguments followed by the evaluation of the inner product in F if and only if it satisfies the finitely positive semi-definite property.

Mercers theorem

- Mercers theorem is also used to construct a feature space for a valid kernel just like the last section.
- Suppose K is a continuous symmetric function $K : R^m * R^m \mapsto R$, than there exist a inner product space H and a mapping $\phi : R^m \mapsto H$ so that

$$K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$$

if for all square-integrable function g

$$\int \int K(x_1, x_2) g(x_1) g(x_2) dx_1 dx_2 \geq 0$$

Example of Mercer Kernel

- Let our kernel be a positive constant function $K(x_1, x_2) = c$. Now we want that

$$\int \int cg(x_1)g(x_2)dx_1dx_2 \geq 0$$

$$c \int \int g(x_1)g(x_2)dx_1dx_2 \geq 0$$

- They both are the same integerals. So we can write it as

$$c(\int g(x_1)dx_1)^2 \geq 0$$

So, now we can say that this kernel satisfies mercer theorem.

Properties of Kernels

- $k(x, y) = k_1(x, y) + k_2(x, y)$
- $k(x, y) = ak_1(x, y)$
- $k(x, y) = f(x)f(y)$
- $k(x, y) = k_1(x, y)k_2(x, y)$

Kernel Construction

- Kernels constructed using the properties of kernels from last slide.
- $k(x, y) = p(k_1(x, y))$ where p(x) is a polynomial
- $k(x, y) = exp(k_1(x, y))$
- $k(x, y) = exp(-||x - y||^2/(2\sigma^2))$
- All kernels mentioned are positive semi-definite.

Polynomial kernels

- The derived polynomial kernel for a kernel k_1 is defined as

$$k(x, y) = p(k_1(x, y))$$

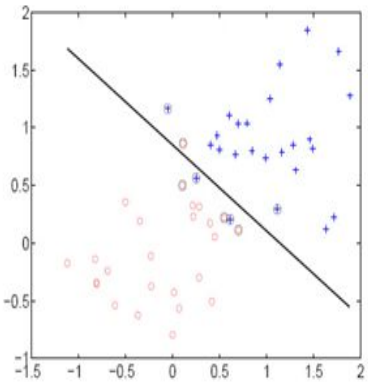
- Frequently, it also refers to the inhomogeneous polynomial

$$k(x, y) = (\langle x, y \rangle + R)^d$$

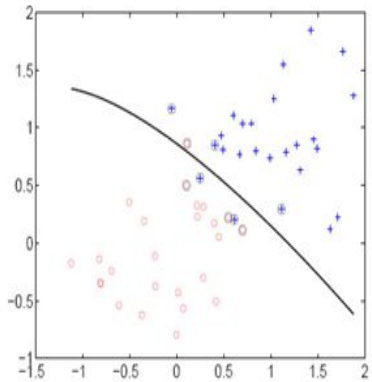
  We can expand it using Binomial Theorem. Features for each component in the sum together form the features of the whole kernel.
- The most common degree is 2 since larger degree tend to overfit.

Polynomial Kernel

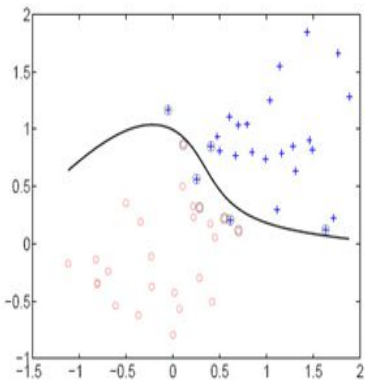
Polynomial kernel: $K(\mathbf{x}, \mathbf{y}) = (c + \mathbf{x} \cdot \mathbf{y})^d$



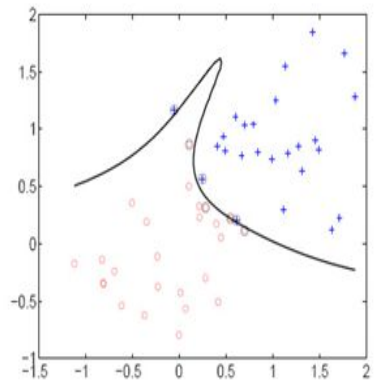
linear



2^{nd} order polynomial



4^{th} order polynomial



8^{th} order polynomial

Radial basis function kernel

- For $\sigma > 0$, the Gaussian kernel is defined as

$$k(x, y) = exp(-||x - y||^2/(2\sigma^2))$$

We can also write it as

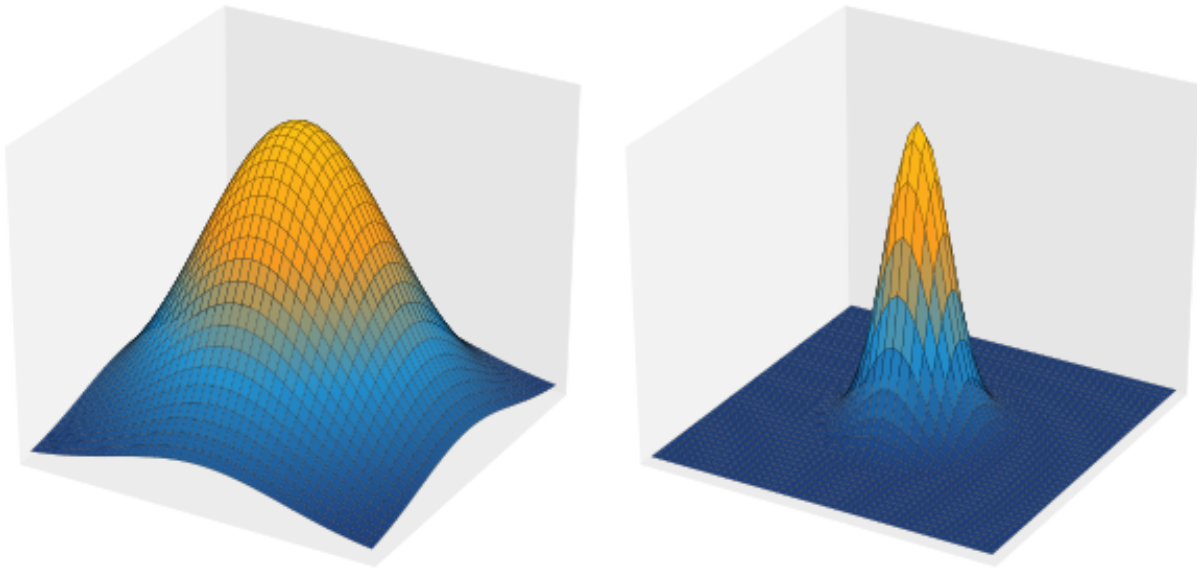
$$k(x, y) = exp(-\gamma||x - y||^2)$$

- Now this kernel signifies a infinite dimensional feature space i.e it works as if we have transformed our features in a infinite dimensional space. Just perform the expansion of the function and this function can have up to infinite length. Although the higher order terms falls very quickly.
- This specify a similarity measure as when x is very close to y the kernel will attain its maximum value and when x is far away the kernel value will tend towards zero. The shape however is smooth.

