10_Support_Vector_Machine

Final Tip:

- Primal & Dual Problem
- How to re-formulate Primal to Dual Problem (easier-to-solve)

10.1 Hyperplane & Binary Classification

10.1.1 Binary Classification

Given

- Training Sample: $\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^{N}$
  - Instances: $\mathbf{x}_t \in \mathbb{R}^m$
  - Labels: $y_t \in \{+1, -1\}$

  ### Do
- Train a prediction function:

$$h : \mathcal{X} \mapsto \{+1, -1\}$$

One intuition is to give a *Linear Discriminant Function*:

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

where the decision boundary is given by:

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

Key Point 1: $\mathbf{w}$ and Hyperplane

$\mathbf{w}$ is normal to the hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$.

Proof.

Suppose that $\mathbf{x}_1$ and $\mathbf{x}_2$ are two different points lying on the hyperplane. Therefore, we know that:

$$\begin{aligned} \mathbf{w}^\top \mathbf{x}_1 + b = 0 \\ \mathbf{w}^\top \mathbf{x}_2 + b = 0 \\ \mathbf{x}_1 \neq \mathbf{x}_2 \end{aligned}$$

Subtracting the two equations:

$$\mathbf{w}^\top (\mathbf{x}_1 - \mathbf{x}_2) = 0$$

where $\mathbf{x}_1 - \mathbf{x}_2$ by definition is an arbitrary line on the hyperplane.

- That is, $\mathbf{w}$ is perpendicular to any arbitrary line in the hyperplane;
- which means that $\mathbf{w}$ is normal to the hyperplane.

Key Point 2: Distance Equation

We need to know the distance from an arbitrary data to the hyperplane.
Consider an arbitrary data point $\mathbf{x}$ as:

$$\mathbf{x} = \mathbf{x}_p + \rho \frac{\mathbf{w}}{||\mathbf{w}||}$$

where:

- $\mathbf{x}_p$ is the orthogonal projection of point $\mathbf{x}$ on the hyperplane.
- $\frac{\mathbf{w}}{||\mathbf{w}||}$ is the unit vector with the same direction as $\mathbf{w}$.
- ρ is the *distance* from the point to the hyperplane.

We could derive ρ by:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{w}^\top \mathbf{x} + b \\ &= \mathbf{w}^\top (\mathbf{x}_p + \rho \frac{\mathbf{w}}{||\mathbf{w}||}) + b \\ &= (\mathbf{w}^\top \mathbf{x}_p + b) + \rho \frac{\mathbf{w}^\top \mathbf{w}}{||\mathbf{w}||} \\ &= 0 + \rho \frac{\mathbf{w}^\top \mathbf{w}}{||\mathbf{w}||} \\ &= \rho \frac{||\mathbf{w}||^2}{||\mathbf{w}||} \\ &= \rho ||\mathbf{w}|| \end{aligned}$$

Namely:

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

10.1.2 Canonical Form

The function $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ actually defines a classification metric. For an arbitrary data point $\mathbf{x}$,

$$f(\mathbf{x}) > 0 \equiv \mathbf{x} \text{ lies "above" the plane;}$$

$$f(\mathbf{x}) = 0 \equiv \mathbf{x} \text{ lies on the plane;}$$

$$f(\mathbf{x}) < 0 \equiv \mathbf{x} \text{ lies "below" the plane;}$$

The term "above" means that $\mathbf{x} - \mathbf{x}_p$ points to the same direction as $\mathbf{w}$, and vise versa.

