Machine Learning from Data

Lecture 2: Spring 2021

Today’s Lecture

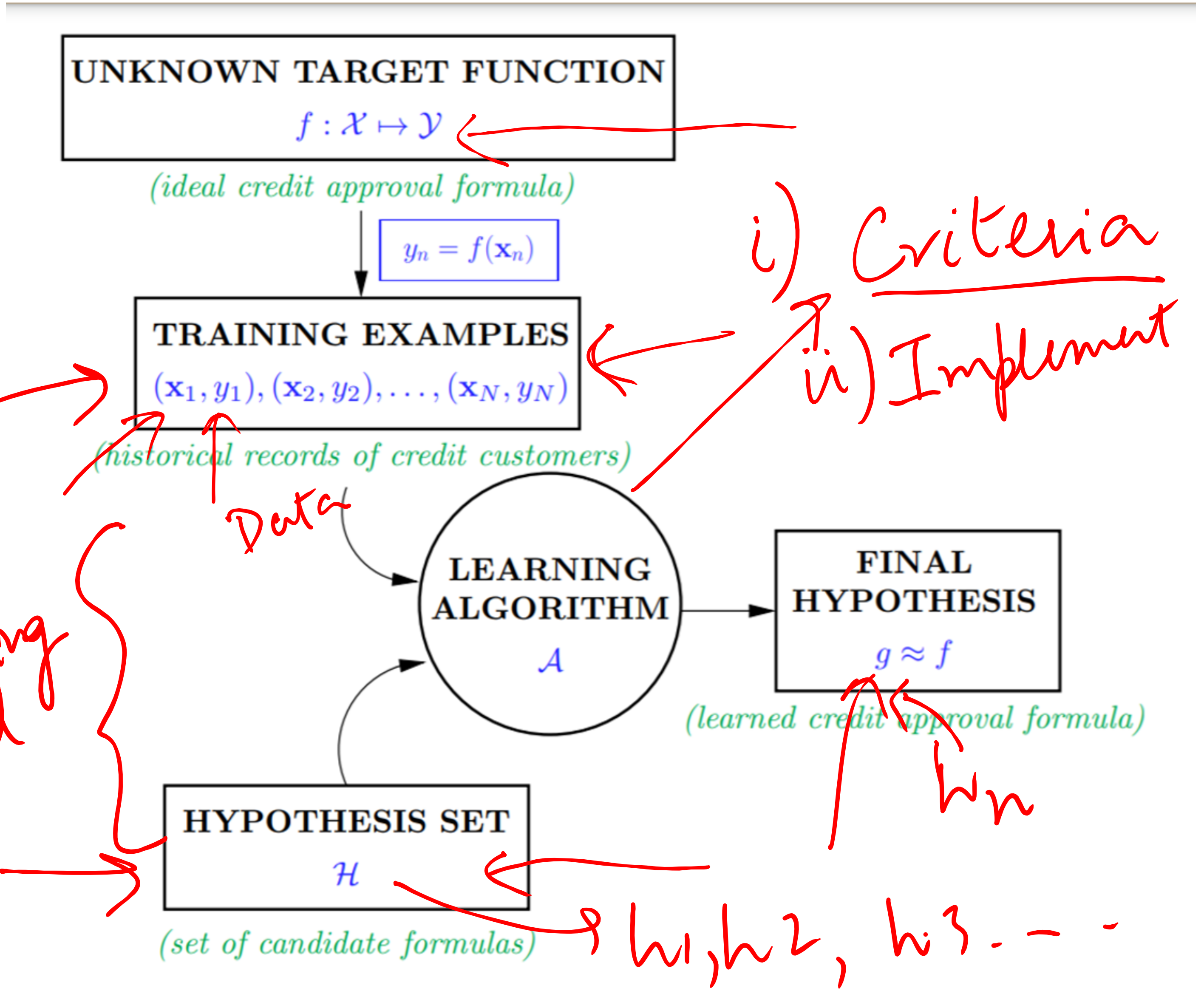
- The Perceptron
- Learning Set-Up
- The PLA
- Other Views of Learning

In the Previous Lecture

- Formalized Components of Learning
- Learning Process

1) Before we see the data

Learning Model



Formalize Components of Learning:

Credit Card

input $\mathbf{x} \in \mathbb{R}^d = \mathcal{X}$.

output $y \in \{-1, +1\} = \mathcal{Y}$.

target function $f : \mathcal{X} \mapsto \mathcal{Y}$.

