Machine Learning 101

Rajdeep Chatterjee, Ph.D.
Amygdala AI, Bhubaneswar, India *

January 2025

Support Vector Model

1 VC Dimension

The Vapnik-Chervonenkis (VC) dimension is a measure of the capacity of a hypothesis space in terms of its ability to shatter a dataset. **Shattering** occurs when a hypothesis set can perfectly classify all possible label combinations of a dataset. Formally:

1.1 Definition

The VC dimension of a hypothesis class $\mathscr{H}$ is the size of the largest set $S = \{x_1, x_2, \ldots, x_n\}$ such that for every possible labeling of S, there exists an $h \in \mathscr{H}$ that correctly classifies S.

Example

Consider a set of points in R^2: 1. For a single line (linear classifier), three non-collinear points can be shattered, but not four. 2. For a circle, four points forming a square can be shattered, but not five.

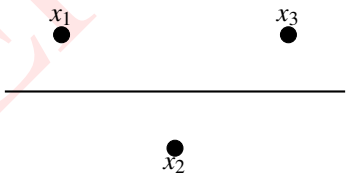

Figure 1: Example of 3 points being shattered by a linear classifier.

2 Precursor of Support Vector Machine

The precursor to Support Vector Machines (SVMs) is the Generalized Portrait Algorithm developed by Vapnik and Lerner in 1963. This method focused on finding the hyperplane that maximizes the margin between two linearly separable classes.

*Amygdala AI, is an international volunteer-run research group that advocates for *AI for a better tomorrow* `http://amygdalaai.org/`.

3 Derivation of the SVM Loss Function

The objective of SVM is to find the hyperplane that maximizes the margin while minimizing classification errors.

4 Wide Margin Problem and Primal Problem

The wide margin problem aims to find a hyperplane that maximizes the margin between two classes of data while minimizing classification errors. The margin is defined as the distance between the hyperplane and the closest data points, known as support vectors.

Primal Problem

For a dataset $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in R^d$ and $y_i \in \{-1, 1\}$, the optimization problem is formulated as:

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

$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i. \tag{2}$$

Here:

- w is the weight vector defining the hyperplane.
- b is the bias term.
- ξ_i are slack variables that allow for soft-margin classification.
- C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing classification errors.

Geometric Interpretation

The hyperplane is represented as $w^T x + b = 0$. Points satisfying $y_i(w^T x_i + b) = 1$ lie on the margin boundaries, while points with $y_i(w^T x_i + b) > 1$ are correctly classified.

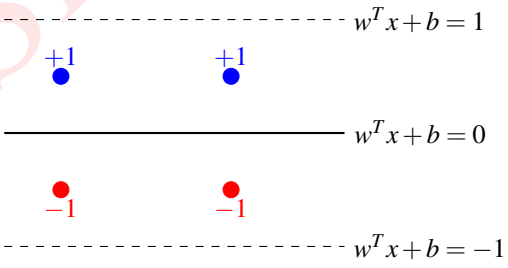

Figure 2: Wide margin hyperplane and support vectors.

5 Lagrangian Dual Problem

The primal problem can be transformed into its dual form by introducing Lagrange multipliers $\alpha_i \geq 0$. The Lagrangian is formulated as:

$$\mathscr{L}(w,b,\xi,\alpha) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \tag{3}$$

$$-\sum_{i=1}^{n}\alpha_i\left[y_i(w^T x_i + b) - 1 + \xi_i\right]. \tag{4}$$

By setting the partial derivatives of $\mathscr{L}$ with respect to w, b, and ξ_i to zero, we derive the following conditions:

$$w = \sum_{i=1}^{n}\alpha_i y_i x_i, \tag{5}$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0, \tag{6}$$

$$\alpha_i = C - \lambda_i \quad (\lambda_i \geq 0). \tag{7}$$

Substituting these conditions into the Lagrangian, we derive the dual problem:

$$\max_{\alpha} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \tag{8}$$

$$\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C. \tag{9}$$

Geometric Interpretation of the Dual

The dual problem focuses on the support vectors, which are the points where $0 < \alpha_i < C$. These points lie on the margin or violate it slightly. The final decision function is:

$$f(x) = \text{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i x_i^T x + b\right). \tag{10}$$

Soft Margin Classifier

The soft margin classifier allows some misclassifications to handle non-linearly separable data. This introduces the slack variables ξ_i, which relax the constraints and allow for a trade-off between maximizing the margin and minimizing classification errors.

