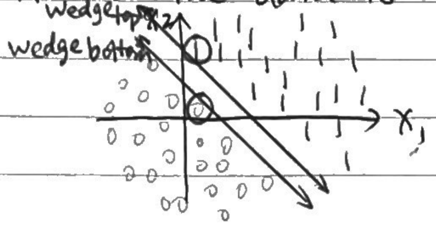
lec #4

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

Assume the data is linearly separable so it looks like:



we need an algorithm that locates the middle of that wadge.

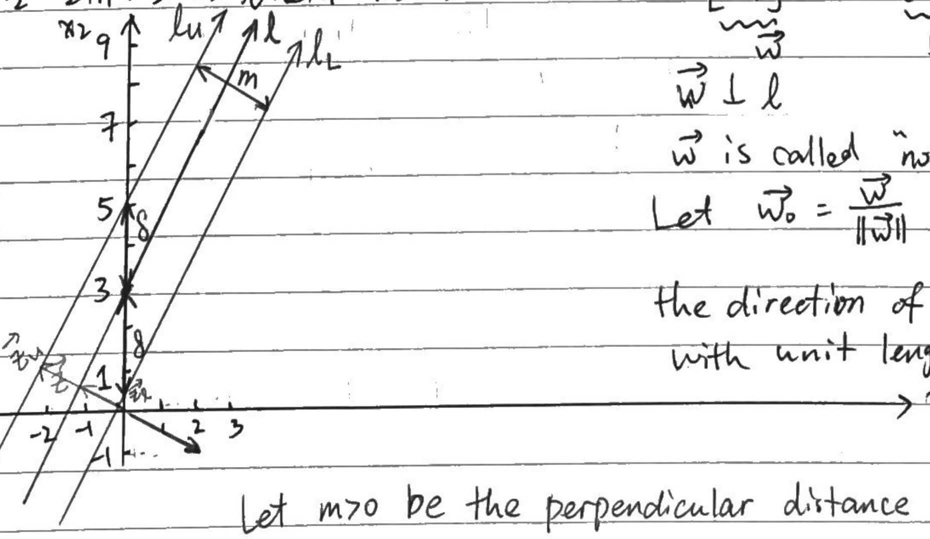
Let the top of the wedge be the linearly separable model "closest" to the $y=1$'s and the bottom of the wedge be the linearly separable model "closest" to the $y=0$'s.
The "max margin hyperplan" is the parallel line in the center of the top of bottom.
Note: there are two critical observations (the circled points). Since observations are x-vectors, these critical observations are called "support vectors" and hence the final model is called a "support vector machine" (SVM). "Machine" is a fancy word meaning "complex model". So "machine learning" just means "learning complex models". To find SVM...

first write: $\mathcal{H} = \{ \mathbb{1}_{\vec{w} \cdot \vec{x} - b \geq 0} : \vec{w} \in \mathbb{R}^3, b \in \mathbb{R} \}$.

Note: $\vec{w} \cdot \vec{x} - b = 0$ defines a line / hyper plane.

Hesse Normal Form

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



$\vec{w} \perp l$

$\vec{w}$ is called "normal vector"

Let $\vec{w}_0 = \frac{\vec{w}}{\|\vec{w}\|}$

the direction of the w vector with unit length.

$\vec{z} = \alpha \vec{w}_0$, $\vec{z} \in l$

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

$\vec{w} \cdot (\alpha \vec{w}_0) - b = 0$

$\frac{\alpha}{\|\vec{w}\|} \|w\|^2 - b = 0$

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

$\vec{z} = \frac{b}{\|\vec{w}\|} \vec{w}_0$

Let $m > 0$ be the perpendicular distance between l_u and l_L and let $\delta > 0$ be the distance between l_u and l (and l_L and L) on the x_2 axis.

$\vec{l_U}: \vec{w}\cdot\vec{x} - (b+\delta) = 0, \quad \vec{z_U} = \frac{b+\delta}{\|\vec{w}\|}\vec{w_0}$

$\vec{l_L}: \vec{w}\cdot\vec{x} - (b-\delta) = 0, \quad \vec{z_L} = \frac{b-\delta}{\|\vec{w}\|}\vec{w_0}$

$$m = \|\vec{z_U} - \vec{z_L}\| = \left\|\frac{b+\delta}{\|\vec{w}\|}\vec{w_0} - \frac{b-\delta}{\|\vec{w}\|}\vec{w_0}\right\|$$

Goal is to make m as large as possible (maximum margin) $\quad = \frac{1}{\|\vec{w}\|} 2\delta \underbrace{\|\vec{w_0}\|}_{\text{unit length}=1} = \frac{2\delta}{\|\vec{w}\|}$

$\Leftrightarrow$ making the w vector as small as possible.

The Hesse Normal form is not unique. There are infinite equivalent specification of a line.

