COMP 4432 Machine Learning

Lesson 5: Support Vector Machines

Agenda

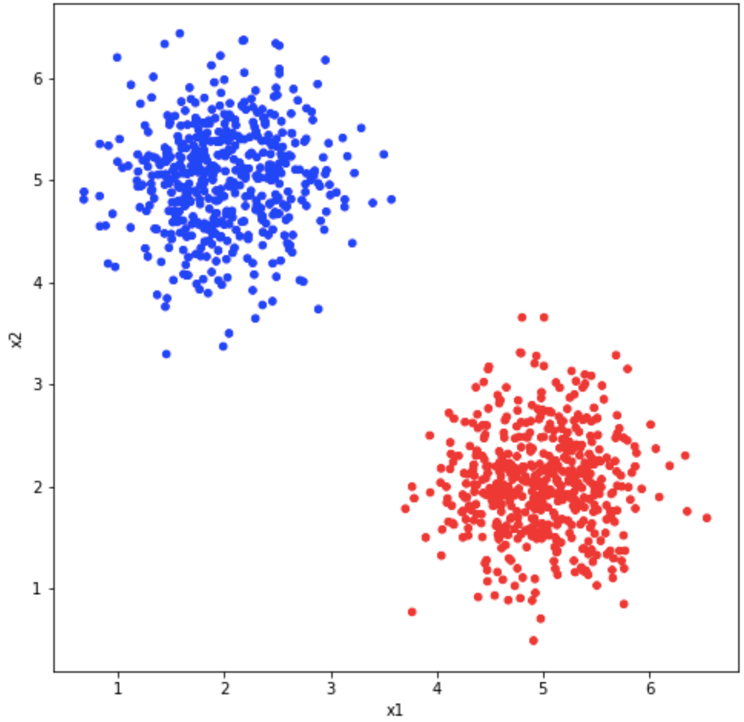
- Assignment 3
- Introduction to SVM
- Maximum Margin with Linearly Separable
- Constraint Problem
- Transformations
- Kernels
- Implementation

Assignment 3

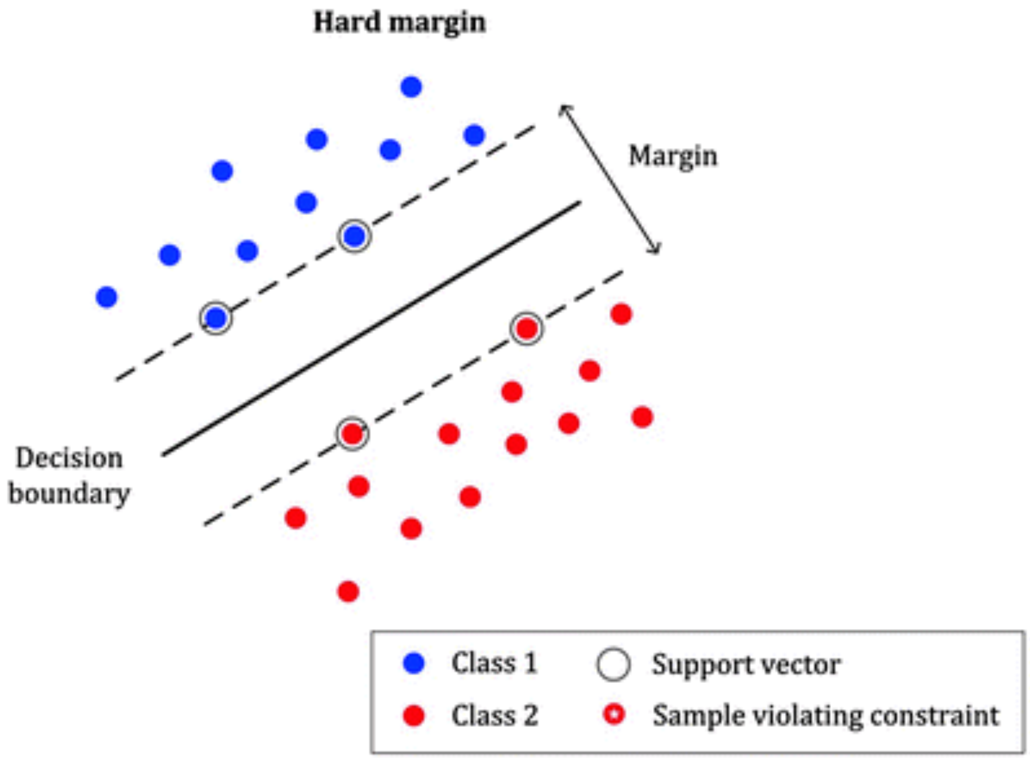
- Included updated instructions for Part 2