(The target f is *unknown*.)

data set $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$.

$(y_n = f(\mathbf{x}_n)$.)

$x_1 \quad x_2 \quad x_3 \; \ldots$

- **Input**: Salary, debt, years, ...
  Output: Approve or not
  Target function: Relationship between X and Y
- Data on customers
- X, Y and D will be given by the learning problem.

$$\text{Input}(X) = \begin{array}{cccc} \text{Salary} & \text{debt} & \text{year, age} & \\ [x_1, & x_2, & x_3 & \cdots x_d] \end{array}$$

d-dimensional space

$$\text{Score} = \underset{w_1}{8}x_1 - \underset{w_2}{4}x_2 + \underset{w_3}{2}x_3 + 5x_4 + \cdots 0x_n + \cdots x_d$$

$$\text{Score} = [w_1 x_1 + w_2 x_2 + w_3 x_3 \cdots w_d x_d]$$

If Score > threshold → Approve

If Score < threshold → Decline

A Simple Learning Model: The Perceptron

- Let us consider the Credit Example: We as Data Scientists/ML Practitioners want to approve/decline an incoming application.
- Formalize the problem:

$x_1 w_1 + x_2 w_2 \dots$

Salary, debt

$$\text{Input} = [\underset{\text{Salary}}{x_1}, \underset{\text{debt}}{x_2}, \dots, x_d]^T$$

$$\text{Credit Score} = \sum_{i=1}^{d} w_i x_i$$

$$\begin{cases} \text{Approve credit if } \sum_{i=1}^{d} w_i x_i > \text{threshold} \\ \text{Decline credit if } \sum_{i=1}^{d} w_i x_i < \text{threshold} \end{cases}$$

Simple Learning Model

- Rule:

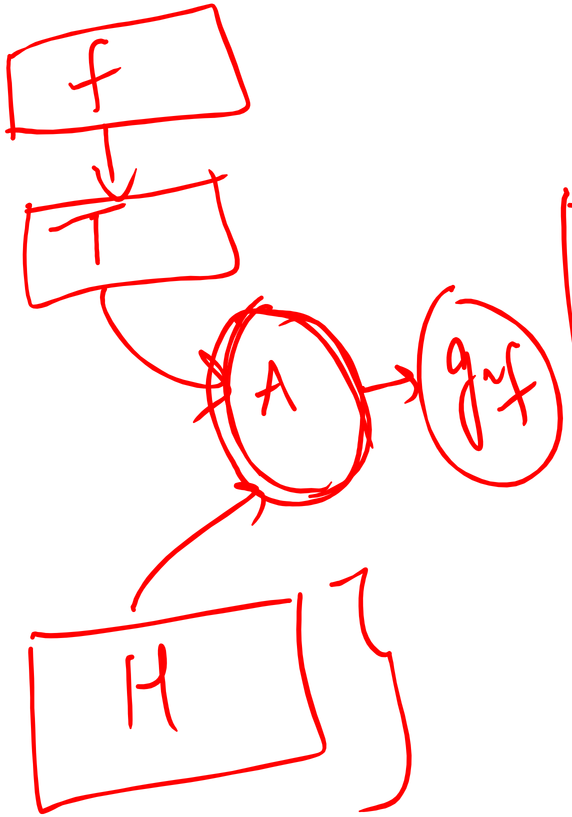
$$\text{sign}\left(\sum_{i=1}^{d} w_i x_i - \text{threshold}\right)$$

w_0

$x_0 = 1$

$$g(x) = \text{sign}\left(\sum_{i=1}^{d}\right)$$

$$\boxed{g(x) = \text{sign}\left(\sum_{i=0}^{d} w_i x_i\right)}$$



The Perceptron Hypothesis Set

- $x \in \mathbb{R}^d$

$$x = \begin{bmatrix} \cancel{1\, x_0} \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \in \underbrace{\{1\} \times \mathbb{R}^d}$$

$$g(x) = \text{sign}(w^T x)$$

$$w \to \begin{bmatrix} 6 \\ -3 \\ \vdots \end{bmatrix}$$

$$\underbrace{\mathcal{H} = \{ h(x) = \text{sign}(w^T x) \}}_{\text{Perceptron}}$$

