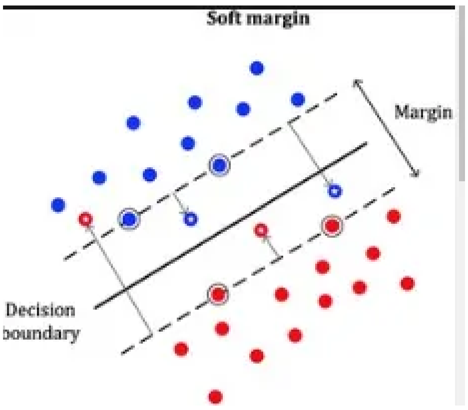
The Maths behind Soft margin classifier — Support vector machines

krishna sri vaibhav · Follow

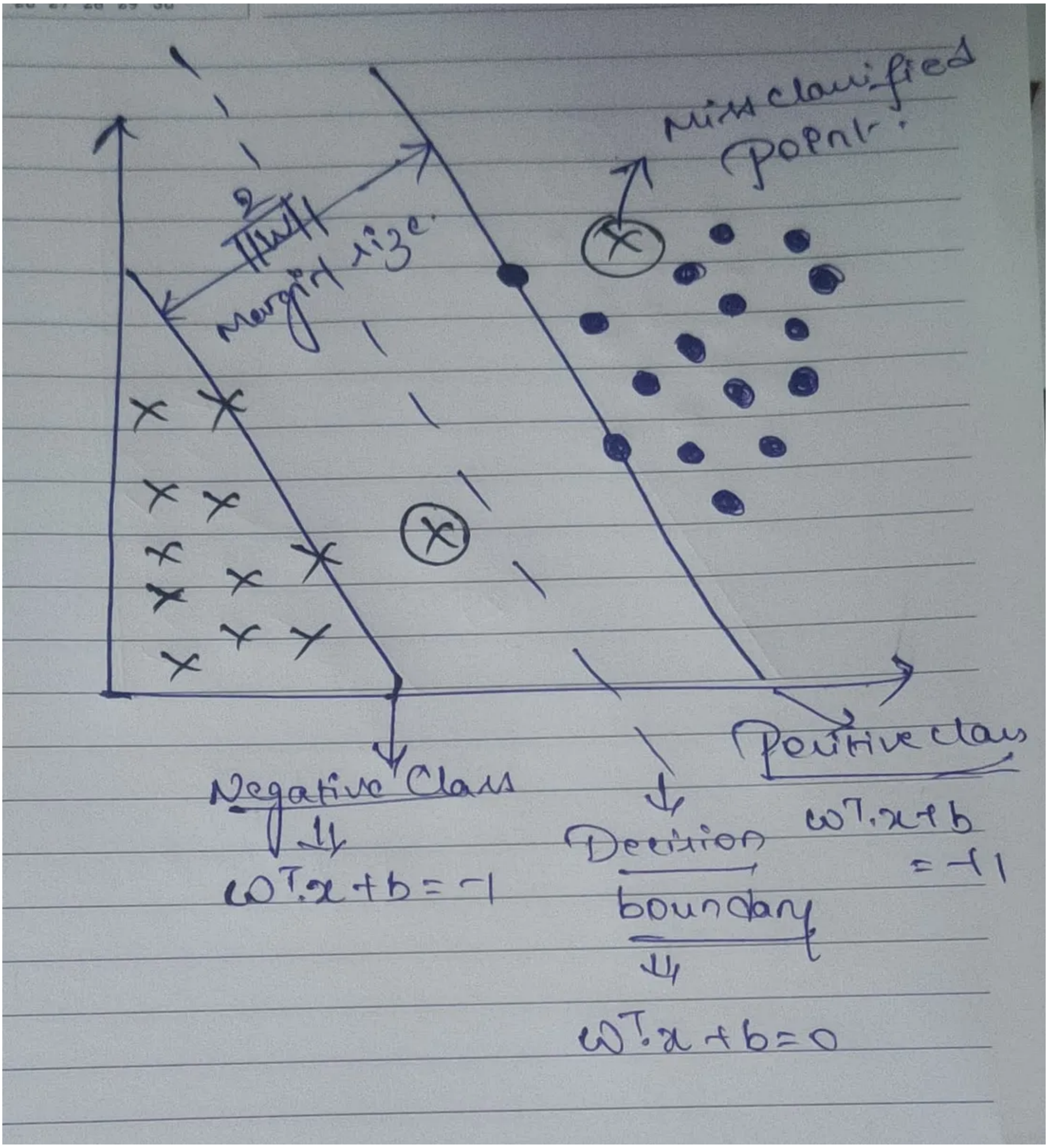
4 min read · Jun 25

In the previous blog, I explained about the Hard margin classifier which is limited only to linearly separable problems. Now I'm gonna tell you about the Soft margin classifier which can also deal with Non-linear data.



