

05_Support_Vector_Machine

5.0 A Quick View

What does it do?

- Find an optimized **separating plane** to
 - Separate samples of 2 classes
 - Maximize the margins

Summary

	G	Constraint	Solution
Reg	$G = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j$	$\begin{cases} \lambda_i \geq 0 \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases}$	$f(\mathbf{x}) = \left(\sum_{i=1}^N \lambda_i y_i \right)$
SfMg	$G = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j$	$\begin{cases} 0 \leq \lambda_i \leq C \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases}$	$f(\mathbf{x}) = \left(\sum_{i=1}^N \lambda_i y_i \right)$
NLin	$G = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot K(\mathbf{x}_i, \mathbf{x}_j)$	$\begin{cases} \lambda_i \geq 0 \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases}$	$f(\mathbf{x}) = \sum_{i=1}^N \lambda_i y_i$

5.1 Margin

5.1.1 Motive

Given

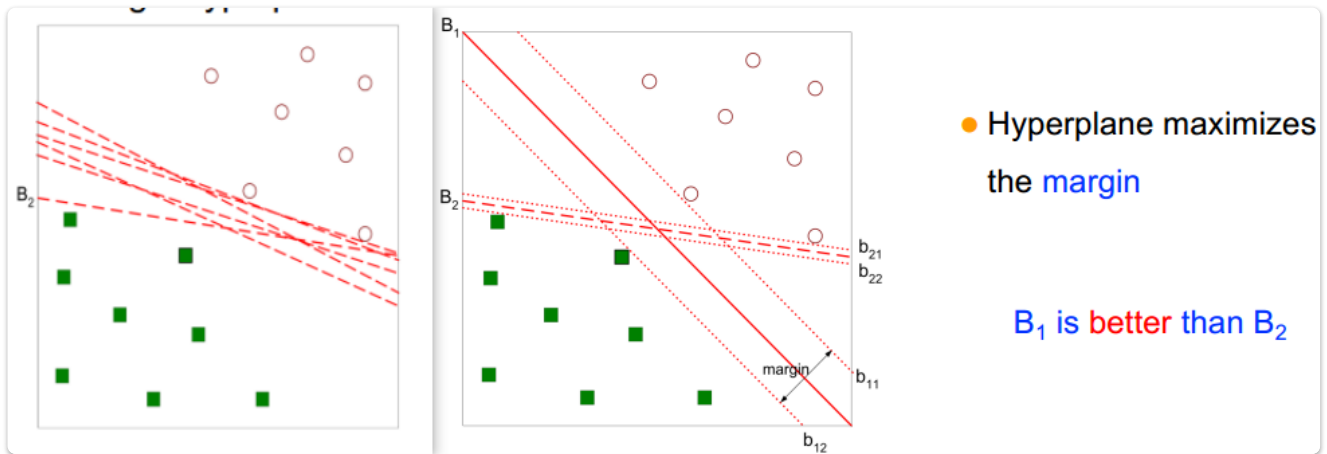
- A set of multi-dimensional linearly separable classes
 - $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$
- Two classes to categorize samples in \mathcal{X} :
 - $\omega = \{\omega_{(1)}, \omega_{(2)}\}$
- A mapping relation of $\mathcal{D} \in \mathcal{X} \times \omega$
 - $\mathcal{D} = \{\langle \mathbf{x}_1, \omega_1 \rangle, \langle \mathbf{x}_2, \omega_2 \rangle, \dots, \langle \mathbf{x}_N, \omega_N \rangle\}$

Do

- Find a hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$ that separates the two classes.

- \mathbf{w} is the normal vector of this hyperplane.

From multiple possible solutions, we want the one that maximizes the margin.



5.1.2 Distance to Hyperplane

i The distance from each sample \mathbf{x}_i to the hyperplane is:

$$r = \frac{\mathbf{w}^\top \mathbf{x}_i + b}{\|\mathbf{w}\|}$$

Proof. Suppose that \mathbf{x}_p is the projection of a data sample \mathbf{x}_i on the hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$. Therefore,