$$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \in \mathbb{R}^{d+1}$$

$$\boxed{g \simeq f}$$

$\boxed{\text{PLA}}$ w_*

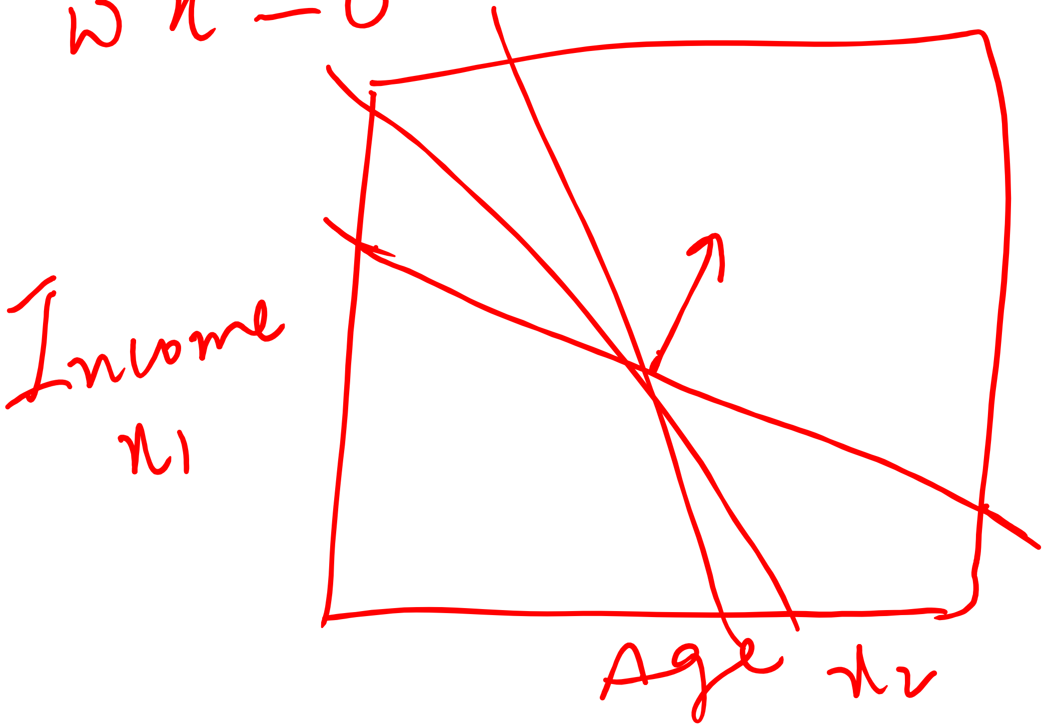
Geometry of the Perceptron (2-d)

- 2-d space $w^Tx \longrightarrow$ line

$\underline{w_1, w_2, w_0} \rightarrow$ position Hyperplane

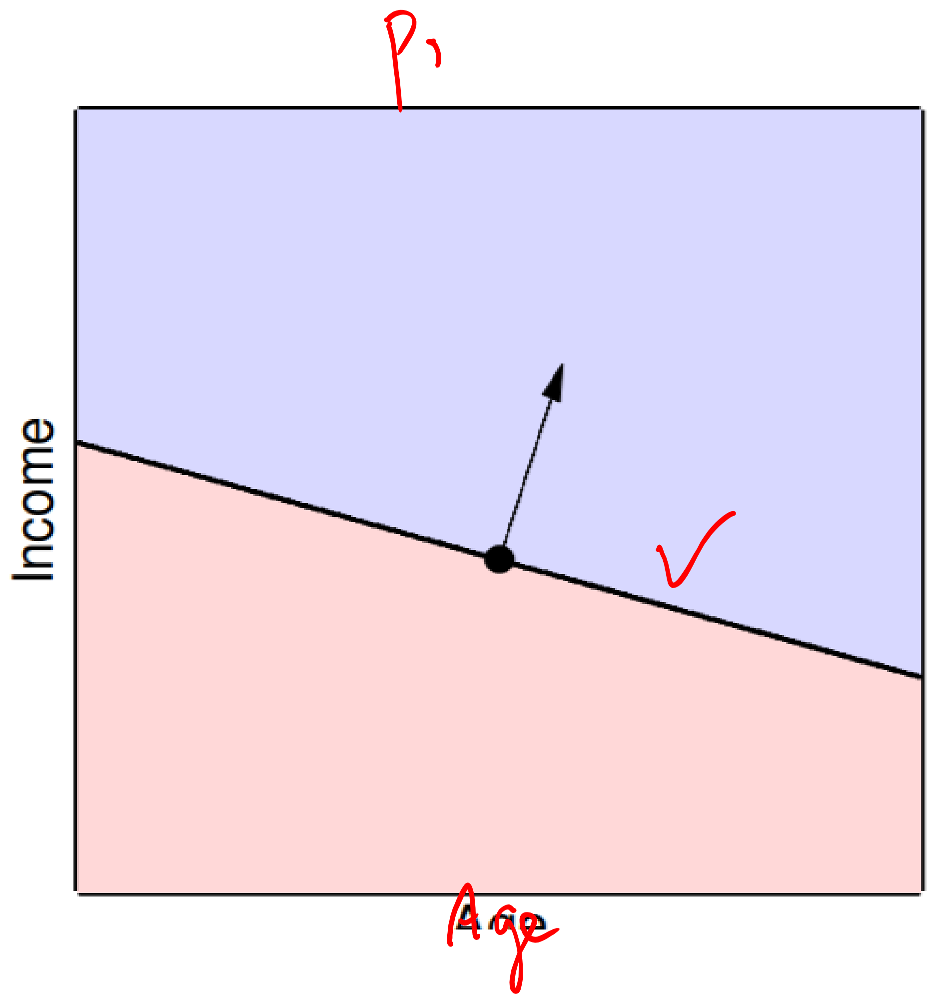
$$\begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$