The main basic thing is that the Hard margin classifier won't allow misclassifications. whereas the soft margin classifier allows misclassifications making it handle the non-linear data.

Consider the below picture



As you guys can see, there's a data point that actually belongs to the negative class but is misclassified as the positive class. and there's another data point that is correctly classified as the negative class but it is inside the margin. so for both data points there's actually some error. That error needs to be minimized.

> *Note: This soft margin classifier deals with maximizing the margin as well as Minimizing the misclassification error.*

Now let's jump into the maths involved in it.

Hinge loss function:

This hinge loss function tells about how much loss or error has been incurred for each data point. For correctly classified data points the hinge loss will always be zero and for misclassified points, the error will be greater than zero.

from the decision rule,

w.r.t,

$$y_i(w^T.x+b) \geq +1$$

This is true for any value of y.

from this:

$$\Rightarrow 1 - y_i(w^T.x+b) \leq 0$$

Now the hinge loss would be.

$$\boxed{L(y, f(x)) = \max(0, 1 - y * f(x))}$$

$$\boxed{f(x) = w^T x + b}$$

SUN 14

This is the Hinge loss function and this will be used to determine the loss for each data point.

Now if you see the first picture there are two data points, one which is misclassified and the other is classified correctly but it is inside the margin.

Let’s apply that hinge loss to those data points and see what the output looks like.

1. For a correctly classified data point that lies in the positive class [y=1]:- max(0,1-y*f(x)) = max(0,1–1*(>=1)) = 0.

2. For a correctly classified data point that lies in the negative class [y=-1]:- max(0,1-y*f(x)) = max(0,1-(-1)*(<=-1)) = max(0,1*(<=-1)) = 0

3. For a negative data point that lies in the positive class [y = -1]:- max(0,1-y*f(x)) = max(0,1-(-1)*(>1)) = max(0,>1) = >1

4. For a negative data point that is correctly classified but it’s inside the margin [y=-1]:- max(0,1-(-1)*([-1 to 0])) = max(0,[-1 to 0]) = [-1 to 0]

So This is how the loss will be generated, for correctly classified points the loss will always be zero, and the negative point which is classified as the positive has a loss greater than 1. The correctly classified negative point which is inside the margin has a loss ranging between -1 to 0.

Now this loss, which is known to be the **Misclassification error** should be minimized.

Final equation:

The final equation looks like this

$$\boxed{\min_{w,b} \frac{1}{n}\sum_{i=1}^{n} \max(0, 1-y_i(w_i^T \cdot x_i + b)) + \lambda \|\vec{w}\|}$$

where $\lambda\|\vec{w}\|$ is the penalty term

$\lambda \rightarrow$ Regularization parameter

$\|\vec{w}\| \rightarrow$ Magnitude of weights

Observe the above equation carefully it infers two things

1. Minimizing the misclassification error

2. Increasing the size of the margin.

We need to choose between these two things, If you guys want to reduce the misclassification error then that error will be reduced but the margin will be small. If you focus on increasing the margin size, then misclassifications might happen. These all are just assumptions. In real-world scenarios, while working with real-time problems, it's up to you to choose what to focus on, and what's your requirement in the end!

Let's get into that equation again, And now if you see I added a term at the end of the equation goes like **lambda multiplied by the magnitude of the weighted vectors.** so that's actually a **Regularization term or penalty term**, If you remember that from the regression concepts, what that actually does is, it alters the magnitude of the coefficients to achieve an optimal solution by preventing overfitting the data.

This **Lambda** is the **Regularization parameter**, This plays a crucial part in the whole equation. so when lambda is set too small it actually decreases the magnitude and increases the size of the margin and pays less attention to misclassification errors. when the lambda is set too large it pays more attention to misclassification errors and pays less attention to the margin size.

As I said above it's up to you to choose Margin or Misclassification error!

So this is all about the soft margin classifier. By reading this I hope you will get some intuition on how this classifier works.

Support Vector Machine

Follow

Written by krishna sri vaibhav

26 Followers

Machine learning student

More from krishna sri vaibhav