AIML - CS 337

Lecture 11: Soft SVM: Non-Separable Classification

12^{th} September, 2022

Lecturer: Abir De *Scribe: Jash, Manish, Chetan, Yashwanth*

In this lecture, we continued our discussion on **non-separable** instances of the classification problem. We saw how we can approach this problem by relaxing the constraints on our previous formulation and transform it into a more easily solvable problem using **Dual Formulation**.

1 Recap

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,

$$\{\boldsymbol{w}^*, b^*\} = argmin||\boldsymbol{w}||^2,\ s.t\ \forall i \in \mathcal{D},\ y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) > 1$$

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

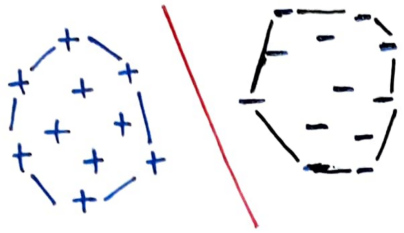


Figure 1: Separable Clusters

In the case of non-separable instances, the condition

$$y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) > 1$$

will not be satisfied for every point as the convex hulls for positive and negative points intersect, so we cannot correctly classify all our points which leads us to relax some of our constraints.

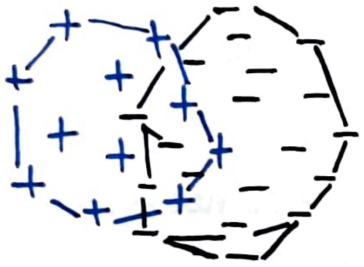


Figure 2: Overlapping Clusters

2 Formulation for Non-Separable Instance

Since $y_i(\boldsymbol{w}^T\boldsymbol{x_i} + \boldsymbol{b}) > 1$ is not satisfied for some $(x_i, y_)$, we can write our constraints in the form,

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

where $\boldsymbol{\xi}_{(x_i,y_i)}$ (slack variables) is separate for each (x_i, y_i), which leads us to the following optimization problem,

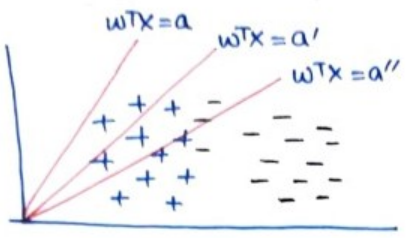
$$\min_{\boldsymbol{w},\, b,\, \boldsymbol{\xi_{(x_i,y_i)}}} ||\boldsymbol{w}||^2 + c\sum_i \boldsymbol{\xi}_{(x_i,y_i)_+}$$

Note that if $y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) > 1$ i.e., our point is correctly classified, then $1 - y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) < 0$, otherwise $1 - y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) > 0$. So, we can use $\boldsymbol{\xi}_{(x_i,y_i} = 1 - y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b)$ (At optimal points, equality is always achieved when minimising $\boldsymbol{\xi}_{(x_i,y_i)}$) to write our formulation in the following equivalent form,

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

If we increase c, the classes would be better separated and can lead to overfitting.

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

$$\vec{\boldsymbol{X}}_{i\_new} = \vec{\boldsymbol{X}}_{i\_old} - \boldsymbol{E}[\vec{\boldsymbol{X}}]$$

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

3.2 Batch Normalization

$$\boldsymbol{X}^{(\boldsymbol{i})}_{j\_new} = \frac{\boldsymbol{X}^{(\boldsymbol{i})}_{i\_old} - \hat{\mu}_j}{\hat{\sigma}_j}$$

$\boldsymbol{X}^{(\boldsymbol{i})}_{j\_new}$: jth feature of ith sample

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

$\hat{\sigma}_j$: Emperical 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.

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

Note that $g(\boldsymbol{w})$ can be a vector and then the inequality would be pointwise ($g_i(w) \leq c_i$)
We can approximate (1) as:

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

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

4.2 Objective Function

SVM problem at hand:

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

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

Rewriting constraints as $1 - \xi_i - y_i(\boldsymbol{w}^\top \boldsymbol{x_i} + b) \leq 0$ 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:

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \lambda ||\boldsymbol{w}||^2 + \sum_{i \in \mathcal{D}} \xi_i + \sum_{i \in \mathcal{D}} \alpha_i (1 - \xi_i - y_i(\boldsymbol{w}^\top \boldsymbol{x_i} + b)) + \sum_{i \in \mathcal{D}} \beta_i (-\xi_i)$$

$$\text{s.t. } \alpha_i \geq 0, \beta_i \geq 0 \ \ \forall i \in \mathcal{D}$$

This is our final optimisation problem in dual space and the function to be maximised is called Lagrangian dual function $g(\boldsymbol{\alpha}, \boldsymbol{\beta})$.
Note : λ used here is different from the λ used in (2) of convex optimisation.

4.3 Optimality Conditions

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

$$\frac{\partial g}{\partial \boldsymbol{w}} = 0 \Rightarrow 2\lambda \boldsymbol{w^*} + \sum_{i \in \mathcal{D}} \alpha_i (-y_i \boldsymbol{x_i}) = 0$$

$$\Rightarrow \boldsymbol{w^*} = \sum_{i \in \mathcal{D}} \frac{\alpha_i y_i \boldsymbol{x_i}}{2\lambda}$$

$$\frac{\partial g}{\partial b} = 0 \Rightarrow \sum_{i \in \mathcal{D}} \alpha_i y_i = 0$$

$$\frac{\partial g}{\partial \xi_i} = 0 \Rightarrow 1 - \alpha_i - \beta_i = 0$$

$$\Rightarrow \alpha_i + \beta_i = 1 \quad \forall i \in \mathcal{D}$$

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

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

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 $1 - y_i(\boldsymbol{w^*}^\top \boldsymbol{x_i} + b^*) > 0 \Rightarrow \alpha_i^* = 0 \Rightarrow \beta_i^* = 1$. This means that penalty is low and point is correctly classified.
   If $1 - y_i(\boldsymbol{w^*}^\top \boldsymbol{x_i} + b^*) = 0$, 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:

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

Thus, our dual problem has become:

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

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