

Supervised Learning: Support Vector Machines

April 16, 2021

1 Motivation

When we split a set of points via a boundary, to ensure that the model generalizes best, we need to pad out said boundary with as much space as possible. If our data is assumed to be representative of the data that shall be used for prediction, then having many points on the boundary means that the data you test with or run in production has a higher chance to be misclassified. How do we maximize the margin?

2 Defining Error

To solve this, we treat each point that falls on the right side but inside the margin and each misclassified point as an error. Essentially,

$$Error = ClassificationError + MarginError$$

Recall the perceptron algorithm. Here, error is defined as the distance from the line, meaning $|\beta^T X|$. All misclassified points are now defined as points $|\beta^T X| = D$ for margin distance D. Thus, for each point x_i misclassified, $Error = \sum_{x_i} \beta^T X$

3 How to Minimize this Error

Some equations to note:

$$Margin = \frac{2D}{|\beta|}$$

$$Error = |\beta D|^2$$

$$DistanceBetweenpoints = \frac{D}{\beta}$$

We know how to minimize this, Gradient Descent.

4 The C Parameter

Naturally not all problem's requirements are created equal. Sometimes we are more willing to tolerate a smaller margin in favor of wanting our model to make less mistakes, or a large margin with some tolerance for mistakes. So, error becomes:

$$Error = C \times ClassificationError + MarginError$$

5 Polynomial Kernel Trick

What if the points cannot be separated easily by a line? If this is the case, we can map the points to an equation and then try drawing a line between those re-mapped points. Here is how we do it:

- Given: $\vec{x} = \begin{bmatrix} x_n \\ \cdots \\ x_1 \\ 1 \end{bmatrix}$

- Choose an equation $f(\vec{x}) = x_{n+1}$ (go up one dimension) that splits the points well in the x_{n+1} dimension
- Choose a value on the x_{n+1} axis, c
- Then, the equation that splits the data into categories becomes $f(\vec{x}) = c$

Note that $f(\vec{x})$ does not have to be a polynomial or even pass the "vertical line test". You can draw circles/irregular shapes as boundaries if you want to, so long as you can write it in a function form.

6 RBF Kernel

You can also do this with different kernel functions, like the normal distribution $y = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. We can assign $\gamma = \frac{1}{2\sigma^2}$, and rewrite the equation as $y = \sqrt{\frac{\gamma}{\pi}}e^{-\gamma(x-\mu)^2}$. Basically, define a normal distribution for each point where the extrema of the function is at the point. So for every point $(\vec{x}, -1)$ (mapping zeroes to negative one) or $(\vec{x}, 1)$, we can map each point to the gamma function in n dimensions, and then drawing an n-plane to cut the points in half will give you an n-1 dimension cross section that is your boundary line.