Radial basis function kernel

- A small gamma means a Gaussian with a large variance so the influence of x_j is more, i.e. even if a point is not close, it will be assigned a good similarity value.
- If gamma is large, then variance is small implying that only near by points are assigned high similarity value. Technically speaking, large σ leads to high bias and low variance models, and vice-versa. Also the scaling by γ occurs the same amount in all directions.

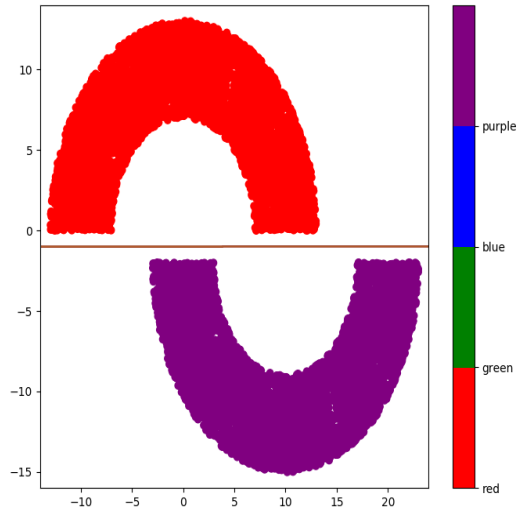
Radial basis function kernel

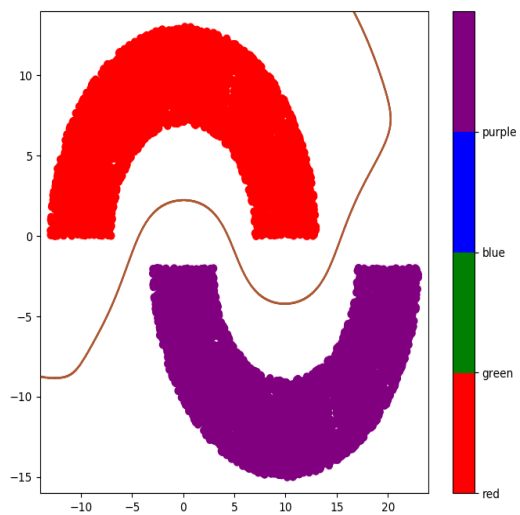


Radial basis function kernel

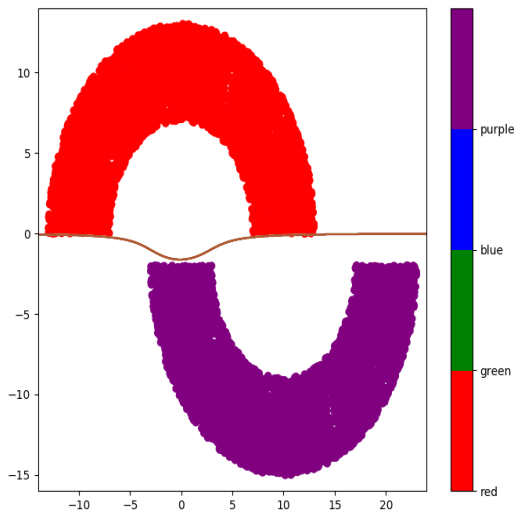
- It is a stationary kernel, which means that it is invariant to translation. Suppose you are computing $K(x, y)$. A stationary kernel will yield the same value $K(x, y)$ for $K(x + c, y + c)$, where c may be vector.For the RBF, this is done by working on the difference of the two vectors. Also note that the linear kernel does not have this property.