The prediction function is defined as:

$$\begin{aligned} h(\mathbf{x}) &= \text{sign}(f(\mathbf{x})) \\ &= \text{sign}(\mathbf{w}^\top \mathbf{x} + b) \end{aligned}$$

Key Point 1: Functional & Geometric Margin

Given:

- An example $(\mathbf{x}_t, y_t) \in \mathbb{R}^m \times \{+1, -1\}$.
  Do:
- ℹ The functional margin of this example with respect to the hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$ is given by:

  $$\begin{aligned} \rho_t &= y_t \cdot f(\mathbf{x}_t) \\ &= y_t \cdot (\mathbf{w}^\top \mathbf{x}_t + b) \end{aligned}$$

  Functional margin can tell if a data point is *Incorrectly Classified.*
- y_t is the ground truth.
- If the data point $\mathbf{x}_t$ is correctly classified, y_t and $\mathbf{w}^\top \mathbf{x}_t + b$ should have the same signs;
- i.e. if the data point $\mathbf{x}_t$ is correctly classified, $\rho_t \geq 0$, otherwise $\rho_t < 0$.
- ℹ The geometric margin of this example with respect to the hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$ is given by:

  $$\begin{aligned} \rho &= y_t \cdot \frac{f(\mathbf{x})}{||\mathbf{w}||} \\ &= y_t \cdot \frac{\mathbf{w}^\top \mathbf{x} + b}{||\mathbf{w}||} \end{aligned}$$

  This tells not only if the data point is incorrectly classified, but also preserves the *Euclidean Distance* from the point to the hyperplane.

Key Point 2: Canonical Form

Note that, for arbitrary $\lambda \neq 0 \in \mathbb{R}$, the following equation denotes the exact same hyperplane:

$$\lambda(\mathbf{w}^\top \mathbf{x} + b) = 0$$

The canonical form of a hyper plane is:

$$\min_{\mathbf{x}_t \in \mathcal{X}} |\mathbf{w}^\top \mathbf{x}_t + b| = 1$$

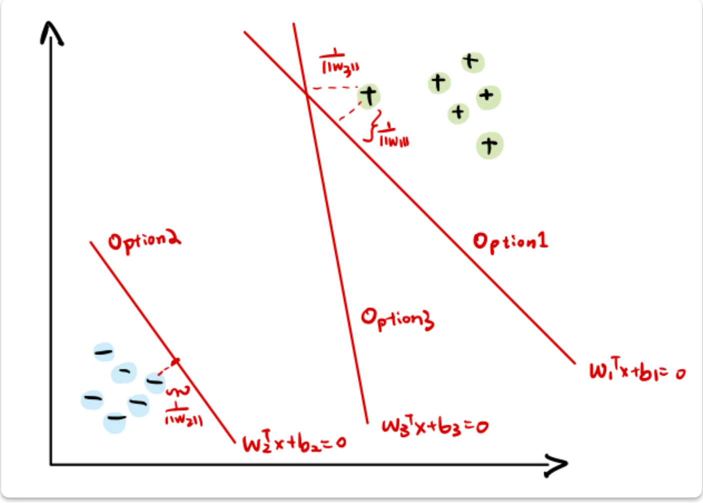
That is, this hyperplane is defined that:

- the *minimum functional margin* from an arbitrary point to the hyper plane is exactly 1.
- i.e., the *minimum euclidean distance* from an arbitrary point to the hyperplane is exactly $\frac{1}{\|\mathbf{w}\|}$.

The canonical form regulates that:

- The classification should not only be correct;
- It should also be robust.

Summary: What to Optimize?



💡 Note that: If the two classes are separated enough, we can easily find infinite options that satisfies the canonical form.

However, to ensure the model's robustness facing unseen data, we need to search for a hyper-plane that:

1. Maximizes the geometric margin $\frac{1}{\|\mathbf{w}\|}$;
2. while maintaining the property $y_i(\mathbf{w}^\top \mathbf{x} + b) \geq 1, \forall i$

10.2 Optimization: Primal Form & Dual Problem

We need a classifier that gives us the max margin.

