# Lecture 12: Support Vector Machine

Shukai Gong

Given a dataset $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, where $\boldsymbol{x}_n \in \mathbb{R}^D$ and $y_n \in \{-1, 1\}$, the goal of SVM is to find a hyperplane $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ with largest separation such that the discriminative power is maximized and the error risk is minimized. As is shown in Figure 1, **the desired hyperplane has the largest distance to the nearest training data points of any class** so that the hyperplane is less likely to be influenced by out-of-sample data points and more robust.

The hyperplane is shifted along two opposite directions $\boldsymbol{w}^\top \boldsymbol{x} = b + d$ and $\boldsymbol{w}^\top \boldsymbol{x} = b - d$ to form a margin of width $\frac{2d}{\|w\|}$. The data points that are on the margin are called **support vectors**. Eventually, the SVM model can be written by

$$f^*(\boldsymbol{x}) = \text{sign}((\boldsymbol{w}^*)^\top \boldsymbol{x} + b^*)$$

where $\boldsymbol{w}^*$ and $b^*$ are optimal parameters.

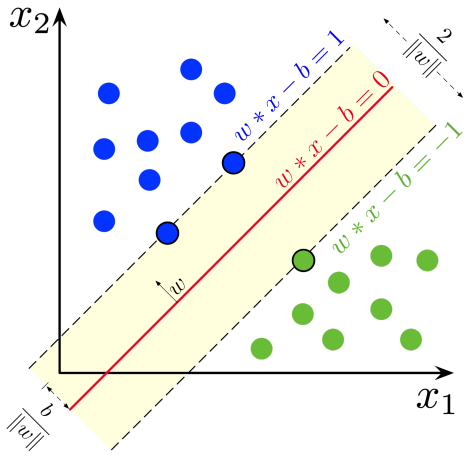


Figure 1: Hard-Margin Support Vector Machine

## 1 Hard-Margin SVM

### 1.1 Formulation of Optimization Goal

For a set of training data points $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N, y_i \in \{-1, +1\}$ that are **2-class linear separable**, the hard-margin SVM problem can be formulated as

$$\begin{aligned} &\max \text{margin}(\boldsymbol{w}, b), \text{ s.t. } \begin{cases} \boldsymbol{w}^\top \boldsymbol{x}_i + b \geq 0, & \text{if } y_i = 1, \\ \boldsymbol{w}^\top \boldsymbol{x}_i + b < 0, & \text{if } y_i = -1. \end{cases}, i = 1, \cdots, N \\ \iff &\max \text{margin}(\boldsymbol{w}, b), \text{ s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 0, i = 1, \cdots, N \end{aligned}$$

Margin can be defined as the shortest distance from the hyperplane to the nearest data point of any class, i.e.

$$\text{margin}(\boldsymbol{w}, b) = \min_{i=1,\cdots,N} \frac{|\boldsymbol{w}^\top \boldsymbol{x}_i + b|}{\|\boldsymbol{w}\|}$$

Therefore our optimization goal can be rewritten as

$$\begin{aligned}
&\max_{\boldsymbol{w},b} \min_{i=1,\cdots,N} \frac{|\boldsymbol{w}^\top \boldsymbol{x}_i + b|}{\|\boldsymbol{w}\|}, && \text{s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 0, i = 1, \cdots, N \\
\iff &\max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|} \min_{i=1,\cdots,N} y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b), && \text{s.t. } \exists r > 0, \min_{i=1,\cdots,N} y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = r, i = 1, \cdots, N \\
\iff &\max_{\boldsymbol{w},b} \frac{r}{\|\boldsymbol{w}\|}, && \text{s.t. } \exists r > 0, \min_{i=1,\cdots,N} y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = r, i = 1, \cdots, N
\end{aligned}$$

Without loss of generality, we can set $r = 1$ since it is just a scaling factor and can be absorbed into $\boldsymbol{w}$ and $b$. Therefore, the optimization problem can be rewritten as:

$$\begin{aligned}
&\max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|}, && \text{s.t. } \min_{i=1,\cdots,N} y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 1, i = 1, \cdots, N \\
\iff &\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2, && \text{s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, i = 1, \cdots, N
\end{aligned}$$