$\forall c \neq 0 \quad c(\vec{w}\cdot\vec{x} - b) = 0 \quad$ Let $c = \frac{1}{\delta} \Rightarrow m = \frac{2}{\|\vec{w}\|}$

Now we need two conditions

1) All $y=1$'s are above or equal to l_U:

$\forall i$ s.t. $y_i = 1$

$$\vec{w}\cdot\vec{x_i} - (b+1) \geq 0$$
$$\vec{w}\cdot\vec{x_i} - b \geq 1$$
$$\tfrac{1}{2}(\vec{w}\cdot\vec{x_i} - b) \geq \tfrac{1}{2}$$
$$(y_i - \tfrac{1}{2})(\vec{w}\cdot\vec{x_i} - b) \geq \tfrac{1}{2}$$

2) All $y=0$'s are below or equal to l_L:

$\forall i$ s.t. $y_i = 0$

$$\vec{w}\cdot\vec{x_i} - (b-1) \leq 0$$
$$\vec{w}\cdot\vec{x_i} - b \leq -1$$
$$\tfrac{1}{2}(\vec{w}\cdot\vec{x_i} - b) \leq -\tfrac{1}{2}$$
$$-\tfrac{1}{2}(\vec{w}\cdot\vec{x_i} - b) \geq \tfrac{1}{2}$$
$$(y_i - \tfrac{1}{2})(\vec{w}\cdot\vec{x_i} - b) \geq \tfrac{1}{2}$$

Note how both inequalities are the same for both (1) and (2). Thus this inequality satisfies both constraints. So all observations will be in their right places.

$\boxed{\forall i \quad (y_i - \tfrac{1}{2})(\vec{w}\cdot\vec{x_i} - b) \geq \tfrac{1}{2}} \Rightarrow$ line is linearly seperable.

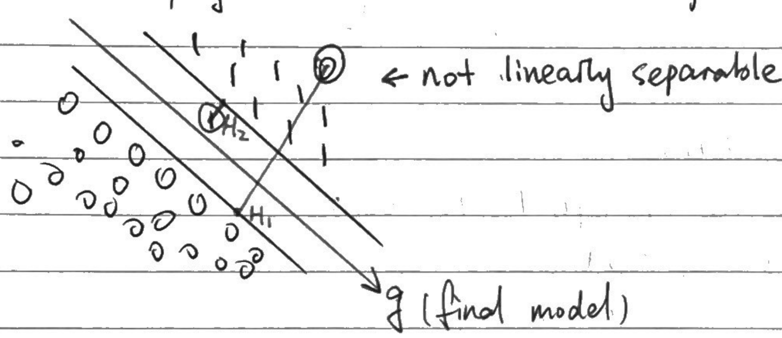
You compute the SVM by optimizing the following problem:

$\min \|\vec{w}\|$ s.t. $\sim$ is true. and reture the resulting $\vec{w}$ and b.

There is no analytical Solution. You need optimization algorithms.
It can be solved with quadratic programming and other procedures as well.
Note: everything we did above generalizes to $p > 2$. Note: most textbooks have 1's in the place of our ½'s that's because they assumed $y = \{-1, 1\}$, but we assumed binary.

What if the data is not linearly separable?
You can never satisfy that constraint... So this whole thing doesn't work. We will use a new objective function / loss function / error-tallying function called "hinge loss", H:

$$H_i := \max\{0, \tfrac{1}{2} - \underbrace{(y_i - \tfrac{1}{2})(\vec{w} \cdot \vec{x}_i - b)}_{\text{should be } \geq \frac{1}{2}}\}$$



Let's say a point is d away from where it should be.

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

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

With this last function, it's clear we wish to minimize the sum of hinge error, $SHE := \sum_{i=1}^{n} \max\{0, \tfrac{1}{2} - (y_i - \tfrac{1}{2})(\vec{w} \cdot \vec{x}_i - b)\}$.

But we also want to maximize the margin. So we combine both ~~con~~ Considerations together into the objective function of Vapnik (1963):

$$\underset{\vec{w}, b}{\operatorname{argmin}} \left\{ \underbrace{\tfrac{1}{n} SHE}_{\text{minimizing distance errors}} + \lambda \underbrace{\|\vec{w}\|^2}_{\text{maximizing the width of the wedge.}} \right\}$$

Once λ is set, the computer can do the optimization to find the resulting SVM. even using out of the box R packages.

What is λ? It's a positive "hyperparameter", "tuning parameter". It's set by you! It controls the tradeoff between these two considerations.

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

What if you have the modeling setting where $\mathcal{Y} = \{1, 2, \dots, L\}$, a nomial categorical response with $L > 2$ levels. The model will still be a "classification model" but not a "binary classification model" and it's sometimes called a "multinomial classification model". What is null model g_0? Again, $g_0 = \text{sampleMode}[y]$.

Consider a model that perdicts on a new x_* by looking through the training data and finding the "closest" $\vec{x}_i$ and returning it's y_i as the predicted response value. This is called a "nearest neighbor" model. Further, you may also want to find the k closest observations and return the mode of these k observations (randomize ties) as the predicted response value. That's called "k nearest neighbors" (KNN) model where k is a natural number hyperparameter. There is another hyperparameter that must be specified, the "distance function" $d: \mathcal{X}^2 \to \mathbb{R}_{\geq 0}$. The typical distance function is Euclidean distance squared:

$$d(\vec{x}_*, \vec{x}_i) := \Sigma (x_{i,j} - x_{*,j})^2$$

What is $\mathcal{H}$? $\mathcal{A}$?