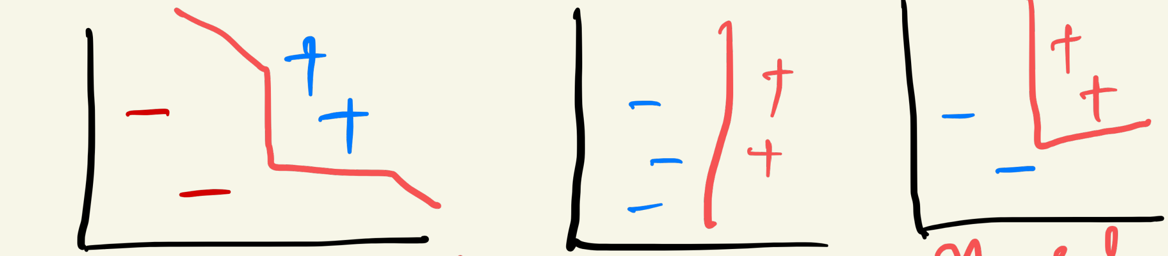
Support Vector Machines

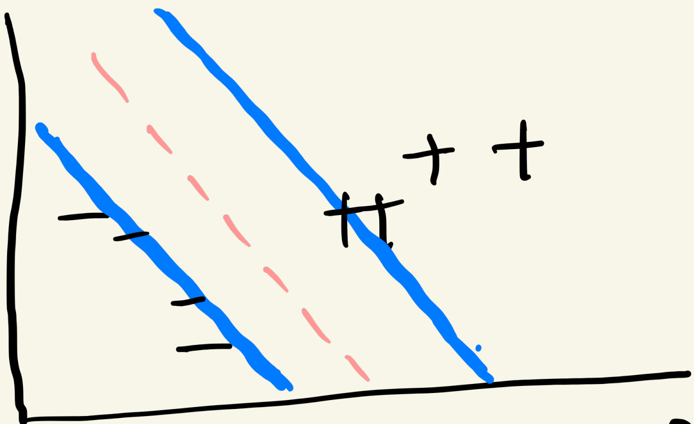
Support Vector Machines



Neural Net | 1 D | Neural Net

{ * When stuck switch to different perspective. }

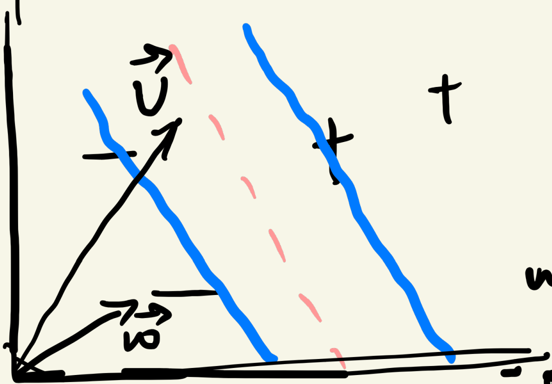
It's all about decision boundaries



We want to draw a straight line, but which straight line is the question.

A straight line which would put widest street to separate +ve from -ve.

that's why it's called widest possible approach!!



$\vec{w} \perp$ median of gutter (street)

$\vec{u} \rightarrow$ unknown

what we are interested is if the unknown is on the right side of the street or left.

$$\boxed{\vec{w}\cdot\vec{u} \geq c} \text{ — (A)}$$

$c = -b$

without loss of generality

decision rule $\rightarrow$ $\boxed{\vec{w}\cdot\vec{u} + b \geq 0 \rightarrow \text{then } +}$ — (1)

we need constraints to calculate b & w

$\vec{w}\cdot x_+ + b \geq 1$ — (B) $\rightarrow$ for positive samples

$\vec{w}\cdot x_- + b \leq -1$ — (C) $\rightarrow$ for negative samples.

to make life a little easier & convinient we introduce y_i such that $y_i = +1$ for +ve
$= -1$ for -ve

$\vec{w} \cdot \bar{x}_+ + b \geq 1$ $\quad\quad$ $y_i(\vec{w} \cdot x_i + b) \geq 1$

$\vec{w} \cdot \bar{x}_- + b \leq -1$ $\quad\quad$ $y_i(\vec{w} \cdot x_i + b) \geq 1$

Oops the two eqnens. are same after multiplyin y_i to (B) & (C) because you multiply -1 to (C) which would revert the inequality.