## 1.2 Dual Problem Optimization

The primal problem can be solved by the Lagrange multiplier method, and the Lagrangian function is a convex optimization problem with $N$ linear constraints,

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2, \text{ s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, i = 1, \cdots, N$$

It's **equivalent** to an unconstrained primal problem with $N$ Lagrange multipliers $\lambda_i \geq 0$,

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) = \min_{\boldsymbol{w},b} \max_{\boldsymbol{\lambda}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \underbrace{\lambda_i}_{\geq 0} \underbrace{(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b))}_{\leq 0}$$

$$\Rightarrow \begin{cases} \min_{\boldsymbol{w},b} \max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) \\ \text{s.t. } \lambda_i \geq 0, i = 1, \cdots, N \end{cases}$$

**Non-negative Lagrangian Multipliers $\lambda_i$**

For arbitary data point $(\boldsymbol{x}_i, y_i)$,

- If $1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) > 0$, then $\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \infty = \infty$. (Meaningless!)
- If $1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq 0$, then $\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{w}\|^2$. (Set all $\lambda_i = 0$ maximizes the non-positive term $\lambda_i(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b))$)

So

$$\min_{\boldsymbol{w},b} \max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) = \min_{\boldsymbol{w},b}(\infty, \frac{1}{2}\|\boldsymbol{w}\|^2) = \min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$

By setting the Lagrangian multipliers $\lambda_i \geq 0$ for all $i = 1, \cdots, N$, we made an equivalent optimization problem to the primal problem.

It can be proved that the dual problem is **equivalent** to the primal problem when the optimization goal is convex and the constraints are linear **(strong duality)**, i.e.

$$\begin{cases} \min_{\boldsymbol{w},b} \max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) \\ \text{s.t. } \lambda_i \geq 0, i = 1, \cdots, N \end{cases} = \begin{cases} \max_{\boldsymbol{\lambda}} \min_{\boldsymbol{w},b} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) \\ \text{s.t. } \lambda_i \geq 0, i = 1, \cdots, N \end{cases}$$

First, we take the derivative of $\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda})$ w.r.t $\boldsymbol{w}$ and $b$ and set them to zero,

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum\limits_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i = 0 \\ \dfrac{\partial \mathcal{L}}{\partial b} = -\sum\limits_{i=1}^{N} \lambda_i y_i = 0 \end{cases} \Rightarrow \begin{cases} \boldsymbol{w} = \sum\limits_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i \\ \sum\limits_{i=1}^{N} \lambda_i y_i = 0 \end{cases}$$

Plugging these back to $\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda})$, we have

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

So the duality problem is simplified to

$$\begin{cases} \max\limits_{\boldsymbol{\lambda}} & \sum\limits_{i=1}^{N} \lambda_i - \dfrac{1}{2} \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j \\ \text{s.t.} & \lambda_i \geq 0, i = 1, \cdots, N \\ & \sum\limits_{i=1}^{N} \lambda_i y_i = 0 \end{cases} = \begin{cases} \min\limits_{\boldsymbol{\lambda}} & \dfrac{1}{2} \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j - \sum\limits_{i=1}^{N} \lambda_i \\ \text{s.t.} & \lambda_i \geq 0, i = 1, \cdots, N \\ & \sum\limits_{i=1}^{N} \lambda_i y_i = 0 \end{cases}$$

A sufficient and necessary condition for **strong duality** (we use without proof) is that the primal problem satisfies the **Karush-Kuhn-Tucker (KKT) conditions**. The KKT conditions for this specific problem are

$$\begin{cases} \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) = 0 & \Rightarrow \boldsymbol{w}^* = \sum\limits_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i \\ \nabla_b \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) = 0 & \Rightarrow \sum\limits_{i=1}^{N} \lambda_i y_i = 0 \\ \lambda_i (1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) = 0, & i = 1, \cdots, N \\ \lambda_i \geq 0, & i = 1, \cdots, N \\ 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq 0, & i = 1, \cdots, N \end{cases}$$

