ML&MEA (2024)
Lecture 8 - SVM with soft margin

Quanying Liu

BME, SUSTech

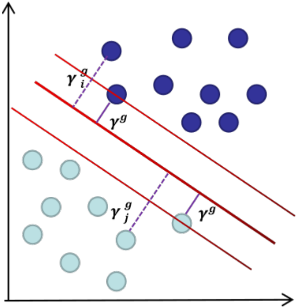
2024.3.21

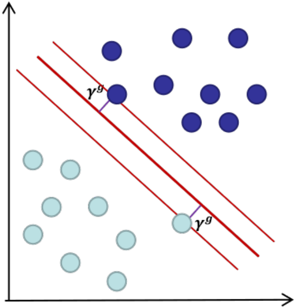
Content

1. Recap
2. Nonseparable patterns
3. Soft Margin
4. Solution
5. Summary

Maximizing the margin

In SVM, for a given dataset $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, the goal is to find the **optimal hyperplane** where has the **maximal margin** over all possible hyperplanes $(\mathbf{w}, b)$.



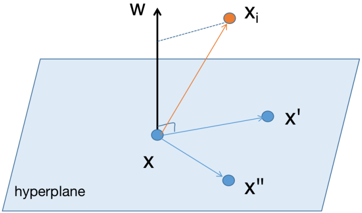


We have to define margin mathematically.

What is margin? Mathematically, ...

Hyperplane: $\mathbf{w}^T\mathbf{x} + b = 0$
Normalization: $|\mathbf{w}^T\mathbf{x}_i + b| = 1$

Any two points in the hyperplane:
$\mathbf{w}^T\mathbf{x} + b = 0, \quad \mathbf{w}^T\mathbf{x}' + b = 0$
$\implies \mathbf{w}^T(\mathbf{x} - \mathbf{x}') = 0$
$\implies \mathbf{w}$ is $\perp$ to the hyperplane



Unit vector $\hat{\mathbf{w}} = \dfrac{\mathbf{w}}{\|\mathbf{w}\|}$

$$\gamma_i = \text{distance} = |\hat{\mathbf{w}}^T(\mathbf{x}_i - \mathbf{x})| = \frac{1}{\|\mathbf{w}\|}|\mathbf{w}^T(\mathbf{x}_i - \mathbf{x})|$$
$$= \frac{1}{\|\mathbf{w}\|}|(\mathbf{w}^T\mathbf{x}_i + b) - (\mathbf{w}^T\mathbf{x} + b)| = \frac{1}{\|\mathbf{w}\|}$$

SVM: the constrained optimization problem

The constrained optimization problem:

$$\mathbf{w}^*, b^* = \arg\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|} \quad \text{s.t.,} \quad \min_{i=1,2,\ldots,n} |\mathbf{w}^T\mathbf{x}_i + b| = 1 \tag{1}$$

Notice: For a two-class SVM classification, $\mathbf{w}^T\mathbf{x}_i + b > 0, d_i = +1$; otherwise, $\mathbf{w}^T\mathbf{x}_i + b < 0, d_i = -1$. $\Longrightarrow d_i(\mathbf{w}^T\mathbf{x}_i + b) > 0$

$$\mathbf{w}^*, b^* = \arg\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} \quad \text{s.t.,} \quad d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, 2, \ldots, n \tag{2}$$

Non-constrained problem

The optimization problem for SVM in Eq. (2):

$$\min_{\mathbf{w},b} \ \frac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$\text{s.t.} \quad 1 - d_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 0$$

The **Lagrangian multipliers**: $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$
The **generalized Lagrangian function** is:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n} \alpha_i \left(1 - d_i\left(\mathbf{w}^T\mathbf{x}_i + b\right)\right) \tag{3}$$

The unconstrained optimization problem:

$$\min_{\mathbf{w},b} \max_{\boldsymbol{\alpha}} \ \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})$$
$$\text{s.t.} \quad \alpha_i > 0 \tag{4}$$

Dual problem

The unconstrained optimization problem in Eq. (4):

$$\min_{\mathbf{w},b} \max_{\boldsymbol{\alpha}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})$$
$$\text{s.t.} \quad \alpha_i > 0$$

Since Eq. (4) is unconstrained *w.r.t.* $\mathbf{w}$ and b, we would like to solve $\mathbf{w}$, b firstly, with $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0$, $\frac{\partial \mathcal{L}}{\partial b} = 0$.