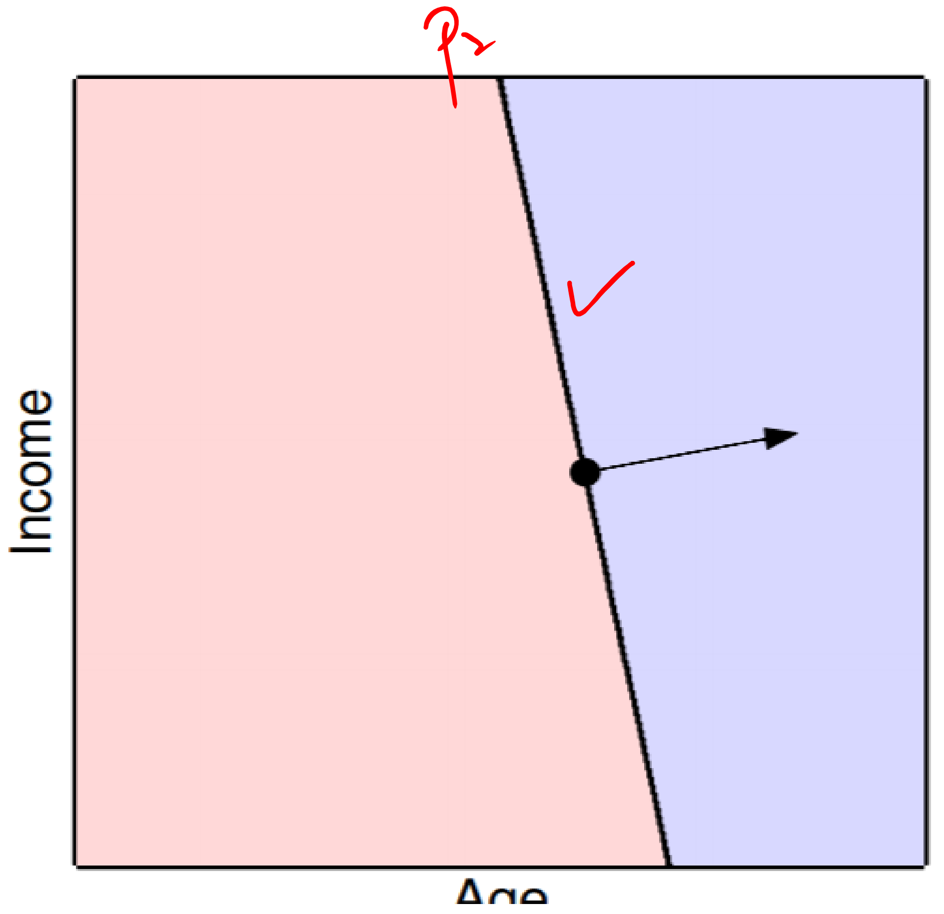
$w^Tx = 0$



Geometry of Perceptron

$h(x) = \underbrace{\text{sign}(w^T x)}$



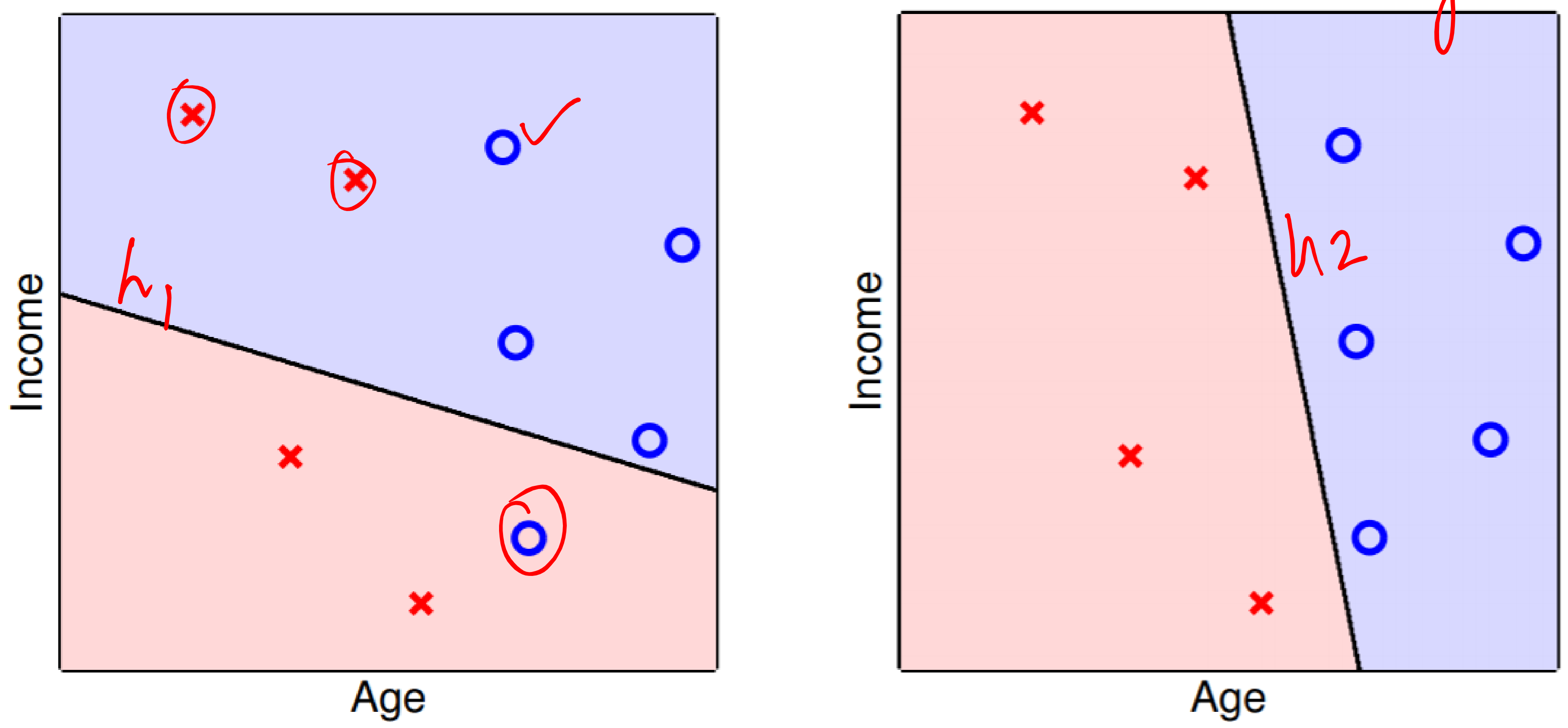


Use Data to Pick the Perfect Line

h(x)

Yes / No

g



How to Learn a Final Hypothesis *g* from *H*

- What do we want? $h(x)$
- What does that mean in terms of the data? Looks better
- How do we find this g in H(which is infinite)? $w^T x$
- How can we get started?

