CS419-Intro to ML

Lecture 8: Support Vector Machine

8 February 2023

Lecturer: Abir De *Scribe: Group 15 and 16*

1 Recap

The Classification Problem
Given data points $x_i; 1 \leq i \leq n$ and possible class choices $c_1, c_2, c_3, ..., c_k$
We want to find a mapping $f : x \rightarrow c_1, c_2, ..c_k$

The Perceptron Algorithm attempts to minimize the following error:

$$E = -\Sigma_{x_p} y \phi^T(x) w \tag{1}$$

where x_p are the misclassified examples.

We are just using the Gradient Descent Algorithm here:

$$\nabla E = -\Sigma_{x_p} y \phi(x) \tag{2}$$

$$w^{k+1} = w^k - \nabla E \tag{3}$$

$$w^{k+1} = w^k + y' \phi(x) \tag{4}$$

As is the situation, given two non-intersecting convex hulls, the following algorithm is guaranteed to give us a hyperplane that separates the two:

And thus we have our Perceptron update rule :

$$w^{k+1} = w^k + y' \phi(x) \tag{5}$$

on multiplying y' and $\phi(x)$, we get :

$$y' w^{k+1^T} \phi(x) = y' w^{k^T} \phi(x) + ||\phi(x)||^2 \tag{6}$$

to control the change which is governed by $||\phi(x)||$, we use η, which is the rate of convergence :

$$y' w^{k+1^T} \phi(x) = y' w^{k^T} \phi(x) + \eta ||\phi(x)||^2 \tag{7}$$

2 Introduction

Hyperplane separates the convex hulls, but we want to find the "best" hyperplane.

What does **"best"** mean?

We want to have some wiggle room around the best hyperplane so that we can perform well on the unseen test data. This means that we want to maximise the margin between the two hulls, i.e. Maximise sum of the distances of the closest points in each class from the hyperplane.

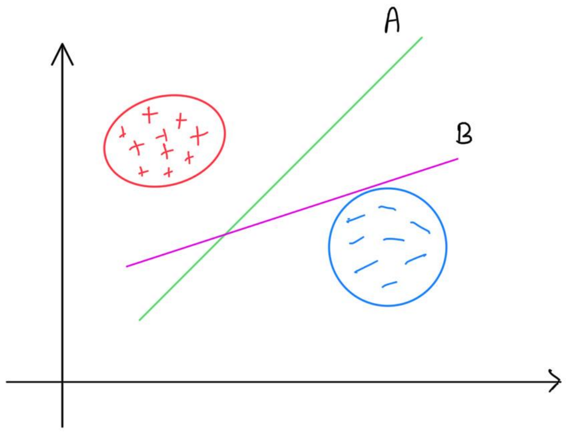

Figure 1: 2 Hyperplanes

The above figure demonstrates why we are interested in finding the best possible hyperplane. Hyperplane A is much a better choice as it has a larger margin than B.

3 SVM - Separable Case

Lets consider the case where there is no overlap between the data points.The necessary condition for the separable case is that the convex hull corresponding to all the positive points and all the negative points do not intersect. To find a hyperplane the following problem can be solved:

$$y(w^T x + b) >= \Delta, \forall (x_i, y_i) \tag{8}$$

where w is the parameter vector, and b is the bias or offset scalar.

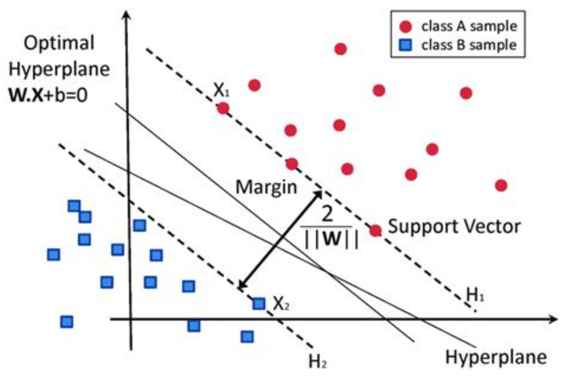

Figure 2: Representation of Support Vector Machine

The use of Δ is to create a margin, which can be likened to breathing space. The issue with the above equation is it will be satisfied by a set of infinite equations and it does not take into account the notion of the "best" hyperplane we had established. To take this into account we first can modify b in the equation and then scale w and b to make the equation look like the following (it is good practice to attempt to prove why such a b will always exist and prove that any separating hyperplane satisfying the previous equation can be converted to this form)