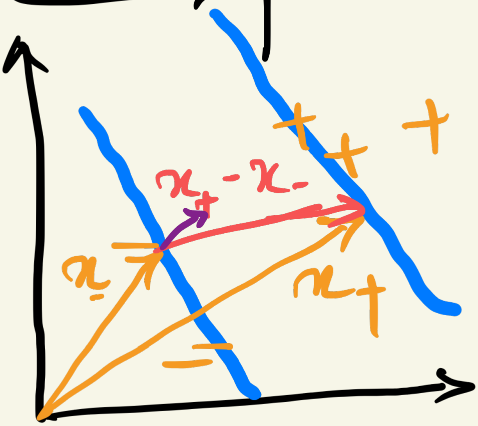
$$y_i(\vec{w} \cdot x_i + b) - 1 \geq 0$$

$$y_i(x_i \cdot \vec{w} + b) - 1 = 0$$

constraints

2

for x_i in the gutter



If I had a unit normal that's normal to the median line of the street, if it's unit normal, then i could take dot product of this unit normal & this diff. and that would be the width of the street.

(3) Width $= (x_+ - x_-) \cdot \frac{\vec{w}}{||w||} = \frac{2}{||w||}$

from (2): $x_+ \to 1-b$, $x_- \to 1+b$

So what we gonna do is max $\frac{2}{||w||}$

max $\frac{1}{||w||} \rightsquigarrow$ min $||w||$

$$\boxed{\min \frac{1}{2} ||w||^2}$$

To find the extremum of a function with constraints requires use of Lagrange multipliers.

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

Lagrangian

the ones which are gonna be non 0 are going to be the ones connected with vectors that lie in gutter, the rest are gonna be 0.

$$\frac{\partial L}{\partial w} = \vec{w} - \sum \alpha_i y_i \vec{x}_i = 0$$

$$\boxed{\vec{w} = \sum_i \alpha_i y_i \vec{x}_i}$$

$\vec{w}$ is a linear sum of samples, some samples.

$$\frac{\partial L}{\partial b} = -\sum_i \alpha_i y_i = 0$$

3

$$\rightarrow \boxed{\sum_i \alpha_i y_i = 0}$$

Plug $\vec{w}$ to L

$$L = \frac{1}{2}\left(\sum \alpha_i y_i \vec{x}_i\right)\left(\sum \alpha_j y_j \vec{x}_j\right) - \left(\sum \alpha_i y_i \vec{x}_i\right)\cdot\left(\sum \alpha_j y_j \vec{x}_j\right) - \underbrace{\sum \alpha_i y_i b}_{0} + \sum \alpha_i$$

4

$$\boxed{L = \sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boxed{x_i \cdot x_j}}$$

optimization depends on dot product of samples

So the decision rule is gonna be

$$\sum \alpha_i y_i \vec{x}_i \cdot \vec{u} + b \geq 0 \rightarrow \text{then +ve}$$

If the optimization algorithm doesn't get stuck in the Local maximum it should be nice.

it's a convex space and so it wont get stuck in local maximum.

What if when samples are linearly inseperable.

I need a transformation $\phi(\vec{x})$ and because the maximization is only dependent on dot product all I need to do is $\boxed{\phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \text{ to map}}$

$$\phi(\vec{x}) \cdot \phi(\vec{u})$$

If I have a func.

$$\underline{K(x_i, x_j)} = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$

Kernel function → which provides me the dot product of those two vectors in another space, i.e I don't need to know the transformation in that space.

Popular Kernels :

linear $\rightarrow \left(\vec{u} \cdot \vec{v} + 1 \right)^n \rightarrow$ $n = 2$ slightly nonlinear

radial basis $\rightarrow e^{-\frac{\| x_i - x_j \|}{\sigma}} \rightarrow$ If σ is very small we might end up overfitting.