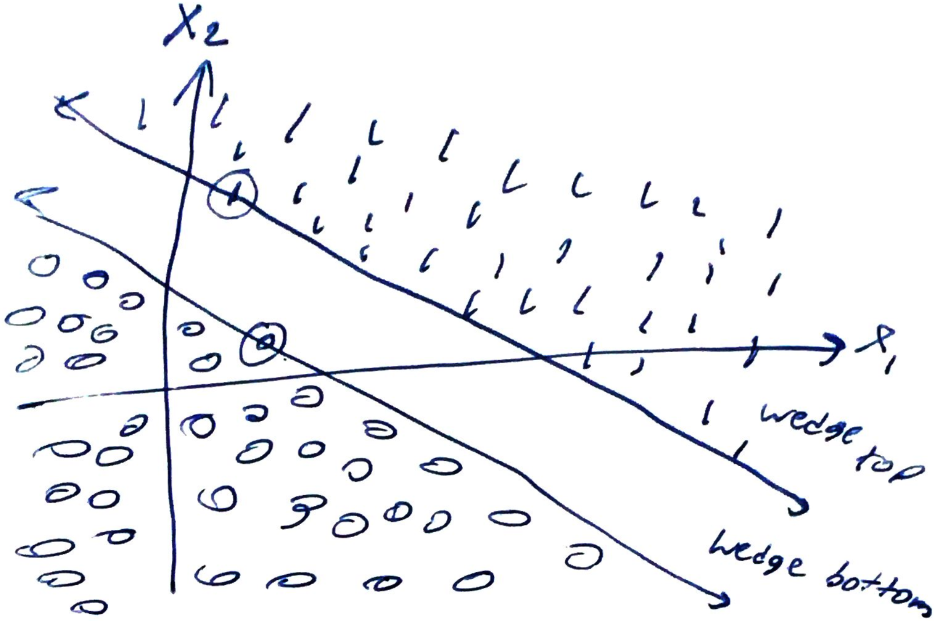
Lecture 05

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

Assume the data is linearly separable so it looks like:



We need an algorithm that locates the middle of the wedge. 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 hyperplane" is the parallel line in the center of the top and bottom.

Note: there are two critical observations (the circled points). Since observations are X-factors, 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...

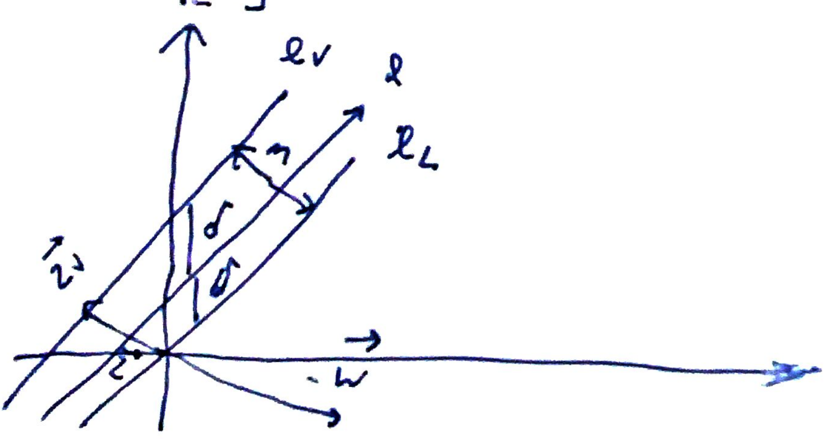
First rewrite

$$H = \{ \mathbb{1}_{\vec{w}\cdot\vec{x}-b \geq 0} : \vec{w} \in \mathbb{R}^p, b \in \mathbb{R} \}$$

Note $\underbrace{\vec{w}\cdot\vec{x}-b=0}_{\text{Hesse Normal form}}$ defines a line / hyperplane

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

$$\ell : \underbrace{\begin{bmatrix} 2 \\ -1 \end{bmatrix}}_{\vec{w}} \cdot \vec{x} - \underbrace{(-3)}_{b} = 0$$



The w vector is perpendicular to line l and called the "normal vector"

Let $\vec{w_0} := \frac{\vec{w}}{\|\vec{w}\|}$

the direction of the w vector with unit length

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-z axis.