6 Support Vector Classifier and Regressor

Support Vector Classifier (SVC)

The Support Vector Classifier seeks to classify data by finding the hyperplane that maximizes the margin. This can be visualized as:

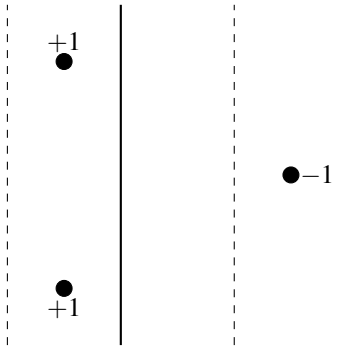

Figure 3: Visualization of the Support Vector Classifier.

Support Vector Regressor (SVR)

The Support Vector Regressor extends the concept of SVC to regression problems. The goal is to find a function $f(x)$ that deviates from the true values y_i by at most ε for all training data, while remaining as flat as possible. The optimization problem is:

$$\min_{w,b,\xi,\xi^*} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{11}$$

$$\text{s.t.} \quad y_i - (w^T x_i + b) \leq \varepsilon + \xi_i, \tag{12}$$

$$(w^T x_i + b) - y_i \leq \varepsilon + \xi_i^*, \tag{13}$$

$$\xi_i, \xi_i^* \geq 0, \quad \forall i. \tag{14}$$

Here, ξ_i and ξ_i^* are slack variables that allow for deviations larger than ε.

7 Karush-Kuhn-Tucker (KKT) Conditions in SVM

The Support Vector Machine (SVM) optimization problem seeks to find the optimal hyperplane that maximizes the margin between two classes while ensuring correct classification. The mathematical formulation and its solution through KKT conditions are fundamental to understanding SVMs.

7.1 Primal Optimization Problem

The primal optimization problem for SVMs can be expressed as:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1,\ldots,n \end{aligned}$$

Here, $\mathbf{w}$ represents the normal vector to the separating hyperplane, b is the bias term, and the constraints ensure that each data point $\mathbf{x}_i$ with label $y_i \in \{-1, 1\}$ is correctly classified with a margin of at least 1.

7.2 Lagrangian Formulation

To solve this constrained optimization problem, we introduce Lagrange multipliers $\alpha_i \geq 0$ and form the Lagrangian:

$$\mathscr{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1]$$

The Lagrangian combines the objective function with the constraints, where each constraint is weighted by its corresponding Lagrange multiplier.

7.3 KKT Conditions

The KKT conditions provide necessary and sufficient conditions for optimality in convex optimization problems like SVMs. These conditions are:

1. Stationarity Conditions

Taking derivatives of the Lagrangian with respect to the primal variables and setting them to zero:

$$\frac{\partial \mathscr{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$$
$$\frac{\partial \mathscr{L}}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

These equations show that the optimal weight vector $\mathbf{w}$ can be expressed as a linear combination of the training examples.

2. Primal Feasibility

The original constraints must be satisfied:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0, \quad \forall i$$

This ensures that all points are correctly classified with the required margin.

3. Dual Feasibility

The Lagrange multipliers must be non-negative:

$$\alpha_i \geq 0, \quad \forall i$$

4. Complementary Slackness

This crucial condition connects the primal and dual problems:

$$\alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] = 0, \quad \forall i$$

Implications of KKT Conditions

The complementary slackness condition leads to important insights about Support Vectors:

- For points exactly on the margin ($y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$), α_i can be positive. These points are the Support Vectors.
- For points beyond the margin ($y_i(\mathbf{w}^T\mathbf{x}_i + b) > 1$), α_i must be zero. These points do not contribute to the solution.