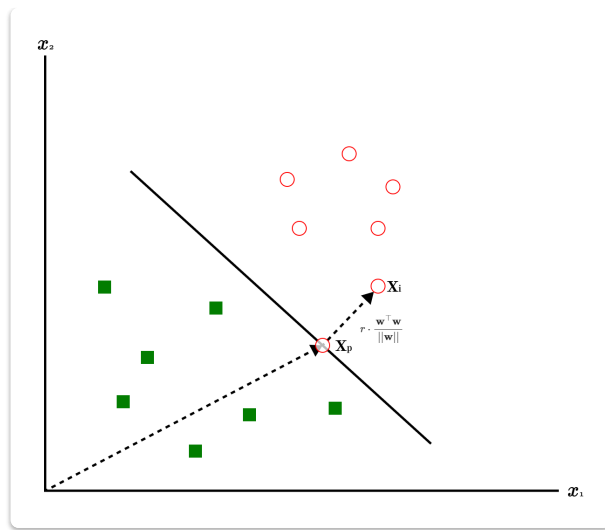
$$\mathbf{x}_i = \mathbf{x}_p + r \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

namely,

$$\begin{aligned} \mathbf{w}^\top \mathbf{x}_i + b &= \mathbf{w}^\top \left(\mathbf{x}_p + r \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \right) + b \\ &= \mathbf{w}^\top \mathbf{x}_p + b + r \cdot \frac{\mathbf{w}^\top \mathbf{w}}{\|\mathbf{w}\|} \\ &= 0 + r \cdot \frac{\mathbf{w}^\top \mathbf{w}}{\|\mathbf{w}\|} \\ &= r \cdot \frac{\mathbf{w}^\top \mathbf{w}}{\|\mathbf{w}\|} \end{aligned}$$

Thus,

$$\begin{aligned} r &= (\mathbf{w}^\top \mathbf{x}_i + b) \cdot \frac{\|\mathbf{w}\|}{\mathbf{w}^\top \mathbf{w}} \\ &= (\mathbf{w}^\top \mathbf{x}_i + b) \cdot \frac{\|\mathbf{w}\|}{\|\mathbf{w}\|^2} \\ &= \frac{\mathbf{w}^\top \mathbf{x}_i + b}{\|\mathbf{w}\|} \end{aligned}$$



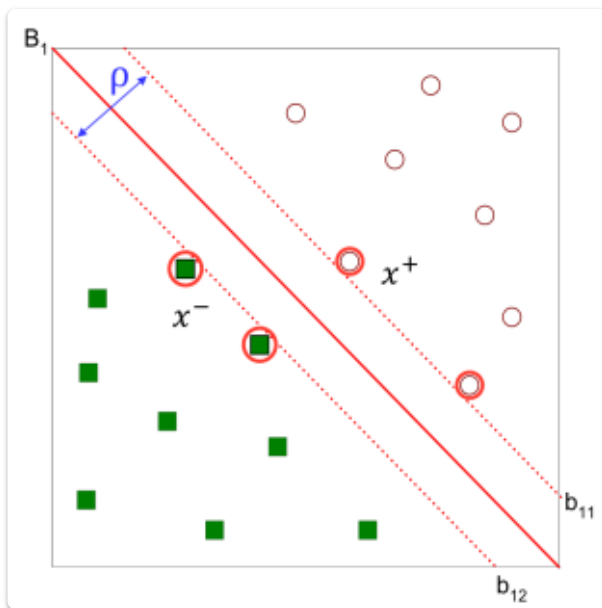
5.1.3 Support Vectors and Margin

i Support Vectors are:

- A subset of training samples
- Samples closest to the hyperplane

i Margin ρ is the distance between support vectors.

- The hyperplane is to maximize the margin ρ .



In the above graph, there are 3 hyperplanes:

$$B_1 : \mathbf{w}^\top \mathbf{x} + b = 0$$

$$b_{11} : \mathbf{w}^\top \mathbf{x} + b = +1$$

$$b_{12} : \mathbf{w}^\top \mathbf{x} + b = -1$$

Where \mathbf{x}^+ and \mathbf{x}^- lies on the hyperplanes b_{11} and b_{12} . Then:

$$\begin{aligned} \mathbf{w}^\top (\mathbf{x}^+ - \mathbf{x}^-) &= \mathbf{w}^\top \mathbf{x}^+ - \mathbf{w}^\top \mathbf{x}^- \\ &= (\mathbf{w}^\top \mathbf{x}^+ + b) - (\mathbf{w}^\top \mathbf{x}^- + b) \\ &= (1) - (-1) \\ &= 2 \end{aligned}$$

The margin would be:

$$\begin{aligned}\rho &= \frac{\mathbf{w}^\top (\mathbf{x}^+ - \mathbf{x}^-)}{\|\mathbf{w}\|} \\ &= \frac{2}{\|\mathbf{w}\|}\end{aligned}$$

5.2 Quadratic Optimization

5.2.1 Formulation

Let

- $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ be the data set
- $y = \{y_1, y_2, \dots, y_N\} \subset \{-1, 1\}^N$ be the class labels of the corresponding data in \mathcal{X} .