Double Moon1 Classifier Image: Linear and RBF kernel

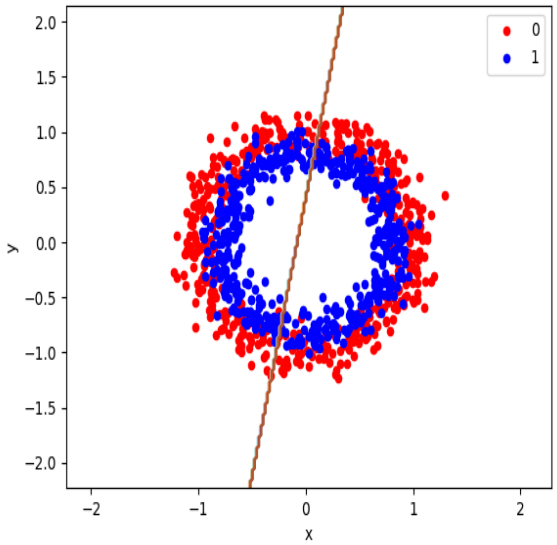


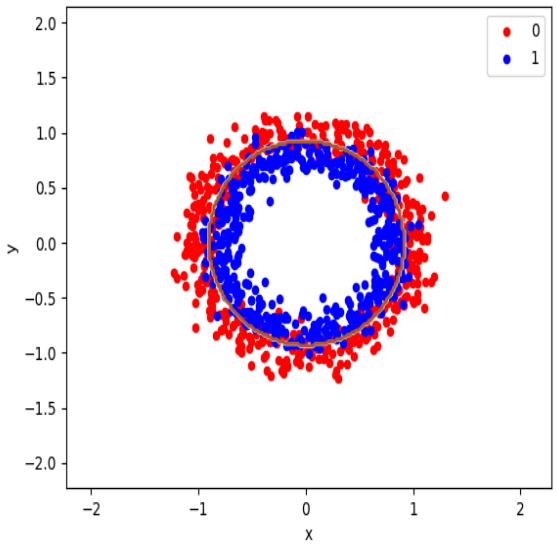


Double Moon1 Classifier Image: Polynomial Kernel

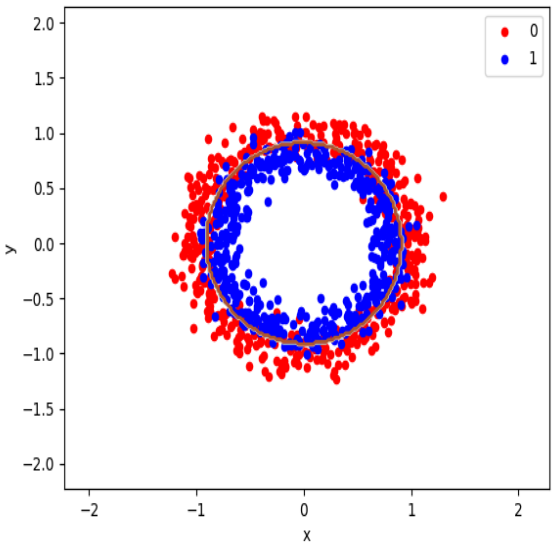


Circle2 Classifier: Linear and RBF Kernel

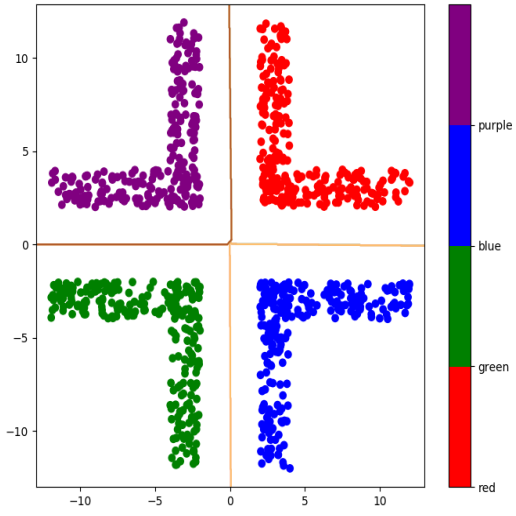


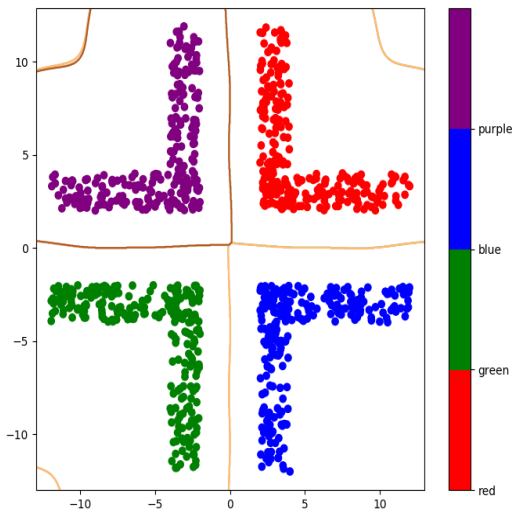


Circle2 Classifier: Polynomial Kernel

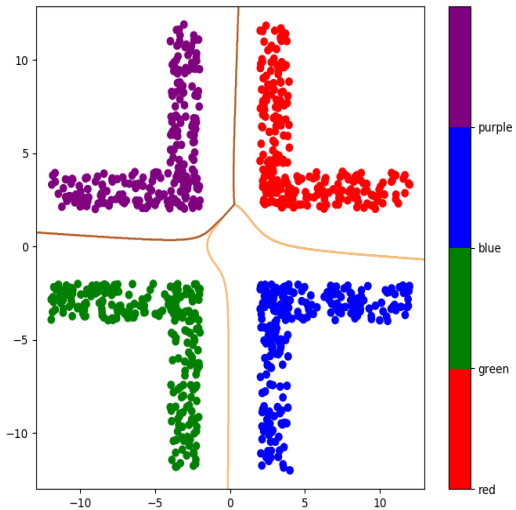


Corners Dataset : Linear and RBF Kernel

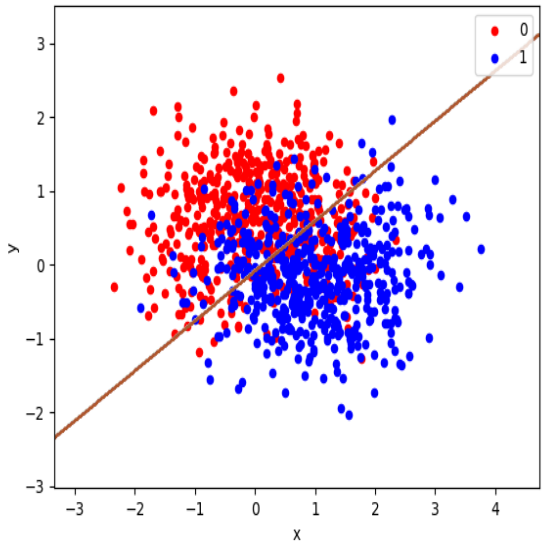


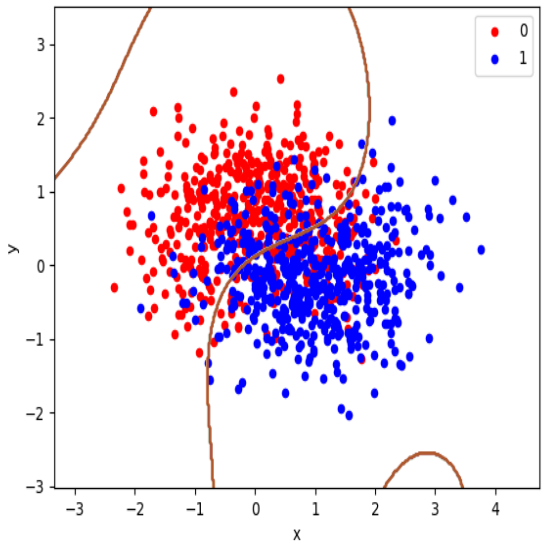


Corner Dataset : Polynomial Kernel

