Math 390.4 Lec 6 2/20/18

Null model Alg.

$\mathcal{H} = \{0, 1\}$

most likely category $\in \{0,1\}$

$\vec{x}$'s don't matter in null model

Always keep

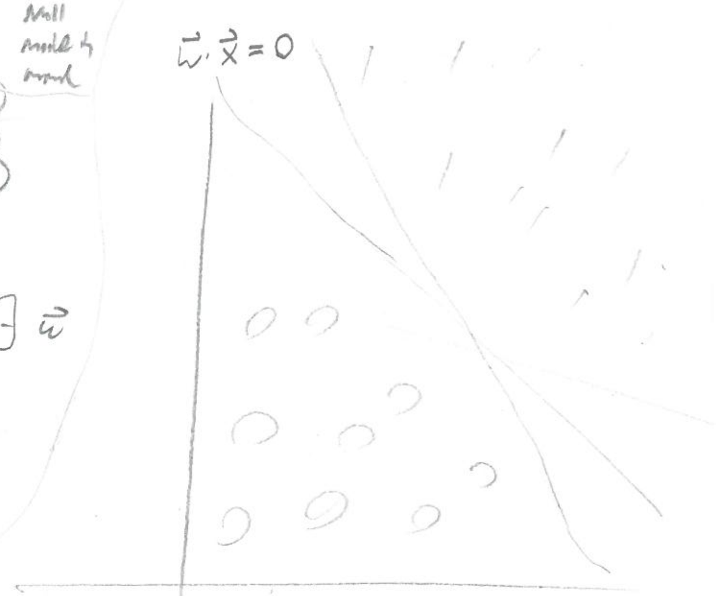
Null model is mod

Let's return to $g = \mathcal{A}(\mathbb{D}, \mathcal{H}) = \text{Mode}[\vec{y}]$

$$\mathcal{H} = \{ \mathbb{1}_{\vec{w} \cdot \vec{x} > 0} : \vec{w} \in \mathbb{R}^{p+1} \}$$

$\vec{x}$ includes a 1

Assume "linearly separable" i.e. $\exists \vec{w}$ s.t. there would be no errors if g were used on obs's in $\mathbb{D}$.

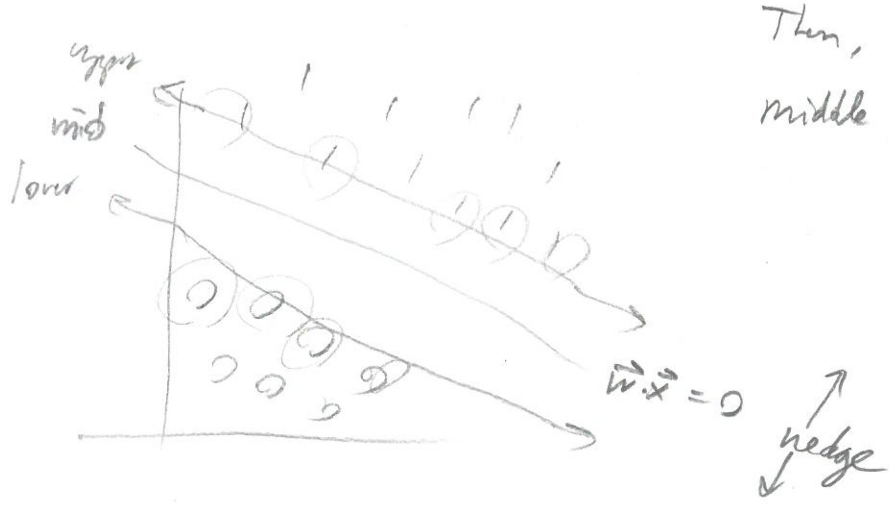


Which hyperplane (i.e. $\vec{w}$) is "best"?

Consider a New $\mathcal{A}$, different from perceptron learning algorithm.