Do

- Find the optimal \mathbf{w} such that:
 - $\rho = \frac{2}{\|\mathbf{w}\|}$ is maximized, and
 - $\begin{cases} \mathbf{w}^\top \mathbf{x}_i + b \geq 1 & \text{if } y_i = +1 \\ \mathbf{w}^\top \mathbf{x}_i + b \leq -1 & \text{if } y_i = -1 \end{cases}$ for $i = 1, 2, \dots, N$

Maximizing the margin $\rho = \frac{2}{\|\mathbf{w}\|}$ is equivalent to minimizing:

$$\frac{1}{2} \|\mathbf{w}\|^2 = \frac{1}{2} \mathbf{w}^\top \mathbf{w}$$

★ The formulated quadratic optimization problem of SVM is:

- Minimize: $\frac{1}{2} \|\mathbf{w}\|^2$
- With constraint: $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall 1, 2, \dots, N$

5.2.2 Lagrangian of Quadratic Optimization

★ The Lagrangian of the quadratic optimization problem is:

$$L(\mathbf{w}, b) = \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \sum_{i=1}^N \lambda_i (1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))$$

where $\lambda_1, \lambda_2, \dots, \lambda_N \geq 0$ is the Lagrangian multiplier of all the data points in \mathcal{X} respectively.

★ At the end, only the support vector's Lagrangian multiplier would be non-zero.

- That is $\lambda_i \neq 0$ if and only if \mathbf{x}_i is a support vector.
- Non-support vectors won't contribute to the hyper plane.

Suppose that we have already found a series of such Lagrangian multipliers. To optimize L , we compute the partial derivatives of L with respect to \mathbf{w} and b .

Optimize L w.r.t. \mathbf{w} .

$$\begin{aligned}
\frac{dL}{d\mathbf{w}} &= \mathbf{w} + \frac{dL}{d\mathbf{w}} \sum_{i=1}^N \lambda_i - (\lambda_i y_i) \mathbf{w}^\top \mathbf{x}_i - (\lambda_i y_i b) \\
&= \mathbf{w} + \sum_{i=1}^N -\lambda_i y_i \mathbf{x}_i \\
&= \mathbf{w} - \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i
\end{aligned}$$

Let $\frac{dL}{d\mathbf{w}} = 0$.

$$\mathbf{w} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i$$

Optimize L w.r.t. b

$$\begin{aligned}
\frac{dL}{db} &= 0 + \frac{dL}{db} \sum_{i=1}^N \lambda_i - (\lambda_i y_i) \mathbf{w}^\top \mathbf{x}_i - (\lambda_i y_i b) \\
&= \sum_{i=1}^N -\lambda_i y_i \\
&= - \sum_{i=1}^N \lambda_i y_i
\end{aligned}$$

Let $\frac{dL}{db} = 0$.

$$\sum_{i=1}^N \lambda_i y_i = 0$$

★ Therefore, the optimized weight and bias of this quadratic is:

$$\mathbf{w} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i$$

★ with a constraint of:

$$\sum_{i=1}^N \lambda_i y_i = 0$$

with respect to $\lambda_1, \lambda_2, \dots, \lambda_N \geq 0$.

5.2.3 Dual Problem

Get λ with optimized L

Substitute $\mathbf{w} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i$ and $\sum_{i=1}^N \lambda_i y_i = 0$ into $L(\mathbf{w}, b)$ would result in:

$$\begin{aligned}
L(\mathbf{w}, b) &= \frac{1}{2} \left(\sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \right)^\top \left(\sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \right) \\
&\quad + \sum_{i=1}^N \lambda_i (1 - y_i) \left(\left(\sum_{j=1}^N \lambda_j y_j \mathbf{x}_j \right)^\top \mathbf{x}_i + b \right) \\
&= \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j \\
&\quad + \sum_{i=1}^N \lambda_i - \sum_{i=1}^N (\lambda_i y_i) \cdot \left(\sum_{j=1}^N (\lambda_j y_j) \cdot \mathbf{x}_j^\top \mathbf{x}_i + b \right) \\
&= \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j \\
&\quad + \sum_{i=1}^N \lambda_i - \sum_{i=1}^N \left(\sum_{j=1}^N (\lambda_i y_i) \cdot (\lambda_j y_j) \cdot \mathbf{x}_j^\top \mathbf{x}_i + b(\lambda_i y_i) \right) \\
&= \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j \\
&\quad + \sum_{i=1}^N \lambda_i - \sum_{i=1}^N \sum_{j=1}^N (\lambda_i y_i) \cdot (\lambda_j y_j) \cdot \mathbf{x}_j^\top \mathbf{x}_i \\
&\quad + b \sum_{i=1}^N \sum_{j=1}^N (\lambda_i y_i) \\
&= -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j \\
&\quad + \sum_{i=1}^N \lambda_i \\
&\quad + b \sum_{i=1}^N \sum_{j=1}^N (\lambda_i y_i) \\
&= \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j
\end{aligned}$$

The original criterion function is now with respect to only λ . That is:

$$G(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j$$

Optimize $G(\lambda)$ by computing:

$$\frac{dG}{d\lambda_i}, \forall i = 2, 3, \dots, N$$

Check the solutions if they satisfy the constraint of:

$$\begin{cases} \lambda_i > 0 \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases}$$

5.2.4 Solutions: Regular

Maximize:

$$G(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j$$

with constraint:

$$\begin{cases} \lambda_i \geq 0 \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases}$$

The dual solution is:

$$f(\mathbf{x}) = \left(\sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \right)^\top \mathbf{x} + b$$

with:

$$\begin{cases} \mathbf{w} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \\ b = y_k - \left(\sum_{i=1}^N \lambda_i y_i \mathbf{x}_i^\top \right) \mathbf{x}_k \end{cases}$$

5.3 Soft Margin Classification

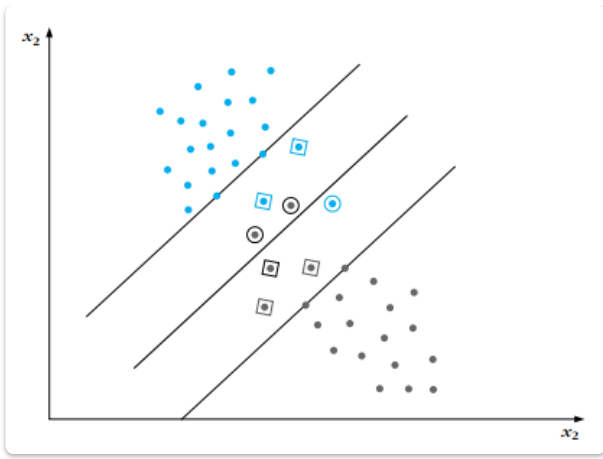
5.3.1 Problem Setup

There exists conditions that training samples can't be separated.

- In this case, **no** hyperplane could satisfy $y_i(\mathbf{w}^\top \mathbf{x} + b) > 1, \forall \mathbf{x}$.
- i.e., $\neg \exists \mathbf{w}, b, \forall \mathbf{x}, y_i(\mathbf{w}^\top \mathbf{x} + b) > 1$

Training samples belong to one of the three possible categories.

- Correctly Classified: Samples outside the margin.
 - $y_i(\mathbf{w}^\top + b) > 1$
- Margin Violation: Samples within the margin, but correctly classified.
 - $y_i(\mathbf{w}^\top + b) > 1$
- Misclassified samples:
 - $y_i(\mathbf{w}^\top + b) < 0$



5.3.2 Slack Variables & Parameter C 松弛因子与C参数

Assignment of ξ_i

Assign slack variables $\xi_1, \xi_2, \dots, \xi_N \geq 0$ to all the samples in \mathcal{X} .

- Correctly Classified: $\xi_i = 0$
- Margin Violation: $0 \leq \xi_i \leq 1$
- Misclassified Variables: $\xi_i > 1$

i About slack variables ξ_i .

- ξ_i allows misclassification of difficult or noisy samples.
 - The resulting is called a **Soft Margin**.
 - If ξ_i is sufficiently large, every constraint will be forced to be satisfied.
- ξ_i is based on the output of the discriminant function $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$.
- ξ_i approximates the number of mis-classified samples.

