ECE421 - Week 1 - Part 3 - linear classification

- It's time to rigorously define a Simple Learning Model
- Consider the example of credit approval
  - Bank needs to determine whether to approve credit to a customer or not (Yes/No)
  - Input: $\underline{x} = (\text{age, salary, years of experience, debt}) \in \mathcal{X}$

    Output (label): $y \in \{+1, -1\} = \mathcal{Y}$

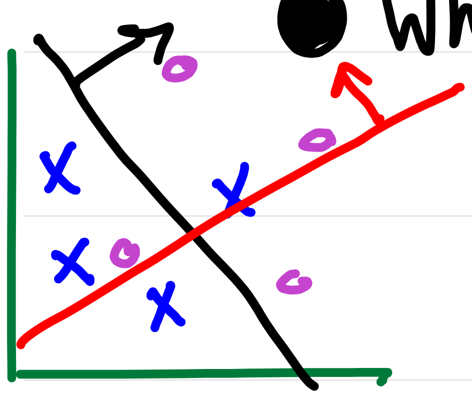
    Let $\mathcal{X}$ denote the input space (i.e. the set of all possible $\underline{x}$)

    Let $\mathcal{Y}$ denote the output space (in this example $\mathcal{Y} = \{+1, -1\}$)
  - Unknown Target function: $f: \mathcal{X} \rightarrow \mathcal{Y}$, maps each input to an output.

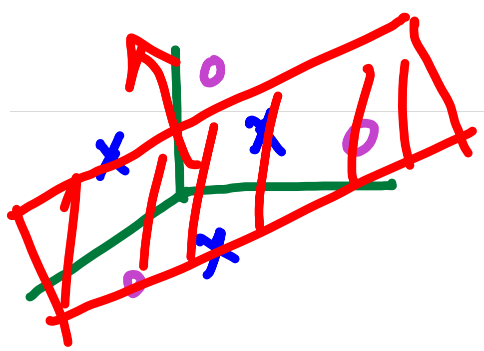
Note: a bar under a parameter indicates that it is a vector.

- **Historical data set:** $\mathcal{D} = \{(\underline{x}_1, y_1), \underbrace{(\underline{x}_2, y_2)}_{\text{2nd input-output pair, 2nd datapoint}}, \ldots, (\underline{x}_N, y_N)\}$

- **Goal:** to design a learning algorithm that uses $\mathcal{D}$ to pick a mapping $g: \mathcal{X} \to \mathcal{Y}$ that approximates f
  - The algorithm chooses g from a set of candidate mapping $\mathcal{H}$ ← hypothesis set



  - What is $\mathcal{H}$ for linear classification problem?
    - Naively speaking, for linear classification $\mathcal{H}$ would be the set of all possible hyperplanes that partition the input space



- We can describe $\mathcal{H}$ through a functional form that is shared among all $h \in \mathcal{H}$

  - In linear classification $h \in \mathcal{H}$ can be described as

$$h(\underline{x}) = \text{sign}\left(\sum_{i=1}^{d} w_i x_i + b\right), \quad \text{where} \quad \text{sign}(z) = \begin{cases} +1 & z > 0 \\ -1 & z < 0 \end{cases}$$

$\underline{w} = (w_1, w_2, \dots, w_d) \in \mathbb{R}^d$ ← weight vector

$b \in \mathbb{R}$ ← bias

$$\sum_{i=1}^{d} w_i x_i \overset{h(\underline{x})=+1}{\underset{h(\underline{x})=-1}{\gtrless}} -b$$

← Threshold

- Training: In linear classification, the goal is to find "good" $g \in \mathcal{H}$ (i.e., a "good" $\underline{w}$ and b), given the data set.

- But "good" to do what?
  - with a good model, we should little difference between the prediction and the true labels.
  - In fact, we want the $\underline{w}$ and b that give us the "minimum" possible error

