Support Vector Machine

Epoch IIT Hyderabad

Ankit Saha
AI21BTECH11004

17 Jul 2022

1. Introduction

Support vector machine (SVM) is a supervised machine learning algorithm that is used for classification. SVMs are mostly used for binary linear classification.

The SVM algorithm aims to find the optimal line (or hyperplane) that separates two groups of data having different labels. Thus, it works best when the given data is linearly separable. In general, there are infinitely many hyperplanes that separate linearly separable data. The optimal among these is usually chosen as the hyperplane that is the furthest away from the closest points on both sides. This is known as the maximum-margin hyperplane.

2. Mathematical Formulation

Suppose we are given m labelled samples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)$ where $x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}$ $\forall i \in [m]$. Assuming the data is linearly separable, our goal is to find the $n-1$ dimensional maximum-margin hyperplane that separates the two categories, i.e., we must find $\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}$ such that

$$\mathbf{w}^\top \mathbf{x}_i + b \geq \frac{\delta}{2} \qquad \forall i : y_i = 1 \tag{2.1}$$

$$\mathbf{w}^\top \mathbf{x}_i + b \leq -\frac{\delta}{2} \qquad \forall i : y_i = -1 \tag{2.2}$$

where $\delta > 0$ is the margin of the classifier
The above inequalities are equivalent to

$$y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq \frac{\delta}{2} \qquad \forall i \in [m] \tag{2.3}$$

The hyperplanes $\mathbf{w}^\top \mathbf{x} + b = \pm\frac{\delta}{2}$ are the support hyperplanes and the vectors $\mathbf{x}_i$ lying on these are called support vectors. These are the only input data points that will determine the optimal weight vector $\hat{\mathbf{w}}$, hence the name, support vectors. In other words, an SVM model is insensitive to new data points that are farther away from the optimal separating hyperplane than the support vectors.

$$\hat{\mathbf{w}} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{2.4}$$

where $\alpha_i = 0$ whenever $\mathbf{x}_i$ is not a support vector.

3. Finding the Optimal Weights

As we have mentioned earlier, the optimal separating hyperplane is the one that is the furthest away from the closest points on both sides. In other words, the distance between the support hyperplanes has to be maximum. The distance between them is given by

$$\frac{\left|\left(\frac{\delta}{2} - b\right) - \left(-\frac{\delta}{2} - b\right)\right|}{||\mathbf{w}||} = \frac{\delta}{||\mathbf{w}||} \tag{3.1}$$

Thus, in order to maximize the distance between them, we have to minimize $||\mathbf{w}||$

Formally,

$$\hat{\mathbf{w}}, \hat{b} = \underset{\mathbf{w},b}{\arg\min} ||\mathbf{w}||^2 \text{ s.t. } y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq \frac{\delta}{2} \ \forall i \in [m] \tag{3.2}$$

This is a standard quadratic programming optimization problem that can be solved by using Lagrange multipliers.

$$\mathcal{L}(\mathbf{w}, b, \mathbf{\Lambda}) = \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{m} \lambda_i \left(y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) - \frac{\delta}{2} \right) \tag{3.3}$$

where $\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_m \end{pmatrix}^\top$ is the Lagrangian.

Setting $\nabla \mathcal{L}(\mathbf{w}, b, \mathbf{\Lambda}) = \mathbf{0}$, we get

$$\frac{\partial}{\partial \mathbf{w}} \left(\mathcal{L}(\mathbf{w}, b, \mathbf{\Lambda}) \right) = 0 \tag{3.4}$$

$$\implies 2\hat{\mathbf{w}} - \sum_{i=1}^{m} \lambda_i y_i \mathbf{x}_i = 0 \tag{3.5}$$

$$\implies \hat{\mathbf{w}} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \text{ where } \alpha_i = \frac{\lambda_i}{2} \tag{3.6}$$

In other words, the optimal weight vector is a linear combination of the input vectors, which is consistent with what we mentioned earlier.

4. Soft-margin Classifiers

The above calculations were assuming that the input data is linearly separable. However, when the data is not linearly separable, a fixed margin δ will not work. Hence, we make the margin variable by introducing another parameter $\boldsymbol{\xi} \in \mathbb{R}^m$, $\boldsymbol{\xi} = \begin{pmatrix} \xi_1 & \xi_2 & \cdots & \xi_m \end{pmatrix}^\top$

$$y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq \frac{\delta}{2} - \xi_i \qquad \forall i \in [m] \tag{4.1}$$

Whenever $\mathbf{x}_i$ is on the correct side of the margin, $\xi_i = 0$ so that it is the same as the previous case. And when it is on the wrong side, we can choose ξ_i to be proportional to the distance of $\mathbf{x}_i$ from the margin. Therefore,

$$\xi_i = \max\left(0, \frac{\delta}{2} - y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right)\right) \tag{4.2}$$

The optimization problem is then reformulated as

$$\hat{\mathbf{w}}, \hat{b}, \hat{\boldsymbol{\xi}} = \underset{\mathbf{w},b,\boldsymbol{\xi}}{\arg\min} ||\mathbf{w}||^2 + C\,||\boldsymbol{\xi}||^2 \text{ s.t. } y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq \frac{\delta}{2} - \xi_i \ \forall i \in [m] \tag{4.3}$$

Here, $C\,||\boldsymbol{\xi}||^2$ is the regularizer that prevents ξ_i from becoming too large.

5. Questions

i) What is the significance of the threshold term b?
ii) How would you extend the concept of support vector machines to regression?
iii) What is the minimum number of support vectors possible for a given data?
iv) Can SVMs be used for multi-class classification?
v) What is the training error of an SVM model when the data is linearly separable?

6. Answers

i) The threshold term b ensures that the optimal separating hyperplane is not constrained to pass through the origin. This provides the model with another degree of freedom.
ii) Using the same concept of support vector machines, in regression, we try to find a hyperplane that best fits the data, i.e., all data points are within a certain fixed distance from the hyperplane.
iii) The minimum number of support vectors possible for a given data is 2, one on either side.
iv) Yes, because a multi-class classification problem can be framed as a series of binary classification problems.
v) When the data is linearly separable, we can always find a hyperplane that perfectly separates the two categories. Hence, the training error will be 0.