Why not create a wedge, large as possible using parallel hyperplanes.

Then, g is built from the $\vec{w}$ in the middle of this wedge.



The "max-margin hyperplane" (proven to be optimal linear classifier in 1990)

Which data pts most matter? Since data pts $\vec{x}_i$ are "vectors", these are called the "support vectors". The middle line model is then called the "support vector machine"

weird name ... if I were naming this now ...

"support vector" $\Rightarrow$ "essential observation"

"machine" $\Rightarrow$ "model" + "separation"

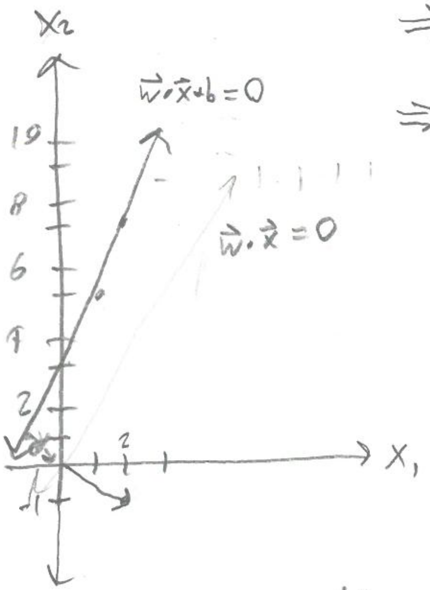
How to fit this?

Unfortunately, it is convenient to revert back to the ugly slope-intercept form,

$$\mathcal{H} = \left\{ \mathbb{1}_{\vec{w}\cdot\vec{x}+b>0} : \vec{w}\in\mathbb{R}^p,\ b\in\mathbb{R} \right\}$$

this is equivalent to before, it's just a slight reparameterization...

Let's first review 8th grade math.. Imagine the line $x_2 = 2x_1 + 3$



$$\Rightarrow 2x_1 - x_2 + 3 = 0$$

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

where is $\vec{w}$ on this graph? It's the "normal vector" i.e. perpendicular to the line

If $p > 2$, $\vec{w}\cdot\vec{x} = 0$ is a plane / hyperplane and $\vec{w}$ is $\perp$ to it.

Note: $\|\vec{w}\|$ indicates length of the vector $:= \sqrt{\sum_{j=1}^{p} w_j^2}$

And the normalized $\vec{w}$ vector is defined as the vector in same direction with length 1.

$$\vec{w}_0 := \frac{\vec{w}}{\|\vec{w}\|} \qquad \text{Proof: } \|\vec{w}_0\| = \sqrt{\sum_{j=1}^{p}\left(\frac{w_j}{\|\vec{w}\|}\right)^2} = \sqrt{\frac{1}{\|\vec{w}\|^2}\sum_{j=1}^{p} w_j^2} = \sqrt{\frac{1}{\|\vec{w}\|^2}\|\vec{w}\|^2} = 1$$

What is the length of the line $\vec{l} = \alpha\vec{w}_0$ where $\alpha\in\mathbb{R}$, constant?

$$\|\vec{x}\| = \sqrt{\sum_{j=1}^{p}(\alpha w_{0,j})^2} = \sqrt{\alpha^2\sum_{j=1}^{p} w_{0_j}^2} = \sqrt{\alpha^2\|\vec{w}_0\|^2} = |\alpha| \quad \text{Makes sense... } \alpha\cdot 1 = 1!$$

let $\vec{l}$ be the vector from the origin to the line $\vec{w}\cdot\vec{x}+b=0$, perpendicular to it