Optimal Solution

From the stationarity condition, we can express the optimal weight vector as:

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

This representation shows that the optimal hyperplane is determined only by the Support Vectors (points with $\alpha_i > 0$), making the SVM solution sparse and computationally efficient.

Practical Significance

The KKT conditions not only provide the theoretical foundation for solving the SVM optimization problem but also lead to practical algorithms for finding the solution. They enable:

- Transformation of the primal problem into its dual form
- Identification of Support Vectors
- Implementation of efficient optimization algorithms
- Extension to non-linear classification through kernels

8 Kernel Trick and Non-Linear SVM

For non-linearly separable data, the kernel trick is used to map the input space into a higher-dimensional feature space where a linear separator can be found. The kernel function $K(x_i, x_j)$ computes the inner product in this feature space without explicitly performing the transformation. Common kernel functions include:

- Linear kernel: $K(x_i, x_j) = x_i^T x_j$
- Polynomial kernel: $K(x_i, x_j) = (x_i^T x_j + c)^d$
- Radial Basis Function (RBF) kernel: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$

The dual problem with the kernel trick becomes:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{15}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C. \tag{16}$$

Support Vector Regressor (SVR)

Support Vector Regression extends the SVM to regression tasks by finding a function $f(x)$ that has at most ε deviation from the actual target.

9 Mercer's Theorem and SVM Kernels

9.1 Mercer's Theorem

Mercer's theorem is a fundamental result in the theory of kernel methods, particularly in Support Vector Machines (SVMs). It provides the mathematical foundation for using kernel functions to implicitly map data into higher-dimensional feature spaces. The theorem states that a symmetric function $k(x,x')$ can be expressed as an inner product in some feature space if and only if it is positive semi-definite. Formally, for a kernel function $k(x,x')$, there exists a feature mapping $\phi : R^d \to \mathscr{H}$ such that:

$$k(x,x') = \langle \phi(x), \phi(x') \rangle. \tag{17}$$

Here:

- $k(x,x')$ is the kernel function.
- $\phi(x)$ is the feature mapping that projects the input data x into a higher-dimensional feature space $\mathscr{H}$.
- $\langle \cdot, \cdot \rangle$ denotes the inner product in the feature space.

9.1.1 Implications of Mercer's Theorem

Mercer's theorem ensures that: 1. **Kernel Functions Represent Inner Products**: Any positive semi-definite kernel function corresponds to an inner product in some feature space. 2. **Implicit Feature Mapping**: We can work in high-dimensional feature spaces without explicitly computing $\phi(x)$, which is computationally efficient. 3. **Non-Linear Transformations**: Kernels enable non-linear decision boundaries by mapping data into spaces where linear separation is possible.

9.1.2 Positive Semi-Definiteness

A kernel function $k(x,x')$ is positive semi-definite if for any finite set of points $\{x_1, x_2, \ldots, x_n\}$, the corresponding kernel matrix K (where $K_{ij} = k(x_i, x_j)$) is positive semi-definite. That is, for any vector $\alpha \in R^n$:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0. \tag{18}$$

9.2 Popular SVM Kernels

Kernel functions are central to SVMs, as they determine the shape of the decision boundary. Below are some commonly used kernels:

1. **Linear Kernel:**

$$k(x,x') = x^T x'. \tag{19}$$

The linear kernel is suitable for linearly separable data. It does not perform any transformation and works directly in the input space.

2. **Polynomial Kernel:**

$$k(x,x') = (x^T x' + c)^d, \tag{20}$$

where c is a constant and d is the degree of the polynomial. This kernel maps the data into a polynomial feature space, allowing for more complex decision boundaries.

3. **Radial Basis Function (RBF) Kernel:**

$$k(x,x') = \exp\left(-\gamma \|x - x'\|^2\right), \tag{21}$$

where $\gamma > 0$ controls the spread of the kernel. The RBF kernel is particularly powerful because it can map data into an infinite-dimensional feature space, enabling highly non-linear decision boundaries.

4. **Sigmoid Kernel:**

$$k(x,x') = \tanh(\alpha x^T x' + c), \tag{22}$$

where α and c are parameters. This kernel is inspired by the activation function used in neural networks and can be used to model non-linear relationships.