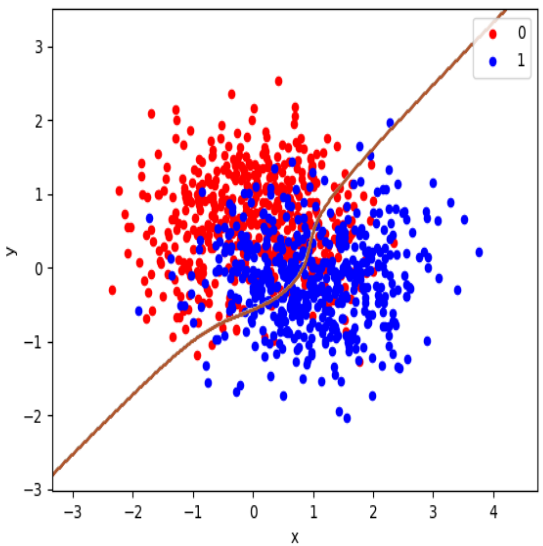


DoubleMoon4: Linear and RBF kernel





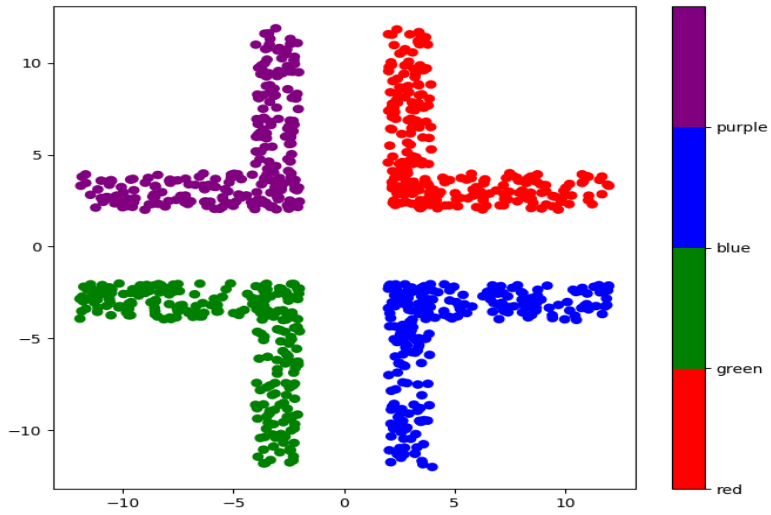
DoubleMoon4: Polynomial Kernel

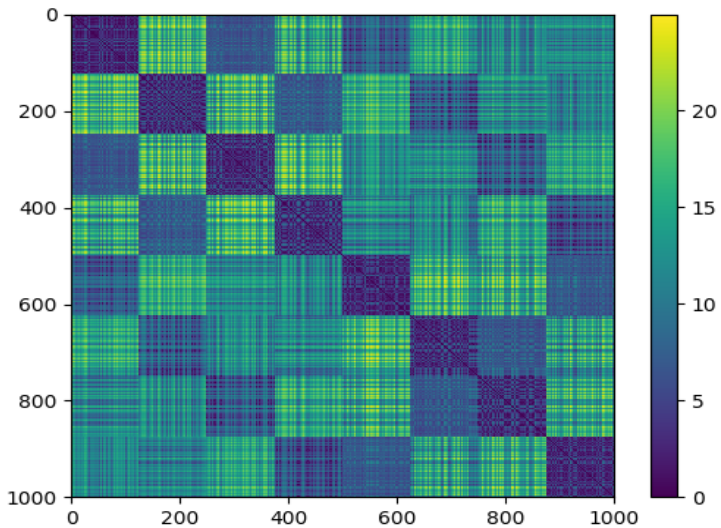


Results: Accuracy of Generated datasets

Dataset	Linear Kernel	RBF Kernel	Polynomial Kernel
DoubleMoon1	100	100	100
DoubleMoon2	100	100	100
Corners	100	100	100
TwoSpirals	64.45	100	51.8
CircleData	48	81.33	97.67
DoubleMoon3	85.33	86	86.33
Blob	85	86	86
CircleData2	49.67	84.66	83.33
DoubleMoon4	78.66	78.33	73.66
SwissRoll	76.458	97.916	97.916

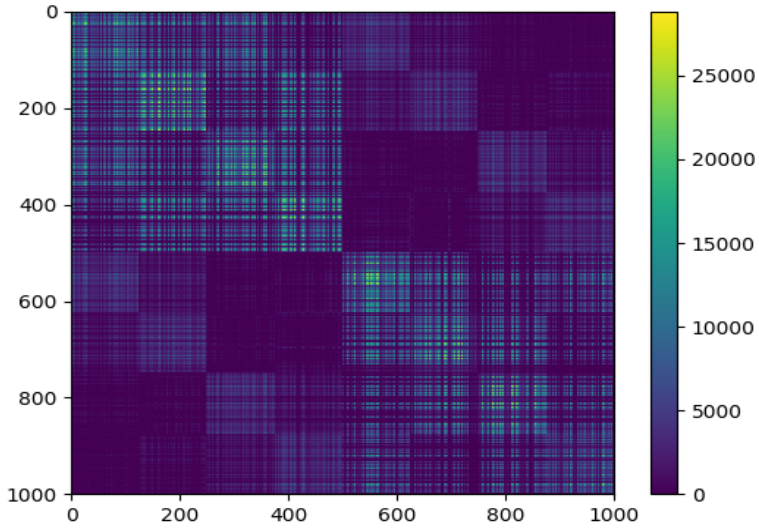
Corner Dataset and euclidian distance matrix