Introduction to SVMs

- Identifies boundary between *linearly* separable classes

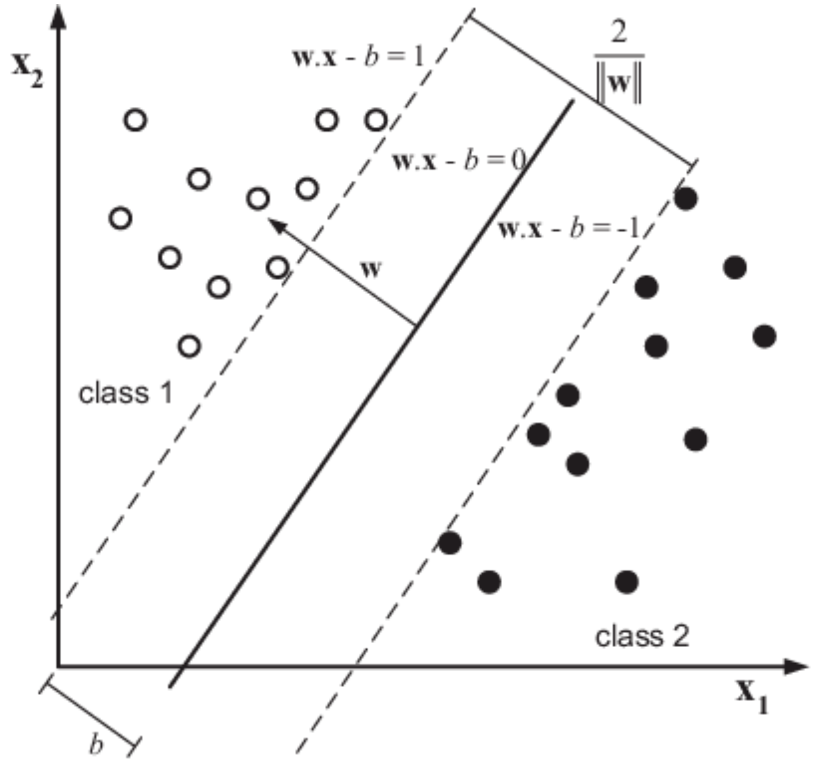


Maximize Margin



- Only support vectors matter

Maximum Hard Margin

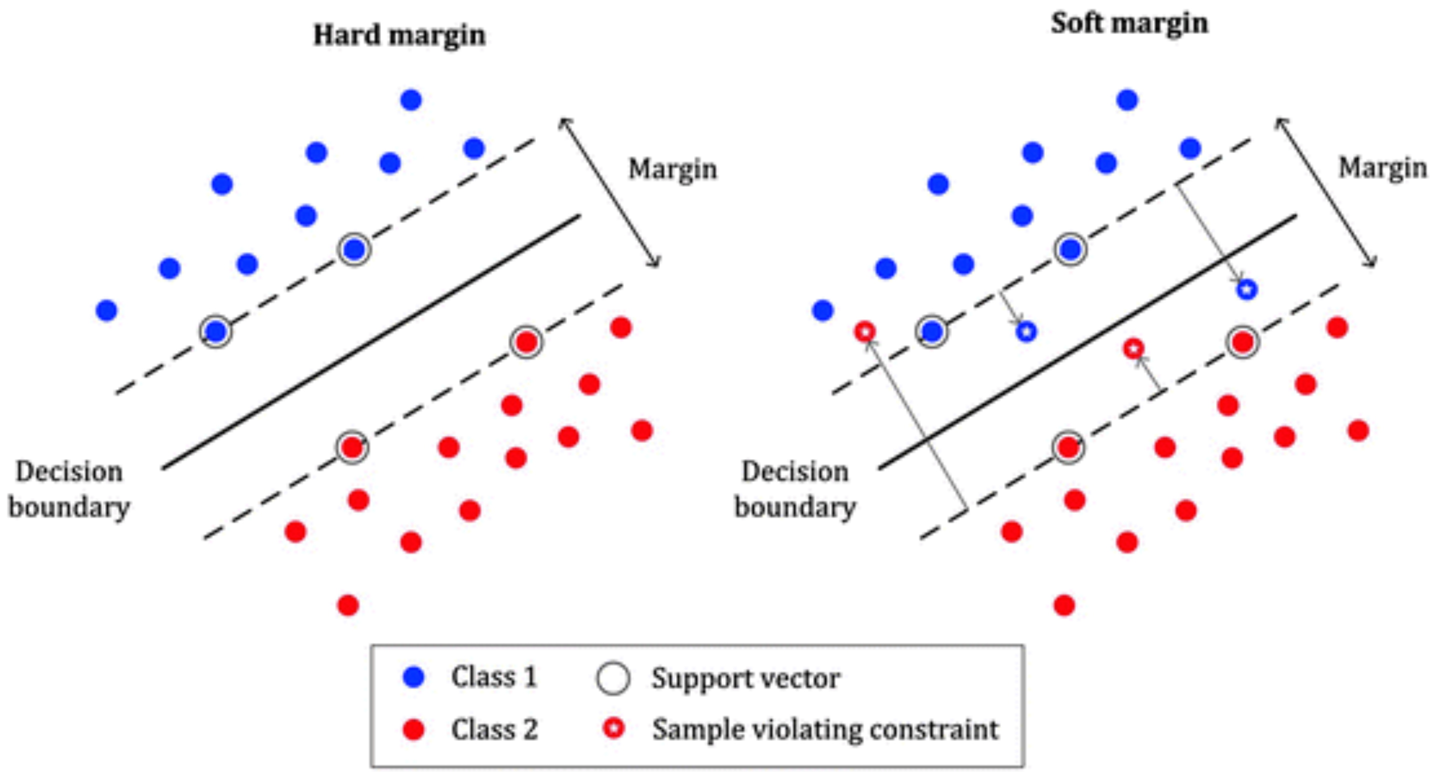


Constraint Problem

- Hard margin

$$
\begin{aligned}
\min_{\mathbf{w},b} \quad & \tfrac{1}{2}\mathbf{w}^t\mathbf{w} \\
\text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 \text{ for } i = 1, 2, \ldots, n
\end{aligned}
$$

Maximize Margin



Constraint Problem

- Soft margin
  - We'll let some missteps occur
  - Maximize margin while minimizing misclassification
- Demo

$$
\begin{aligned}
\min_{\mathbf{w},b,\xi} \quad & \tfrac{1}{2}\mathbf{w}^t\mathbf{w} + C\textstyle\sum_{i=1}^{n}\xi_i \\
\text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 - \xi_i \text{ for } i = 1, 2, \ldots, n \\
\text{and} \quad & \xi_i \geq 0
\end{aligned}
$$

Constraint Problem

- Dual Problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}\mathbf{w}^t\mathbf{w} \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 \text{ for } i = 1, 2, \ldots, n \end{aligned}$$

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i - b) - 1)$$

Constraint Problem

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i - b) - 1)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = & \quad \mathbf{w} - \textstyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \\ & \quad \mathbf{w} = \textstyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \quad -\textstyle\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\begin{aligned} \min_{\alpha} \quad & \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to} \quad & \alpha_i \geq 0 \text{ for } i = 1, 2, \ldots n \\ \text{and} \quad & \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

Constraint Problem

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i - b) - 1)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = &\quad \mathbf{w} - \textstyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \\ &\quad \mathbf{w} = \textstyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \quad -\textstyle\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\begin{aligned} \min_{\alpha} &\quad \textstyle\sum_{i=1}^{n} \boxed{\alpha_i} - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to} &\quad \alpha_i \geq 0 \text{ for } i = 1, 2, \ldots n \\ \text{and} &\quad \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

Constraint Problem

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i - b) - 1)$$

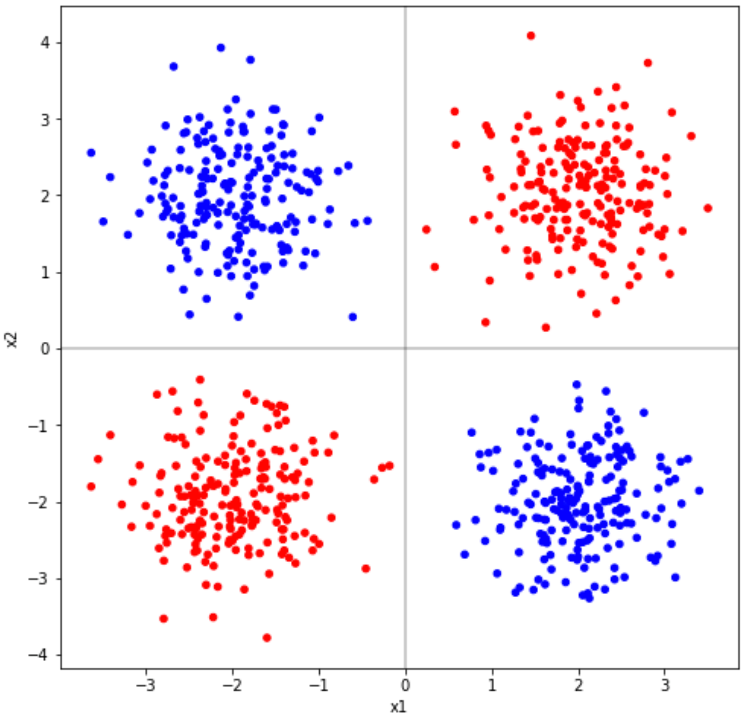
$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= \quad \mathbf{w} - \textstyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \\ & \quad \mathbf{w} = \textstyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \quad -\textstyle\sum_{i=1}^{n} \alpha_i y_i = 0$$

