SVM

SVM, or Support Vector Machine, is a popular supervised machine learning algorithm used for both classification and regression tasks. It's particularly effective for classification problems and is widely used in various fields.

Basic Concept

SVM aims to find the best possible decision boundary (hyperplane) that separates different classes in the feature space. This hyperplane should maximize the margin between the closest points (support vectors) of different classes.

Margin

The distance between the hyperplane and the nearest data point from either class. SVM tries to maximize this margin.

Support Vectors

These are the data points nearest to the hyperplane that influence its position and orientation. They "support" the hyperplane.

Advantages

- Effective in high-dimensional spaces
- Memory efficient
- Versatile (different kernel functions for various decision boundaries)

Disadvantages

- Not suitable for large datasets (computationally expensive)
- Sensitive to noise

Hard Margin SVM

1. Definition: Hard margin SVM assumes that the data is linearly separable in the feature space.
2. Goal: Find the maximum margin hyperplane that perfectly separates the classes without any misclassification.
3. Mathematical Formulation: Minimize: $(1/2)||w||^2$ Subject to: $y_i(w^T x_i + b) \geq 1$ for all i
4. Characteristics:
   - No tolerance for misclassification
   - Works only with linearly separable data
   - Very sensitive to outliers
5. Advantages:
   - Simple and intuitive
   - Provides a unique solution
6. Disadvantages:
   - Not applicable to non-linearly separable data
   - Overfitting in the presence of noise or outliers

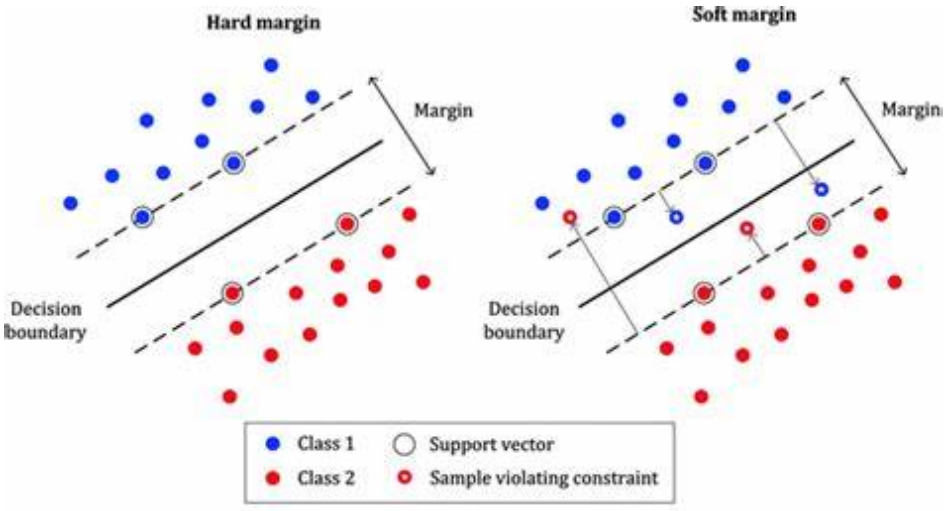
Soft Margin SVM

1. Definition: Soft margin SVM allows for some misclassification, making it more robust and applicable to non-linearly separable data.
2. Goal: Find a balance between maximizing the margin and minimizing the classification error.
3. Mathematical Formulation: Minimize: $(1/2)||w||^2 + C \Sigma_i \xi_i$ Subject to: $y_i(w^T x_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

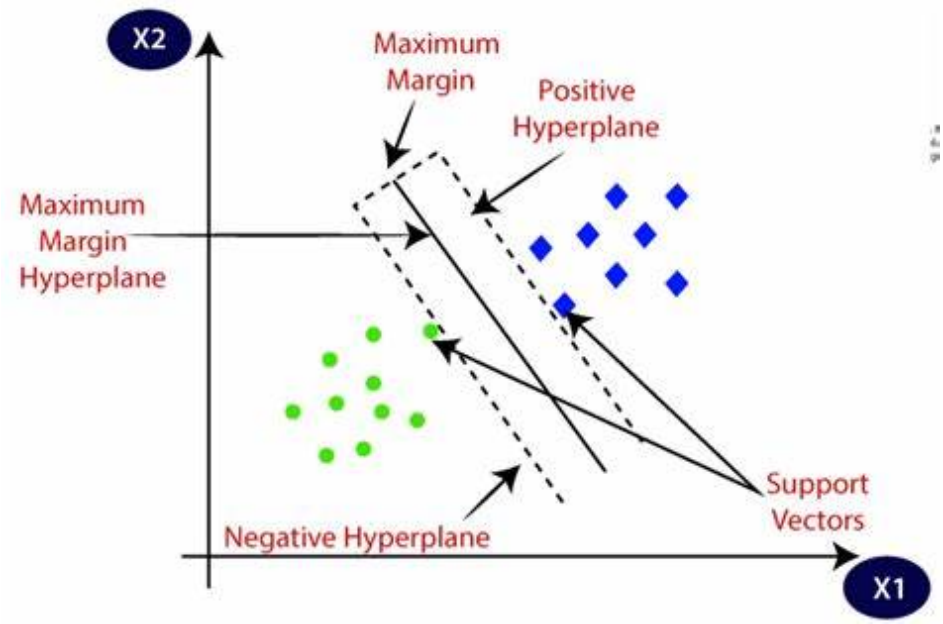
   Where:
   - ξ_i are slack variables
   - C is the regularization parameter
4. Characteristics:
   - Tolerates some misclassification