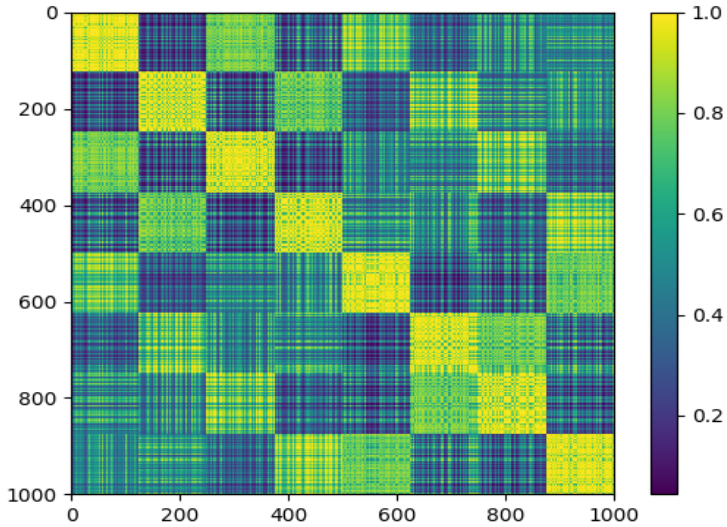




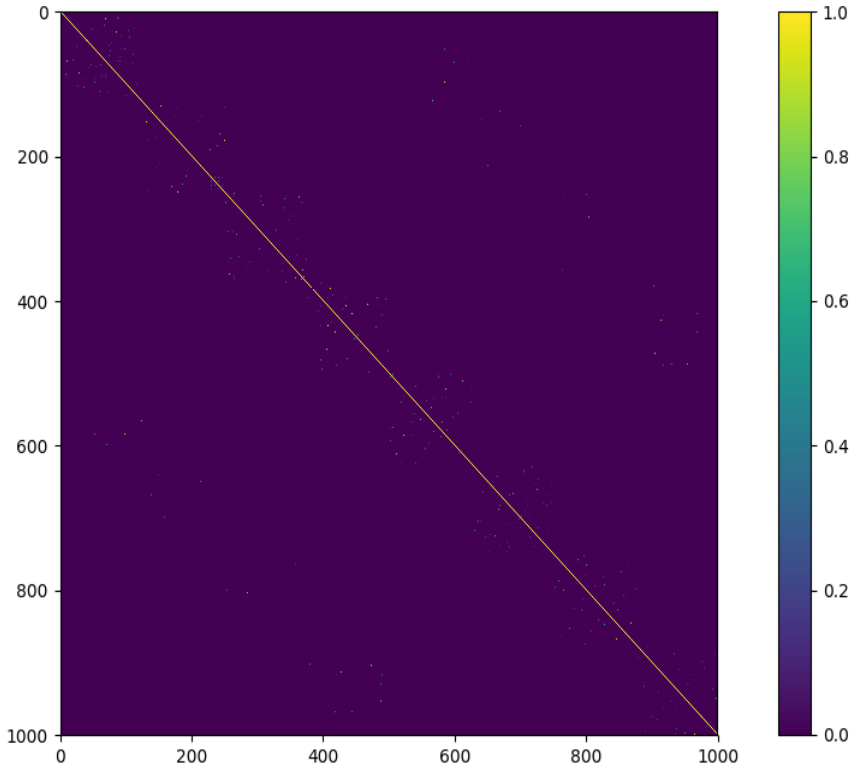
Euclidian Distance Matrix

Corners Polynomial and RBF(10) matrix





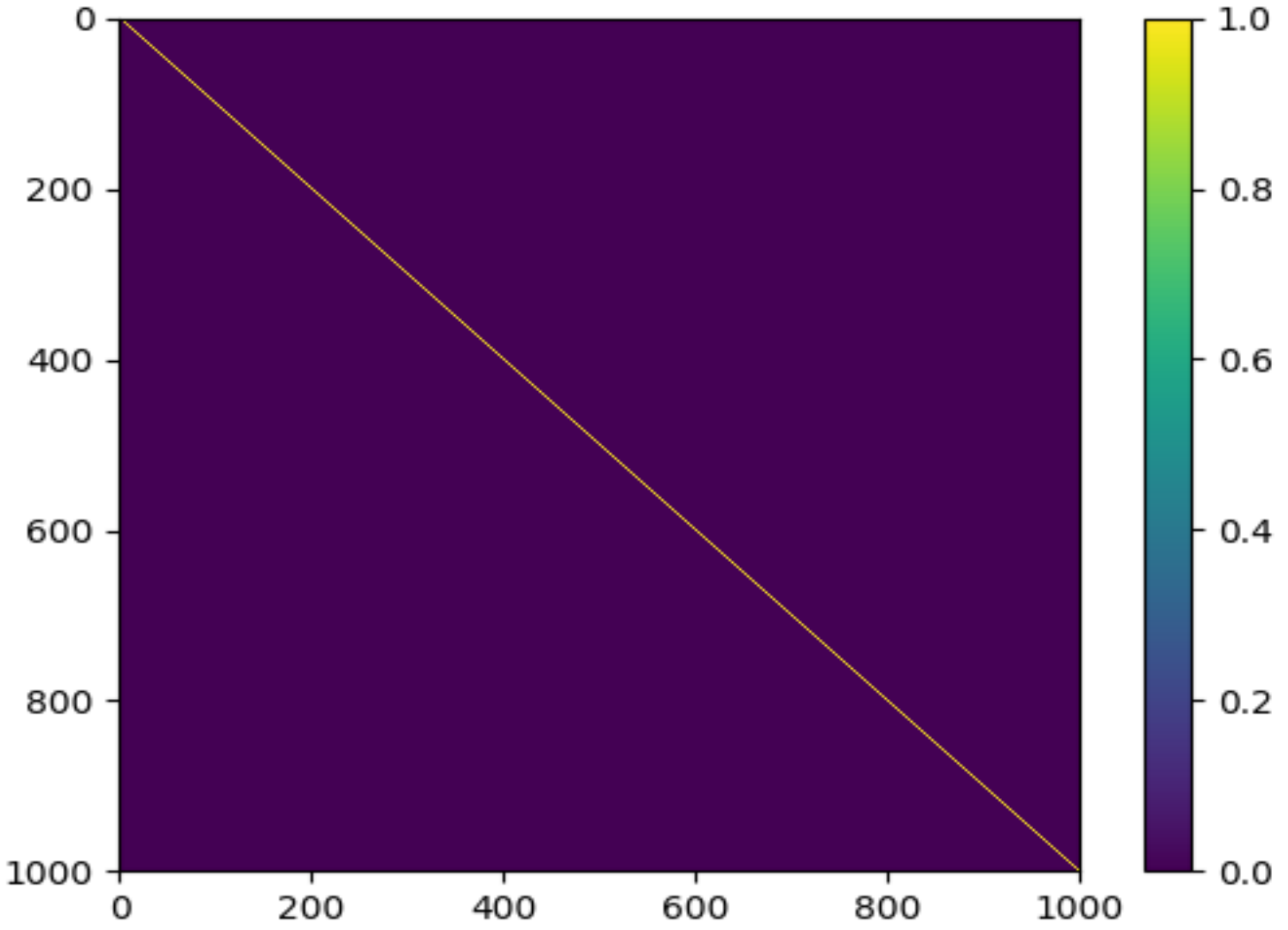
Corners:RBF(0.1)



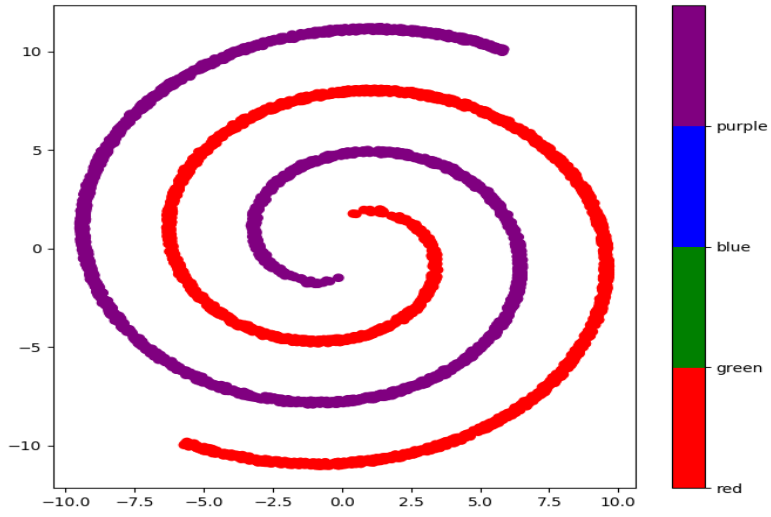
▶

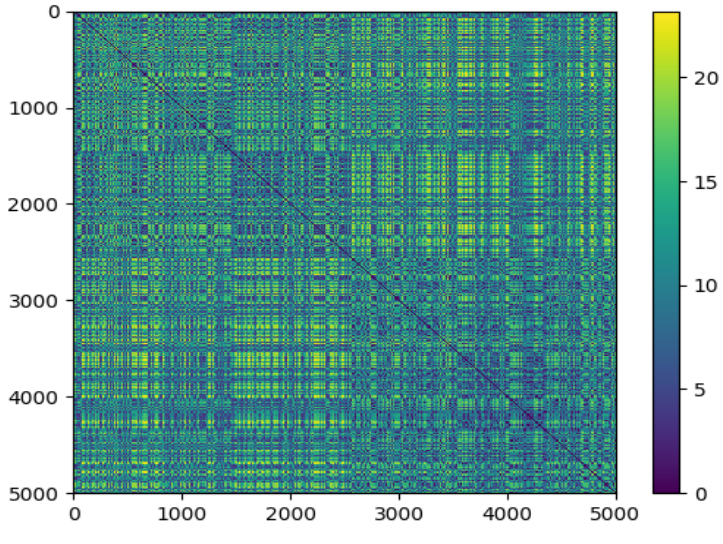
Corners Dataset:RBF(0.001)