10.2.1 Primal Form: Optimization with Constraints

We need to find an optimized weight $\mathbf{w}^*$ such that the geometric distance from one of the buffer to the hyper plane:

$$\frac{1}{\|\mathbf{w}\|}$$

is maximized. Maximizing this value is equivalent to *minimizing* the objective function of:

$$\mathcal{J}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$$

with the constraints:

$$y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1,\ y_i \in \{1, -1\};\ i = 1, \cdots, N$$

The constraints regulates that all the points should be *out of or on* the two *functional margins.*

10.2.2 Primal Lagrangian

We use the Primal Lagrangian to combine the optimization and the constraint.
The **Primal Lagrangian** is given by:

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \alpha_i\left(1 - y_i(\mathbf{w}^\top \mathbf{x} + b)\right)$$

We need to *Minimize* this function.

We introduce a Lagrange multiplier α_i for all the data samples $\mathbf{x}_i \in \mathcal{X}$. These multiplier terms enforce the constraint $y_i\left(\mathbf{w}^\top \mathbf{x} + b\right) \geq 1$ by:

- Penalizing the objective if the constraint is violated.
  - If the objective is violated, $1 - y_i(\mathbf{w}^\top \mathbf{x} + b)$ would be larger than 0.
  - The minimization is then *prevented slightly*.
- The more it violates, the more term is added to the Lagrangian function, scaled by the penalization factor of α_i.

10.2.3 Dual Problem

Key Point 1: Why do we need a "Dual Problem"?

The original Primal Lagrangian has a total of $N + 2$ parameters:

- A weight $\mathbf{w}$
- A bias b
- The N Lagrangian parameters $\{\alpha_N\}_1^N$ corresponding to N datapoints.
  With this many parameters, minimizing this function is costly.

However, we could mimic the optimal solutions by converting the original Primal Lagrangian into a Dual Problem. To solve the Dual Problem:

- First, optimize $\mathcal{L}(\mathbf{w}, b, \alpha)$ w.r.t. $\mathbf{w}$ and b.
  - *Assume* that all the Lagrangian parameters α_i are already found.
  - Then, minimize the function w.r.t. $\mathbf{w}$ and b, regarding all the α_i as constants.
  - The original function $\mathcal{L}(\mathbf{w}, b, \alpha)$ is thus converted to $\mathcal{G}(\alpha)$, which is a function that only contains the Lagrangian parameters.
- After this, optimize $\mathcal{G}(\alpha)$ w.r.t. $\{\alpha\}_i^N$.

> *The effect of the Dual Problem could be witnessed, but is is not yet been formally proven. Note that the solution to the Dual Problem could be similar to the Primal Problem, but they are highly likely not be exactly the same.*

Step 1: Optimize $\mathcal{L}(\mathbf{w}, b, \alpha)$ w.r.t. $\mathbf{w}$ and b.

To optimize, we minimize $\mathcal{L}$ by setting $\dfrac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0$ and $\dfrac{\partial \mathcal{L}}{\partial b} = 0$.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0$$

$$\implies \frac{\partial}{\partial \mathbf{w}} \left[\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^{N} \alpha_i \Big(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \Big) \right] = 0$$

$$\implies \mathbf{w} + \sum_{i=1}^{N} -\alpha_i y_i \mathbf{x}_i = 0$$

$$\implies \mathbf{w} = \sum_{i=1}^{N} (\alpha_i \cdot y_i) \cdot \mathbf{x}_i$$

$$
\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial b} = 0 \\
\implies &\frac{\partial}{\partial b}\left[\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \alpha_i \Big(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\Big)\right] = 0 \\
\implies &0 + \frac{\partial}{\partial b} \sum_{i=1}^{N} \alpha_i - \alpha_i y_i \mathbf{w}^\top \mathbf{x}_i - \alpha_i y_i b = 0 \\
\implies &\sum_{i=1}^{N} 0 - \alpha_i y_i = 0 \\
\implies &\sum_{i=1}^{N} \alpha_i \cdot y_i = 0
\end{aligned}
$$

★ In summary, the first optimization yields the following:

$$
\mathbf{w}^* = \sum_{i=1}^{N} (\alpha_i \cdot y_i) \cdot \mathbf{x}_i
$$

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

Step 2: Substitute optimal $\mathbf{w}$ into $\mathcal{L}$.