$$y(w^T x + b) >= 1, \forall (x_i, y_i) \tag{9}$$

Now we can see in the figure that the distance between the separating hyper planes is given by $2/||w||$ and to maximise this, which means to get the "best" solution we wanted with maximum margin for error due to noise, we need to minimise $||w||$. There is a slight issue here that the problem mentioned before is not a convex problem making it tougher to solve and increasing difficulty in showing results for the same. Thus we phrase the problem as

$$\begin{aligned} &\text{minimize} \quad ||w||^2 \\ &\text{subject to} \quad y(w^T x + b) \geq 1, \forall (x, y) \end{aligned} \tag{10}$$

Now, distance from (x,y) to $w^T x + b = 0$ is given by:

$$\frac{w^T x + b}{||w||} \tag{11}$$

So we want to do the following:

$$\underset{w,b}{\text{maximize}} \quad \underset{i}{\text{minimize}} \quad \frac{w^T x + b}{||w||} \tag{12}$$

Here we will show the **equivalence between (10) and (12)** :

We will make an observation here, say (w*, b*) is a solution to :

$$\underset{w,b}{\text{maximize}} \quad \underset{i}{\text{minimize}} \quad \frac{w^T x + b}{||w||} \tag{13}$$

$(\mu w^*, \mu b^*)$ is a solution too.
i.e w and b can be scaled, we can just focus on:

$$\text{min} \quad w^T x + b = 1 \tag{14}$$

$$\implies \quad y_i(w^T x_i + b) \geq 1 \quad \forall i \tag{15}$$

$$\text{We have : } \quad \max_w \frac{1}{||w||} \ni y_i(w^T x_i + b) \geq 1 \quad \forall i \tag{16}$$

$$\implies \min_w ||w^2|| \quad \ni y_i(w^T x_i + b) \geq 1 \quad \forall i \tag{17}$$

which is just (10).

4 SVM - Non-Separable Case

It is not always the case that different labelled clusters are away from each other or non-overlapping. There are cases where these labelled data points are mixed up, that is, their clusters overlap.

In these Non-Separable cases, there is an overlap between the differently labelled data points. An easier way to visualize this overlap is to assume the points to be coloured pins, now you try to bind a class or a label by wrapping the exterior pins in a rubber band, you do the same for the different class but with a different rubber band, if these rubber bands overlap anywhere, you are free to say you are dealing with a Non-separable case.

Now the question arises *How do we deal with these cases ?*. Using the method for separable cases, won't fetch us the **w,b** parameters, as here

$$y_i(w^T x_i + b) \geq 1, \forall (x_i, y_i) \tag{18}$$

is not satisfied by all of the points, as the ***convex hull*** for the two classes ***overlap***.
One way to find the optimal hyperplane in such a case is to ignore the overlapping points. This cannot always be done since computing overlap is difficult, especially in higher dimensions. Another drawback is the case shown in the given below figure.

Here if we try ignoring or deleting the overlapped points, we would end up having minimal data from one class causing a useless classification problem. Another way to deal with this problem is to introduce a relaxation parameter, which we'll discuss shortly.

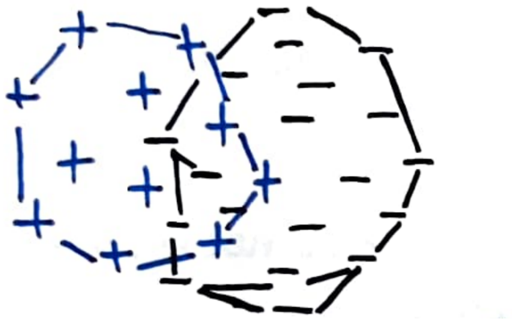
Figure 3: Non-Seperable Case

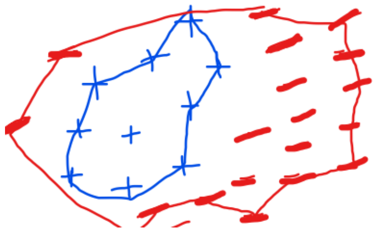
Figure 4: Non-Seperable Case

4.1 Using the Method of Ignoring Overlapping Points

Here we aim to ignore or delete the minimal no. of points from the data such that no overlapping is seen.
Step 1:

Let us begin with constructing a variable

$N_{overlap}\left(D = \{x_i, y_i\}\right)$

This variable measures the number of overlapping points in domain D. Now we would start with deleting two points (|S|), and check if our $N_{overlap}$ is zero, if not we would increment the value of S, i.e delete 3 points and so on until our $N_{overlap}$ is zero. the value of s which results in zero overlaps is our **min s**.

$$\underset{S:|S|\leq b}{\text{minimize}} \quad N_{overlap}(D|S) \tag{19}$$

This program will keep on iterating until the Set of points S is searched which makes the overlap zero.

Step 2:

Here we try solving the same optimization problem as done for the separable case.

$$\begin{aligned} &\text{minimize} \quad ||w||^2 \\ &\text{subject to} \quad y(w^T x + b) \geq 1, \forall (x, y) \in D| \end{aligned} \tag{20}$$

or

$$\max_{w,b} \min_{i} \frac{|w^T x_i + b|}{||w||} \tag{21}$$

But this algorithm would take high computational power in higher dimensions and a very long time to compute as the complexity of the algorithm is $O(n^2)$. We need something faster and more reliable, here is where the concept of relaxation or slackness comes.

4.2 Using Relaxation $\xi_{x,y}$

We know that the condition $y_i(w^T x_i + b) \geq 1$ causes the problem. *How about separate margins*, that is we add a **relaxation/slackness** parameter as a function of x_i, y_i so as to make all of the points satisfy the variable condition? In this method, we use relaxation ξ_{x_i,y_i} at all points. It can be thought of as how far a misclassified point is from the separating hyperplane. This can be described by:

$$y_i(w^T x_i + b) \geq 1 - \xi_{x_i,y_i} \tag{22}$$

where, $\xi_{x_i,y_i} \geq 0$

Using the previous optimization problem would let ξ_{x_i,y_i} freely vary and attain very large values, which would then cause any w to satisfy the conditions, which completely defeats the purpose of optimization. Thus we must also **minimize** ξ_{x_i,y_i} in our optimization problem.

So the optimization problem becomes:

$$\text{minimize} \quad ||w||^2 + C \sum_{x,y \in D} \xi_{x_i,y_i} \tag{23}$$

Let $w^*, b^*, \xi^*_{x_i,y_i}$ be the optimal solution.

Here the minimization of the $C \sum_{x,y \in D} \xi_{x_i,y_i}$ ensures the equality relation obtained.

So

$$y_i(w^{*T} x_i + b^*) = 1 - \xi^*_{x_i,y_i} \tag{24}$$

We know if our point is correctly classified (a good point) $1 - y_i(w^T x_i + b) \leq 0$, otherwise $1 - y_i(w^T x_i + b) \geq 0$. Thus we can set $\xi_{x_i,y_i} = max(0, 1 - y_i(w^T x_i + b))$ as we would want ξ_{x_i,y_I} to be **zero for optimal points** (good points).

As $max(x, 0)$ is equivalent to $Relu(x)$, we can replace it to make the following optimization problem.

The final optimization problem becomes:

$$\text{minimize} \quad ||w||^2 + C \sum_{i \in D} Relu(1 - y_i(w^T x_i + b)) \tag{25}$$

This is called **Hinge Loss**.

It is to note here that the parameter C is to be set by us, and as we keep on increasing it the classes would be separated more and more accurately but this would cause **overfitting** to our data. Thus a **validation set** should tell us its performance, and the best C must be decided.

References

CS725 : Foundations of Machine learning - Lecture Notes by Ajay Nagesh, 2011