

Lecture 11: Soft SVM: Non-Separable Classification

12th September, 2022

Lecturer: Abir De

Scribe: Group 21, Group 22 and TA team

1 Recap

1.1 The Problem

Our dataset is of the form $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$.

The x_i are data points, usually in \mathbb{R}^k for some $k \in \mathbb{N}$.

The y_i s are discrete, and each $y_i \in \{+1, -1\}$.

1.2 Separable Case

In the previous lecture, we assumed that there will always be a hyperplane separating our data, in such a way that, for every point, the classification error will be 0. We saw how we can solve such an instance with the following formulation,

$$\{\mathbf{w}^*, b^*\} = \|\mathbf{w}\|^2, \text{ s.t. } \forall i \in \mathcal{D}, y_i(\mathbf{w}^T \mathbf{x}_i + b) > 1$$

Necessary and Sufficient Condition for separability: Convex Hull corresponding to all the positive points and that corresponding to all the negative points do not intersect.

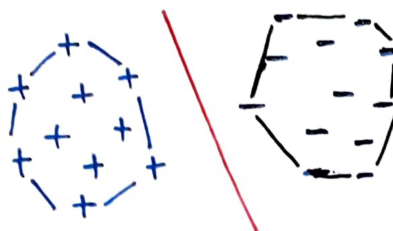


Figure 1: Separable Clusters

1.3 Non-Separable Case

In the case of non-separable instances: we will not be able to find a \mathbf{w} such that the condition $y_i(\mathbf{w}^T \mathbf{x}_i + b) > 1$ is satisfied for every point - this is because the convex hulls for positive and negative points intersect, and hence we cannot correctly classify all our points.

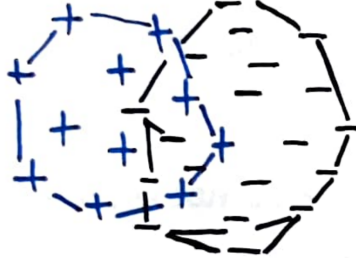


Figure 2: Overlapping Clusters

One way to deal with these is to remove points in the overlap.

Another way is to work with a new formulation where we relax some of our constraints. However, we can't simply handcraft these constraints. We'll have to mathematically model these constraints and try to learn them.

2 Formulation for Non-Separable Instance

We may proceed like this: If the given expression isn't greater than 1 for all (x_k, y_k) , there must exist some (x, y) for which it doesn't.

To accommodate this, we may replace the 1 in $y(w^T x + b) > 1 \quad \forall x, y$
by $y(w^T x + b) > 1 - \xi \quad \forall x, y$

We want $\xi > 0$, to be as small as possible. The ξ will be different for different points.

That is, finally, we can write our constraints in the form,

$$y_i(w^T x_i + b) \geq 1 - \xi_{(x_i, y_i)}$$

where $\xi_{(x_i, y_i)}$ (known as slack variables) are separate for each (x_i, y_i) , which leads us to the following optimization problem,

$$\min_{w, b, \xi_{(x_i, y_i)}} ||w||^2 + c \sum_i \xi_{(x_i, y_i)}$$

Note that if $y_i(w^T x_i + b) > 1$ i.e., our point is correctly classified, then $1 - y_i(w^T x_i + b) < 0$, otherwise $1 - y_i(w^T x_i + b) > 0$. So, we can use $\xi_{(x_i, y_i)} = 1 - y_i(w^T x_i + b)$

