

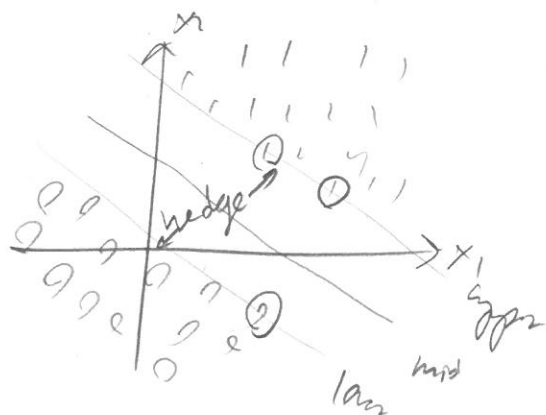
Math 3904 Lect 2/2/19

1

$p=2$ $y = \{0, 1\}$

Create a linear threshold model for lin. sep data set.

$$\mathcal{H} = \{ \mathbb{I} \vec{w} \cdot \vec{x} \geq 0 : \vec{w} \in \mathbb{R}^3 \}$$



Why not create a wedge?

The top is the lin. sep line "closest" to the $y=1$'s and the bottom is the lin. sep line "closest" to the $y=0$'s.

The model would also see the middle line.

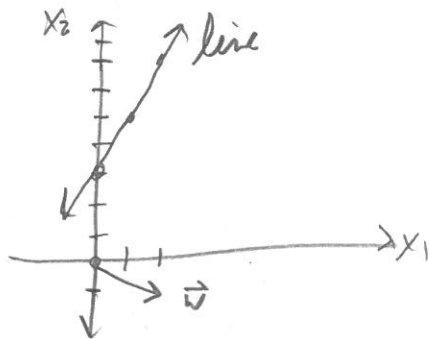
"Max margin hypothesis" - Proven to be optimal linear classifier in 1998.

Which data points matter most? the ones on the upper and lower lines. Since each point is a vector \vec{x}_i , these are called the "support vectors" and the classifier is called the "support vector machine".

most important observation model

To fit this, let's rewrite $\mathcal{H} = \{ \mathbb{I} \vec{w} \cdot \vec{x} + b \geq 0 : \vec{w} \in \mathbb{R}^p, b \in \mathbb{R} \}$

Let's talk about lines. eg li: $x_2 = 2x_1 + 3 \Rightarrow 1 \cdot 2x_1 - x_2 + 3 = 0$



$$\Rightarrow \text{li: } \underbrace{\begin{bmatrix} 2 \\ -1 \end{bmatrix}}_{\vec{w}} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\vec{x}} - \underbrace{3}_{b} = 0$$

Hesse Normal Form!
 $\vec{w} \cdot \vec{x} - b = 0$

What is \vec{w} on the plot?

It's the normal vector, i.e.

\perp to the line (or hyperplane)

let $\|\vec{w}\| = \sqrt{\sum_{i=1}^p w_i^2} = \sqrt{\vec{w} \cdot \vec{w}}$, the length of the vector.

let $\vec{w}_0 = \frac{\vec{w}}{\|\vec{w}\|}$, the normalised normal vector

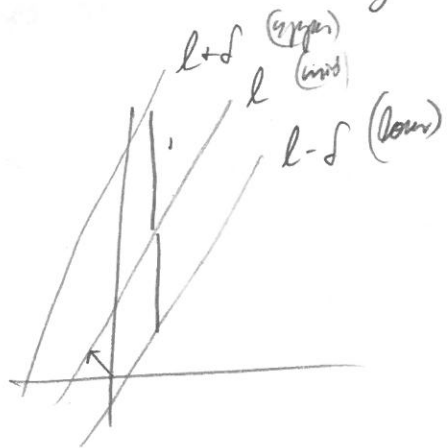
$x \in \mathbb{R}$, $\vec{z} = \alpha \vec{w}_0$, the normal vector of length $|\alpha|$.