The dual problem:

$$\max_{\boldsymbol{\alpha}} \min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}) \qquad (5)$$
$$\text{s.t.} \quad \alpha_i > 0$$

Watch video for proof:

`https://www.bilibili.com/video/BV1aE411o7qd?p=32`

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

Recall the generalized Lagrangian function:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n} \alpha_i \left(1 - d_i \left(\mathbf{w}^T\mathbf{x}_i + b\right)\right)$$

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

$$\begin{aligned}
\frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}} \left(\frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i d_i \mathbf{w}^T\mathbf{x}_i - b\sum_{i=1}^{n} \alpha_i d_i + \sum_{i=1}^{n} \alpha_i\right) \\
&= \frac{1}{2}\frac{\partial\left(\mathbf{w}^T\mathbf{w}\right)}{\partial \mathbf{w}} - \sum_{i=1}^{n} \alpha_i d_i \left(\frac{\partial\left(\mathbf{w}^T\mathbf{x}_i\right)}{\partial \mathbf{w}}\right) \\
&= \mathbf{w} - \sum_{i=1}^{n} \alpha_i d_i \mathbf{x}_i = \mathbf{0}
\end{aligned}$$

Solving $\frac{\partial \mathcal{L}}{\partial b} = 0$

Recall the generalized Lagrangian function:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n} \alpha_i \left(1 - d_i \left(\mathbf{w}^T\mathbf{x}_i + b\right)\right)$$

Considering $\frac{\partial \mathcal{L}}{\partial b} = 0$,

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \frac{\partial}{\partial b}\left(\frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i d_i \mathbf{w}^T\mathbf{x}_i - b\sum_{i=1}^{n} \alpha_i d_i + \sum_{i=1}^{n} \alpha_i\right)$$

$$= \sum_{i=1}^{n} \alpha_i d_i = 0$$

Solving the Dual problem

Therefore, it becomes a dual problem of $Q(\boldsymbol{\alpha})$

$$
\begin{aligned}
\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i \left(d_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) - 1 \right) \\
&= \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i d_i \mathbf{w}^T \mathbf{x}_i - b \sum_{i=1}^{n} \alpha_i d_i + \sum_{i=1}^{n} \alpha_i \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \\
&\triangleq Q(\boldsymbol{\alpha})
\end{aligned}
$$

We **maximize** $Q(\alpha)$ to get the optimal Lagrange multipliers $\boldsymbol{\alpha}^*$

The $\boldsymbol{\alpha}^*$

The optimal Lagrange multipliers $\boldsymbol{\alpha}^*$:

$$
\begin{aligned}
\boldsymbol{\alpha}^* &= \arg\max_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}) \\
&= \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j d_i d_j \mathbf{x}_i^T\mathbf{x}_j - \sum_{i=1}^{n}\alpha_i
\end{aligned}
$$

$$
\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i d_i = 0, \qquad \alpha_i \geq 0
$$

It can be solved by quadratic programming using **Linear Quadratic Solver**.

Non-separable Cases

Recall: Two classes of data are **linearly separable**, if and only if there exists a hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ that separates two classes.

In reality, there are non-separable cases. How to classify these two classes?

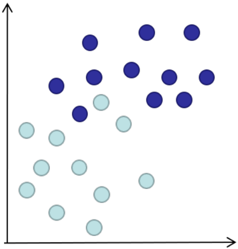
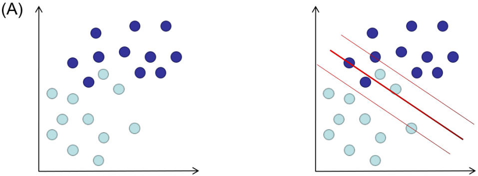


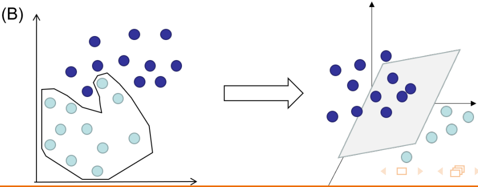
图 1: Non-separable cases

Soft Margin & Kernel Method

1. Find the optimal hyperplane to minimize classification error with some tolerance (Soft Margin).



2. Transform data into a higher-dimension space where two classes are separable (Kernel Method).



The non-negative slack variable ξ_i

Introduce an **non-negative slack variable** ξ_i for $i = \{1, \ldots, N\}$ that satisfies:

1. $\xi_i = 0$: Data point i is not inside the margin of separation
2. $0 \leq \xi_i \leq 1$: Data point is inside the margin of separation and on the correct side of the hyperplane (e.g., ξ_1 and ξ_2)
3. $\xi_i > 1$: Data point is inside the margin of separation but on the wrong side of the hyperplane (e.g., ξ_3 and ξ_4)

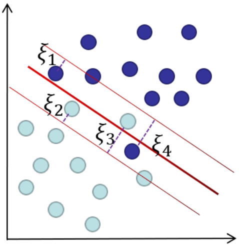
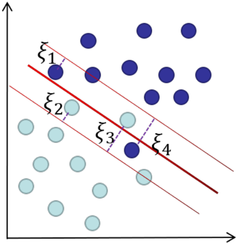


图 2: Slack variable ξ_i

The new loss function with Soft margin

The optimal hyperplane must penalize the classification error $\sum_{i=1}^{N} \xi_i$.
The new loss function $\mathcal{L}(\mathbf{w}, \boldsymbol{\xi})$ is:

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda \sum_{i=1}^{N} \xi_i$$

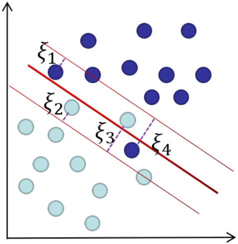


- λ is a hyperparameter, reflecting the cost of violating the margin constraints.
  1. A large λ generally leads to a smaller margin but also fewer misclassification of training data
  2. A small λ generally leads to a larger margin but more misclassification of training data
- As a design parameter, λ can be set by user with $\lambda > 0$.

The new constraints with Soft margin

Given a data $\mathbf{x}_i$, we have

$$d_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 - \xi_i$$



If $\xi_i = 0$, then the constraint is the same as that in basic version (Hard Margin).
SVM with soft margin:

Primal problem:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda\sum_{i=1}^{N}\xi_i \\ &s.t.\ \ d_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - 1 + \xi_i \geq 0 \\ &\xi_i \geq 0 \end{aligned} \tag{6}$$

Hinge loss: $\mathcal{L} = \max\{0, 1 - d\hat{y}\}$

Hinge loss $\mathcal{L}()$ is a loss function, which is usually used in the classification task of "maximum-margin".

$$\text{Hinge loss } \mathcal{L} = \max\{0, 1 - d\hat{y}\},$$

where d represents the label (-1 or 1), and $\hat{y}$ represents the prediction output. For a data point i:

$$\mathcal{L}(\mathbf{w}, b) = \max\{0, 1 - d_i\left(\mathbf{w}^T\mathbf{x}_i + b\right)\}$$

1. $1 - d_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \leq 0$: Data point i is classified correctly.
2. $1 - d_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) > 0$: Data point i is classified incorrectly.

Reading material: `https://zhuanlan.zhihu.com/p/347456667`

The dual problem with soft margin

Let α_i and β_i be the Lagrange multipliers for the two constraints.

The **generalized Lagrangian function** is: $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$

$$
\begin{aligned}
&= \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(d_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{N}\beta_i\xi_i \\
&= \frac{\mathbf{w}^T\mathbf{w}}{2} + \lambda\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i d_i\mathbf{w}^T\mathbf{x}_i - b\sum_{i=1}^{N}\alpha_i d_i + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\beta_i\xi_i
\end{aligned}
$$

The constrained Primal problem becomes the **dual problem**:

$$
\begin{aligned}
\min_{\mathbf{w},b,\xi}\ \max_{\alpha,\beta}\quad & \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
s.t.\quad & \alpha_i \geq 0, \quad \beta_i \geq 0
\end{aligned}
\tag{7}
$$

KKT conditions

The **KKT conditions** are as follows,

$$\partial \mathcal{L}/\partial \mathbf{w} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i d_i \mathbf{x}_i = \mathbf{0}$$

$$\partial \mathcal{L}/\partial b = -\sum_{i=1}^{N} \alpha_i d_i = 0$$

$$\partial \mathcal{L}/\partial \xi_i = \lambda - \alpha_i - \beta_i = 0$$

$$d_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) - 1 + \xi_i \geq 0$$

$$\alpha_i \left(d_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) - 1 + \xi_i \right) = 0$$

$$\beta_i \xi_i = 0$$

$$\alpha_i \geq 0$$

$$\beta_i \geq 0$$

Solving the dual problem: Step 1

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i d_i \mathbf{w}^T \mathbf{x}_i - b \sum_{i=1}^{N} \alpha_i d_i + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i \xi_i - \sum_{i=1}^{N} \beta_i \xi_i$$

