Support Vector Machines (SVM)

1. Problem Setup

We are given a dataset:

$$\{(x_i, y_i)\}_{i=1}^{N}$$

Where:
- $x_i \in R^d \rightarrow$ feature vector
- $y_i \in \{-1, +1\} \rightarrow$ class label

The goal of SVM is to find a decision boundary that separates the two classes with the maximum margin.

2. Decision Function

A linear SVM models a hyperplane:

$$f(x) = w \cdot x + b$$

- $w \rightarrow$ weight vector (controls orientation)
- $b \rightarrow$ bias (controls position)

Prediction rule:

$$\hat{y} = \text{sign}(w \cdot x + b)$$

3. Margin and Geometric Interpretation

The margin is the distance between the hyperplane and the closest data points.

- For a point x_i:

$$\text{Functional margin} = y_i(w \cdot x_i + b)$$

- SVM enforces:

$$y_i(w \cdot x_i + b) \geq 1$$

- This defines two margin boundaries:

$$w \cdot x + b = +1 \quad \text{and} \quad w \cdot x + b = -1$$

- Distance between them:

$$\text{Margin width} = \frac{2}{||w||}$$

Maximizing the margin $\Leftrightarrow$ minimizing $||w||$.

4. Hard-Margin SVM (Linearly Separable Case)

Objective:

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

Subject to:

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

This finds the maximum-margin separating hyperplane.

---

5. Soft-Margin SVM (Non-Separable Data)

Introduce slack variables $\xi_i \geq 0$:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

Optimization problem:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i$$

- C controls trade-off between:
  - large margin
  - classification errors.

---

6. Hinge Loss

The soft-margin SVM can be written using hinge loss:

$$L(y, f(x)) = \max(0, 1 - y f(x))$$

Total loss:

$$L = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y_i(w \cdot x_i + b))$$

This is what gradient-descent SVM implementations minimize

7. Support Vectors

Support vectors are data points for which:

$$y_i (w \cdot x_i + b) \leq 1$$

They lie:

- on the margin
- or inside the margin

Only these points influence:

- the position of the hyperplane
- the final model

All other points are irrelevant once the margin is set.

8. Dual Formulation (Key Insight)

Using Lagrange multipliers α_i, the dual problem becomes:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

Subject to:

$$0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i y_i = 0$$

Weight vector:

$$w = \sum_i \alpha_i y_i x_i$$

Only points with $\alpha_i > 0$ are support vectors.

9. Kernel Trick

Replace dot product:

$$x_i \cdot x_j \rightarrow K(x_i, x_j)$$

Decision function becomes:

$$f(x) = \sum_i \alpha_i y_i K(x_i, x) + b$$

This allows SVM to create non-linear decision boundaries.

10. Common Kernels

- Linear:

$$K(x, z) = x \cdot z$$

- Polynomial:

$$K(x, z) = (x \cdot z + c)^d$$

- RBF (Gaussian):

$$K(x, z) = \exp(-\gamma \|x - z\|^2)$$

Why SVM is Powerful

- Maximizes margin → better generalization
- Depends only on support vectors
- Works well in high dimensions
- Kernel trick handles complex non-linear data

- <u>One-Line Summary</u>

SVM finds the hyperplane with the largest margin by solving a constrained optimization problem, and kernels allows this idea to work in non-linear spaces.