9.2.1 Choosing the Right Kernel

The choice of kernel depends on the nature of the data and the problem: - Use the **linear kernel** for linearly separable data or when interpretability is important. - Use the **polynomial kernel** for data with polynomial relationships. - Use the **RBF kernel** for highly non-linear data or when no prior knowledge about the data structure is available. - Use the **sigmoid kernel** for problems where neural network-like behavior is desired.

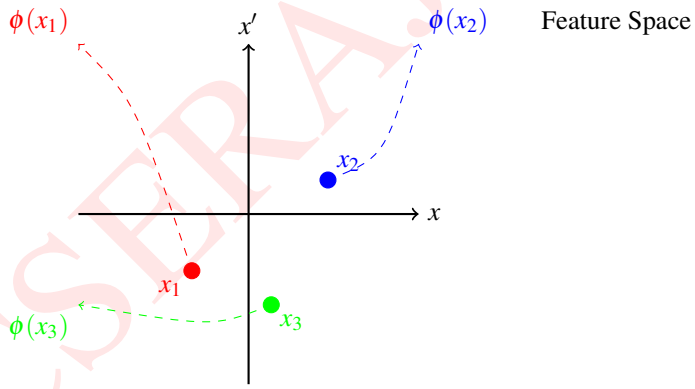

Figure 4: Kernel transformation from input space to feature space. The kernel function $k(x,x')$ implicitly maps input points x_1, x_2, x_3 into a higher-dimensional feature space where linear separation is possible.

9.3 Kernel Trick in Practice

The kernel trick allows SVMs to operate in high-dimensional feature spaces without explicitly computing the feature mapping $\phi(x)$. Instead, the kernel function $k(x,x')$ computes the inner product directly. This is particularly useful for non-linear classification and regression tasks.

For example, in the dual formulation of the SVM optimization problem, the kernel trick replaces the inner product $x_i^T x_j$ with $k(x_i, x_j)$:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{23}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C. \tag{24}$$

Mercer's theorem and kernel functions are essential tools in SVMs, enabling the handling of non-linear data by mapping it into higher-dimensional spaces. The choice of kernel function significantly impacts the performance of the SVM, and understanding their properties is crucial for effective model design.

10 Toy Dataset

In this document, we will walk through the process of finding support vectors using a toy dataset. Support vectors are the data points that lie closest to the decision boundary in a Support Vector Machine (SVM). These points are crucial in defining the hyperplane that separates the classes.

Consider the following toy dataset with two features x_1 and x_2, and binary labels $y \in \{-1, +1\}$:

x_1	x_2	y
1	2	+1
2	3	+1
2	1	-1
3	2	-1

Table 1: Toy Dataset

Step 1: Plot the Data

First, let's plot the data points on a 2D plane to visualize the separation between the two classes.

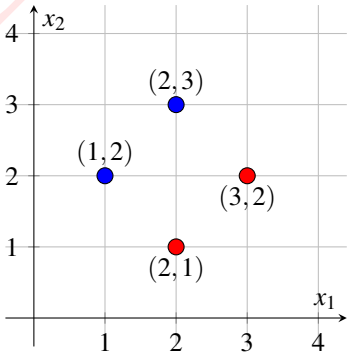

Figure 5: Plot of the Toy Dataset

Step 2: Define the Optimization Problem

The goal of SVM is to find the hyperplane that maximizes the margin between the two classes. The optimization problem can be formulated as:

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

subject to:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 \quad \forall i$$

where $\mathbf{w}$ is the weight vector, b is the bias term, and $\mathbf{x}_i$ are the feature vectors.

