Support Vector Machines (SVM) for Classification in Machine Learning

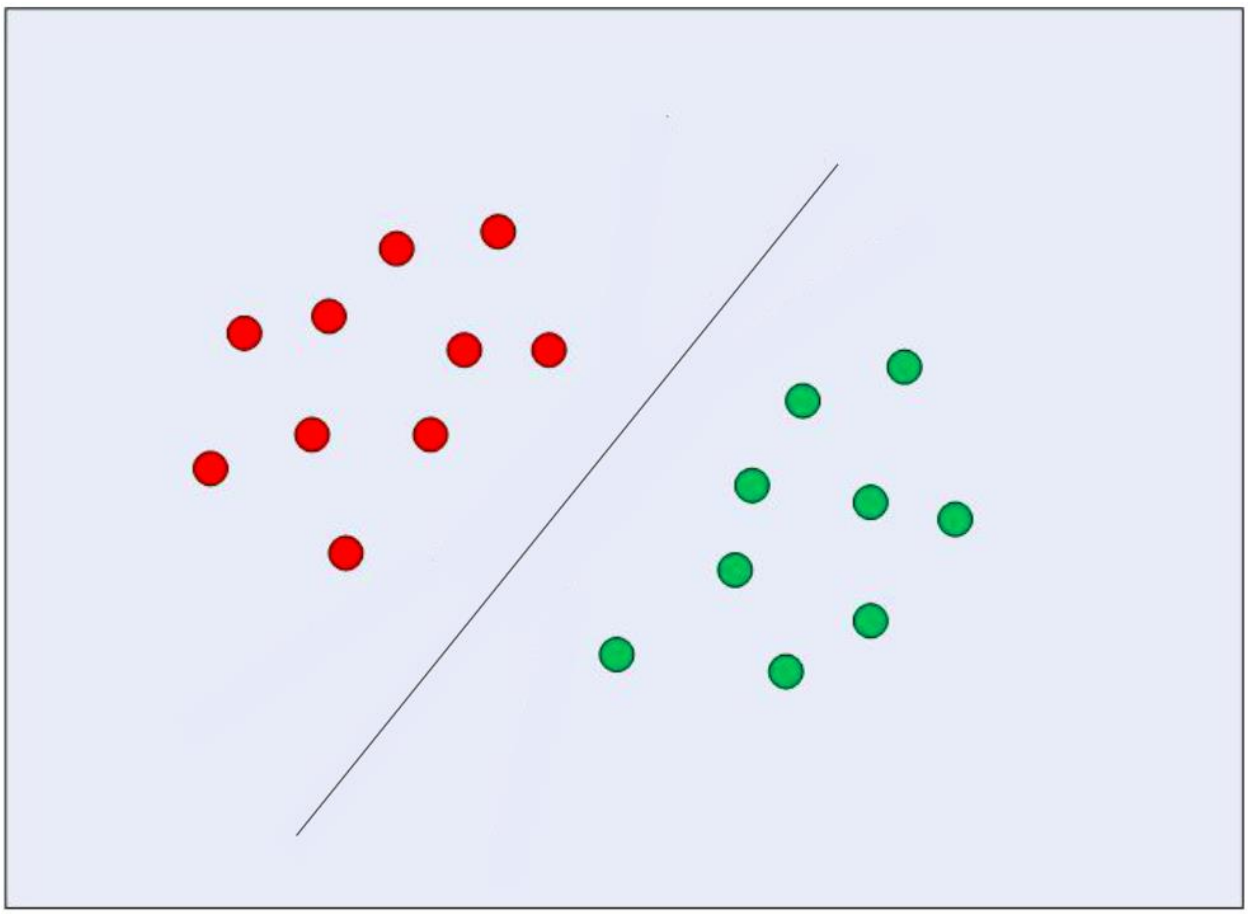
Support Vector Machines (SVM) are a powerful class of supervised learning algorithms used for classification (and regression). In classification tasks, SVM works by finding the best boundary, called a **hyperplane**, that separates different classes in the feature space. SVM tries to maximize the margin, which is the distance between the hyperplane and the closest data points from each class, known as **support vectors**.

1. What is a Hyperplane?

In a **2D feature space**, a hyperplane is simply a **line** that separates the data points into two classes. In higher dimensions (like 3D or more), the hyperplane is a plane or a higher-dimensional surface.

For example:

- Imagine you have two classes of points in a 2D space: **Class A** and **Class B**.
- A **hyperplane** is a line that best divides the two classes.



2. Maximizing the Margin

The idea of SVM is to **maximize the margin** between the two classes. The margin is the distance between the hyperplane and the closest data points from each class, called **support vectors**.

The goal of SVM is to find the hyperplane that **maximizes** this margin.