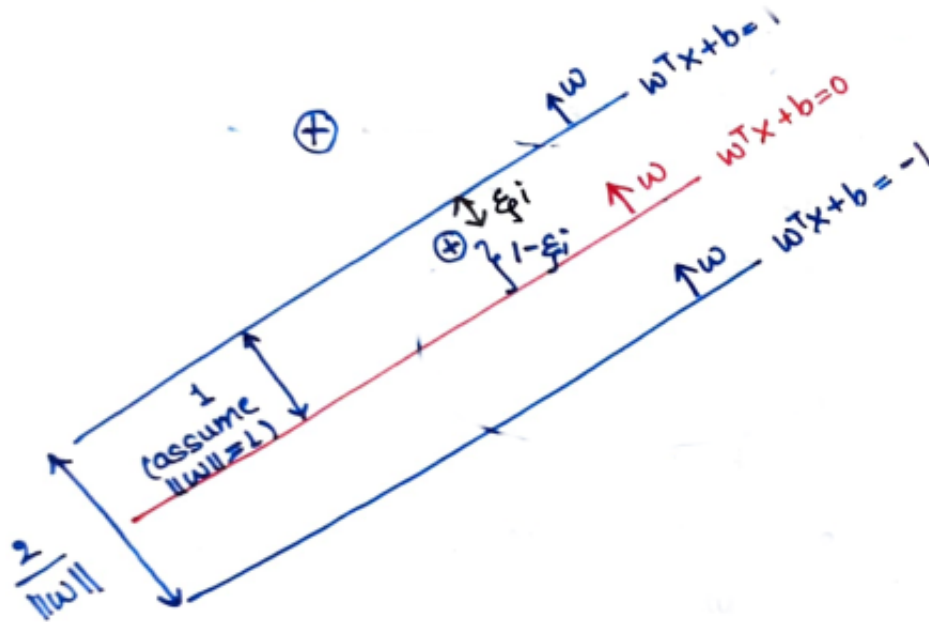
At optimal points, equality is always achieved when minimising $\xi_{(x_i, y_i)}$. This allows us to write our formulation in the following equivalent form,

$$\min_{w, b} (||w||^2 + c \sum_{i \in \mathcal{D}} \max(1 - y_i(w^T x_i + b), 0))$$

If we increase c , the classes would be better separated and might lead to over-fitting.

A Geometric Perspective

We have that $y_i(w^T x_i + b) > 1$ is not satisfied for all points of the dataset. If it were satisfied then we would be dealing with the separable case anyway. So, For the case, When $y_i(w^T x_i + b) > 1$ doesn't hold, We are trying to find the minimum $\xi_{(x_i, y_i)}$ s.t. $y_i(w^T x_i + b) \geq 1 - \xi_{(x_i, y_i)}$. ($\xi_{(x_i, y_i)}$ can be simply written as ξ_i)

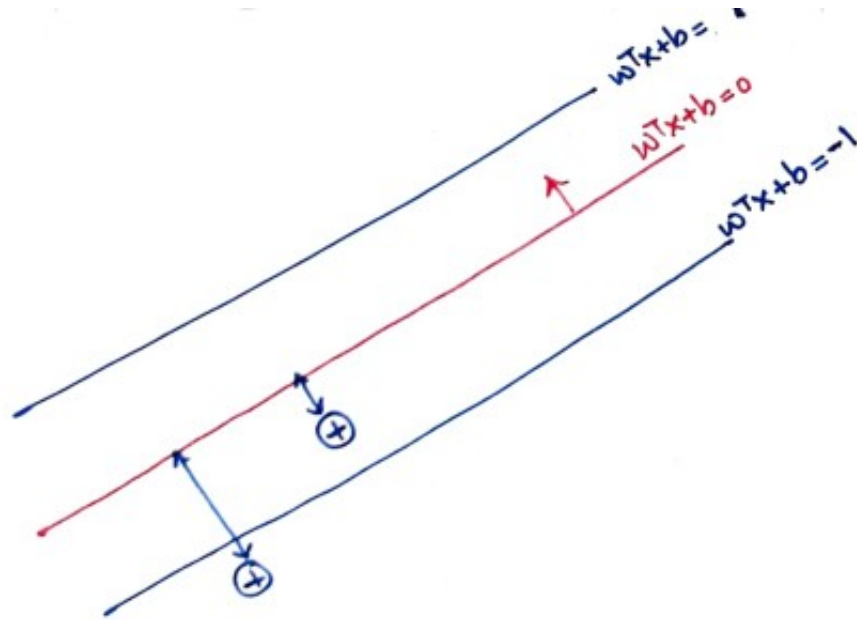


Note here that for the point that lies above the positive hyperplane,

We have that $y_i(w^T x_i + b) > 1 \Rightarrow \xi_{(x_i, y_i)} = 0$

For the other point, which is labelled as +ve but is below the said hyperplane, We'll have that $y_i(w^T x_i + b) = 1 - \xi_{(x_i, y_i)}$, in any such case, We'll find that $0 \leq 1 - y_i(w^T x_i + b) \leq 1$

