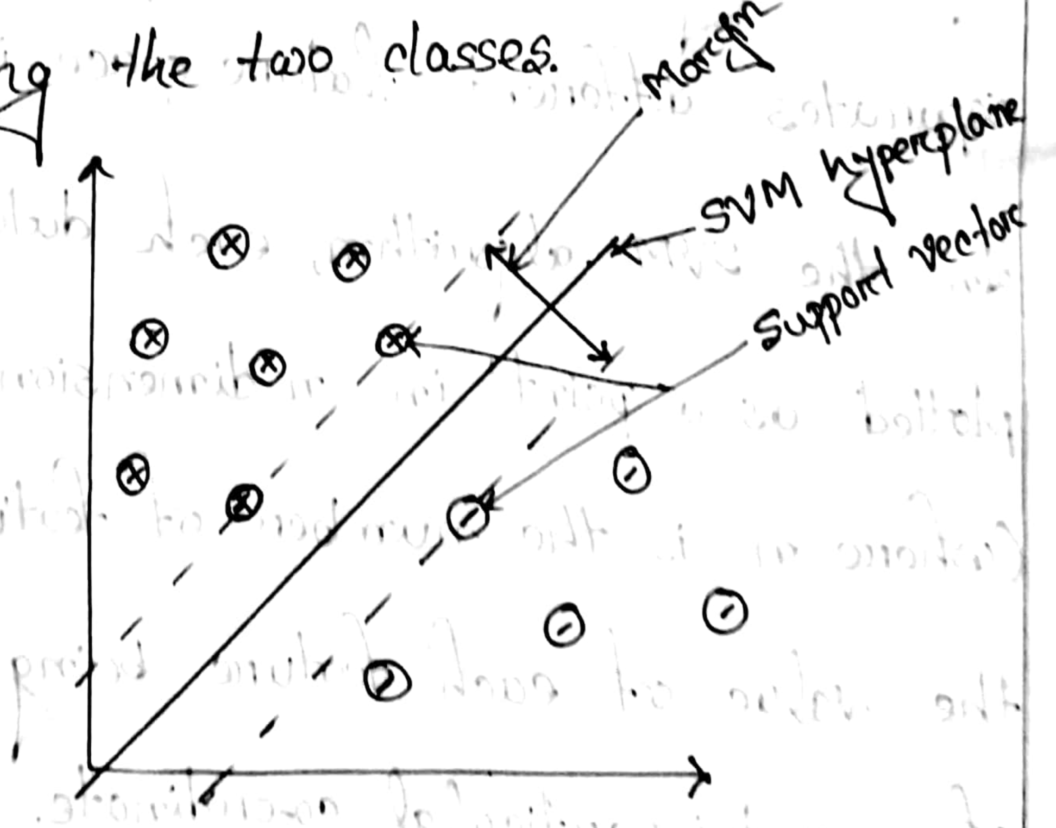
Assignment on Support Vector Machine

ID: 19701070
Md. Masud Mozumder

Introduction: Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for classification and regression challenges. However, it is mostly used in classification problems. It mainly works by finding the optimal hyperplane that separates different feature space into classes. In the SVM algorithm, each data item is plotted as a point in n-dimensional space (where n is the number of features), with the value of each feature being the value of a ~~padrei~~ particalal coordinate. Then, classification is performed by finding the optimal hyperplane that differentiates the two classes well. In this explanation, I'll focus on the binary classification case for simplicity. Below is an example of linear SVM classifier separating the two classes.



Basic Concept: Given a set of labeled data points $\{(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)\}$, where x_i represents the feature vector and y_i represents its corresponding class lebel ($y_i \in \{-1, 1\}$), the goal of SVM is to find the hyperplane that maximally separates the samples of different classes while maximizing the margin.

Hyperplane and Margin: A hyperplane in an n-dimensional space is an $(n-1)$-dimensional subspace. In the case of SVM, in a two dimensional feature space ($n=2$), the hyper-plane is a line. The margin is the distance between the hyperplane and the nearest data points from either class. The SVM algorithm aims to find the hyper-plane that maximized this margin. The margin can be of two types considering allowing misclassifications. These are:

1. Hard margin
2. Soft margin

Mathematical Formulation:

Let's denote the hyperplane as $w \cdot x + b = 0$, where w is the weight vector (normal to the hyperplane) and b is the bias term.