Let's focus on the third condition (**complementary slackness condition**). For arbitary data point $(\boldsymbol{x}_i, y_i)$,

- If the data point is a support vector, i.e. $1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 0$, then $\lambda_i > 0$.
- If the data point is beyond the margin, i.e. $1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) < 0$, then $\lambda_i = 0$.

which means that non-support-vector data points won't show up in $\boldsymbol{w}^* = \sum\limits_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i$ since their $\lambda_i = 0$. Taking advantage of this, by picking one support vector $(\boldsymbol{x}_k, y_k)$ , we can derive $b$ by

$$1 - y_k(\boldsymbol{w}^\top \boldsymbol{x}_k + b) = 0 \Rightarrow y_k^2(\boldsymbol{w}^\top \boldsymbol{x}_k + b) = y_k \Rightarrow b^* = y_k - \boldsymbol{w}^\top \boldsymbol{x}_k$$

The hard-margin SVM is therefore

$$f^*(\boldsymbol{x}) = \text{sign}((\boldsymbol{w}^*)^\top \boldsymbol{x} + b^*), \quad \begin{cases} \boldsymbol{w}^* = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i \\ b^* = y_k - \boldsymbol{w}^\top \boldsymbol{x}_k \end{cases}$$

which only **depends on the support vectors** and the corresponding Lagrange multipliers.

## 2 Soft-Margin SVM

Hard-margin SVM is too idealized about the separability of the data points. In practice, the data points are usually either not linearly separable, or subject to noise. As is shown in Figure 2, Soft-margin SVM **allows for some data points to be misclassified** by introducing a slack variable $\xi_i \geq 0$ for each data point $(\boldsymbol{x}_i, y_i)$ to measure the degree of misclassification.

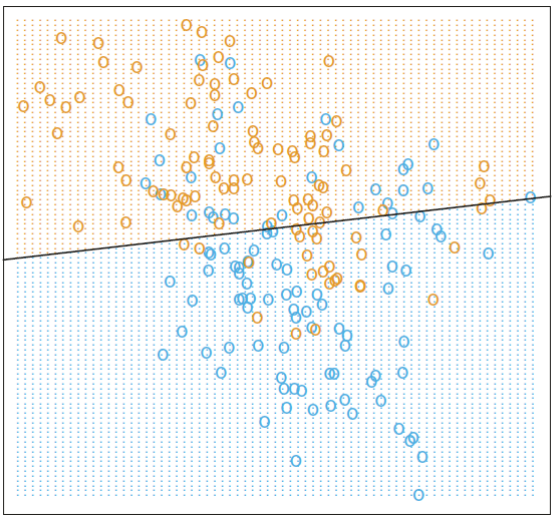

Figure 2: Soft-Margin Support Vector Machine

Recall that $y_i(\boldsymbol{w}^\top \boldsymbol{x} + b) \geq 1$ indicates that the data point is correctly classified. The misclassification is presented by $y_i(\boldsymbol{w}^\top \boldsymbol{x} + b) < 1$. Some methods of defining slackness $\xi_i$ is presented below:

- **Counts**: $\xi_i = \mathbf{I}(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) < 1)$ (Discontinuous!)
- **Hinge Loss**: $\xi_i = \max(0, 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b))$ (Recommended)

We can formulate the optimization problem as

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)), \qquad \text{s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, i = 1, \cdots, N$$

$$\iff \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \xi_i, \qquad \text{s.t.} \begin{cases} y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, & i = 1, \cdots, N \\ \xi_i \geq 0, & i = 1, \cdots, N \end{cases}$$

Similar to the hard-margin SVM, we construct the Lagrangian function and derive its dual problem.