Substituting $\mathbf{w}^*$ in $\mathcal{L}$ yields:

$$
\mathcal{L}(\mathbf{w}^*, b, \alpha) = \frac{1}{2}\|\mathbf{w}^*\|^2 + \sum_{i=1}^{N} \alpha_i \Big(1 - y_i(\mathbf{w}^{*\top} \mathbf{x}_i + b)\Big)
$$

Respectively:

1. First, we substitute the first term:

$$
\begin{aligned}
\frac{1}{2}\|\mathbf{w}^*\|^2 &= \frac{1}{2}\left[\sum_{i=1}^{N} (\alpha_i \cdot y_i) \cdot \mathbf{x}_i\right]^\top \left[\sum_{i=1}^{N} (\alpha_i \cdot y_i) \cdot \mathbf{x}_i\right] \\
&= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i \cdot \alpha_j) \cdot (y_i \cdot y_j) \cdot (\mathbf{x}_i^\top \mathbf{x}_j)
\end{aligned}
$$

2. Then, we substitute the second term:

$$\begin{aligned}\sum_{i=1}^{N} \alpha_i\Big(1-y_i(\mathbf{w}^{*\top}\mathbf{x}_i+b)\Big) &= \sum_{i=1}^{N} \alpha_i\Big[1-y_i(\Big[\sum_{j=1}^{N} \alpha_j\cdot y_j\cdot \mathbf{x}_j\Big]^{\top} \mathbf{x}_i+b)\Big] \\ &= \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N} \alpha_i\cdot y_i(\Big[\sum_{j=1}^{N}\alpha_j\cdot y_j\cdot\mathbf{x}_j\Big]^{\top}\mathbf{x}_i+b) \\ &= \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i\cdot\alpha_j)\cdot(y_i\cdot y_j)\cdot(\mathbf{x}_i^{\top}\mathbf{x}_j)\end{aligned}$$

Therefore, the optimal function with respect to the weights $\mathbf{w}$ and the bias b would be:

$$\mathcal{L} = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i\cdot\alpha_j)\cdot(y_i\cdot y_j)\cdot(\mathbf{x}_i^{\top}\mathbf{x}_j)+\sum_{i=1}^{N}\alpha_i$$

The function is already optimal with respect to the $\mathbf{w}$ and b, under the given Lagrangian multipliers $\{\alpha_i\}_{i=1}^{N}$.

★ The new function is given by:

$$\mathcal{G}(\alpha) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i\cdot\alpha_j)\cdot(y_i\cdot y_j)\cdot(\mathbf{x}_i^{\top}\mathbf{x}_j)+\sum_{i=1}^{N}\alpha_i$$

with the constraints:

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

Step 3: Optimize $\mathcal{G}(\alpha)$ w.r.t. $\{\alpha_i\}_i^N$.

By the constraint, we know that:

$$\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_N\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_N\end{bmatrix} = 0$$

The original $\mathcal{G}$ could be expressed in the form of :

$$\mathcal{G}(\alpha) = -\frac{1}{2}\alpha^{\top}H\alpha+\alpha^{\top}\mathbf{1}$$

where:

- $\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}$

To optimize $\mathcal{G}$, we compute $\frac{\partial \mathcal{G}}{\partial \alpha_i}$ for all α_i.

10.2.4 Support Vectors

Remark: The Primal Form:

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \alpha_i \Big(1 - y_i(\mathbf{w}^\top \mathbf{x} + b)\Big)$$

We optimized $\mathbf{w}$ and b as we assumed that α is already known.

The optimized Lagrangian parameters will satisfy:

- $\alpha_i \neq 0$:
  - if $\mathbf{x}_i$ is a support vector.
  - i.e., $1 - y_i(\mathbf{w}^\top \mathbf{x} + b) = 0$
- $\alpha_i = 0$:
  - if $\mathbf{x}_i$ is not a support vector.
  - i.e., $1 - y_i(\mathbf{w}^\top \mathbf{x} + b) \neq 0$.
  Only support vectors influence the computation of $\mathbf{w}$.