We have derived:

$\partial \mathcal{L}/\partial \mathbf{w} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i d_i \mathbf{x}_i = \mathbf{0} \quad \Longrightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x}_i$

$\partial \mathcal{L}/\partial b = -\sum_{i=1}^{N} \alpha_i d_i = 0 \quad \Longrightarrow \sum_{i=1}^{N} \alpha_i d_i = 0$

$\partial \mathcal{L}/\partial \xi_i = \lambda - \alpha_i - \beta_i = 0 \quad \Longrightarrow \lambda = \alpha_i + \beta_i$

$$\mathbf{w}^T\mathbf{w} = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i d_i \alpha_j d_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\sum_{i=1}^{N} \alpha_i d_i \mathbf{w}^T \mathbf{x}_i = \sum^{N} \sum^{N} \alpha_i d_i \alpha_j d_j \mathbf{x}_i^T \mathbf{x}_j$$

Solving the dual problem: Step 2

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i d_i \mathbf{w}^T\mathbf{x}_i - b\sum_{i=1}^{N}\alpha_i d_i + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\beta_i\xi_i$$

$$= \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N}\alpha_i d_i \mathbf{w}^T\mathbf{x}_i + \lambda\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\beta_i\xi_i$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i d_i \alpha_j d_j \mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{N}\underbrace{(\alpha_i + \beta_i)}_{\lambda}\xi_i + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\beta_i\xi_i$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i d_i \alpha_j d_j \mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{N}\alpha_i \quad \triangleq Q(\boldsymbol{\alpha})$$

Solving the dual problem: Step 3

Formulate the Dual Problem (with soft margin):

$$\underset{\boldsymbol{\alpha}}{\text{Maximize}}: \quad Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{Subject to:} \quad \sum_{i=1}^{N} \alpha_i d_i = 0 \quad \text{and} \quad 0 \le \alpha_i \le \lambda$$

- Note: $Q(\boldsymbol{\alpha})$ is same as the dual problem without soft margin.
- The effect of the error penalty $\sum_{i=1}^{N} \xi_i$ on the optimization problem is to set an upper bound for Lagrange multipliers α_i.
- In KKT conditions, $\lambda = \alpha_i + \beta_i$ with $\alpha_i \ge 0$ and $\beta_i \ge 0$. $\Longrightarrow 0 \le \alpha_i \le \lambda$.

Solving α^* with Quadratic programming

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{N} \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i d_i = 0 \quad \text{and} \quad 0 \le \alpha_i \le \lambda$$

Solving with **Linear Quadratic Solver**:

$$\arg\min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^T \begin{bmatrix} d_1 d_1 \mathbf{x}_1^T \mathbf{x}_1 & d_1 d_2 \mathbf{x}_1^T \mathbf{x}_2 & \cdots & d_1 d_n \mathbf{x}_1^T \mathbf{x}_n \\ d_2 d_1 \mathbf{x}_2^T \mathbf{x}_1 & d_2 d_2 \mathbf{x}_2^T \mathbf{x}_2 & \cdots & d_2 d_n \mathbf{x}_2^T \mathbf{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ d_n d_1 \mathbf{x}_n^T \mathbf{x}_1 & d_n d_2 \mathbf{x}_n^T \mathbf{x}_2 & \cdots & d_n d_n \mathbf{x}_n^T \mathbf{x}_n \end{bmatrix} \boldsymbol{\alpha} + (-\mathbf{1})^T \boldsymbol{\alpha}$$

$$\text{s.t.} \quad \mathbf{d}^T \boldsymbol{\alpha} = 0 \quad \text{and} \quad \mathbf{0} \le \boldsymbol{\alpha} \le [\lambda, \lambda, \ldots, \lambda]^T$$

Using ‘cvxopt’ to solve α

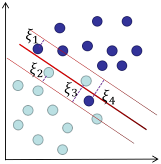
```
from cvxopt import matrix, solvers
Define the quadratic part of the objective function P and the linear part q
P = matrix([[?, ?], [?, ?]])
q = matrix([?, ?])

Define the inequality constraints G and h (Gx <= h)
G = matrix([[?],[?]])
h = matrix([?,?])

Define the equality constraints Ax = b (empty in this case)
A = matrix([[?, ?], (1,2)])
b = matrix(?)

Solve the quadratic programming problem
solution = solvers.qp(P, q, G, h, A, b)

print('Solution:')
print(solution['x'])
```

What is P, q, G, h, A, b in the code?

