Support Vector Machine—Theory

Decision Rule:

$$\boxed{\vec{w} \cdot \vec{x} + b \geq 0 \quad Then \quad +}$$

w is the vector that is prependicular to the decision boundary.

Maximize the Width of the Street

$\boxed{y_i(\vec{x}_i \cdot \vec{w}_i + b) - 1 = 0}$ for x_i in gutter

y_i : such that $y_i = +1 for + samples$

$y_i = -1 for - samples$

Width of the street = $(\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{\|w\|}$

$\vec{x}_+$ and $\vec{x}_-$ are the samples in gutter.

Therefore it satisfys equation 2, so we have:

max (Width = $\frac{2}{\|w\|}$)

The problem here we have is to maximize the Width. With the equation *Width* = $\frac{2}{\|w\|}$, it involves the calcualtion of square root, which is complex. So to make calculation more convenient, we transform this maximization problem to minimization problem:

min ($\frac{1}{2}\|w\|^2$)

Lagrange

Then it becomes the problem of finding extremum with constraints. Thus Lagrange method is employed to solve the problem

So we have Lagrange:

$L = \frac{1}{2}\|\vec{w}\|^2 - \sum \alpha_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1]$

To find out the extremum of Lagrange above:

$\frac{dL}{d\vec{w}} = \vec{w} - \sum \alpha_i y_i \vec{x}_i = 0 \Longrightarrow \boxed{\vec{w} = \sum \alpha_i y_i \vec{x}_i}$

$\frac{dL}{db} = \sum \alpha_i y_i = 0 \Longrightarrow \boxed{\sum \alpha_i y_i = 0}$

Plug those back to Lagrange, then we have Lagrange like:

$\boxed{L = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \vec{x}_i \vec{x}_j}$

It shows that the optimization only depends on the dot production of samples.

Back to the decision rule:

$\sum \alpha_i y_i \vec{x}_i \cdot \vec{u} + b \geq 0$ Then +

where $\vec{u}$ represents unknown value that we want to predict whether it's positive or negative.

It is consistently showing that the decision rule also depends only on the dot product of those sample.

Kernel Function

For non-linear classification problem, **Kernel Function** is introduced.

For non-linear classification, the basic idea is to transform dots into another space ($\phi(\vec{x}_i)$ and $\phi(\vec{x}_j)$) that makes the seperation possible.

So the Lagrange is becomes to:

$L = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \phi(x_i) \phi(x_j)$

To make this Lagrange maximized, just make $\phi(\vec{x}_i) \phi(\vec{x}_j)$ maximized.

Kernel Function: provides the dot product of two vectors in another space, so that we don't have to know the transformation into the other space and keep the complexity not exploding due to high dimensional transformation.

Some Common Kernels:

$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d$

$$K(\vec{x}_i, \vec{x}_j) = exp(-\gamma ||\vec{x}_i - \vec{x}_j||^2)$$

It was 30 years in between the concept and anybody ever hearing about it. It was 30 years between Vapnik's understanding of Kernels and his appreciation of their importance. And that's the way things often go, great ideas followed by long periods of nothing happening, followed by an epiphanies moment when the original idea seemed to have great power with just a little bit of a twist. And then, the world never looks back. And Vapnik, who nobody ever heard of until the early 90s, becomes famous for something that everybody knows about today who does machine learning.