$$\vec{z} = \alpha \vec{w_0}, \quad \vec{z} \in l$$

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

$$\Downarrow$$

$$\vec{w} \cdot (\alpha \vec{w_0}) - b = 0 \Rightarrow \frac{\alpha}{\|\vec{w}\|} \|\vec{w}\| - b = 0$$

$$\Rightarrow \alpha = \frac{b}{\|\vec{w}\|} \Rightarrow \vec{z} = \frac{b}{\|\vec{w}\|} \vec{w_0}$$

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

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

$$m = \|\vec{z}_U - \vec{z}_L\| = \left\| \frac{b+\delta}{\|\vec{w}\|} \vec{w_0} - \frac{b-\delta}{\|\vec{w}\|} \vec{w_0} \right\|$$

$$= \frac{1}{\|\vec{w}\|} 2\delta \|\vec{w_0}\| = \frac{2\delta}{\|\vec{w}\|}$$

Goal is to make M as large as possible (maximum margin) $\iff$ making the w vector as small as possible

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

$$\forall c \neq 0 \quad c(\vec{w}\cdot\vec{x} - b) = 0 \quad \text{Let } c = \frac{1}{d}$$

$$\Downarrow$$

$$M = \frac{2}{\|\vec{w}\|}$$

Now we need two conditions

(I) All $y=1$'s are above or equal to l_+:

$\forall_i$ s.t. $y_i = 1$

$$\vec{w}\cdot\vec{x}_i - (b+1) \geq 0 \Rightarrow \vec{w}\cdot\vec{x}_i - b \geq 1$$

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

$$\Downarrow$$

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

(II) All $y=0$'s are bellow or equal to l_-:

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

$$\Rightarrow \vec{w}\cdot\vec{x}_i - b \leq -1$$

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

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

Note how both inequalities are the same for both I and II. Thus this inequality satisfies both constraints, so all observations will be in their right place

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

=> line is linearly separable

You compute the svm by optimizing the following problem:

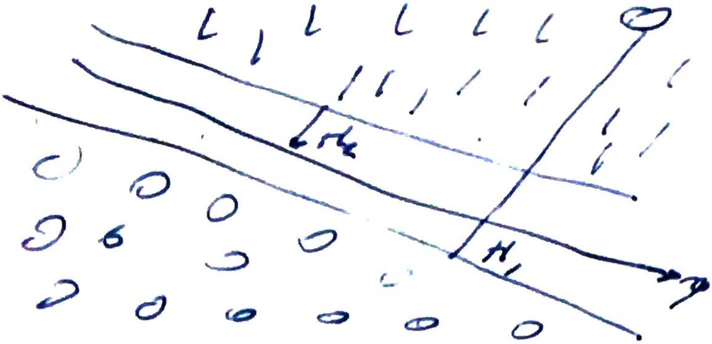
$\min \|\vec{w}\|$ s.t

true

and return the resulting w vector 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 $p \geq 2$.
Note: most textbooks have 1's in the place of our 1/2'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 constraints ... So this whole thing doesn't work, we will use a new objective functions / loss function / error-tallying function called "hinge loss", H:



$$H_1 := \max\left\{0, \frac{1}{2} - \underbrace{(y_1 - \tfrac{1}{2})(\vec{w}\cdot\vec{x}_1 - b)}_{\text{should be} \geq \frac{1}{2}}\right\}$$

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

$$(y_1 - \frac{1}{2})(\vec{w}\cdot\vec{x}_1 - b) = \frac{1}{2} - d$$

With this loss function, it is clear we wish to minimize the sum of the hinge errors:

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

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

$$\underset{\vec{w},\, b}{\operatorname{argmin}} \left\{ \underbrace{\frac{1}{n} SHE}_{\text{minimizing distance errors}} + \lambda \|\vec{w}\|^2 \right\}$$

maximize the width of the wedge

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. Is it a possible "hyperparameter", "tuning parameter". It is set by you! It controls the tradeoff between these two considerations

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

What if you have the modelling setting where $y = \{1, 2, \ldots, L\}$, a nominal categorical response with $L \geq 2$ 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 the null model g_0? Again g_0 = SampleMode[y].

Consider a model that predicts on a new x_* by looking through the training data and finding the "closest" x_i vector and returning its 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 as the predicted response value (randomize ties). 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.

$$d(\vec{x}_*, \vec{x}_i) := \sum_{j=1}^{p} (x_{i,j} - x_{*,j})^2$$

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