Intuitive Optimization

The optimization problem becomes:

- Minimize: $\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$
- With constraint: $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \forall 1, 2, \dots, N$

i The parameter C is a user-selected regularization parameter.

- A trade-off parameter between:
 - error and
 - margin
- Effects of parameter C :
 - Smaller C = Large Margin & More error, allows constraints to be easily ignored.
 - Larger C = Narrow Margin & Less error, constraints is hard to ignore.
 - $C = \infty$ = Hard Margin, which enforces all constraints.

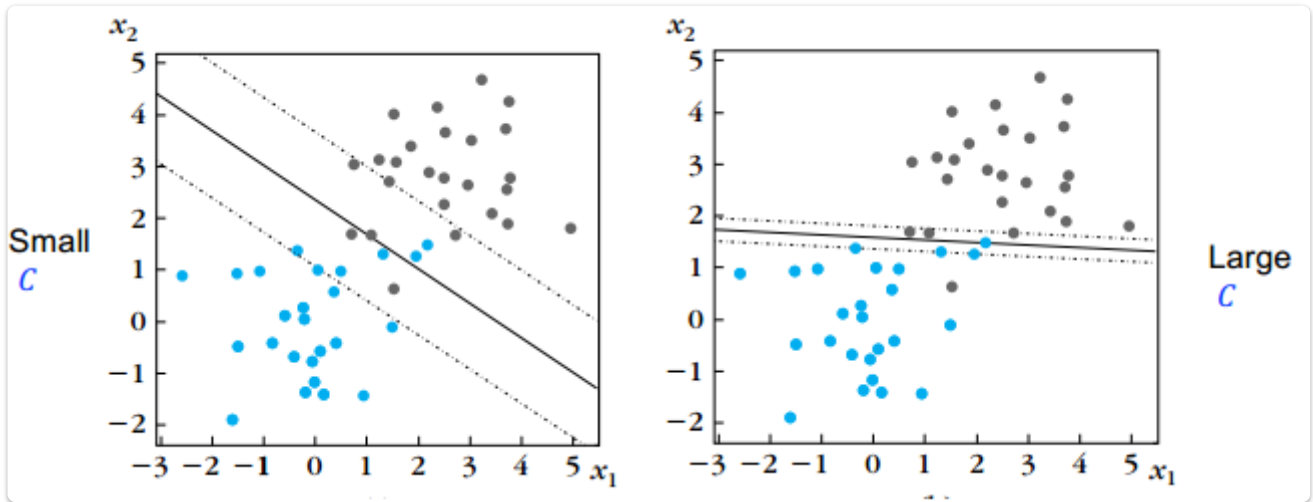
Corresponding Dual Problem

Which transfer to the dual problem as

$$G(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j$$

with the constraints of:

$$\begin{cases} 0 \leq \lambda_i \leq C \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases}$$



5.3.3 Solutions: Soft Margin

Maximize:

$$G(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j$$

with constraint:

$$\begin{cases} 0 \leq \lambda_i \leq C \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases}$$

The dual solution is:

$$f(\mathbf{x}) = \left(\sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \right)^\top \mathbf{x} + b$$

with:

$$\begin{cases} \mathbf{w} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \\ b = y_k (1 - \xi_k) - \left(\sum_{i=1}^N \lambda_i y_i \mathbf{x}_i^\top \right) \mathbf{x}_k \end{cases}$$

5.4 Non-Linear SVM

5.4.0 Basics

Why Non-Linear SVM?

- Datasets may be too hard for linear separation.

What does it do?

- Transform data into a higher dimensional space \mathbf{H} ,
 - via a mapping function Φ
 - such that the data appears of the form $\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j)$.
- Linear separation in \mathbf{H} is *equivalent* to non-linear separation in the original input space.

Problems

- High dimensionality results in high computation burden.
- Hard to obtain a good estimation.

5.4.1 Kernel Function

i A kernel of two data is:

- The inner product between the vectors
- $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$

Suppose that each data point is mapped into high-dimensional space via transformation of:

$$\Phi : \mathbf{x} \mapsto \phi(\mathbf{x})$$

Then, the kernel of two data becomes:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

We could compute $K(\mathbf{x}_i, \mathbf{x}_j)$ without computing $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ explicitly.

5.4.2 Compute Kernel

We could compute kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ in the original space. Suppose that the original space is of 2 dimensions. Let $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^\top \mathbf{x}_j)^2$, then $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^\top \mathbf{x}_j)^2$.

