Kernel Methods

# Kernel Functions
# Kernel Ridge Regression

# Reproducing Kernel Hilbert Spaces (RKHS) I

## Properties of RKHS:

1. **Normed Vector Space:**
   - RKHS is equipped with a well-defined norm $\|\cdot\|$, which satisfies the usual norm properties.
2. **Completeness (Banach Space):**
   - The space is complete, meaning it behaves nicely in the sense that every Cauchy sequence in the space converges to an element within the space.
3. **Inner Product Space:**
   - RKHS has an inner product $\langle\cdot,\cdot\rangle$, and the norm is induced by this inner product, i.e. $||\mathbf{x}|| = \sqrt{\langle\mathbf{x},\mathbf{x}\rangle}$. For instance, in $\mathbb{R}^N$, the inner product between two vectors $\mathbf{x}$ and $\mathbf{y}$ is $\mathbf{x}^T\mathbf{y}$.

# Reproducing Kernel Hilbert Spaces (RKHS) II

4. **Reproducing Kernel:**
   - There exists a kernel function $\kappa(\cdot, \mathbf{x})$ that belongs to $\mathbb{H}$ for each $\mathbf{x} \in \mathcal{X}$ and possesses the reproducing property:

$$f(\mathbf{x}) = \langle f, \kappa(\cdot, \mathbf{x}) \rangle \quad \text{for all } f \in \mathbb{H} \text{ and } \mathbf{x} \in \mathcal{X}.$$

   - This property implies that the evaluation of any function $f$ at any point $\mathbf{x}$ can be represented as an inner product.

# Reproducing Kernel Hilbert Spaces (RKHS) III

**Feature Mapping and Kernel Trick:**

- *Feature Map*: $\phi(\mathbf{x}) := \kappa(\cdot, \mathbf{x})$ maps each point in $\mathcal{X}$ to a function in $\mathbb{H}$.
  - Embeds the input data into a higher-dimensional space (potentially infinite-dimensional)
  - Linear algorithms (like inner products) can be applied to solve nonlinear problems in the original space
- The *Kernel Trick* allows inner products in $H$ to be computed as:

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \kappa(\mathbf{x}, \mathbf{y})$$

This simplifies computations by replacing inner product calculations with kernel evaluations.

# Example

**Dimensional Mapping:**

- Maps two-dimensional vectors from $\mathbb{R}^2$ into a three-dimensional space $\mathbb{R}^3$.
- Transformation:

$$\phi(\mathbf{x}) = [x_1^2, \sqrt{2}x_1 x_2, x_2^2]$$

**Inner Product in Transformed Space:**

- Inner product between $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$ in $\mathbb{R}^3$:

$$\phi(\mathbf{x})^T \phi(\mathbf{y}) = x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 = (\mathbf{x}^T \mathbf{y})^2$$

- Shows that the inner product in $\mathbb{R}^3$ is the square of the inner product in $\mathbb{R}^2$.

# Examples of Kernel Functions I

- **Gaussian Kernel:**

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right)$$

- **Polynomial Kernels:**
  - *Homogeneous:* $\kappa(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y})^r$
  - *Inhomogeneous:* $\kappa(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + c)^r$

- **Laplacian Kernel:**

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp(-t\|\mathbf{x} - \mathbf{y}\|)$$

**Kernel Construction:**

- Kernels can be combined, modified, or constructed to fit specific needs.

# Kernel Ridge Regression in RKHS I

## Problem Setup:

- Nonlinear regression task modeled as:

$$y_n = g(\mathbf{x}_n) + \eta_n$$

- Estimate $f$ as:

$$f(\mathbf{x}) = \sum_{n=1}^{N} \theta_n \kappa(\mathbf{x}, \mathbf{x}_n)$$

## Objective Function:

- Minimize the following loss function:

$$J(\boldsymbol{\theta}) = (\mathbf{y} - \mathcal{K}\boldsymbol{\theta})^T(\mathbf{y} - \mathcal{K}\boldsymbol{\theta}) + C\boldsymbol{\theta}^T\mathcal{K}^T\boldsymbol{\theta}$$

where $\mathcal{K}$ is the kernel matrix, $C$ is the regularization parameter.

