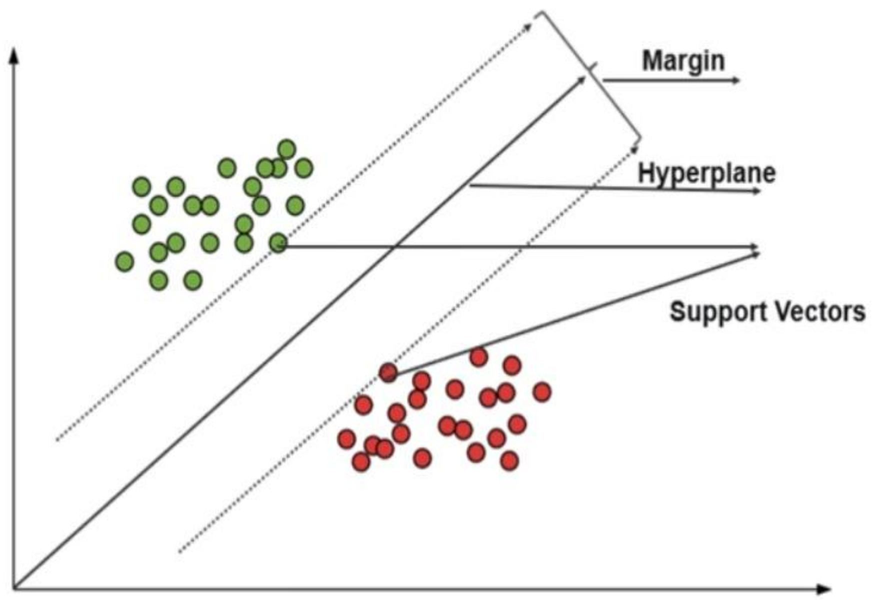
Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression. In classification, SVM finds a hyperplane that separates classes with the **maximum margin**. Unlike logistic regression, SVM focuses on maximizing geometric separation rather than modeling probabilities.

The core idea is:

Find the hyperplane that maximizes the distance between the closest points of different classes.



1. Problem Formulation

This section defines the classification problem mathematically.

Input Space $\mathcal{X} \subseteq \mathbb{R}^p$

Each input vector: $x \in \mathbb{R}^p$

Output Space $\mathcal{Y} = \{-1, +1\}$

Binary classification setting.

Data Distribution

Training samples: $(x_i, y_i) \sim P_{X,Y}$ are assumed i.i.d.

Learning Objective (Expected Risk Minimization)

Minimize classification error while maximizing margin:

$$\min_{w,b} \ \mathbb{E}[\ell(y, w^T x + b)]$$

Approximated using empirical risk.

2. Model Specification

Hypothesis Function : $f(x) = w^T x + b$

Decision rule: $\hat{y} = \text{sign}(w^T x + b)$

Parameter Space $w \in \mathbb{R}^p, b \in \mathbb{R}$

Structural Assumptions

1. Linear separability (for hard margin case)
2. Margin maximization principle
3. Only support vectors influence solution

3. Loss Function

This section defines how classification error is measured.

Hinge Loss

$$\ell(y, w^T x) = \max(0, 1 - y(w^T x + b))$$

Why This Loss?

1. Convex surrogate for 0–1 loss
2. Enforces margin constraint
3. Ignores correctly classified points beyond margin

Convexity

Hinge loss is convex but not differentiable at:

$$y(w^T x + b) = 1$$

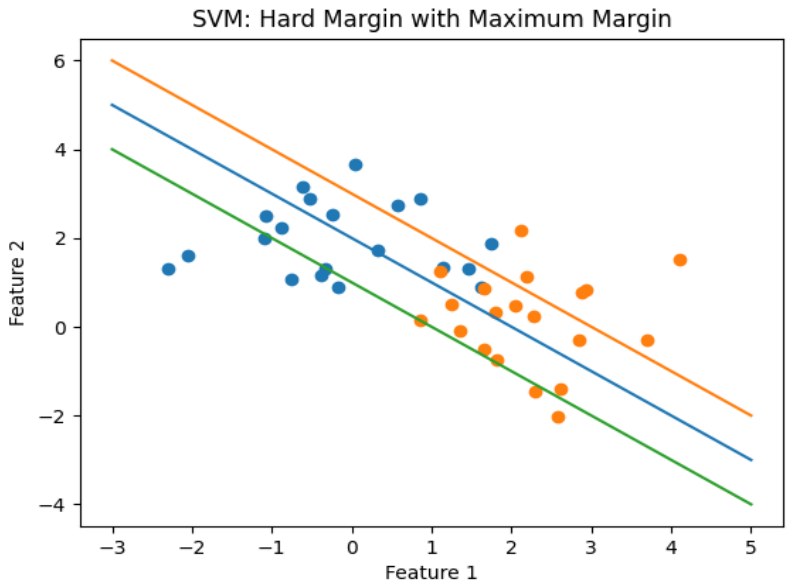
4. Objective Function

Hard Margin SVM (Linearly Separable Case)

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

subject to: $y_i(w^T x_i + b) \geq 1$

Maximizing margin is equivalent to minimizing $\| w \|^2$.