Where are the support vectors?

From KKT condtions:

$$d_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)-1+\xi_i\geq 0$$
$$\beta_i\xi_i=0$$
$$\alpha_i+\beta_i=\lambda$$



If the data $\mathbf{x}_i$ is a support vector (*i.e.*, , $0\leq\alpha_i\leq\lambda$), then

$$d_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)-1+\xi_i=0\Longrightarrow d_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)=1-\xi_i$$

- Support vector $\mathbf{x}_i$ with $\alpha_i<\lambda$ and $\beta_i=\lambda-\alpha_i>0$
  1. To satisfy $\beta_i\xi_i=0$, we must have $\xi_i=0$
  2. Support vector is located on the border of the margin
- Support vector $\mathbf{x}_i$ with $\alpha_i=\lambda$ and $\beta_i=\lambda-\alpha_i=0$
  1. Condition $\beta_i\xi_i=0$ is still satisfied, when $\xi_i\neq 0$
  2. Support vector is located inside the margin

SVM with Soft Margin

	Primal Problem	Dual Problem
Find :	$\mathbf{w}, b$	α_i
Minimizing :	$\mathcal{L}(\mathbf{w}, b, \xi)$	$\|\mathbf{w}\|$
Maximizing :	NaN	$Q(\boldsymbol{\alpha})$
Subject to :	$d_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 - \xi_i$	$\sum_{i=1}^{N} \alpha_i d_i = 0$
	$\xi_i \geq 0$	$0 \leq \alpha_i \leq \lambda$

Primal Problem: $$\mathcal{L}(\mathbf{w}, b, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda\sum_{i=1}^{N}\xi_i$$

Dual Problem: $$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j \mathbf{x}_i^T\mathbf{x}_j$$

Solving α^* with Quadratic programming

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j \mathbf{x}_i^T\mathbf{x}_j - \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.} \sum_{i=1}^{N}\alpha_i d_i = 0 \quad \text{and} \quad 0 \le \alpha_i \le \lambda$$

Solving with **Linear Quadratic Solver**:

$$\arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T \begin{bmatrix} d_1 d_1 \mathbf{x}_1^T\mathbf{x}_1 & d_1 d_2 \mathbf{x}_1^T\mathbf{x}_2 & \cdots & d_1 d_n \mathbf{x}_1^T\mathbf{x}_n \\ d_2 d_1 \mathbf{x}_2^T\mathbf{x}_1 & d_2 d_2 \mathbf{x}_2^T\mathbf{x}_2 & \cdots & d_2 d_n \mathbf{x}_2^T\mathbf{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ d_n d_1 \mathbf{x}_n^T\mathbf{x}_1 & d_n d_2 \mathbf{x}_n^T\mathbf{x}_2 & \cdots & d_n d_n \mathbf{x}_n^T\mathbf{x}_n \end{bmatrix} \boldsymbol{\alpha} + (-\mathbf{1})^T\boldsymbol{\alpha}$$

$$\text{s.t.} \quad \mathbf{d}^T\boldsymbol{\alpha} = 0 \quad \text{and} \quad \mathbf{0} \le \boldsymbol{\alpha} \le [\lambda, \lambda, \ldots, \lambda]^T$$

Algorithm

Step 1: Solve α^* with Linear Quadratic Solver.

Step 2: Calculate $\mathbf{w}^*$ as follows:

$$\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i^* d_i \mathbf{x}_i$$

Step 3: Calculate b^* as follows:

- For each data point $\mathbf{x}_i$ with $0 < \alpha_i \leq \lambda$,

$$b_i^* = \frac{1}{d_i} - {\mathbf{w}^*}^T \mathbf{x}_i$$

- Take b^* as the average of all such b_i^*

$$b^* = \frac{1}{m} \sum_{i=1}^{m} b_i^*$$

where m is the total number of $\mathbf{x}_i$ with $0 < \alpha_i \leq \lambda$.