$\therefore$ For a given data point x_i, the signed distance from x_i to the hyperplane is given by:

$$\text{distance}_i = \frac{w \cdot x_i + b}{\|w\|}$$

The goal of SVM is to find the optimal hyperplane that maximizes the margin. Mathematically, this can be formulated as an optimization problem:

$$\text{maximize} \quad \text{margin} = \frac{2}{\|w\|}$$

subject to the constraint:

$$y_i(w \cdot x_i + b) \geq 1, \quad \text{for all } i = 1, 2, \ldots n$$

This constraint ensures that all data points are correctly classified and lie on the correct side of the hyperplane with a margin of at least 1. Hence this is the case for hard margin.

In real world senarios, the data may not be perfectly separable, or there may be outliers. To handle such cases, we introduce the notion of a soft margin, where we allow for some misclassification.

Introducing a slack variable ξ to allow for missclassification, the optimization problem becomes:

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

subject to the constraint:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \qquad \forall i, \; i = 1, 2, \ldots, n$$

$$\xi_i \geq 0 \qquad \forall i, \; i = 1, 2, \ldots, n$$

where C is a regularization parameter.

Lagrange Duality and Support Vectors: The above optimization problem is a constraint optimization problem, which is easier to solve. The lagrange multiplier or function for the SVM problem is:

$$L(\omega, b, \alpha) = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\omega \cdot x_i + b) - 1]$$

where α_i is the lagrange multipliers.

$$\therefore \frac{\delta L}{\delta \omega} = \omega - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\Rightarrow \omega = \sum_{i=1}^{n} \alpha_i y_i x_i$$

and $\frac{\delta L}{\delta b} = \sum_{i=1}^{n} \alpha_i y_i = 0$

The optimal solution of the dual problem involves maximizing the dual objective function:

$$\text{maximize} \quad L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

subject to the constraints:

$$0 \leq \alpha_i \leq C, \text{ for all } i = 1, 2, \ldots, n$$

and

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

where C is a regularization parameter.

The support vectors are the data points that lie on the margins or are misclassified, and they have corresponding non-zero Lagrange multipliers α_i. These support vectors are crucial in determining the hyperplane.

Kernel Tricks:

In cases where the data is not linearly separable in the input space, we can use kernel tricks to map the data into a higher-dimensional feature space where it may become linearly separable. This is achieved by replacing the inner product $x_i \cdot x_j$ with a kernel function $K(x_i, x_j)$. Common kernel functions include linear, polynomial, Gaussian (RBF), and sigmoid kernels.

The dual optimization problem becomes:

maximize $L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$

Prediction:

Once the optimal α_i values are obtained, we can compute the weight vector w and the bias term b using:

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

and

$$b = y_j - w \cdot x_j \text{ for any support vector } j.$$

Given a new data point x, the prediction is made by:

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

Conclusion:

In summary, SVM is a powerful algorithm for classification tasks that works by finding the optimal hyperplane that maximally separates different classes in the feature space. It achives this by maximizing the margin while ensuring that all data points are correctly classified. The dual optimization problem is solved to obtain the optimal solution, and the support vectors play a crucial role in determining the hyperplane.

Mathematical Example

Suppose we are given the following positively labeled data points:

$$\left\{ \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} 6 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ -1 \end{pmatrix} \right\}$$

and the following negatively labeled data points

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}$$

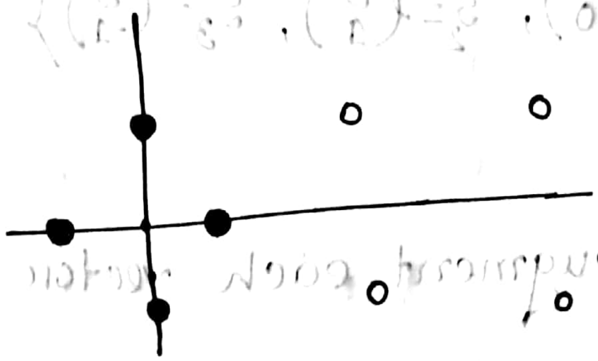


Fig: Sample data points.

We would like to discover a simple SVM that accurately discriminates the two classes.

Since the data is linearly separable, we can use linear SVM.

By inspection, it should be obvious that there are three support vectors,

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ s_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \ s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \right\}$$

Next we augment each vector with a 1 as a bias input.