Proof.

$$\begin{aligned}
K(\mathbf{x}_i, \mathbf{x}_j) &= (1 + \mathbf{x}_i^\top \mathbf{x}_j)^2 \\
&= (1 + [x_{i1} \quad x_{i2}] \begin{bmatrix} x_{j1} \\ x_{j2} \end{bmatrix})^2 \\
&= \left(1 + (x_{i1}x_{j1} + x_{i2}x_{j2})\right)^2 \\
&= 1 + (x_{i1}x_{j1} + x_{i2}x_{j2})^2 - 2((x_{i1}x_{j1} + x_{i2}x_{j2})) \\
&= x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 1 \\
&= \begin{bmatrix} x_{i1}^2 & x_{i2}^2 & \sqrt{2}x_{i1}x_{i2} & \sqrt{2}x_{i1} & \sqrt{2}x_{i2} & 1 \end{bmatrix} \begin{bmatrix} x_{j1}^2 \\ x_{j2}^2 \\ \sqrt{2}x_{j1}x_{j2} \\ \sqrt{2}x_{j1} \\ \sqrt{2}x_{j2} \\ 1 \end{bmatrix} \\
&= \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)
\end{aligned}$$

where

$$\phi(\mathbf{x}) = \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ 1 \end{bmatrix}$$

5.4.3 Kernel Examples

	Kernel	Mapping
Linear	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$	$\Phi : \mathbf{x} \mapsto \phi(\mathbf{x}) = \mathbf{x}$
Polynomial	$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^\top \mathbf{x}_j)^p$	$\Phi : \mathbf{x} \mapsto \phi(\mathbf{x}) \in \mathbb{R}^{\frac{(d+p)!}{p!d!}}$
Sigmoid	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^\top \mathbf{x}_j + \beta_1)$	
Gaussian	$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\sigma}}$	$\Phi : \mathbf{x} \mapsto \phi(\mathbf{x}) \in \mathbb{R}^\infty$

5.4.4 Solutions: Non-Linear

Maximize:

$$G(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot K(\mathbf{x}_i, \mathbf{x}_j)$$

with constraint:

$$\begin{cases} \lambda_i \geq 0 \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases}$$

The dual solution:

$$f(\mathbf{x}) = \sum_{i=1}^N \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

5.5 Examples

Non-Linear

Given

- Suppose we have five 1-D datapoints:
 - $x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 5, x_5 = 6$
- Their corresponding classes are:
 - $y_1 = 1, y_2 = 1, y_3 = -1, y_4 = -1, y_5 = 1$
- Use the degree-2 polynomial kernel:
 - $K(x_i, x_j) = (x_i x_j + 1)^2$

Do

Step 1. Find λ_i .

Maximize:

$$G(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot (1 + x_i x_j)^2$$

with constraint:

$$\sum_{i=1}^N \lambda_i y_i = 0$$

Using a quadratic problem solver, we obtain:

$$\lambda_1 = 0, \lambda_2 = 2.5, \lambda_3 = 0, \lambda_4 = 7.333, \lambda_5 = 4.833$$

Step 2. Calculate.

The support vectors are:

$$x_2 = 2, x_4 = 5, x_5 = 6$$

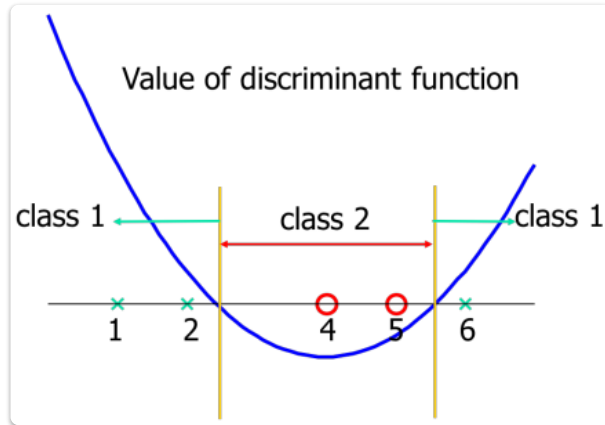
The discriminant function is:

$$\begin{aligned}
 f(x) &= \sum_{i=1}^N \lambda_i y_i K(x_i, x) + b \\
 &= \sum_{i=1}^N \lambda_i y_i (1 + x_i x)^2 + b \\
 &= 2.5 \cdot 1 \cdot (1 + 2x)^2 + 7.333 \cdot (-1) \cdot (1 + 5x)^2 + 4.833 \cdot 1 \cdot (1 + 6x)^2 + b \\
 &= 0.6667x^2 - 5.333x + b
 \end{aligned}$$

Given that $f(6) = 1$, $b = 9$.