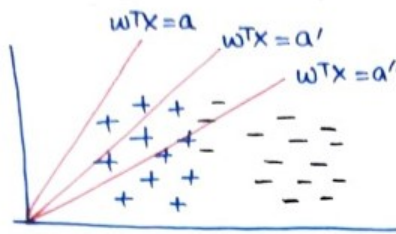
If the case were rather like this :



We'll instead have that $0 \leq 1 - y_i(w^T x_i + b) \leq 1 \Rightarrow \xi_{(x_i, y_i)} \geq 1$

Thus, we may now conclude that the quantity $1 - y_i(w^T x_i + b)$ is negative, When the point is correctly classified, and positive if its incorrectly specified.

3 Tuning of b



In the separable case we had to use b , because if we forced a hyperplane separating all points to that pass through origin we will always incur some misclassification error.

In non-separable case, we have not regularized b so it is not stable, i.e. change x slightly and b changes significantly. So we try to get rid of b . We are already incurring error due to misclassified points so we allow some error due to b to make our optimisation task easier.

We have two options via preprocessing route.

3.1 Shift the origin

$$\vec{X}_{i_new} = \vec{X}_{i_old} - E[\vec{X}]$$

$E[\vec{X}]$: Empirical mean vector incase of data

3.2 Batch Normalization

$$X_{j_new}^{(i)} = \frac{X_{j_old}^{(i)} - \hat{\mu}_j}{\hat{\sigma}_j}$$

$X_{j_new}^{(i)}$: jth feature of ith sample

$\hat{\mu}_j$: Empirical mean over all the samples for jth feature

$\hat{\sigma}_j$: Empirical standard deviation over all samples for jth feature

Batch normalization works because the data along each feature dimension is scaled properly.

Batch normalization also makes the bias tend to 0, training becomes stable and we achieve very good accuracy.

One disadvantage of batch normalization is that mean and standard deviation of each batch may differ significantly if we have lots of data, leading to increase in error. Thus, we have to find the ideal number of data points, which cannot be too high or too low.

4 Dual Formulation

4.1 Convex Optimization

Some background of convex optimization is needed before we move forward with our new formulation of SVM.

$$\min_{\mathbf{w}} f(\mathbf{w}) \text{ such that } g(\mathbf{w}) \leq \mathbf{c} \quad (1)$$

Note that $g(\mathbf{w})$ can be a vector and then the inequality would be pointwise ($g_i(w) \leq c_i$)

We can approximate (1) as:

$$\max_{\lambda \geq 0} \min_{\mathbf{w}} f(\mathbf{w}) + \lambda^\top (g(\mathbf{w}) - \mathbf{c}) \quad (2)$$

When $f(\mathbf{w})$ and $g(\mathbf{w})$ both are strictly convex, (1) and (2) are exactly equivalent. This means that either $\lambda = 0$ or $g(\mathbf{w}) = \mathbf{c}$. This is known as Slater's condition.

4.2 Objective Function

SVM problem at hand:

$$\min_{\mathbf{w}, b, \xi} \lambda \|\mathbf{w}\|^2 + \sum_{i \in \mathcal{D}} \xi_i$$

$$\text{Constraints: } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \quad \forall i \in \mathcal{D}$$

Rewriting constraints as $1 - \xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0$ and $-\xi_i \leq 0$ to make them of the form $g(w) \leq c$.

Using convex optimisation of (2) we rewrite our formulation as:

$$\begin{aligned} \max_{\alpha, \beta} \min_{\mathbf{w}, b, \xi} \lambda \|\mathbf{w}\|^2 + \sum_{i \in \mathcal{D}} \xi_i + \sum_{i \in \mathcal{D}} \alpha_i (1 - \xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b)) + \sum_{i \in \mathcal{D}} \beta_i (-\xi_i) \\ \text{s.t. } \alpha_i \geq 0, \beta_i \geq 0 \quad \forall i \in \mathcal{D} \end{aligned}$$

This is our final optimisation problem in dual space and the function to be maximised is called Lagrangian dual function $g(\alpha, \beta)$.

Note : λ used here is different from the λ used in (2) of convex optimisation.