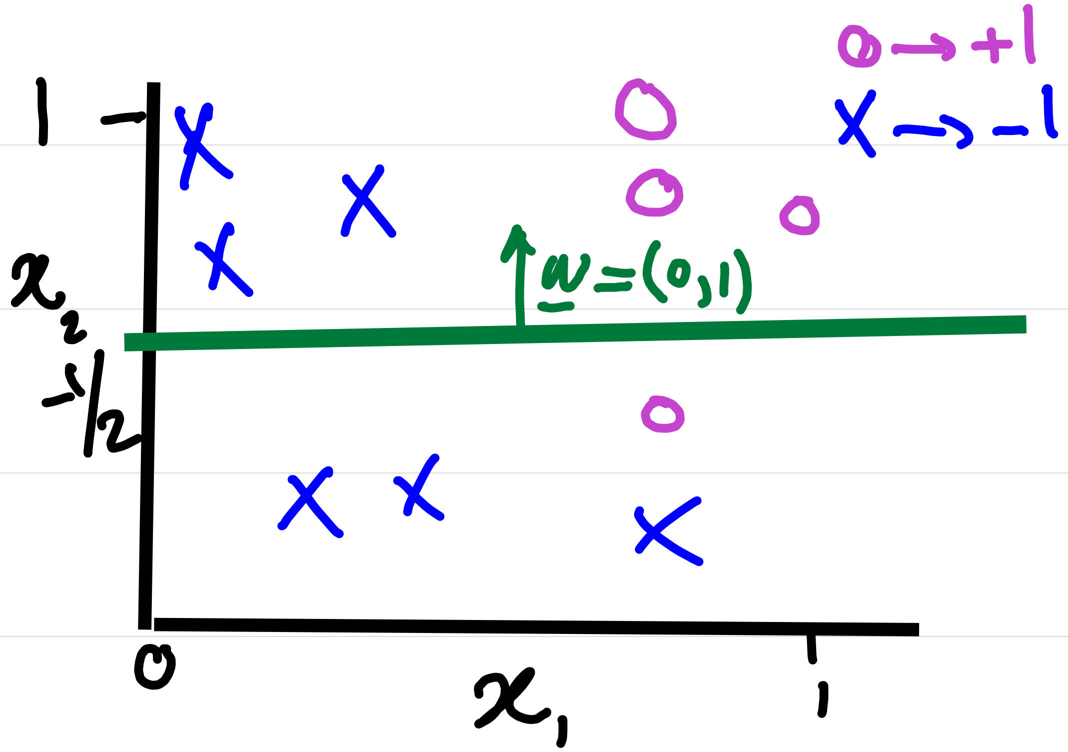
$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\left(f(\underline{x}_n) \neq h(\underline{x})\right) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\left(y_n \neq \text{sign}\left(\sum_{i=1}^{d} w_i x_{ni} + b\right)\right)$$

$$\underline{x}_n = (x_{n1}, x_{n2}, \dots, x_{nd})$$

indicitator function. $\mathbb{1}$(statement) is equal to 1 if the "statemente" is true, otherwise it returns 0.

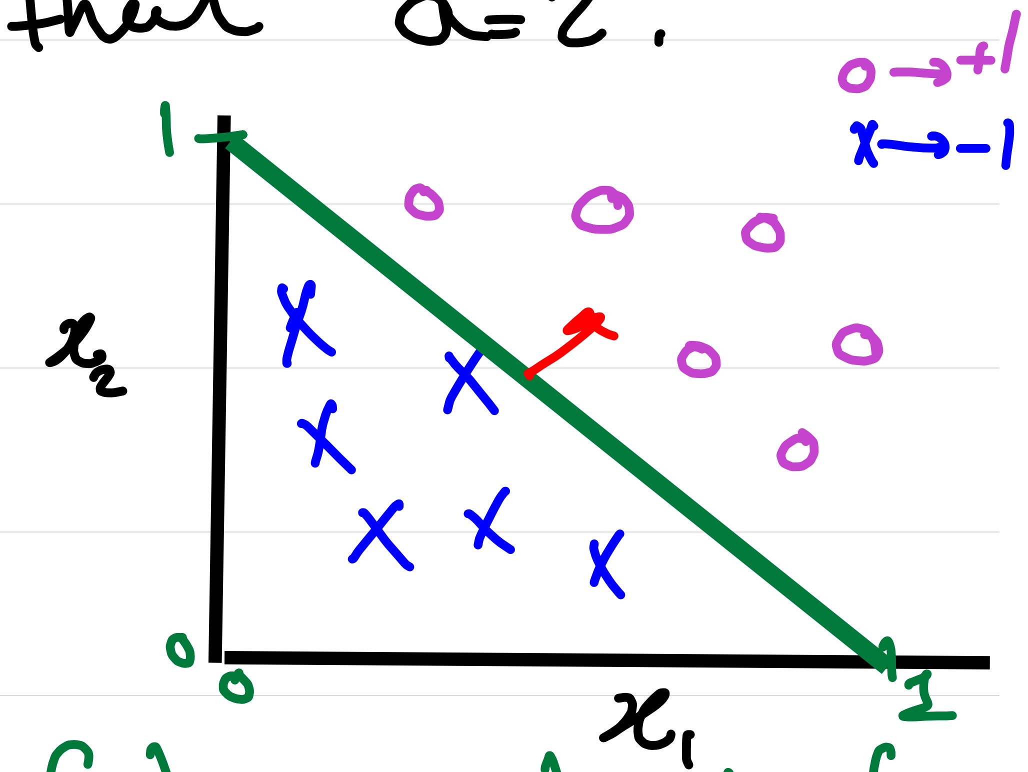
- **Prediction:** Given new customer $\underline{x}$, use $\sum_{i=1}^{d} w_i x_i \underset{\hat{y}=-1}{\overset{\hat{y}=+1}{\gtrless}} -b$ to determine $\hat{y}$.