-

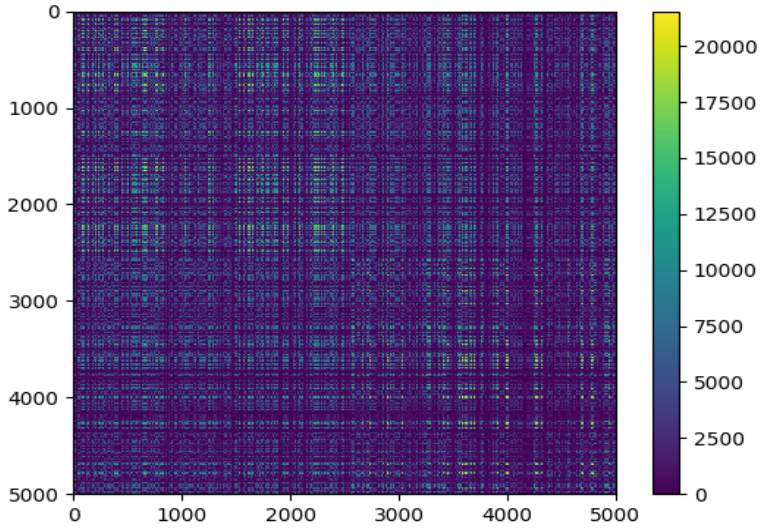


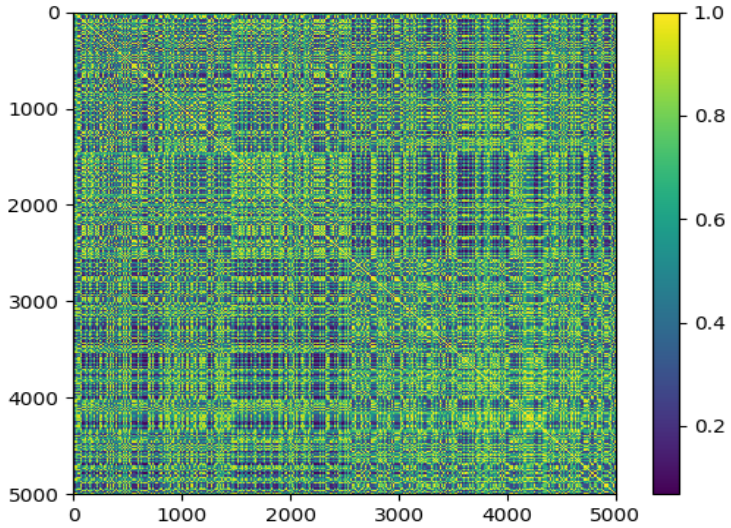
Two Spirals Dataset and euclid distance matrix





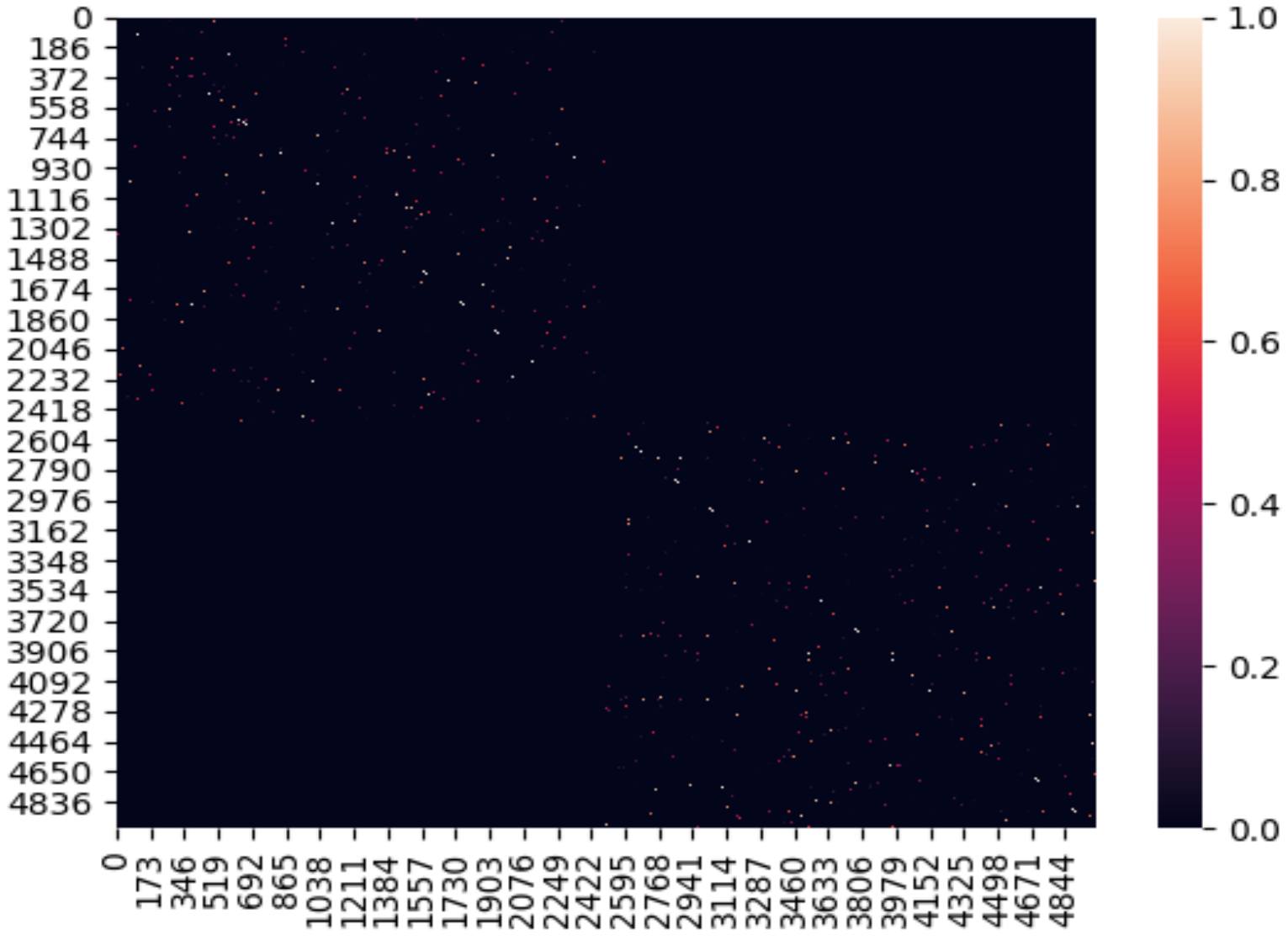
Two Spirals Polynomial and RBF(10) matrix