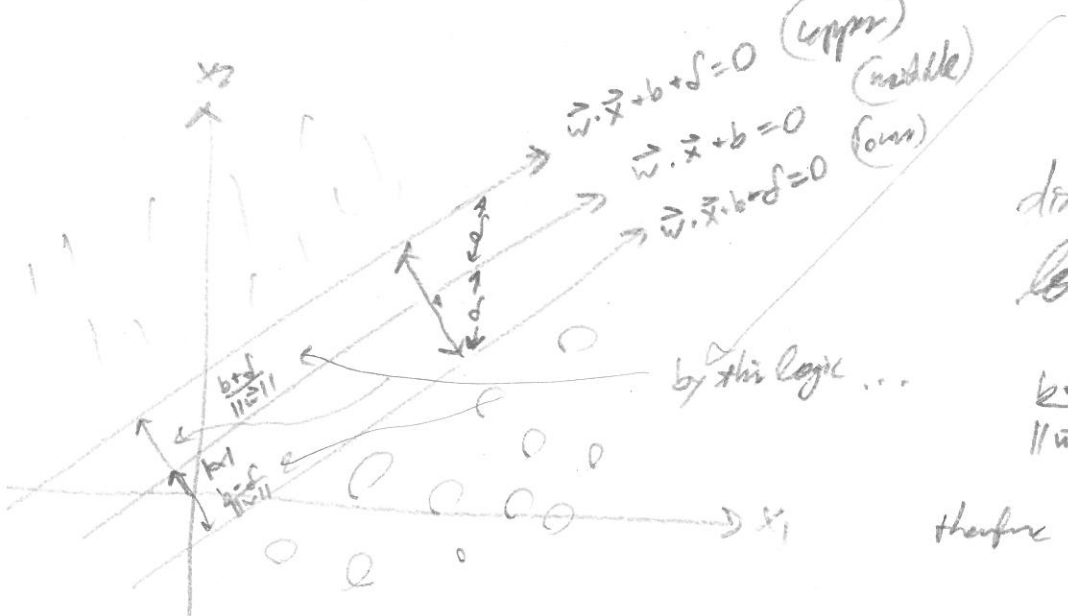
Let's solve for α.. $\vec{l}$ is on the line $\vec{w}\cdot\vec{x}+b=0$

$$\Rightarrow \vec{w} \cdot \vec{l} + b = 0 \Rightarrow \vec{w} \cdot \alpha \vec{w}_0 + b = 0$$

$$\Rightarrow \vec{w} \cdot \alpha \frac{\vec{w}}{\|\vec{w}\|} + b = 0 \Rightarrow \alpha \frac{\|\vec{w}\|^{\cancel{2}}}{\cancel{\|\vec{w}\|}} + b = 0 \Rightarrow \alpha = -\frac{b}{\|\vec{w}\|}$$

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

Now back to finding that "best" line:



distance between upper & lower lines is

$$\frac{b+\delta}{\|\vec{w}\|} - \frac{b-\delta}{\|\vec{w}\|} = \frac{2\delta}{\|\vec{w}\|}$$

therefore

Since $c(\vec{w}\,\vec{x} + b) = 0$, there are infinite solutions since $c \in \mathbb{R}$

Coerce $\delta = 1$ ... now there's a unique solution to the equation

$$\vec{w} \cdot \vec{x} + b + \delta = 0 \Rightarrow \vec{w} \cdot \vec{x} + b + 1 = 0$$

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

Constrain all $y=1$'s to be $\geq$ upper; constrain all $y=0$'s to be $\leq$ lower

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

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

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

Now $y_i = 1$

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

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

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

$(y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i + b) \geq (y_i - \frac{1}{2})$ $\quad$ Now $y_i = 0 \Rightarrow \underbrace{(y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i + b) \geq -\frac{1}{2}}$

Same condition for $y \in \{0, 1\} \Rightarrow \forall i$

Condition of perfect separability. If the pt on the line, $= -\frac{1}{2}$. If not, $> -\frac{1}{2}$.