Mathematically, the margin is given by:

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

Where:

- $\mathbf{w}$ is the weight vector that defines the hyperplane.
- $||\mathbf{w}||$ is the magnitude (length) of this vector.

SVM tries to maximize this margin by adjusting the hyperplane until it separates the classes with the largest possible gap.

---

3. SVM Mathematical Formulation for Linear Classification

Let's assume that we have a set of training data points:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

Where:

- x_i represents the feature vector of the i^{th} data point,
- $y_i \in \{+1, -1\}$ is the class label (1 for Class A, -1 for Class B).

Hyperplane Equation

The hyperplane can be represented by the equation:

$$f(x) = \mathbf{w}^T\mathbf{x} + b = 0$$

Where:

- $\mathbf{w}$ is the weight vector,
- $\mathbf{x}$ is the input feature vector,
- b is the bias term that shifts the hyperplane.

Optimization Problem

The goal of SVM is to **find the optimal hyperplane** that separates the two classes with the maximum margin, while ensuring that the points are correctly classified.

This can be expressed as an **optimization problem**:

$$\text{Maximize} \quad \frac{2}{||\mathbf{w}||}$$

Subject to the constraints:

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

This constraint ensures that each point is correctly classified, where:

- If $y_i = 1$, the point should lie on the side of the hyperplane where $\mathbf{w}^T\mathbf{x}_i + b \geq 1$,
- If $y_i = -1$, the point should lie on the other side, where $\mathbf{w}^T\mathbf{x}_i + b \leq -1$.

To solve this, we **minimize** the following objective function:

$$\text{Minimize} \quad \frac{1}{2}||\mathbf{w}||^2$$

Minimizing $\frac{1}{2}||\mathbf{w}||^2$ is equivalent to maximizing the margin since the margin is inversely proportional to $||\mathbf{w}||$.

4. Numerical Example of Linear SVM

Let's consider a small 2D dataset with two classes (Class A = +1 and Class B = -1):

x_1	x_2	Class (y)
2	3	+1
3	3	+1
3	2	-1
4	1	-1

Step 1: Set up the problem

The goal is to find a hyperplane that separates the two classes. We can express the hyperplane as:

$$f(x) = w_1x_1 + w_2x_2 + b = 0$$

We need to find values for w_1, w_2, and b.

Step 2: Set the Constraints

Using the constraints for the classes, we write the following for each point:

1. For point (2, 3), $y_1 = 1$, so:

$$w_1 \cdot 2 + w_2 \cdot 3 + b \geq 1$$

2. For point (3, 3), $y_2 = 1$, so:

$$w_1 \cdot 3 + w_2 \cdot 3 + b \geq 1$$

3. For point (3, 2), $y_3 = -1$, so:

$$w_1 \cdot 3 + w_2 \cdot 2 + b \leq -1$$

4. For point (4, 1), $y_4 = -1$, so:

$$w_1 \cdot 4 + w_2 \cdot 1 + b \leq -1$$

Step 3: Solve the Optimization Problem

To solve this problem, we would normally use a **quadratic programming** solver or use an optimization method like **SMO** (Sequential Minimal Optimization). However, let's assume we get the optimal values for w_1, w_2, and b. For simplicity, let's say that the optimal solution is:

$$w_1 = 1, \quad w_2 = -1, \quad b = -2$$

Step 4: Final Decision Function

The final decision function for classification will be:

$$f(x) = 1 \cdot x_1 - 1 \cdot x_2 - 2 = 0$$

This represents the hyperplane that separates the two classes. Now, to classify a new point, we simply plug in the values of x_1 and x_2 into the equation:

$$f(x) = x_1 - x_2 - 2$$

If $f(x) > 0$, classify the point as Class A ($+1$); if $f(x) < 0$, classify the point as Class B (-1).

Step 5: Classifying New Points

Let's classify a new point $(x_1 = 2, x_2 = 2)$:

$$f(2, 2) = 2 - 2 - 2 = -2$$

Since $f(x) < 0$, we classify the point as **Class B (-1)**.

5. Non-Linear SVM with Kernels

In many real-world scenarios, the data may not be linearly separable. In this case, we use the **kernel trick** to map the data into a higher-dimensional space where it becomes linearly separable.

A commonly used kernel is the **Radial Basis Function (RBF)** kernel, which transforms the data into a higher-dimensional feature space:

$$K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$$

This allows SVM to create **non-linear decision boundaries**.

6. Conclusion

To summarize:

- SVM is a powerful algorithm that works by finding the hyperplane that maximizes the margin between two classes.
- In a **linear SVM**, we aim to find a hyperplane that best separates the classes.
- SVM can handle **non-linearly separable data** using the kernel trick, mapping the data into higher-dimensional spaces.
- The optimization problem is solved to maximize the margin and ensure correct classification using the support vectors.