- Assume for the sake of illustration that $d=2$.



$\underline{w} = (1,1)$

$b = -1$



Draw the decision boundary for $\underline{w}=(0,1)$ and $b=-1/2$. Use arrow to show the positive prediction side.

Find $\underline{w}$ and b for the decision boundary above. Note the direction of arrow indicating +1 prediction side.

Basic Setup of Learning Problem of Supervised Learning

Input: Data points : $\underline{x} = (x_1, \dots, x_d) \in \mathcal{X}$

e.g., customer $\underline{x} \in \mathbb{R}^4$

Output: Label $y \in \mathcal{Y}$

Classification: if the label has discrete values

Regression: if the label is Continuous

Unknow Mapping: Target function $f : \mathcal{X} \longrightarrow \mathcal{Y}$

$$y = f(\underline{x})$$

Learning Task. Given training data

$$\mathcal{D} = \{(\underline{x}_1, y_1), (\underline{x}_2, y_2), \ldots, (\underline{x}_N, y_N)\}$$

produce a function $g: \mathcal{X} \to \mathcal{Y}$ to make predictions on new inputs (i.e., $\hat{y} = g(\underline{x})$)

- How do we do this? We have to assume a model

Learning Model: Hypothesis Set: $\mathcal{H} = \{h_1, h_2, \ldots, h_m\}$

each being a candidate function ← $h_i : \mathbb{R}^d \to \mathbb{R}$, $\quad y = h_i(\underline{x})$

e.g.: $\underline{x} \xrightarrow{\text{sign}(\underline{w}^T \underline{x} + b)} \pm 1$

Learning Algorithm: Select $g \in \mathcal{H}$ using the training set

Summary.

Unknown target
$y = f(\underline{x})$

$\downarrow$

training Examples
$\{(\underline{x}_1, y_1), \dots, (\underline{x}_N, y_N)\}$

$\downarrow$

Hypothesis Set $\mathcal{H}$ $\longrightarrow$ Learning Algorithm $\mathcal{A}$ $\longrightarrow$ Find hypothesis $g \approx f$

Prediction: $y = g(\underline{x})$ "testing"

Basic setup of Learning Problem of Binary Linear Classification

- **Training Set:** $\mathcal{D} = \{(\underline{x}_1, y_1), \cdots, (\underline{x}_N, y_N)\}$

  $\underline{x}_n \in \mathcal{X}$, $\underline{x}_n = (x_{n1}, x_{n2}, \cdots, x_{nd})$, $y_n \in \{-1, +1\} = \mathcal{Y}$

- **Task:** Given any $\underline{x} \in \mathcal{X}$, output $y \in \{-1, +1\} = \mathcal{Y}$

- **Hypothesis (Decision Rule):**

weight vector: $\underline{w} = (w_1, \cdots, w_d) \in \mathbb{R}^d$

bias: $b \in \mathbb{R}$

Given any data point $\underline{x} = (x_1, \cdots, x_d)$,

$$\left.\begin{array}{ll} \text{if } \sum_{i=1}^{d} w_i x_i + b > 0, & \text{then } \hat{y} = +1 \\ \text{if } \sum_{i=1}^{d} w_i x_i + b < 0, & \text{then } \hat{y} = -1 \\ \text{if } \sum_{i=1}^{d} w_i x_i + b = 0, & \text{output either } +1 \text{ or } -1 \\ & \text{(Unimportant)} \end{array}\right\} \quad h(\underline{x}) = \text{Sign}\left(\sum_{i=1}^{d} w_i x_i + b\right)$$