# Kernel Ridge Regression in RKHS II

## Solution:

- The solution to the minimization problem is given by:

$$(\mathcal{K}^T\mathcal{K} + C\mathcal{K}^T)\hat{\boldsymbol{\theta}} = \mathcal{K}^T\mathbf{y}$$

$$(\mathcal{K} + CI)\hat{\boldsymbol{\theta}} = \mathbf{y}$$

## Prediction:

- For an unknown vector $\mathbf{x} \in R^l$, the prediction is:

$$\hat{\mathbf{y}} = \hat{\boldsymbol{\theta}}^T\kappa(\mathbf{x})$$

$$\hat{\mathbf{y}}(\mathbf{x}) = \mathbf{y}^T(\mathcal{K} + CI)^{-1}\kappa(\mathbf{x})$$

# Exercise 12 I

1. Generate points for two classes
2. Apply non-linear transformation
   - `PHI = np.vstack((X[:, 0]**2, 2**0.5*X[:, 0]*X[:, 1], X[:, 1]**2)).T`
3. Calculate the homogeneous polynomial kernel matrix
   - 
     ```
     for i in range(2*N):
         for j in range(2*N):
           K[i, j] = (X[i, :]  @ X[j, :])**r
     ```
4. Apply representer theorem with $\boldsymbol{\theta} = \mathbf{0}$

# Exercise 12 II

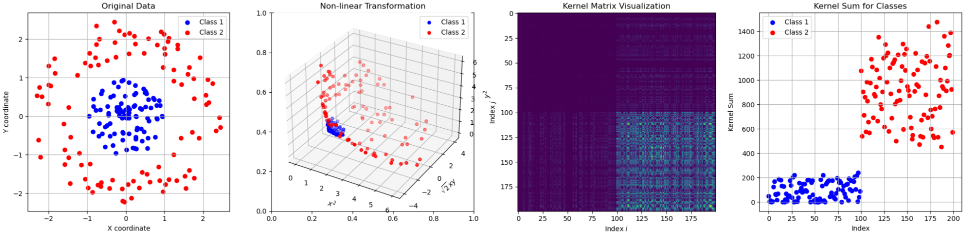


Figure: Homogeneous polynomial kernel

# Exercise 12 III

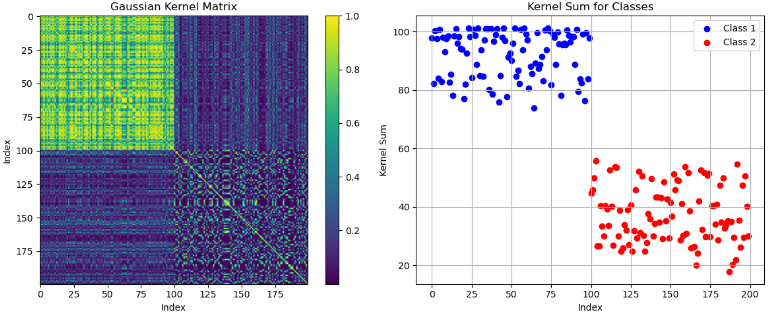


Figure: Gaussian kernel $\sigma = 1$

# Exercise 12 IV

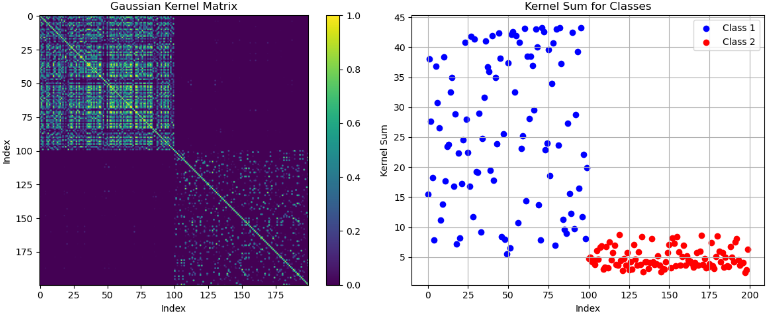


Figure: Gaussian kernel $\sigma = 0.1$