4.3 Optimality Conditions

Differentiating $g(\alpha, \beta)$ w.r.t \mathbf{w}, b, ξ_i and equating to 0, we get:

$$\begin{aligned} \frac{\partial g}{\partial \mathbf{w}} = 0 &\Rightarrow 2\lambda \mathbf{w}^* + \sum_{i \in \mathcal{D}} \alpha_i (-y_i \mathbf{x}_i) = 0 \\ &\Rightarrow \mathbf{w}^* = \sum_{i \in \mathcal{D}} \frac{\alpha_i y_i \mathbf{x}_i}{2\lambda} \\ \frac{\partial g}{\partial b} = 0 &\Rightarrow \sum_{i \in \mathcal{D}} \alpha_i y_i = 0 \\ \frac{\partial g}{\partial \xi_i} = 0 &\Rightarrow 1 - \alpha_i - \beta_i = 0 \\ &\Rightarrow \alpha_i + \beta_i = 1 \quad \forall i \in \mathcal{D} \end{aligned}$$

We also see that our objective function is quadratic in \mathbf{w} and constraints are linear in \mathbf{w}, ξ_i . Therefore, all functions are strictly convex and we can apply Slater's condition. Therefore,

$$\begin{aligned} \alpha_i^* (1 - \xi_i^* - y_i(\mathbf{w}^{*\top} \mathbf{x}_i + b^*)) &= 0 \quad \forall i \in \mathcal{D} \\ \beta_i^* \xi_i^* &= 0 \quad \forall i \in \mathcal{D} \end{aligned}$$

Let's look at 2 cases:

1. $\xi_i^* > 0 \Rightarrow$ point is incorrectly classified or it is inside the margin of hyperplanes
 $\Rightarrow \beta_i^* = 0 \Rightarrow \alpha_i^* = 1$.
 α_i^* being high can be seen as penalty being high, which intuitively means that the point is misclassified.
2. $\xi_i = 0 \Rightarrow$ point is on or outside the margin of hyperplanes
 If $1 - y_i(\mathbf{w}^{*\top} \mathbf{x}_i + b^*) > 0 \Rightarrow \alpha_i^* = 0 \Rightarrow \beta_i^* = 1$. This means that penalty is low and point is correctly classified.
 If $1 - y_i(\mathbf{w}^{*\top} \mathbf{x}_i + b^*) = 0$, then we cannot comment anything about α_i^* .

4.4 Dual Problem at Optimality

Let us substitute optimal values into g and rewrite our dual problem. We have:

$$\begin{aligned}
 \mathbf{w}^* &= \sum_{i \in \mathcal{D}} \frac{\alpha_i y_i \mathbf{x}_i}{2\lambda}, \quad \sum_{i \in \mathcal{D}} \alpha_i y_i = 0, \quad \alpha_i + \beta_i = 1, \quad \alpha_i \geq 0, \quad \beta_i \geq 0 \\
 g(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \lambda \|\mathbf{w}^*\|^2 + \sum_{i \in \mathcal{D}} \xi_i^* (1 - \alpha_i - \beta_i) + \sum_{i \in \mathcal{D}} \alpha_i (1 - y_i(\mathbf{w}^{*\top} \mathbf{x}_i + b^*)) \\
 &= \lambda \mathbf{w}^{*\top} \mathbf{w}^* + \sum_{i \in \mathcal{D}} \xi_i^* (0) + \sum_{i \in \mathcal{D}} \alpha_i - \mathbf{w}^{*\top} \sum_{i \in \mathcal{D}} \alpha_i y_i \mathbf{x}_i \\
 &= \sum_{i \in \mathcal{D}} \alpha_i - \frac{1}{4\lambda} \sum_{i \in \mathcal{D}} \alpha_i y_i \mathbf{x}_i^\top \sum_{j \in \mathcal{D}} \alpha_j y_j \mathbf{x}_j
 \end{aligned}$$

Thus, our dual problem has become:

$$\begin{aligned}
 &\max_{\boldsymbol{\alpha}} \sum_{i \in \mathcal{D}} \alpha_i - \frac{1}{4\lambda} \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{D}} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\
 &\text{s.t. } 0 \leq \alpha_i \leq 1 \quad \forall i \in \mathcal{D} \quad \text{and} \quad \sum_{i \in \mathcal{D}} \alpha_i y_i = 0
 \end{aligned}$$

We have conveniently got rid of $\boldsymbol{\beta}$ and derived a clean optimization problem.