■ Training: Compare decision rule with training data, to choose the "best" parameter values for decision rule – "best" hypothesis

Given $\mathcal{D}$ find $(\underline{w}, b)$ to minimize the training error: Average error on training set.

$$\underbrace{E_{in}}_{\text{in-sample error}}(\underline{w}, b) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\left(f(\underline{x}_n) \neq h(\underline{x}_n)\right) = \sum_{n=1}^{N} \mathbb{1}\left(y_n \neq \underbrace{\text{Sign}\left(\sum_{i=1}^{d} w_i x_{ni} + b\right)}_{\hat{y}_n}\right)$$

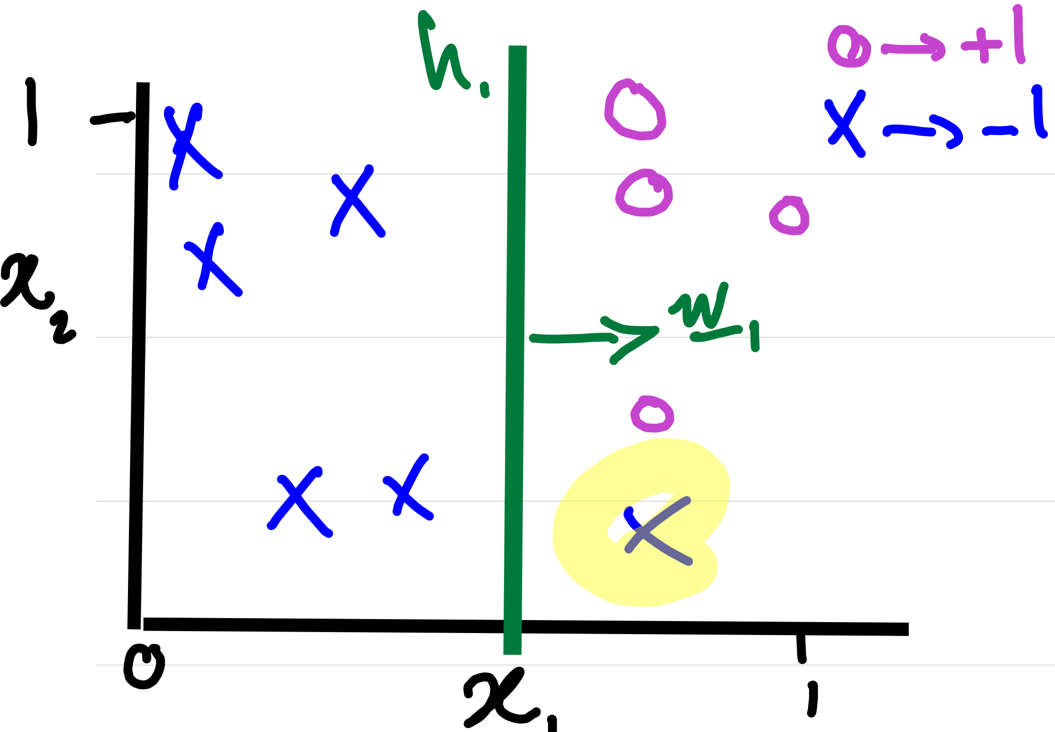
$E_{in}(h)$

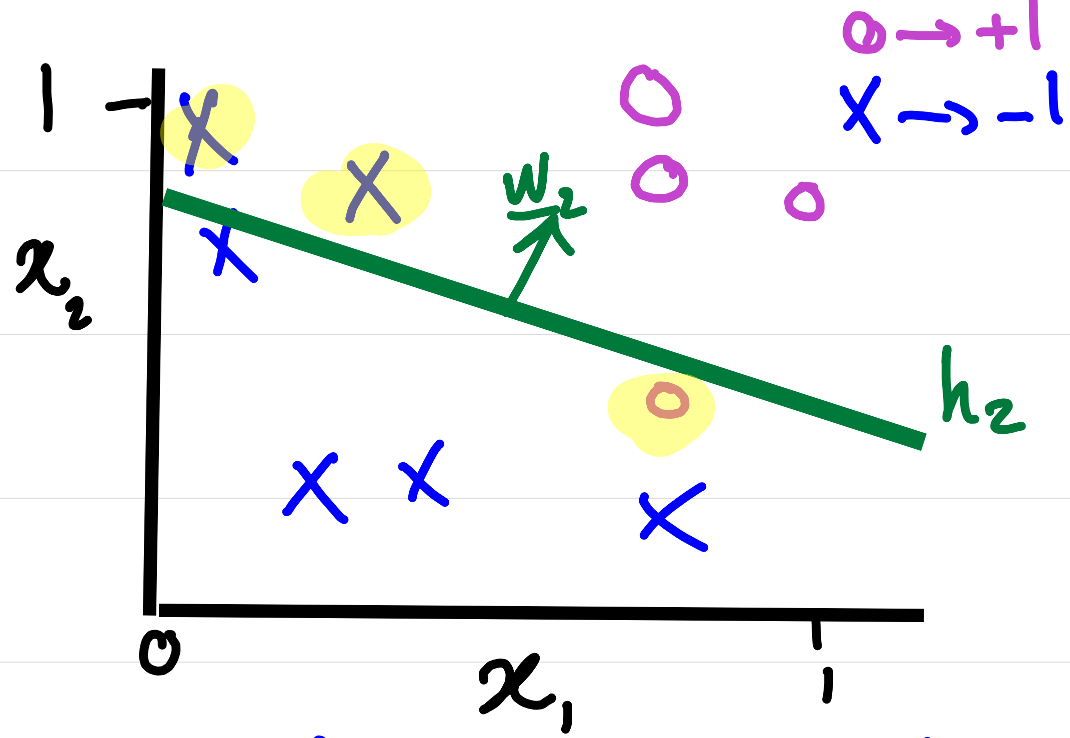
$\mathbb{1}(\cdot)$: Indicator function

y_n: true label for $\underline{x}_n$