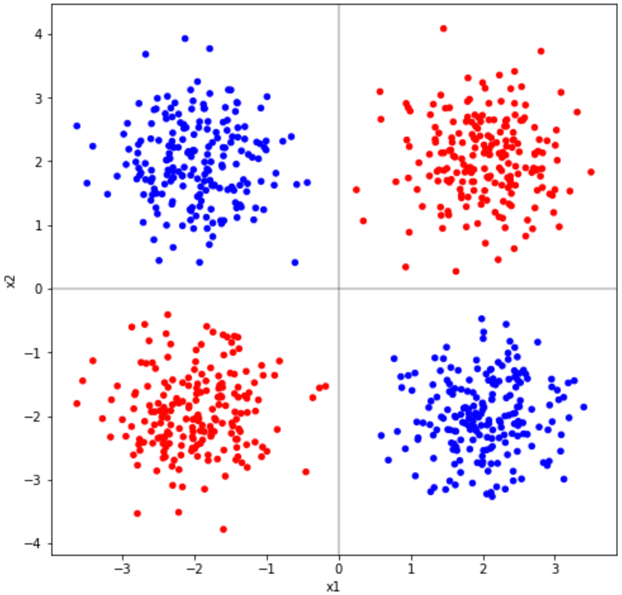
$$\begin{aligned} \min_{\alpha} \quad & \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i \cdot \mathbf{x}_j} \\ \text{subject to} \quad & \alpha_i \geq 0 \text{ for } i = 1, 2, \ldots n \\ \text{and} \quad & \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

Transformations

- Introduce new features that result in linearly separable data (Project data)
- XOR Demo



Transformations

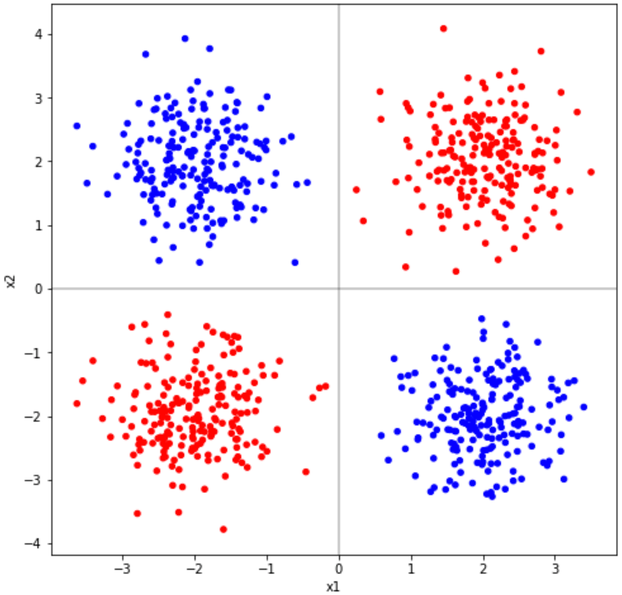


$$
\begin{aligned}
X_1 &\rightarrow x_1^2 \\
X_2 &\rightarrow x_2^2 \\
X_3 &\rightarrow \sqrt{2}\; x_1 x_2
\end{aligned}
$$

$$
\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}
$$

$$
\mathbf{X} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\; x_1 x_2 \end{pmatrix}
$$

