Batch Gradient Descent (GD) Algorithm for Linear SVM

Following is the loss/cost function for the Linear SVM model.

$$J = \frac{1}{2}\vec{w}^T\vec{w} + C\sum_{i=1}^{N}\xi_i$$

The loss/cost is measured by ξ_i. Its range is:

$$\xi_i \geq 0$$

$$\xi_i \geq 1 - \ y_i(\vec{w}^T\vec{x}_i + b\ )$$

Hence, the loss ξ_i can be expressed using a **max (.) function**, which is known as the **Hinge loss** function.

$$\xi_i \geq \max\{0, 1 - \ y_i(\vec{w}^T\vec{x}_i + b\ )\}$$

Using hinge loss, the Linear SVM cost function is given by:

$$J = \frac{1}{2}\vec{w}^T\vec{w} + C\sum_{i=1}^{N}\max\{0, 1 - \ y_i(\vec{w}^T\vec{x}_i + b\ )\}$$

Here ξ_i represents the deviation from the margin. For most of the data points that are away from both margins have their $\xi_i = 0$.

Only for the **support vectors** (that are on the decision surface, hence non-separable): $\xi_i > 0$

The goal of GD is to reduce the number of support vectors with $\xi_i > 0$.

Thus, we can reformulate the cost function by using only the support vectors, as follows.

$(1 - t_{sv} * \vec{X}_{sv} \cdot \vec{w})$

$$J = \frac{1}{2}\vec{w}^T\vec{w} + C\left[\sum \cancel{(1 - \vec{X}_{sv}.\vec{w})} - b * \sum \vec{t}_{sv}\right]$$

Here, $\vec{X}_{sv}$ contains the support vectors and $\vec{t}_{sv}$ contains the labels of the support vectors. Using the Gradient Descent (GD) algorithm, we minimize this cost function. Calculation of $\vec{X}_{sv}$ and $\vec{t}_{sv}$ are shown below.

Calculate the Support Vectors

First, we state the decision rule for a single sample $\vec{x}$:

- If $\vec{w}^T.\vec{x} + b \geq 1$; then class = positive (+1), otherwise negative (-1).

We assume the class labels (+1 or -1) for the entire dataset $\vec{X}$ are stored in a 1D vector $\vec{t}$.

The decision boundary is perpendicular to $\vec{w}$ and its displacement from the origin is controlled by the bias b.

For the entire dataset $\vec{X}$ and class label vector $\vec{t}$ (represents the class labels -1/+1 of each sample in the dataset), the decision rule can be written as follows.

$$\vec{t} * (\vec{w}^T.\vec{x} + b) \geq 1$$

For sample i, if $t_i = +1$ (belongs to positive class), then

$$\vec{w}^T.\vec{x} + b \geq 1$$

However, if $t_i = -1$ (belongs to negative class), then

$$\vec{w}^T.\vec{x} + b \leq -1$$

From the above decision rule, we derive the fact that the support vectors that reside on the decision surface satisfy the following equation:

$$\vec{t} * (\vec{w}^T.\vec{x} + b) < 1$$

In matrix notation:

$$\cancel{(\vec{t} * \vec{X}).\vec{w}} < 1 \qquad \vec{t} * (\vec{X}.\vec{w} + b) < 1$$

We will use this equation to compute the support vectors.

The derivative of the cost function with respect to the weight vector $\vec{w}$ and the intercept/bias b:

$$\nabla_{\vec{w}} J = \vec{w} - C * \sum \cancel{\vec{X}_{sv}} \quad \vec{t}_{sv} * \vec{X}_{sv}$$

$$\nabla_b J = -C * \sum \vec{t}_{sv}$$

Note, in the calculation of $\nabla_{\vec{w}} J$, we need to take the sum of each column of $\vec{X}_{sv}$ (a column represents a component of $\vec{w}$). The sum will give a 1D row vector of dimension d (it is the dimension of $\vec{w}$). We need to reshape this vector as a 1D column vector to match the dimension of $\vec{w}$, which is a 1D column vector.

Finally, we update both $\vec{w}$ and b.

$$\vec{w} := \vec{w} - \eta * \nabla_{\vec{w}} J$$

$$b := b - \eta * \nabla_b J$$

Gradient Descent Algorithm Pseudocode for Linear SVM

Initialize the weight vector $\vec{w}$ with small random numbers and the scalar intercept/bias b with zero.

Then, iterate the following steps until a stopping criterion or max number of epochs is fulfilled.

1. Find the support vector matrix $\vec{X}_{sv}$ and their label vector $\vec{t}_{sv}$ using the following equation.

$$\cancel{(\vec{t} * \vec{X}) \vec{w}} < 1 \qquad t * (\vec{X}\vec{w} + b) < 1$$

2. Compute the cost J:

$$J = \frac{1}{2}\vec{w}^T\vec{w} + C\left[\sum \overset{(1 - t_{sv} * \vec{X}_{sv} \cdot \vec{w})}{\cancel{(1 - \vec{X}_{sv}.\vec{w})}} - b * \sum \vec{t}_{sv}\right]$$

3. Compute the derivative of the cost function with respect to the weight vector $\vec{w}$ and the intercept/bias b:

$$\nabla_{\vec{w}} J = \vec{w} - C * \sum \cancel{\vec{X}_{sv}} \quad \vec{t}_{sv} * \vec{X}_{sv}$$

$$\nabla_b J = -C * \sum \vec{t}_{sv}$$

4. Update both $\vec{w}$ and b.

$$\vec{w} := \vec{w} - \eta * \nabla_{\vec{w}} J$$

$$b := b - \eta * \nabla_b J$$