$\hat{y}_n$: output of decision rule on example $\underline{x}_n$

x_{ni}: the i-th cordinate of the n-th input, i.e. $\underline{x}_n$





$E_{in}(h_1) = 1/10$

$E_{in}(h_2) = 3/10$

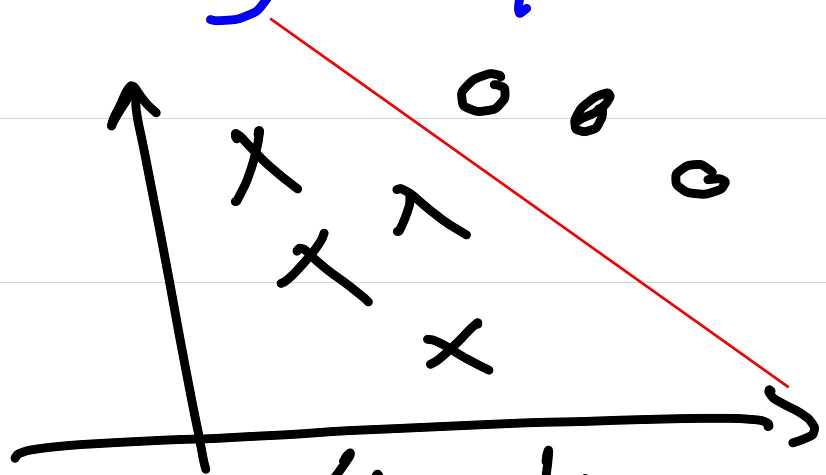
. . . .

How hard is it to find the best decision boundary, i.e. solving

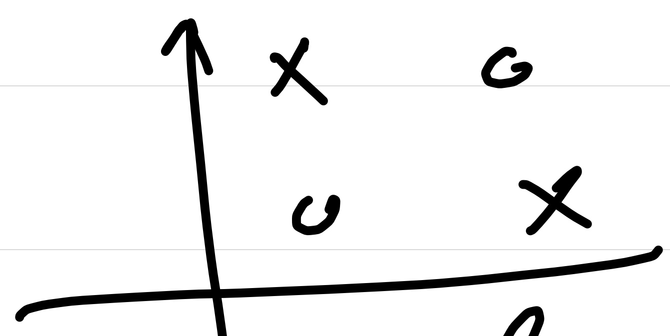
$$\min_{\underline{w}, b} \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}\left(y_n \neq \operatorname{sign}\left(\sum_{i=1}^{d} w_i x_{ni} + b\right)\right)$$

Bad news: finding the best linear classifier is NP-hard, in general

Good news: If D is linearly separable, we can find the solution efficiently.



linearly separable



not linearly separable