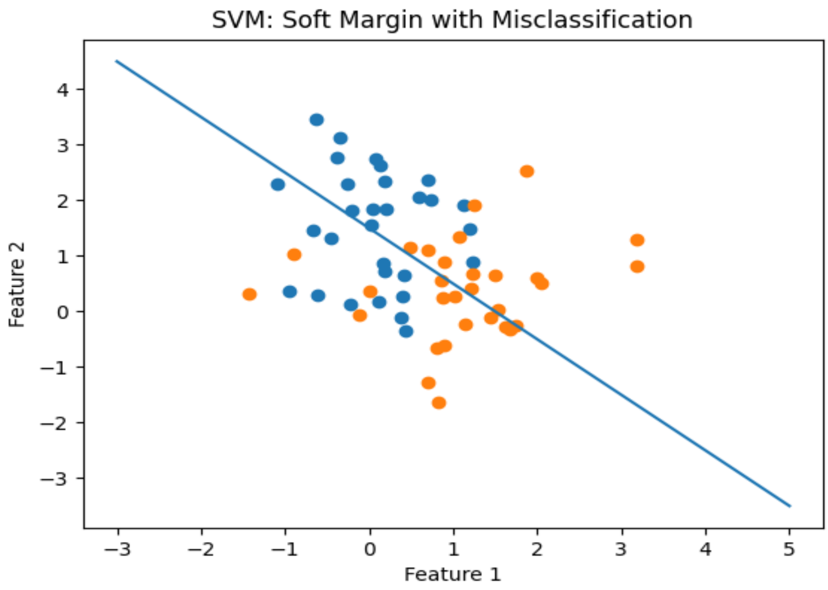
Soft Margin SVM (Non-separable Case)

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

subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

C controls trade-off between margin and misclassification.



5. Optimization Method

SVM is solved using:

- Quadratic Programming (QP)
- Dual formulation
- Sequential Minimal Optimization (SMO)

Dual Formulation

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \, \alpha_j y_i y_j x_i^T x_j$$

subject to:

$$\sum_{i=1}^{n} \alpha_i \, y_i = 0$$
$$0 \leq \alpha_i \leq C$$

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

Computational Complexity

Training:

- Between $O(n^2)$and $O(n^3)$
- Depends on solver

Inference: $O(s)$

where s= number of support vectors.

6. Statistical Interpretation

Maximum Margin Principle

SVM maximizes geometric margin:

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

Minimizing $\| w \|^2$maximizes margin.

Regularization Interpretation

Objective:

$$\frac{1}{2} \| w \|^2 + C\sum \text{hinge loss}$$

This is equivalent to:

- Regularized risk minimization
- Structural Risk Minimization principle

Probabilistic Meaning

Standard SVM does NOT output probabilities.
Probabilities can be estimated using Platt scaling.

7. Regularization & Generalization

Bias–Variance Tradeoff

- Small C → larger margin → higher bias, lower variance
- Large C → smaller margin → lower bias, higher variance

Capacity Control

Controlled by:

- Margin size
- Regularization parameter C

SVM has strong generalization due to margin maximization.

8. Theoretical Properties

Convexity / Global Optimality

Optimization problem is convex → unique global solution.

Consistency

Under appropriate conditions, SVM is statistically consistent.

Stability

Solution depends only on support vectors.
Removing non-support vectors does not change model.

9. Computational Complexity

Training Time

- Quadratic programming
- Typically $O(n^2)$or worse

Inference Time $O(s)$

where s = number of support vectors.

Memory Complexity

- Must store support vectors
- Memory = $O(s \cdot p)$

10. Limitations

When It Fails

1. Very large datasets (training slow)
2. Poor choice of kernel

3. Heavy overlap between classes
4. Extreme class imbalance

Assumption Violations

- Data not separable in chosen feature space
-

Sensitivity Issues

- Sensitive to choice of C
- Sensitive to kernel parameters
- Requires feature scaling