- Can handle non-linearly separable data
- Less sensitive to outliers

5. Advantages:
    - More flexible and generalizable
    - Can handle noisy data and outliers
    - Applicable to a wider range of real-world problems
6. Disadvantages:
    - Requires tuning of the C parameter
    - May not provide a unique solution



Math behind



1. Linear Hyperplane Equation:

$$w^T x + b = 0$$

2. Decision Function:

$$f(x) = sign(w^T x + b)$$

3. Linear SVM Classifier:

$$For\ y = +1 : w^T x + b \geq +1$$

$$For\ y = -1 : w^T x + b \leq -1$$

4. Combined Constraint:

$$y(w^T x + b) \geq 1$$

5. Proof of w being orthogonal to the hyperplane:

- For two points u1 and u2 on the hyperplane:

$$-$$

$$w^T u1 + b = 0$$

$$w^T u2 + b = 0$$

- Subtracting these equations:

$$w^T(u1 - u2) = 0$$

- Since (u1 - u2) is on the hyperplane and w^T (u1 - u2) = 0, w is orthogonal to the hyperplane.

6. Distance Calculation:

$$distance = |x(support) - u1| * (w/|w|)$$

Adding and subtracting b:

$$distance = |w^T x(support) + b - u1^T w - b| * (1/|w|)$$

Given:

$$w^T x(support) + b = 1 (on\ support\ vector)$$

$$u1^T w + b = 0 (on\ hyperplane)$$

Therefore:

$$distance = |1 - 0| * (1/|w|) = 1/|w|$$

7. Margin:

$$Margin = 2/||w||$$

8. Optimization Problem (Hard Margin):

Minimize:

$$(1/2)||w||^2$$

Subject to:

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

9. Optimization Problem (Soft Margin):

Minimize:

$$(1/2)||w||^2 + C\Sigma_i\xi_i$$

Subject to:

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

Where:

- ξ_i are slack variables
- C is the regularization parameter

10. Lagrangian Formulation:

$$L(w, b, \xi, \alpha, r) = (1/2)||w||^2 + C\Sigma_i\xi_i - \Sigma_i\alpha_i[y_i(w^T x_i + b) - 1 + \xi_i] - \Sigma_i r_i\xi_i$$

Where:

- α_i and r_i are Lagrange multipliers

11. KKT (Karush-Kuhn-Tucker) Conditions:

$$\partial L/\partial w = 0 => w = \Sigma_i\alpha_i y_i x_i$$

$$\partial L/\partial b = 0 => \Sigma_i\alpha_i y_i = 0$$

$$\partial L/\partial \xi_i = 0 => C = \alpha_i + r_i$$

12. Dual Problem: Maximize:

$$\Sigma_i\alpha_i - (1/2)\Sigma_i\Sigma_j\alpha_i\alpha_j y_i y_j (x_i^T x_j)$$

Subject to:

$$0 \leq \alpha_i \leq C, \Sigma_i\alpha_i y_i = 0$$

kernel trick

The kernel trick is a fundamental concept in Support Vector Machines (SVMs) and other machine learning algorithms. It's a clever mathematical technique that allows SVMs to operate in high-dimensional feature spaces without explicitly computing the coordinates in that space.

Mathematical Formulation

$$K(x, y) = \varphi(x) \cdot \varphi(y)$$

Where:

- K is the kernel function
- φ is the mapping from input space to feature space
- x and y are input vectors

Common Kernel Functions

1. Linear: K(x, y) = x · y
2. Polynomial: K(x, y) = (γx · y + r)^d
3. Radial Basis Function (RBF): K(x, y) = exp(-γ||x - y||^2)
4. Sigmoid: K(x, y) = tanh(γx · y + r)