Let us find α s.t. \vec{z} is the vector from the origin to the line L .

$$\Rightarrow \vec{w} \cdot \vec{z} - b = 0 \Rightarrow \vec{w} \cdot (\alpha \vec{w}_0) - b = 0 \Rightarrow \alpha \frac{\vec{w} \cdot \vec{w}}{\|\vec{w}\|} - b = 0$$

$$\Rightarrow \alpha \|\vec{w}\| - b = 0 \Rightarrow \alpha = \frac{b}{\|\vec{w}\|}$$

Let us draw a wedge around L by adding and subtracting δ .



The normal vector α to the upper is

$$\vec{w} \cdot \vec{z} - (b + \delta) = 0 \Rightarrow \vec{w} \cdot (\alpha \vec{w}_0) = b + \delta$$

$$\Rightarrow \alpha_u = \frac{b + \delta}{\|\vec{w}\|}$$

lower is

margin :=

$$\alpha_u - \alpha_l = \frac{b + \delta}{\|\vec{w}\|}$$

\Rightarrow distance between upper and lower is $\alpha_u - \alpha_l = \frac{2\delta}{\|\vec{w}\|}$

Now recall that

$\vec{w} \cdot \vec{x} + b = 0$ is overparametrised with infinite solutions

Since $C(\vec{w} \cdot \vec{x} + b) = 0 \quad \forall C \in \mathbb{R}$. If we let $\delta = 1$, there is only one solution

$$\Rightarrow \text{margin} = \frac{2}{\|\vec{w}\|}$$

Force all $y=1$'s to be $\geq l+1$

1, 1, 1, 1, 0's, ..., $\leq l-1$

$$\forall i \text{ s.t. } y_i = 1, \quad \vec{w} \cdot \vec{x}_i - (b+1) \geq 0 \Rightarrow \vec{w} \cdot \vec{x}_i - b \geq 1$$

multiply both sides by $(y_i - \frac{1}{2})$

$$\Rightarrow (y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i - b) \geq y_i - \frac{1}{2}$$

Since $y_i = 1 \forall i$

$$\Rightarrow (y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i - b) \geq \frac{1}{2}$$

$$\forall i \text{ s.t. } y_i = 0, \quad \vec{w} \cdot \vec{x}_i - (b-1) \leq 0 \Rightarrow \vec{w} \cdot \vec{x}_i - b \leq -1$$

multiply both sides by $-(y_i - \frac{1}{2})$

$$-(y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i - b) \leq -\frac{1}{2}$$

$$-(y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i - b) \leq y_i - \frac{1}{2}$$

\forall

$$(y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i - b) \geq \frac{1}{2}$$

Since $y_i = 0 \forall i$

$$\Rightarrow (y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i - b) \leq \frac{1}{2}$$

this inequality satisfies all y_i either 0 or 1!

Now we can frame the problem:

$$\text{maximize } \frac{2}{\|\vec{w}\|} \quad \text{s.t.} \quad \forall i \quad (y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i - b) \geq \frac{1}{2} \quad \text{then } \vec{w} \in \mathbb{R}^p, b \in \mathbb{R}$$

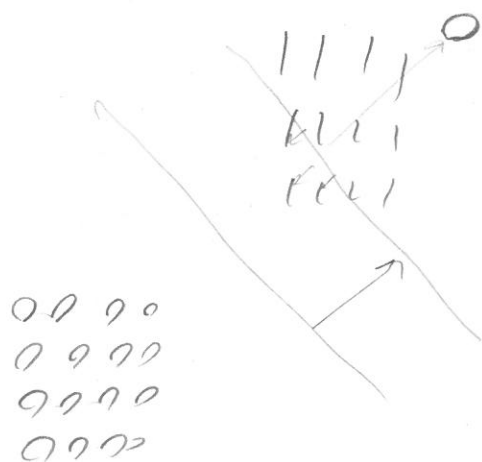
$$\Leftrightarrow \text{minimize } \|\vec{w}\| \quad \text{s.t.} \quad \dots$$

This will find the best perceptron model

What if data not lin. sep? No solution!