Step 3: Compute the Lagrangian

To solve the constrained optimization problem, we introduce Lagrange multipliers $\alpha_i \geq 0$ and form the Lagrangian:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\right]$$

Step 4: Derive the Dual Problem

By taking the partial derivatives of L with respect to $\mathbf{w}$ and b and setting them to zero, we obtain the dual problem:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

subject to:

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{and} \quad \alpha_i \geq 0 \quad \forall i$$

Step 5: Solve the Dual Problem

Using the toy dataset, we can set up the dual problem and solve for α_i. For simplicity, let's assume we have found the following Lagrange multipliers:

Data Point	α_i
(1, 2)	0.5
(2, 3)	0.0
(2, 1)	0.5
(3, 2)	0.0

Table 2: Lagrange Multipliers

Step 6: Identify Support Vectors

Support vectors are the data points for which $\alpha_i > 0$. From the table above, we can see that the support vectors are:

$$\mathbf{x}_1 = (1,2), \quad \mathbf{x}_3 = (2,1)$$

Step 7: Compute the Weight Vector and Bias

Using the support vectors, we can compute the weight vector $\mathbf{w}$ and bias b:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0.5 \cdot (+1) \cdot (1,2) + 0.5 \cdot (-1) \cdot (2,1) = (0.5-1, 1-0.5) = (-0.5, 0.5)$$

To find b, we use the support vectors:

$$b = y_i - \mathbf{w} \cdot \mathbf{x}_i = 1 - (-0.5 \cdot 1 + 0.5 \cdot 2) = 1 - (-0.5+1) = 0.5$$

Step 8: Final Decision Boundary

The final decision boundary is given by:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \implies -0.5x_1 + 0.5x_2 + 0.5 = 0$$

This can be simplified to:

$$x_2 = x_1 - 1$$

11 Toy Dataset 2

Consider the following toy dataset with two features x_1 and x_2, and binary labels $y \in \{-1, +1\}$:

x_1	x_2	y
1	2	+1
2	3	+1
2	1	-1
3	2	-1
4	5	+1
5	4	+1
5	2	-1
6	3	-1

Table 3: Toy Dataset with 8 Samples

Step 1: Plot the Data

First, let's plot the data points on a 2D plane to visualize the separation between the two classes.

To move the legend outside the plot in TikZ/pgfplots, you'll need to modify the legend settings in the axis options. Here's how to do it: latexCopy

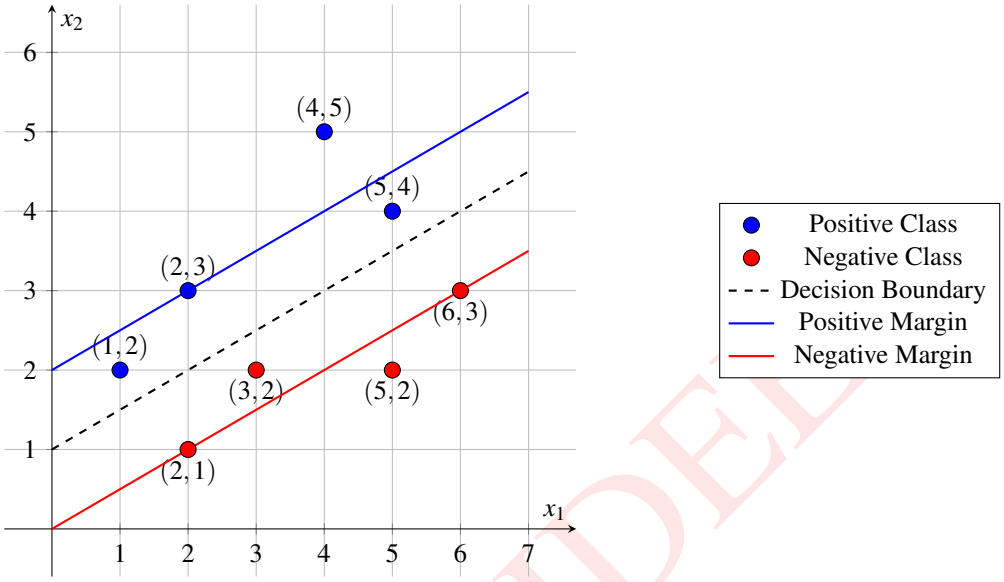

Figure 6: Plot of the Toy Dataset with SVM Decision Boundary and Margins

Step 2: Explanation of the Plot

The plot includes:

- **Decision Boundary**: The dashed black line represents the hyperplane $x_2 = 0.5x_1 + 1$, which separates the two classes.
- **Positive Margin**: The blue line represents the positive margin $x_2 = 0.5x_1 + 2$, which is parallel to the decision boundary and passes through the closest positive support vector.
- **Negative Margin**: The red line represents the negative margin $x_2 = 0.5x_1$, which is parallel to the decision boundary and passes through the closest negative support vector.