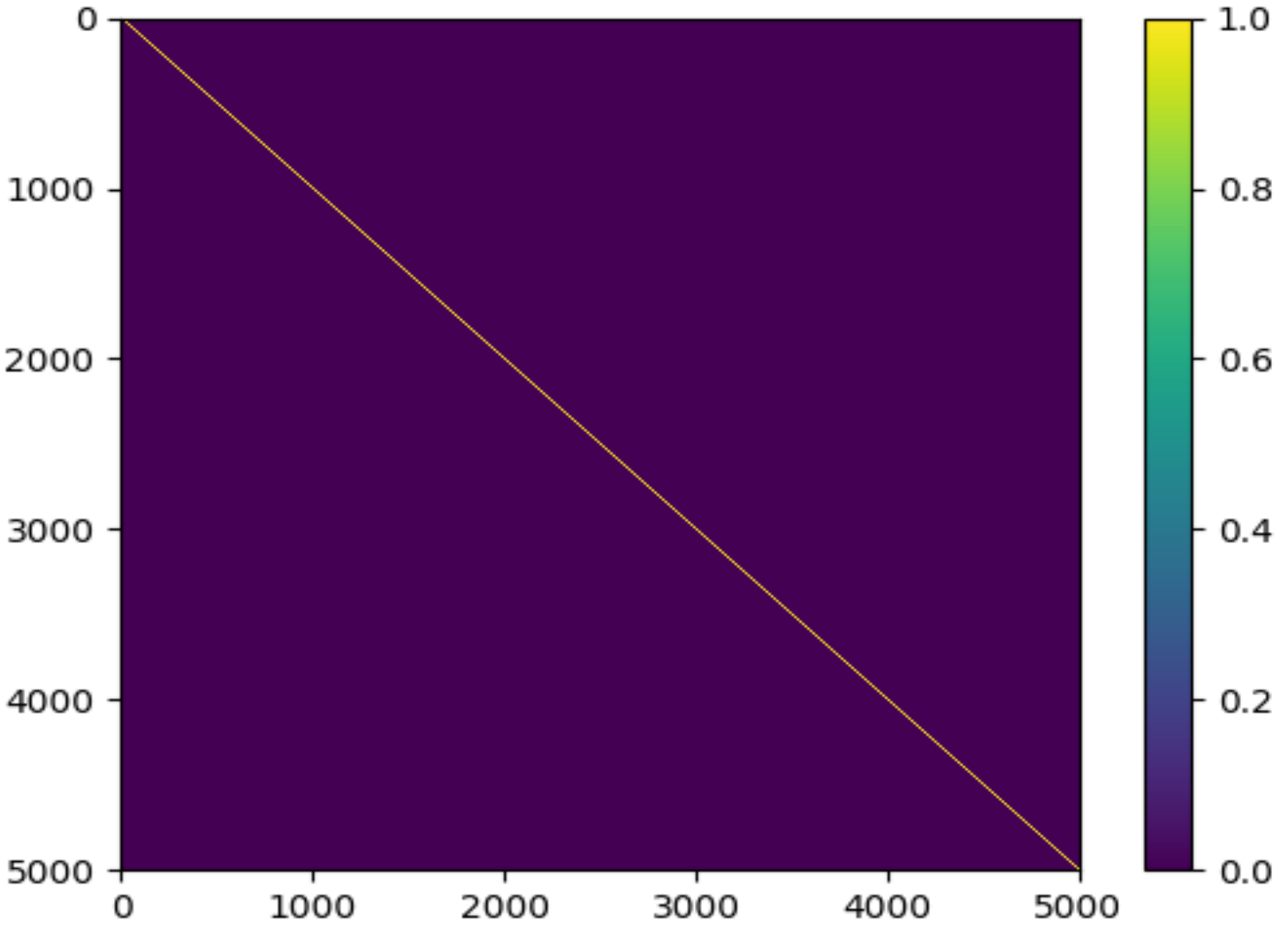
Two Spirals RBF(0.1)

-



Two Spirals RBF(0.001)