$$
\begin{aligned}
&\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \max_{\boldsymbol{\lambda},\boldsymbol{\mu}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\lambda_i(1-\xi_i - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) - \sum_{i=1}^{N}\mu_i\xi_i \\
=&\max_{\boldsymbol{\lambda},\boldsymbol{\mu}} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\lambda_i(1-\xi_i - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) - \sum_{i=1}^{N}\mu_i\xi_i \\
\Rightarrow&\begin{cases}
\dfrac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum\limits_{i=1}^{N}\lambda_i y_i \boldsymbol{x}_i = 0 \\
\dfrac{\partial \mathcal{L}}{\partial b} = -\sum\limits_{i=1}^{N}\lambda_i y_i = 0 \\
\dfrac{\partial \mathcal{L}}{\partial \xi_i} = C - \lambda_i - \mu_i = 0
\end{cases} \Rightarrow \begin{cases}
\boldsymbol{w} = \sum\limits_{i=1}^{N}\lambda_i y_i \boldsymbol{x}_i \\
\sum\limits_{i=1}^{N}\lambda_i y_i = 0 \\
\lambda_i + \mu_i = C
\end{cases}
\end{aligned}
$$

and the dual problem of the primal soft-margin SVM is

$$
\begin{aligned}
&\begin{cases}
\max\limits_{\boldsymbol{\lambda}} & \mathcal{L}(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}) \\
\text{s.t.} & \lambda_i \geq 0, i = 1,\cdots,N \\
& \mu_i \geq 0, i = 1,\cdots,N
\end{cases} = \begin{cases}
\min\limits_{\boldsymbol{\lambda}} & \frac{1}{2}\sum_{i=1}^{N}\sum\limits_{j=1}^{N}\lambda_i\lambda_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j - \sum\limits_{i=1}^{N}\lambda_i \\
\text{s.t.} & \lambda_i \geq 0, \mu_i \geq 0, i = 1,\cdots,N \\
& \sum\limits_{i=1}^{N}\lambda_i y_i = 0 \\
& \lambda_i + \mu_i = C, i = 1,\cdots,N
\end{cases} \\
&= \begin{cases}
\min\limits_{\boldsymbol{\lambda}} & \frac{1}{2}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N}\lambda_i\lambda_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j - \sum\limits_{i=1}^{N}\lambda_i \\
\text{s.t.} & 0 \leq \lambda_i \leq C, i = 1,\cdots,N \\
& \sum\limits_{i=1}^{N}\lambda_i y_i = 0
\end{cases}
\end{aligned}
$$

Compare to the hard-margin SVM case, the soft-margin SVM has an additional constraint $\lambda_i \leq C$ and $\mu_i \geq 0$ to control the trade-off between the margin and the misclassification. The KKT condition for the primal soft-margin SVM is

$$
\begin{cases}
\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}) = 0 & \Rightarrow \boldsymbol{w}^* = \sum\limits_{i=1}^{N}\lambda_i y_i \boldsymbol{x}_i \\
\nabla_b\mathcal{L}(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}) = 0 & \Rightarrow \sum\limits_{i=1}^{N}\lambda_i y_i = 0 \\
\nabla_{\xi_i}\mathcal{L}(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}) = 0 & \Rightarrow \lambda_i + \mu_i = C, i = 1,\cdots,N \\
\lambda_i(1-\xi_i - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) = 0, & i = 1,\cdots,N \\
\mu_i\xi_i = 0, & i = 1,\cdots,N \\
\lambda_i \geq 0, \mu_i \geq 0, \xi_i \geq 0, & i = 1,\cdots,N \\
1 - \xi_i - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq 0, & i = 1,\cdots,N
\end{cases}
$$

Let's break down this huge KKT conditions: For arbitary data point $(\boldsymbol{x}_i, y_i)$, either $\lambda_i = 0$ or $1-\xi_i - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 0$.

$$
\begin{cases}
\text{If } \lambda_i = 0 \Rightarrow \text{No effect on } \boldsymbol{w}^* \\
\text{If } \lambda_i > 0 \Rightarrow (\boldsymbol{x}_i, y_i) \text{ is a SV} \begin{cases}
\text{If } \lambda_i < C \Rightarrow \mu_i > 0 \Rightarrow \xi_i = 0 \Rightarrow b^* = y_i - \boldsymbol{w}^\top \boldsymbol{x}_i \\
\text{If } \lambda_i = C \Rightarrow \mu_i = 0 \begin{cases}
\text{If } \xi_i \leq 1 \Rightarrow \text{Tolerated Misclassification} \\
\text{If } \xi_i > 1 \Rightarrow \text{Misclassification}
\end{cases}
\end{cases}
\end{cases}
$$