We obtained the discriminant function of:

$$f(x) = 0.6667x^2 - 5.333x + 9$$



Linear

Given

- Suppose we have three 2-D data points.
 - $\mathbf{x}_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, $\mathbf{x}_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\mathbf{x}_3 = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$
- Their corresponding labels are:
 - $y_1 = 1$, $y_2 = -1$, $y_3 = -1$
- Use the trivial kernel:
 - $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$

Do

Step 1. Find λ .

$$G(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j$$

1-1. Constraint:

$$\begin{aligned}
 \sum_{i=1}^N \lambda_i y_i = 0 &\implies \lambda_1 - \lambda_2 - \lambda_3 = 0 \\
 &\implies \lambda_1 = \lambda_2 + \lambda_3
 \end{aligned}$$

1-2. Express $G(\lambda)$ with λ_1 , λ_2 and λ_3 .

$$\frac{1}{2} \sum_{i=1}^N (\lambda_i \lambda_j) \cdot (y_i y_j) \cdot \mathbf{x}_i^\top \mathbf{x}_j = \sum_{i=1}^N \lambda_i - G(\lambda)$$

$$\begin{aligned} \Rightarrow \frac{1}{2} \sum_{i=1}^N \Big(& (\lambda_i \lambda_1) \cdot (y_i y_1) \cdot \mathbf{x}_i^\top \mathbf{x}_1 + \\ & (\lambda_i \lambda_2) \cdot (y_i y_2) \cdot \mathbf{x}_i^\top \mathbf{x}_2 + \\ & (\lambda_i \lambda_3) \cdot (y_i y_3) \cdot \mathbf{x}_i^\top \mathbf{x}_3 \\ & \Big) = \sum_{i=1}^N \lambda_i - G(\lambda) \end{aligned}$$

$$\begin{aligned} \Rightarrow \frac{1}{2} \Big(& (\lambda_1 \lambda_1) \cdot (y_1 y_1) \cdot \mathbf{x}_1^\top \mathbf{x}_1 + (\lambda_1 \lambda_2) \cdot (y_1 y_2) \cdot \mathbf{x}_1^\top \mathbf{x}_2 + (\lambda_1 \lambda_3) \cdot (y_1 y_3) \cdot \mathbf{x}_1^\top \mathbf{x}_3 + \\ & (\lambda_2 \lambda_1) \cdot (y_2 y_1) \cdot \mathbf{x}_2^\top \mathbf{x}_1 + (\lambda_2 \lambda_2) \cdot (y_2 y_2) \cdot \mathbf{x}_2^\top \mathbf{x}_2 + (\lambda_2 \lambda_3) \cdot (y_2 y_3) \cdot \mathbf{x}_2^\top \mathbf{x}_3 + \\ & (\lambda_3 \lambda_1) \cdot (y_3 y_1) \cdot \mathbf{x}_3^\top \mathbf{x}_1 + (\lambda_3 \lambda_2) \cdot (y_3 y_2) \cdot \mathbf{x}_3^\top \mathbf{x}_2 + (\lambda_3 \lambda_3) \cdot (y_3 y_3) \cdot \mathbf{x}_3^\top \mathbf{x}_3 \\ & \Big) = \sum_{i=1}^N \lambda_i - G(\lambda) \end{aligned}$$

$$\begin{aligned} \Rightarrow \frac{1}{2} \Big(& \lambda_1^2 \cdot (1 \cdot 1) \cdot \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \lambda_1 \lambda_2 \cdot (1 \cdot -1) \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \lambda_1 \lambda_3 \cdot (1 \cdot -1) \cdot \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \end{bmatrix} + \\ & \lambda_2 \lambda_1 \cdot (-1 \cdot 1) \cdot \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \lambda_2^2 \cdot (-1 \cdot -1) \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \lambda_2 \lambda_3 \cdot (-1 \cdot -1) \cdot \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \end{bmatrix} + \\ & \lambda_3 \lambda_1 \cdot (-1 \cdot 1) \cdot \begin{bmatrix} 3 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \lambda_3 \lambda_2 \cdot (-1 \cdot -1) \begin{bmatrix} 3 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \lambda_3^2 \cdot (-1 \cdot -1) \cdot \begin{bmatrix} 3 & 3 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \end{bmatrix} \\ & \Big) = \sum_{i=1}^N \lambda_i - G(\lambda) \end{aligned}$$