Transformations

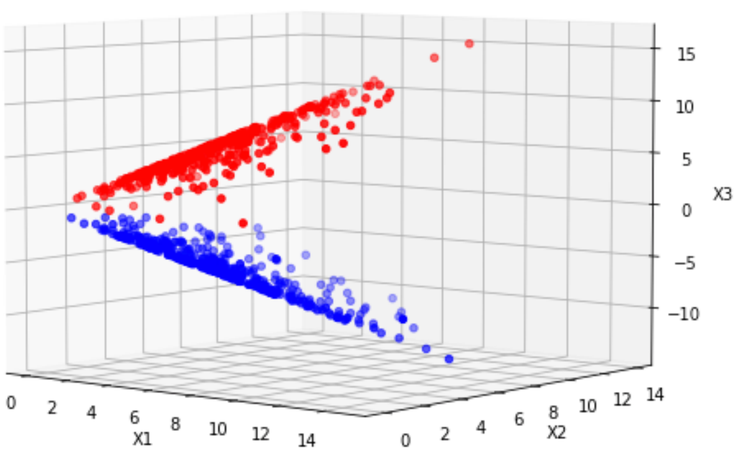


$$\begin{aligned} X_1 &\rightarrow x_1^2 \\ X_2 &\rightarrow x_2^2 \\ X_3 &\rightarrow \sqrt{2}\ x_1 x_2 \end{aligned}$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\mathbf{X} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\ x_1 x_2 \end{pmatrix}$$

Feature space can get very large



Transformations

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$2D \rightarrow 3D$$

$$\mathbf{X} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\ x_1 x_2 \end{pmatrix}$$

$$\mathbf{X}_i \cdot \mathbf{X}_j = (X_1^{(i)} X_1^{(j)} + X_2^{(i)} X_2^{(j)} + X_3^{(i)} X_3^{(j)})$$

$$\begin{array}{rl} \mathbf{X}_i & 4 \text{ calculations} \\ \mathbf{X}_j & 4 \text{ calculations} \\ \mathbf{X}_i \cdot \mathbf{X}_j & 5 \text{ calculations} \end{array}$$

Kernel Functions

$$
\begin{aligned}
K(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i \cdot \mathbf{x}_j)^2 \\
&= (x_{i,1} x_{j,1} + x_{i,2} x_{j,2})^2
\end{aligned}
$$

Number of calculations ?

Kernel Functions

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i \cdot \mathbf{x}_j)^2 \\ &= (\boxed{x_{i,1}x_{j,1}} \boxed{+} \boxed{x_{i,2}x_{j,2}})^{\boxed{2}} \end{aligned}$$

Number of calculations ?

Kernel Functions

$$\begin{aligned}
K(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i \cdot \mathbf{x}_j)^2 \\
&= (x_{i,1}x_{j,1} + x_{i,2}x_{j,2})^2 \\
&= x_{i,1}^2 x_{j,1}^2 + x_{i,2}^2 x_{j,2}^2 + 2x_{i,1}x_{j,1}x_{i,2}x_{j,2} \\
&= (x_{i,1}, x_{i,2}, \sqrt{2}\ x_{i,1}x_{i,2}) \cdot (x_{j,1}, x_{j,2}, \sqrt{2}\ x_{j,1}x_{j,2}) \\
&= \mathbf{X}_i \cdot \mathbf{X}_j
\end{aligned}$$

Transforms the inner product of the original data
Avoids projections into higher dimension

Kernel Functions

$$\begin{aligned}
K(a,b) &= exp\left[-\gamma ||a-b||^2\right] \\
&= exp\left[-\gamma (a^T a + b^T b - 2a^T b)\right] \\
&= exp\left[-\gamma (a^T a + b^T b)\right] \; exp\left[2\gamma \; a^T b\right] \\
&= exp\left[-\gamma (a^T a + b^T b)\right] \; \sum_{k=0}^{\infty} \frac{(2\gamma \; a^T b)^k}{k!}
\end{aligned}$$

Kernel Functions

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$T(\mathbf{x}) = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \end{pmatrix}$$

Transform and then inner product

$$T(\mathbf{x}_i) \cdot T(\mathbf{x}_j)$$

$$\begin{matrix} 1 \\ x_{i,1}x_{j,1} \\ x_{i,2}x_{j,2} \\ x_{i,1}x_{j,1}x_{i,2}x_{j,2} \\ x_{i,1}^2 x_{j,1}^2 \\ x_{i,2}^2 x_{j,2}^2 \end{matrix}$$

Kernel Function: Inner products of original data

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^2 \\ &= (1 + x_{i,1}x_{j,1} + x_{i,2}x_{j,2})^2 \end{aligned}$$

$$\begin{matrix} 1 \\ x_{i,1}x_{j,1} \\ x_{i,2}x_{j,2} \\ x_{i,1}x_{j,1}x_{i,2}x_{j,2} \\ x_{i,1}^2 x_{j,1}^2 \\ x_{i,2}^2 x_{j,2}^2 \end{matrix}$$

Kernels

- Transform original data expands the feature space substantially
- Equivalent to taking inner product of transformed data
- Get the same result without the transformation
- Saves on computation power

Implementation

- Scale your data
- Search for optimum hyperparameters
- Consider multiple kernels…
  - Polynomial / Linear / RBF
- With their respective hyperparameters
  - C
  - Gamma
  - Degree
  - Coef0