Parametrically, it follows the KKT complimentary slackness condition of:

$$\alpha_i \Big(1 - y_i(\mathbf{w}^\top \mathbf{x} + b)\Big) = 0$$

10.3 Soft Margin

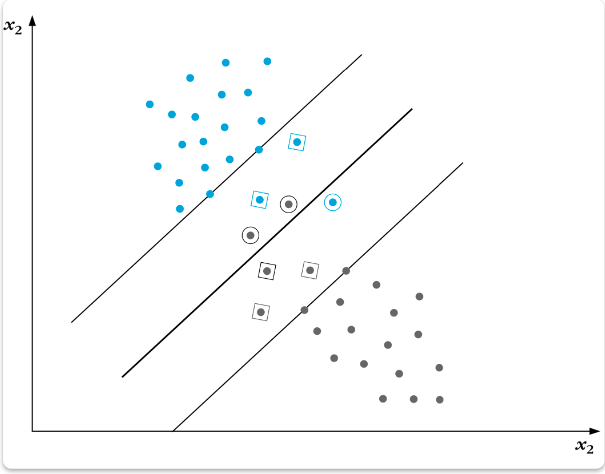
10.3.1 Problems

Remark: In the Primal Optimization Problem, we want to find a hyperplane such that:

$$\frac{1}{\|\mathbf{w}\|} \text{ is maximized}$$

$$\forall i,\ y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \text{ is maintained}$$

Problem: Such hyperplane may not always exist.



We assumed in 10.2 that the data is always separated enough.

- However, there may be cases where datas are slightly mixed together.
- i.e., not linearly separable.

The data samples are divided into 3 categories:

$$\text{Correctly Classified: } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1.$$

$$\text{Correct but violated margin: } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \in [0, 1].$$

$$\text{Incorrectly Classified: } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0.$$

10.3.2 Slack Variables 松弛因子

Key Point 1: Slack Variables

Since a unified limit can't cover all the points, we vary the limits for each point.

- We introduce slack variables $\xi_1, \xi_2, \cdots, \xi_N \geq 0$ to all data samples in $\mathcal{X}$.
- We allow the property to be violated by sample-wise manner.
- 允许这一性质被不同程度地违反。等于是给某些数据“开后门”。

$$y_i(\mathbf{x}^\top \mathbf{x} + b) \geq 1 - \xi_i$$

Where we assign:

$$\text{Correctly Classified: } \xi_i = 0$$

$$\text{Correct but violated margin: } \xi_i \in [0, 1]$$

$$\text{Incorrectly Classified: } \xi_i \geq 1$$

Given such permissions, even if some datapoints may violate the hard margin:

$$y_i(\mathbf{w}^\top \mathbf{x} + b) \geq 1$$

they do not violate the soft margin:

$$y_i(\mathbf{w}^\top \mathbf{x} + b) \geq 1 - \xi_i$$

Key Point 2: New Optimization & Parameter C

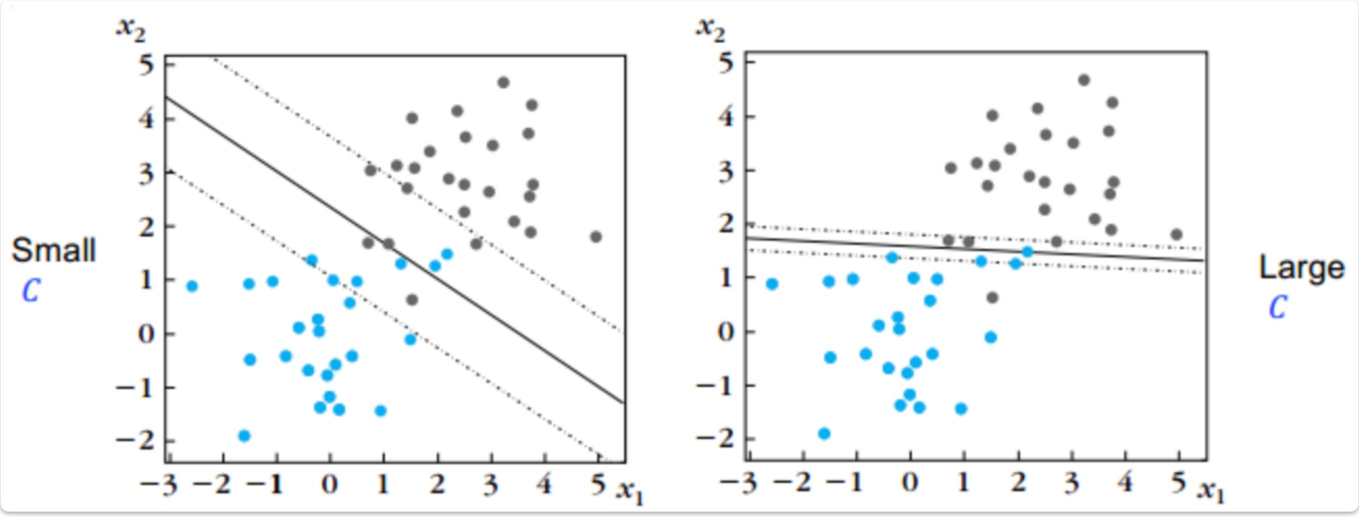
Find a hyperplane where:

$$\text{Minimize: } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i^2$$

$$\text{Maintain: } \forall i,\ y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$$

Here, parameter C is a user-selected regularization parameter.

- It is a tradeoff between:
  - A larger margin, and
  - A smaller classification error
- Effects of parameter C:
  - Smaller C: Larger Margin, More Error
  - Larger C: Smaller Margin, Fewer Error



10.3.3 L_2 Norm

Optimization Problem

$$\text{Minimize: } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i^2$$

$$\text{Maintain: } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$$

$$i = 1, \cdots, N$$

Primal Lagrangian

$$\mathcal{L}^{(L_2)} = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 + \sum_{i=1}^{N}\alpha_i\Big[(1-\xi_i) - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\Big]$$

Optimization of the primal Lagrangian:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \implies \mathbf{w} = \sum_{i=1}^{N}(\alpha_i \cdot y_i) \cdot \mathbf{x}_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \implies \sum_{i=1}^{N}\alpha_i \cdot y_i = 0$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} \implies \xi_i = \frac{\alpha_i}{C}$$

Dual Problem

$$\text{Max: } \mathcal{G}(\alpha) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i \cdot \alpha_j) \cdot (y_i \cdot y_j) \cdot (\mathbf{x}_i^\top \mathbf{x}_j + \frac{1}{C}\delta_{ij}) + \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.: } \sum_{i=1}^{N}\alpha_i \cdot y_i = 0$$

$$\alpha_i \geq C, \;\; i = 1, \cdots, N$$

where:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

10.3.4 L_1 Norm

Optimization Problem

$$\text{Minimize: } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i^2$$

$$\text{Maintain: } y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0,\ i = 1, \cdots, N$$

Primal Lagrangian

$$\mathcal{L}^{(L_1)} = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 + \sum_{i=1}^{N}\alpha_i\Big[(1-\xi_i) - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\Big] - \sum_{i=1}^{N}\beta_i\xi_i$$

$$= \mathcal{L}^{(L_2)} - \sum_{i=1}^{N}\beta_i\xi_i$$

Here, a second set of Lagrangian multipliers $\{\beta_i\}_{i=1}^{N}$ is introduced to ensure that $\xi_i \geq 0$.

Dual Problem

$$\text{Max: } \mathcal{G}(\alpha) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i \cdot \alpha_j)\cdot(y_i \cdot y_j)\cdot(\mathbf{x}_i^\top\mathbf{x}_j) + \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.: } \sum_{i=1}^{N}\alpha_i \cdot y_i = 0,$$

$$0 \leq \alpha_i \leq C,\ \ i = 1, \cdots, N$$