Criteria to pick

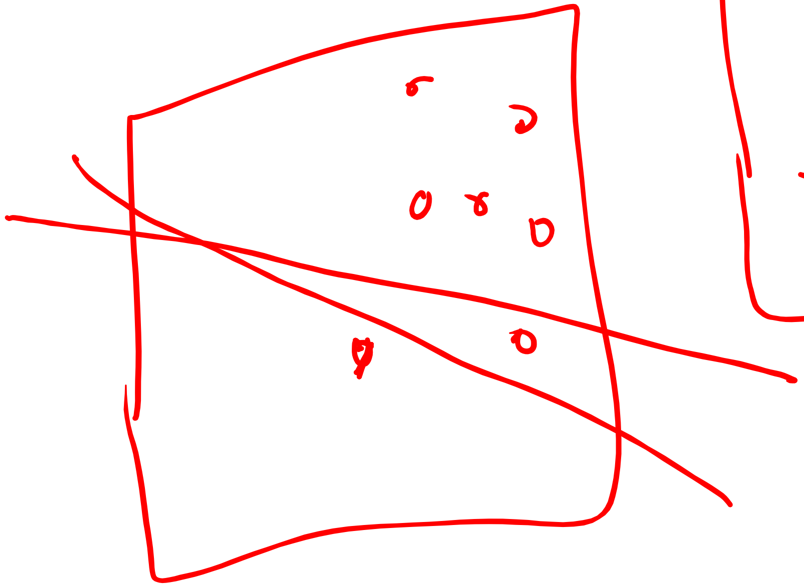
Perceptron Learning Algorithm $\rightarrow w(t)$