$$\begin{aligned} \Rightarrow \frac{1}{2} \Big(& 5\lambda_1^2 - 4\lambda_1 \lambda_2 - 9\lambda_1 \lambda_3 \\ & - 4\lambda_1 \lambda_2 + 5\lambda_2^2 + 9\lambda_2 \lambda_3 \\ & - 9\lambda_1 \lambda_3 + 9\lambda_3 \lambda_3 + 18\lambda_3^2 \\ & \Big) = \lambda_1 + \lambda_2 + \lambda_3 - G(\lambda) \end{aligned}$$

$$\begin{aligned} \Rightarrow G(\lambda) = & -\frac{1}{2} (5\lambda_1^2 + 5\lambda_2^2 + 18\lambda_3^2 - 8\lambda_1 \lambda_2 - 18\lambda_1 \lambda_3 + 18\lambda_2 \lambda_3) \\ & + (\lambda_1 + \lambda_2 + \lambda_3) \end{aligned}$$

$$\begin{aligned} \Rightarrow G(\lambda) = & -\frac{1}{2} (5(\lambda_2 + \lambda_3)^2 + 5\lambda_2^2 + 18\lambda_3^2 - 8(\lambda_2 + \lambda_3)\lambda_2 - 18(\lambda_2 + \lambda_3)\lambda_3 + 18\lambda_2 \lambda_3) \\ & + ((\lambda_2 + \lambda_3) + \lambda_2 + \lambda_3) \end{aligned}$$

$$\Rightarrow G(\lambda) = -\lambda_2^2 - \frac{5}{2} \lambda_3^2 - \lambda_2 \lambda_3 + 2\lambda_2 + 2\lambda_3$$

1-3. Minimize $G(\lambda)$

$$\frac{\partial G}{\partial \lambda_3} = 0$$

$$\implies \frac{\partial}{\partial \lambda_3} \left(-\lambda_2^2 + (2 - \lambda_3)\lambda_2 + \left(2\lambda_3 - \frac{5}{2}\lambda_3^2\right) \right) = 0$$

$$\implies 2\lambda_2 + \lambda_3 - 2 = 0$$

$$\frac{\partial G}{\partial \lambda_2} = 0$$

$$\implies \frac{\partial}{\partial \lambda_2} \left(-\frac{5}{2}\lambda_3^2 + (2 - \lambda_2)\lambda_3 + (2\lambda_2 - \lambda_2^2) \right) = 0$$

$$\implies \lambda_2 + 5\lambda_3 - 2 = 0$$

1-4. Summarize.

$$\begin{cases} \lambda_1 - \lambda_2 - \lambda_3 = 0 \\ 2\lambda_2 + \lambda_3 = 2 \\ \lambda_2 + 5\lambda_3 = 2 \end{cases}$$

$$\implies \begin{bmatrix} 1 & -1 & -1 \\ 0 & 2 & 1 \\ 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix}$$

$$\implies \begin{cases} \lambda_1 = \frac{10}{9} \\ \lambda_2 = \frac{8}{9} \\ \lambda_3 = \frac{2}{9} \end{cases}$$

Step 2. Calculate.

The discriminant function is

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i^\top \mathbf{x} + b \\ &= \left(\frac{10}{9} [2 \ 1] - \frac{8}{9} [1 \ 2] - \frac{2}{9} [3 \ 3] \right) \mathbf{x} + b \\ &= \left[\frac{2}{3} \quad -\frac{4}{3} \right] \mathbf{x} + b \end{aligned}$$

Get b :

$$f(\mathbf{x}_1) = 1$$

$$\implies f\left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = 1$$

$$\implies \begin{bmatrix} \frac{2}{3} & -\frac{4}{3} \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + b = 1$$

$$\implies \frac{4}{3} - \frac{4}{3} + b = 1$$

$$\implies b = 1$$

Therefore, the discriminant function would be:

$$f(\mathbf{x}) = \begin{bmatrix} \frac{2}{3} & -\frac{4}{3} \end{bmatrix} \mathbf{x} + 1$$