-



Results:Accuracy for Image Datasets

- **GHIM**

Linear Kernel	RBF kernel	Polynomial Kernel
79.067	82.67	80.2
88.43	85.16	82.73

- **C1K**

Linear Kernel	RBF Kernel	Polynomial Kernel
82.33	86.33	78
87.33	81.33	80.66

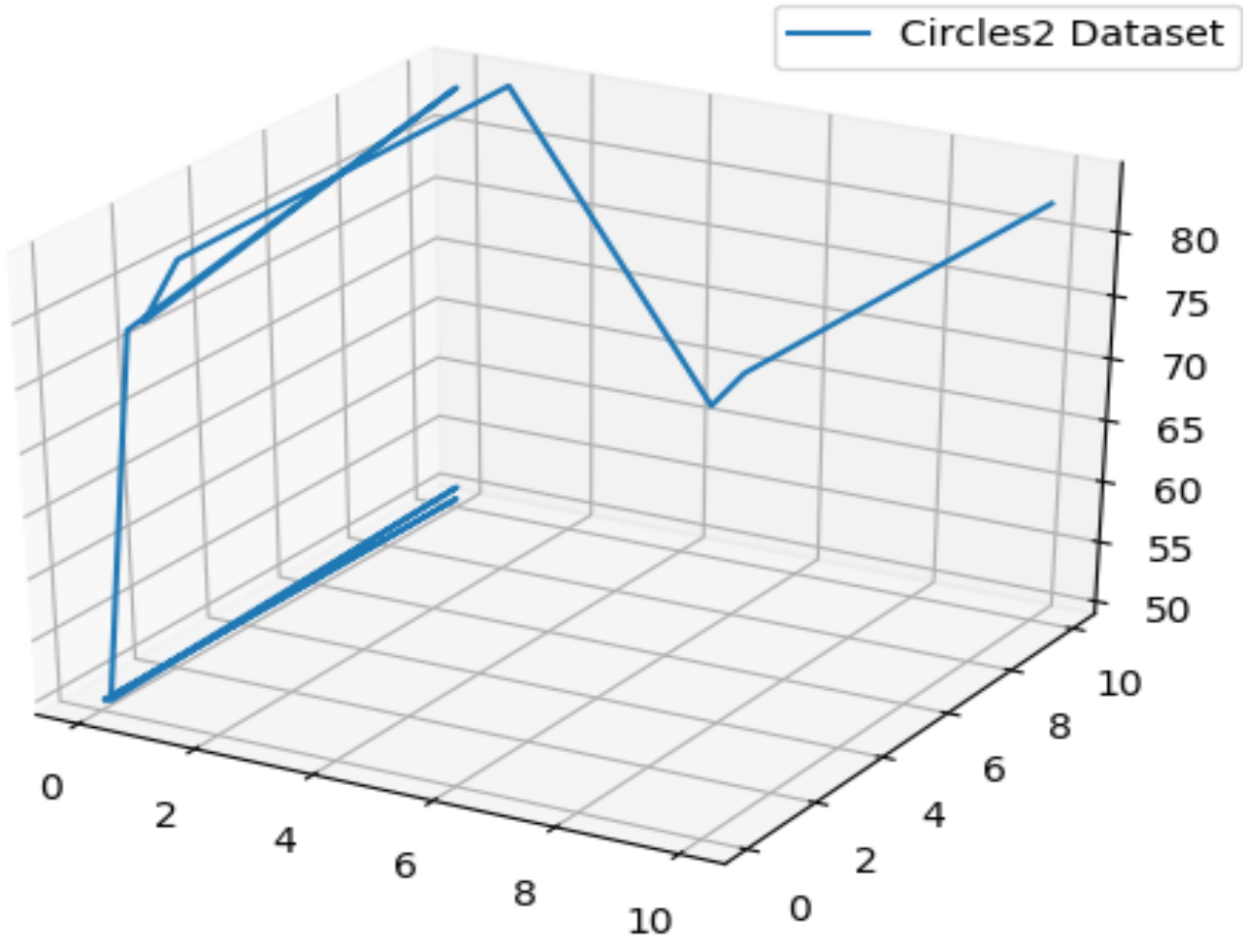
- **CIFAR10**

Linear Kernel	RBF Kernel	Polynomial Kernel
93.48	93.51	93.42
93.45	93.46	93.47

Results : Accuracy for different parameters for Polynomial kernel

Dataset	Degree 2	Degree 3	Degree 4	Degree 5
Circle2	83.33	54	83	61.3
DoubleMoon4	62	73.67	59.33	71.67
Blob	84	86	84.33	84.66

Accuracy for different parameters for RBF kernel:Circles2 Dataset

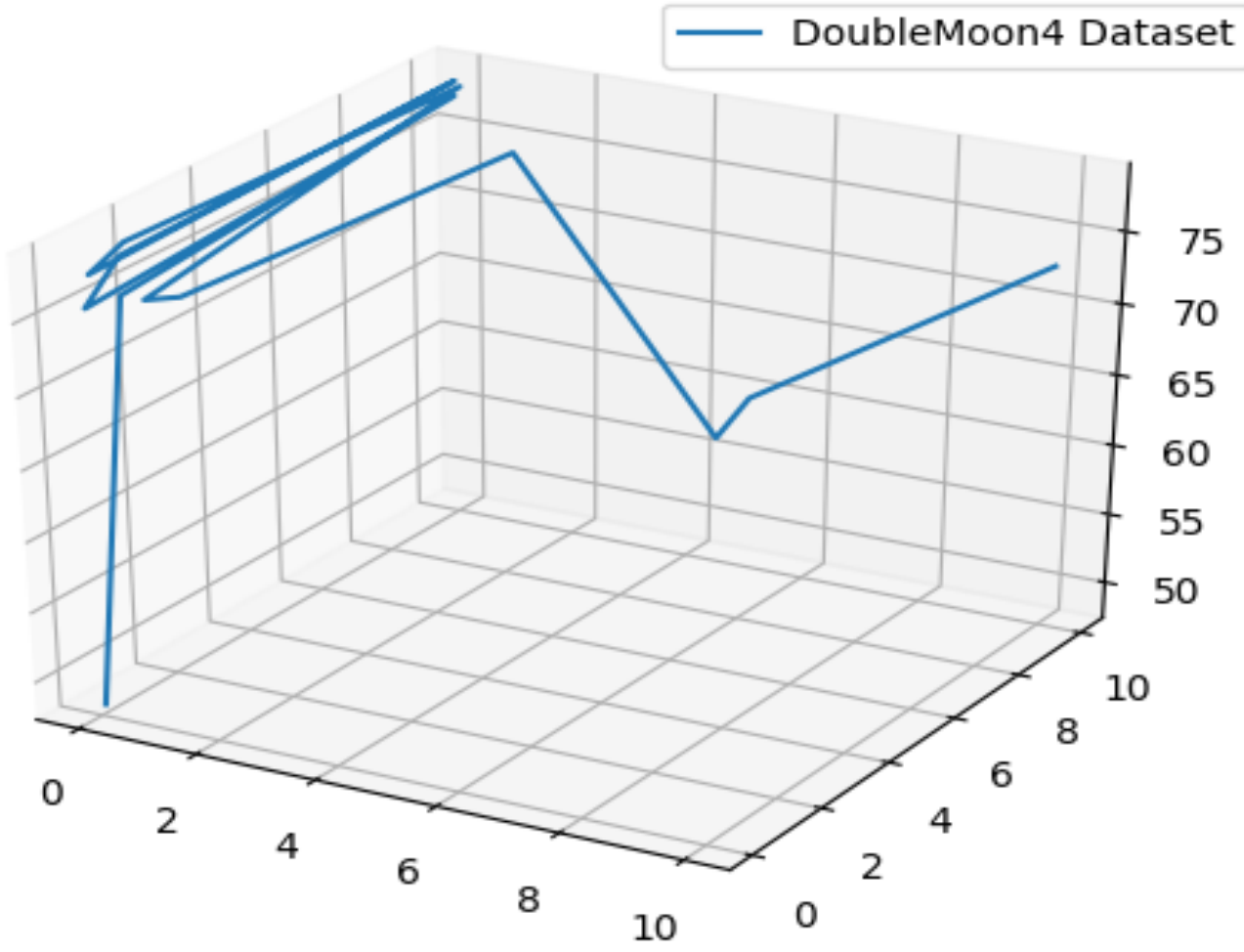


Accuracy for different parameters for RBF kernel

- **Circle2 Dataset**

Gamma	C	Accuracy
0.001	0.1	49.67
0.001	1	49.67
0.001	10	49.67
0.01	0.1	49.67
0.01	1	49.67
0.01	10	50.67
0.1	0.1	49.67
0.1	1	77.33
0.1	10	83.33
1	0.1	80.66
1	1	84
1	10	84.66
10	0.1	82.33
10	1	83.33
10	10	82.66

DoubleMoon4 Dataset



- **DoubleMoon4 Dataset**

Gamma	C	Accuracy
0.001	0.1	48
0.001	1	75.33
0.001	10	77.6
0.01	0.1	75.67
0.01	1	78
0.01	10	78.66
0.1	0.1	78
0.1	1	79
0.1	10	78.33
1	0.1	77
1	1	76
1	10	74.33
10	0.1	75
10	1	76.33
10	10	73

References

Thomas Hofmann, Bernhard Schlkopf and Alexander J. Smola [*KERNEL METHODS IN MACHINE LEARNING*]. The Annals of Statistics, 36(3):1171-1220, 2008.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, United States of America, 2004.