Then we solve the following optimization problem:

Maximize $\frac{2}{\|\vec{w}\|} \Rightarrow$ Minimize $\|\vec{w}\|$ over $\vec{w} \in \mathbb{R}^p, b \in \mathbb{R}$ subj. to. $\forall i \; (y_i - \frac{1}{2})(\vec{w} \cdot \vec{x}_i + b) \geq -\frac{1}{2}$

the $\{\vec{w}, b\}$ solution is the support vector machine

This approach assumes perfect linear separability. In the real world... who has this luxury? We need to "upgrade" $\mathcal{A}$.

We need a loss function. Previously we employed $SAE = \sum_{i=1}^{n} \mathbb{1}_{\hat{y}_i \neq y_i}$ and then allowed the computer to find $\vec{w}$. This is okay but we can do better. We can make the loss depend on how bad the error is. Consider the following for the ith obs:

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

termed "hinge loss"

Let's see why this makes sense...

Imagine the pt is correctly classified and thus it respects the inequality. Consider it is above the inequality by $d \geq 0$.

$$\Rightarrow \left(y_i - \tfrac{1}{2}\right)(\vec{w} \cdot \vec{x}_i + b) = -\tfrac{1}{2} + d \underbrace{\geq}_{\text{correct!}} -\tfrac{1}{2}$$

$$\Rightarrow H_i = \max\left\{0, -\tfrac{1}{2} - \left(-\tfrac{1}{2} + d\right)\right\} = \max\{0, -d\} = 0$$

makes sense .. if it is correctly classified then there should be zero error

Imagine the pt. is incorrectly classified and hence does <u>not</u> respect the inequality. Let's say its below by $d > 0$.

$$\Rightarrow \left(y_i - \tfrac{1}{2}\right)(\vec{w} \cdot \vec{x}_i + b) = -\tfrac{1}{2} - d \underbrace{<}_{\text{incorrect!}} -\tfrac{1}{2}$$

$$\Rightarrow H_i = \max\left\{0, -\tfrac{1}{2} - \left(-\tfrac{1}{2} - d\right)\right\} = \max\{0, d\} = d > 0$$

makes sense .. you make a mistake, the worse it is, the more you pay.

minimize:

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

One more wrinkle... Recall we were trying to maximize the margin $|\alpha| = \frac{1}{\|\vec{w}\|}$ i.e. minimize $\|w\|$. You can't minimize two things! You need <u>one</u> objective function / loss function. Here is Vapnik's (1963) idea: Minimize:

$$\underbrace{\frac{1}{n} \sum \max\left\{0, -\tfrac{1}{2} - \left(y_i - \tfrac{1}{2}\right)(\vec{w} \cdot \vec{x}_i + b)\right\}}_{\text{min. avg. hinge loss}} + \underbrace{\lambda \|\vec{w}\|^2}_{\text{max. margin}}$$

tradeoff between these two goals.

What are the params? $\vec{w} \in \mathbb{R}^p$, $b \in \mathbb{R}$. Still $p+1$ parameters.

What is λ? A predefined constant called a "hyperparameter."

$g = \mathcal{A}(\mathbb{D}, \mathcal{H}, \lambda_1, \lambda_2, \ldots)$

It is considered a tuning knob on the $\mathcal{A}$ itself. It is a meta idea. Recall $g(\vec{x}) = \mathbb{1}_{\vec{w} \cdot \vec{x} + b}$ there is no λ here!

Perceptron's max iter could be a λ.

λ only affects which $g \in \mathcal{H}$ will be selected.

We will discuss how the value of hyperparameters are selected later in the course. For now, what does λ do?

- If $\lambda \approx 0$, we only care about errors and not a max. margin. One error far away can ruin our nice separation line.
- If $\lambda \approx \infty$, we only care about the best line of separation and not about errors... this makes no sense! I can just make $\|\vec{w}\| \approx 0$

Considering λ is picked "reasonably". How do we solve for $\{\vec{w}, b\}$ our params? Use modern numerical optimization methods:

- quadratic programming
- sub-gradient descent
- coordinate descent

which we will likely not study. Lucky for us, R packages already implemented this for us.