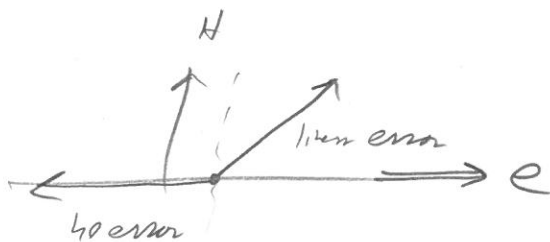
We can allow for violations in this condition. Makes sense to penalize based on how far the line is from the pt.

Consider this following loss function:



$$H_i := \max \left\{ 0, \frac{1}{2} - (y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i - b) \right\}$$

↑
"hinge loss"



Let's make sure this works. Imagine x_i, y_i is above the region by $+d$

$$(y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i + b) = \frac{1}{2} + d \geq d \quad \leftarrow \text{correct}$$

$$H_i = \max \left\{ 0, \frac{1}{2} - (\frac{1}{2} + d) \right\} = \max \{ 0, -d \} = 0 \quad \text{no error!}$$

Imagine x_i, y_i is below the region by $+d$:

$$(y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i + b) = \frac{1}{2} - d \neq d$$

$$H_i = \max \left\{ 0, \frac{1}{2} - (\frac{1}{2} - d) \right\} = \max \{ 0, d \} = d \quad \leftarrow \text{penalized by } d!$$

we wish to minimize:

$$SHE = \sum_{i=1}^n \max \left\{ 0, \frac{1}{2} - (y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i - b) \right\}$$

Sum

hinge errors in D

↑ the further the pt. is away, the more you pay

But we also want to minimize $\|\vec{w}\|$ i.e. make margin margin.
 Let's put everything together:

anymin $\left\{ \underbrace{\frac{1}{n}SHE}_{\text{minimize any huge loss}} + \underbrace{\lambda \|\vec{w}\|^2}_{\text{maximize margin}} \right\}$ objective function of Vapnik (1963)

There is a tradeoff between these two goals.

What is λ ? It is a "hyperparameter", "tuning parameter". It is set by you! (soon we will revisit this). It controls how the tradeoff between these goals is to be done. $\lambda \uparrow \Rightarrow$ maximum margin a priority

$\lambda \downarrow \Rightarrow$ minimize mistakes. If $\lambda \rightarrow \infty \Rightarrow \|\vec{w}\| \rightarrow 0$. This is silly!

$g = A(D, \mathcal{H}, \lambda)$ hyperparameters

eg gradient descent, etc.

If λ is selected "reasonably", we can use numerical methods to solve for \vec{w}, b which minimize. We can use packages in R that do this.

What if $y \in \{0, 1, \dots, L-1\}$, ^{var. with} nominal $L > 2$ levels

A model to predict is a classification model. Can we use

SVM? No... just for binary classification (as is now)

Null Model $g = \text{Mode}[y]$ (again).

How about g : Given that finds closest \vec{x}_i and assigns the y_i ? "Nearest Neighbor".

$$g(\vec{x}_i) = y_j \text{ s.t. } j = \underset{j \neq i}{\operatorname{argmin}} \{d(\vec{x}_i, \vec{x}_j)\}$$

$\mathcal{U} = \{ \} ?$ } Distinct so greatly precisely what they look like,
 $\mathcal{A} = ?$ } but they are looking in the distance.

We have a hyperparameter, $\dim. d$. The default is to apply Euclidean:

$$d(\vec{x}_i, \vec{x}_k) := \|\vec{x}_i - \vec{x}_k\|^2 = \sum_{j=1}^d (x_{ij} - x_{kj})^2$$

Variant: Instead of returning the ^{y_i & d} closest \vec{x}_i , find the K nearest \vec{x}_i 's and return the mode of the K y_i 's.