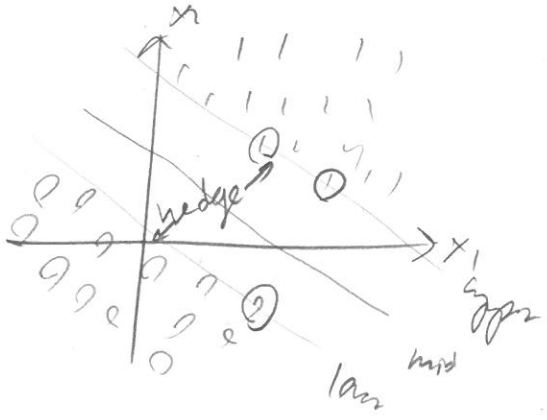
Math 3909 Lec4 2/7/19

$p=2$ $y=\{0,1\}$. Create a linear threshold model for this. sep data set.

$\mathcal{H} = \{ \mathbb{1}_{\vec{w}\cdot\vec{x}\geq 0} : \vec{w}\in\mathbb{R}^3 \}$



Why not create a wedge?

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

The model & would choose the midline.

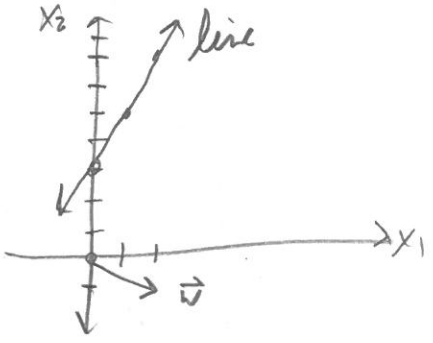
"Max margin hyperplane". Proven to be optimal linear classifier in 1998.

Which data points matter most? the ones on the hyperplane and large 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 observations) (model)

To fit this, let's rewrite $\mathcal{H} = \{ \mathbb{1}_{\vec{w}\cdot\vec{x}+b\geq 0} : \vec{w}\in\mathbb{R}^p, b\in\mathbb{R} \}$

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



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

Hesse Normal Form!

$\vec{w}\cdot\vec{x} - b = 0$

Where is $\vec{w}$ on the plot?

It's the normal vector, i.e.

$\perp$ to the line (or hyperplane)

$= \sqrt{\vec{w}\cdot\vec{w}}$

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

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

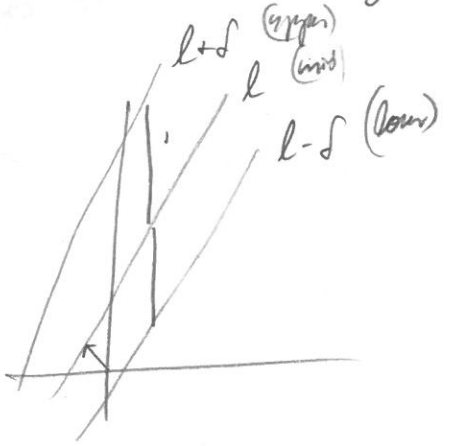
$\forall \alpha \in \mathbb{R}$, $\vec{z} = \alpha \vec{w}_0$, the normal vector of length $|\alpha|$.

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

$$\Rightarrow \vec{w}\cdot\vec{z} - b = 0 \Rightarrow \vec{w}\cdot(\alpha\vec{w}_0) - b = 0 \Rightarrow \alpha \underbrace{\frac{\vec{w}\cdot\vec{w}}{\|\vec{w}\|}}_{\|w\|^2} - b = 0$$

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

Let us draw a wedge around l by adding and subtracting δ



The normal vector to the upper is

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

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

/ / / / / / / lower is

$$\text{margin} := \alpha_l = \frac{b-\delta}{\|w\|}$$

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

Now recall that $\vec{w}\cdot\vec{x} + b = 0$ is overparameterized 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

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

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

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

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

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

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

Since $y_i = 1$ $\forall i$ $\Rightarrow$ $\left(y_i - \frac{1}{2}\right)\left(\vec{w}\cdot\vec{x}_i - b\right) \geq \frac{1}{2}$

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

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

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

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

$$\Downarrow$$

$$(y\_i - 1/2)\left(\vec{w}\cdot\vec{x} - b\right) \geq \frac{1}{2}$$

Since $y_i = 0$ $\forall i$ $\Rightarrow$ $\left(y_i - \frac{1}{2}\right)\left(\vec{w}\cdot x_i - b\right)$ >= $\frac{1}{2}$

this inequality satisfies all y_i either 0 or 1!

Now we can frame the problem:

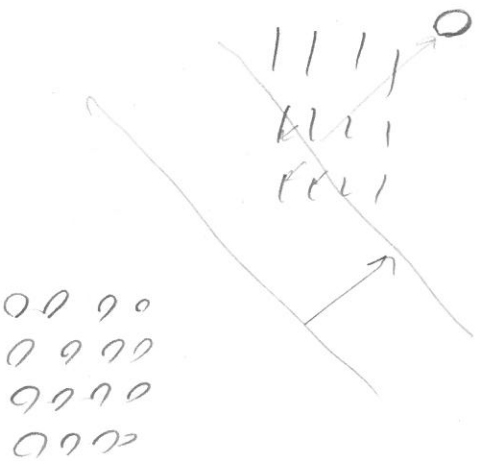
maximize $\frac{2}{\|w\|}$ s.t. $\forall i$ $\left(y_i - \frac{1}{2}\right)\left(\vec{w}\cdot\vec{x} - b\right) \geq \frac{1}{2}$ where $\vec{w} \in \mathbb{R}^p$, $b \in \mathbb{R}$

$\Leftrightarrow$ minimize $\|w\|$ s.t. , , , , , , $\rightarrow$

This will find the best perceptron model

What if data not lin. sep? No solution!

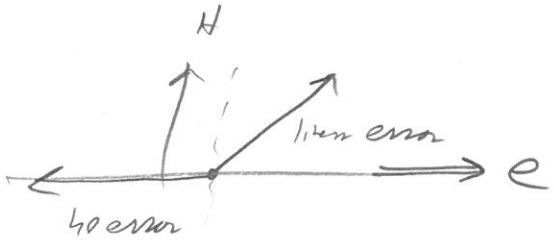
We can allow for violations in this condition. Makes sense to penalize based on how far the line is from the para.



Consider this following loss function:

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

"hinge loss"



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

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

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

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

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

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

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

We wish to minimize:

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

Sum hinge errors in D

Here we also want to minimize $\|\vec{w}\|$ i.e. make maximum margin.

Let's put everything together:

objective function of Vapnik (1963)

$$\underset{\vec{w}, b}{\text{argmin}} \left\{ \frac{1}{n} SHE + \lambda \|\vec{w}\|^2 \right\}$$

SHE: minimize any hinge loss; $\|\vec{w}\|^2$: maximize margin

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 + priority

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

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

(λ: hyperparameter)

If λ is selected "reasonably", we can use numerical methods (e.g. gradient descent, etc.) to solve for $\vec{w}, b$ which minimize. We can use packages in R that do this.

What if $y \in \{0, 1, \ldots, L-1\}$, nominal var. with $L > 2$ 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: find the $\vec{x}_i$ closest and assigns that y_i? "Nearest Neighbor"!

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

$\mathcal{H} = \{\}$?
$A =$?
} Difficult to specify precisely what these look like, but they are lurking in the shadows.

We have a hyperparameter, fun. d. The default is to apply Euclidean:

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

Variant: instead of returning the closest x_i, find the K nearest x_i's and return the mode of the K y_i's. $\hat{y}_i$ of d