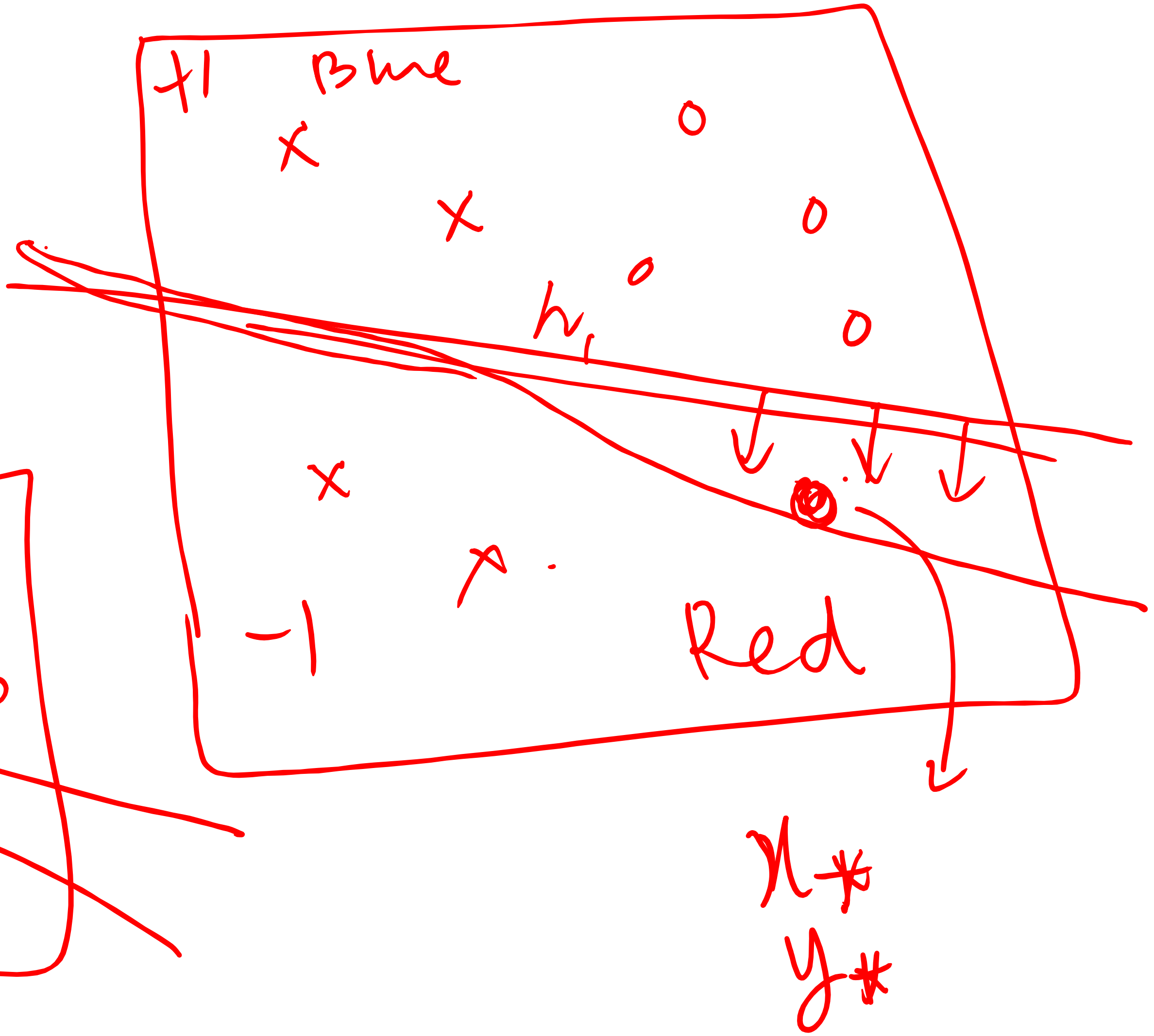
- Pick any $w(0)$

  $[0, 0, 0]$

  Move $w^T x$

  How?



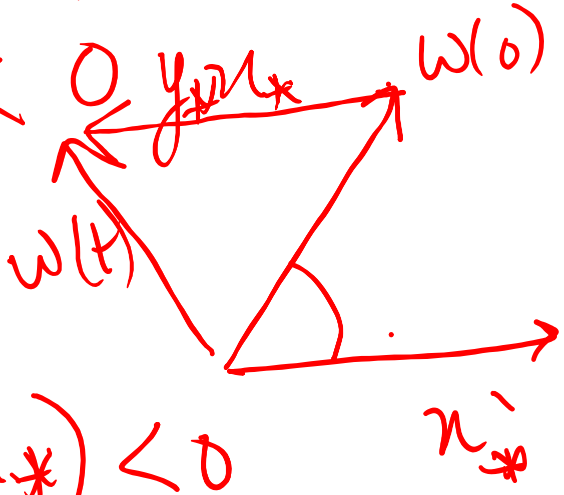


Let, misclassified point be x_*, $w(0)$

$\text{sign}(w(0) . x_*) > 0$

Update Rule

$\underline{y_*} < 0$

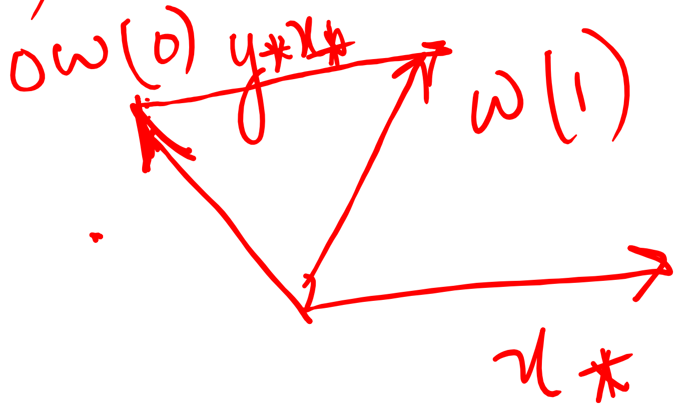


$$\boxed{w(1) = w(0) + y_* x_*}$$

Repeat

$\text{sign}(w(0) . x_*) < 0$

$y_* > 0$



PLA Summarized

Data is linearly separable

A simple iterative method.

PLA

$\mathbf{w}(1) = \mathbf{0}$
for iteration $t = 1, 2, 3, \ldots$
the weight vector is $\mathbf{w}(t)$.
From $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ pick any misclassified example.
Call the misclassified example $(\mathbf{x}_*, y_*)$,

$$\boxed{\text{sign}\,(\mathbf{w}(t) \bullet \mathbf{x}_*) \neq y_*.}$$

Update the weight:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + y_*\mathbf{x}_*.$$

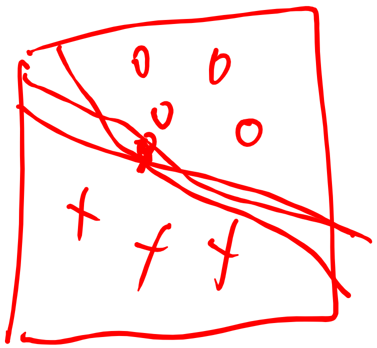
$t \leftarrow t + 1$

PLA Convergence

- Theorem: If the data can be fit by a linear separator, then after some finite number of steps, PLA will find one.