So we need to find a support vector $(\boldsymbol{x}_k, y_k)$ with $0 < \lambda_k < C$ to derive $b^*$. Still, the soft-margin SVM is

$$f^*(\boldsymbol{x}) = \text{sign}((\boldsymbol{w}^*)^\top \boldsymbol{x} + b^*), \quad \begin{cases} \boldsymbol{w}^* = \sum_{i=1}^N \lambda_i y_i \boldsymbol{x}_i \\ b^* = y_k - \boldsymbol{w}^\top \boldsymbol{x}_k \end{cases}$$

# 3 Kernel SVM

We have assumed in hard and soft SVM that data points are linearly separable. For non-linearly separable data points, we can use the **kernel trick to map the data points to a higher-dimensional feature space where they are linearly separable.**

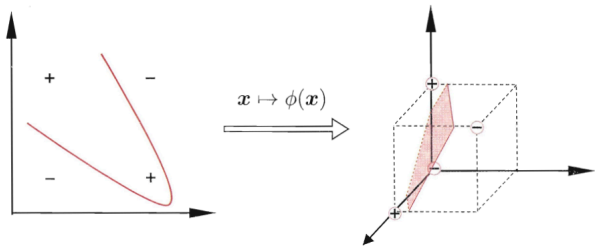


Figure 3: Kernel Trick in SVM

Denote $\phi(\boldsymbol{x})$ as the feature vector of $\boldsymbol{x}$ in the higher-dimensional space, the SVM model corresponding to the hyperplane in the feature space is $\boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = 0$ and the optimization problem can be formulated as (take soft-margin SVM as an example)

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^N \xi_i, \qquad \text{s.t.} \begin{cases} y_i(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, & i = 1, \cdots, N \\ \xi_i \geq 0, & i = 1, \cdots, N \end{cases}$$

The dual problem of the primal kernel soft-margin SVM is

$$\begin{cases} \min\limits_{\boldsymbol{\lambda}} & \frac{1}{2}\sum\limits_{i=1}^N \sum\limits_{j=1}^N \lambda_i \lambda_j y_i y_j \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) - \sum\limits_{i=1}^N \lambda_i \\ \text{s.t.} & 0 \leq \lambda_i \leq C, i = 1, \cdots, N \\ & \sum\limits_{i=1}^N \lambda_i y_i = 0 \end{cases}$$

**We can observe that the only difference between the kernel SVM and the linear SVM is that the dot product $\boldsymbol{x}_i^\top \boldsymbol{x}_j$ is replaced by the kernel function $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$.** As an analogy, $\boldsymbol{w}^* = \sum_{i=1}^N \lambda_i y_i \phi(\boldsymbol{x}_i)$ and $b^* = y_k - \boldsymbol{w}^\top \phi(\boldsymbol{x}_k)$, and the kernel SVM model is written by

$$\begin{aligned} f^*(\boldsymbol{x}) =& \text{sign}((\boldsymbol{w}^*)^\top \phi(\boldsymbol{x}) + b^*) \\ =& \text{sign}\left(\sum_{i=1}^N \lambda_i y_i \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}) + b^*\right) \\ =& \text{sign}\left(\sum_{i=1}^N \lambda_i y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b^*\right) \end{aligned}$$

**Similar to other kernel methods, we don't have to find $\phi$ explicitly, but only need to introduce the kernel function $\kappa$ to derive the hyperplane.**

## 4 Noticable Facts about SVM

- Both primal and dual problems are quadratic programming problems, which can be solved by numerical methods. Ultimately we need to know certain $\lambda_k$ to derive the hyperplane.
- When $C \to \infty$, the soft-margin SVM "behaves like" the hard-margin SVM. (From $0 < \lambda_k < C$ to $0 < \lambda_k$)
- Linear SVM is a special case of kernel SVM with a linear kernel: $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$.

## References


- Machine Learning, Zhi-Hua Zhou
- Support Vector Machine