So, $s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, then $\tilde{s}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$

Similarly,

$$\tilde{s}_2 = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}, \quad \tilde{s}_3 = \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

Now, we have,

$$\alpha_1 \tilde{S}_1 \cdot \tilde{S}_1 + \alpha_2 \tilde{S}_2 \cdot \tilde{S}_1 + \alpha_3 \tilde{S}_3 \cdot \tilde{S}_1 = -1$$

$$\alpha_1 \tilde{S}_1 \cdot \tilde{S}_2 + \alpha_2 \tilde{S}_2 \cdot \tilde{S}_2 + \alpha_3 \tilde{S}_3 \cdot \tilde{S}_2 = +1$$

$$\alpha_1 \tilde{S}_1 \cdot \tilde{S}_3 + \alpha_2 \tilde{S}_2 \cdot \tilde{S}_3 + \alpha_3 \tilde{S}_3 \cdot \tilde{S}_3 = +1$$

Now, computing the dot products results in;

$$2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$$

$$4\alpha_1 + 11\alpha_2 + 9\alpha_3 = +1$$

$$4\alpha_1 + 9\alpha_2 + 11\alpha_3 = +1$$

Solving the above equations, we have

$$\alpha_1 = -3.5$$

$$\alpha_2 = 0.75$$

$$\alpha_3 = 0.75$$

Next,

$$\tilde{\omega} = \sum_i \alpha_i \tilde{S}_i$$

$$= -3.5\begin{pmatrix}1\\0\\1\end{pmatrix} + 0.75\begin{pmatrix}3\\1\\1\end{pmatrix} + 0.75\begin{pmatrix}3\\-1\\1\end{pmatrix}$$

$$= \begin{pmatrix}1\\0\\-2\end{pmatrix}$$

Finally, remembering our vectors are augmented with a bias, we can equate the last entry in $\tilde{\omega}$ as the hyperplane offset b and write the separating hyperplane equation $y = \omega x + b$ with

$$\omega = \begin{pmatrix}1\\0\end{pmatrix} \text{ and } b = -2$$

Plotting the line gives the expected decision surface shown in the figure below:

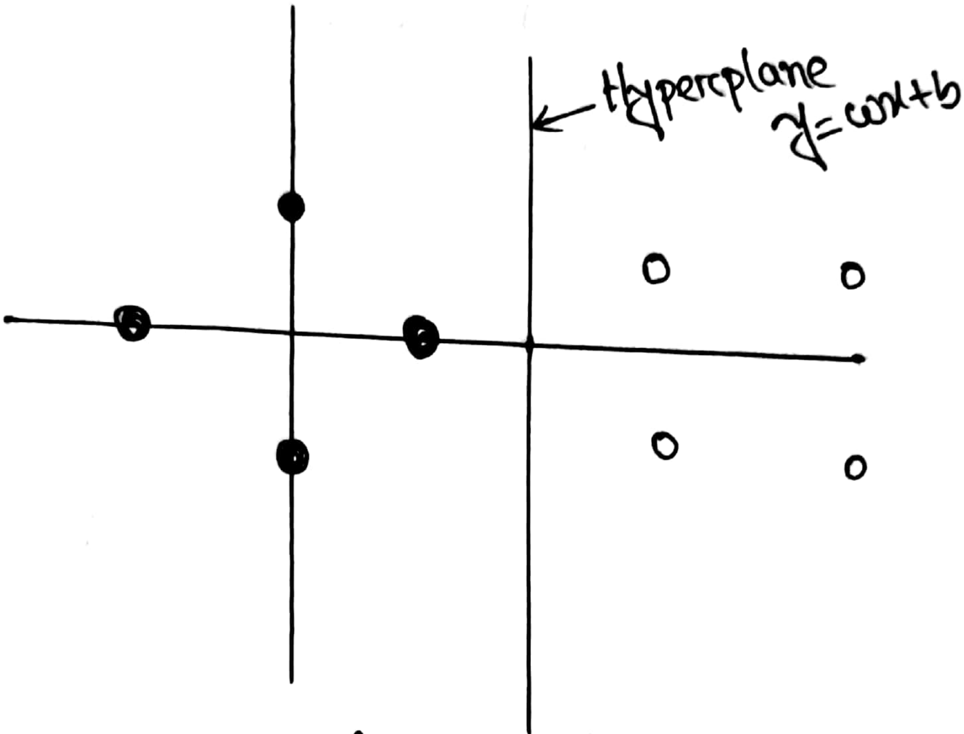


Fig: Optimal hyperplane separating data points into two classes.