The SVM decision boundary and margins are successfully plotted for the toy dataset. The support vectors lie on the margins, and the decision boundary maximizes the separation between the two classes.

Step 3: Define the Optimization Problem

The goal of SVM is to find the hyperplane that maximizes the margin between the two classes. The optimization problem can be formulated as:

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

subject to:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 \quad \forall i$$

where $\mathbf{w}$ is the weight vector, b is the bias term, and $\mathbf{x}_i$ are the feature vectors.

Step 4: Identify Support Vectors

Support vectors are the data points that lie closest to the decision boundary. For this toy dataset, let's assume the support vectors are:

$$\mathbf{x}_1 = (2,3), \quad \mathbf{x}_2 = (3,2), \quad \mathbf{x}_3 = (4,5), \quad \mathbf{x}_4 = (5,2)$$

Step 5: Compute the Weight Vector and Bias

Using the support vectors, we can compute the weight vector $\mathbf{w}$ and bias b. For simplicity, assume the Lagrange multipliers α_i for the support vectors are:

$$\alpha_1 = 0.5, \quad \alpha_2 = 0.5, \quad \alpha_3 = 0.5, \quad \alpha_4 = 0.5$$

The weight vector $\mathbf{w}$ is calculated as:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

Substituting the values:

$$\mathbf{w} = 0.5\cdot(+1)\cdot(2,3) + 0.5\cdot(-1)\cdot(3,2) + 0.5\cdot(+1)\cdot(4,5) + 0.5\cdot(-1)\cdot(5,2)$$

$$\mathbf{w} = (1,1.5) + (-1.5,-1) + (2,2.5) + (-2.5,-1) = (-1,2)$$

To find b, use one of the support vectors, e.g., $\mathbf{x}_1 = (2,3)$:

$$b = y_i - \mathbf{w}\cdot\mathbf{x}_i = 1 - (-1\cdot 2 + 2\cdot 3) = 1 - (-2+6) = -3$$

Step 6: Final Decision Boundary

The final decision boundary is given by:

$$\mathbf{w}\cdot\mathbf{x} + b = 0 \implies -x_1 + 2x_2 - 3 = 0$$

This can be simplified to:

$$x_2 = \frac{x_1 + 3}{2}$$

We have successfully identified the support vectors and derived the decision boundary for the toy dataset. The support vectors are the key data points that define the margin and the optimal separating hyperplane.

12 Valid Values of Lagrange Multipliers in SVM

The Lagrange multipliers α_i in SVMs must satisfy the following constraints:

1. Range of α_i

$$0 \leq \alpha_i \leq C \quad \forall i$$

where C is the regularization parameter.

2. Interpretation of α_i Values

- $\alpha_i = 0$: The data point $\mathbf{x}_i$ is **not a support vector**.
- $0 < \alpha_i < C$: The data point $\mathbf{x}_i$ is a **support vector on the margin**.
- $\alpha_i = C$: The data point $\mathbf{x}_i$ is a **support vector inside the margin** or **misclassified**.

3. Constraints

The Lagrange multipliers must satisfy:

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

and

$$0 \leq \alpha_i \leq C \quad \forall i.$$

4. Role of C

The regularization parameter C controls the trade-off between maximizing the margin and minimizing classification errors:

- **Small** C: Emphasizes a larger margin (soft margin).
- **Large** C: Emphasizes correct classification (hard margin).

13 Conclusion

Support Vector Machines (SVMs) are powerful tools for classification and regression tasks. By maximizing the margin and using the kernel trick, SVMs can handle both linearly and non-linearly separable data. The dual formulation provides a computationally efficient way to solve the optimization problem, especially in high-dimensional spaces.