linearly separable

Problem 1.3 in textbook



w^* (optimal)

$w(t) \longrightarrow w^*$

Other Views of Learning

- Design: learning is from data, design is from specs and a model.
- Statistics → ML
- Data Mining → Unsupervised

Types of Learning

ML

- Supervised
- Unsupervised
- Reinforcement Learning

x_n , y_n

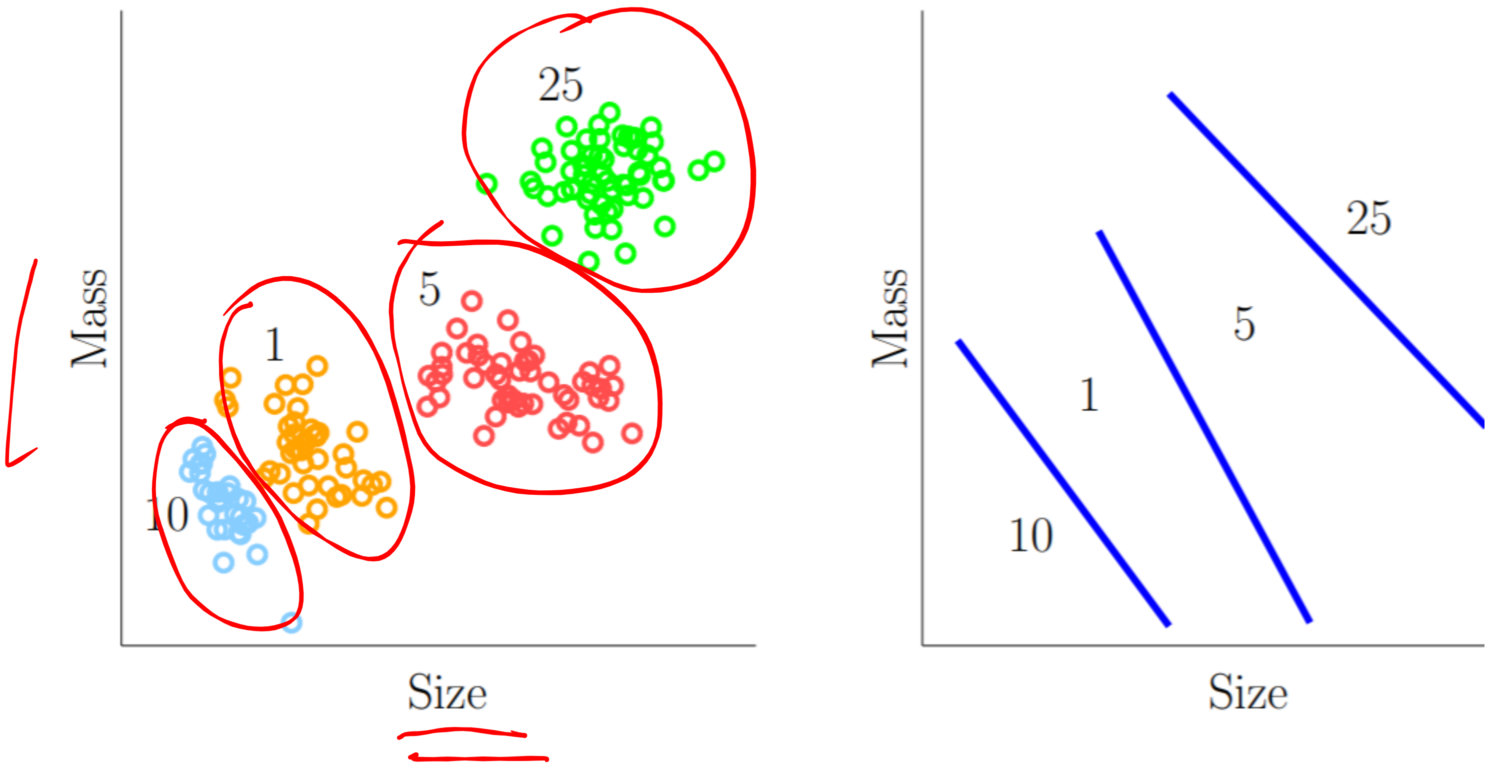
Answer

Organizing.

$\{x \rightarrow$ try something $\rightarrow$

get feedback.

Unsupervised Learning



Categorizing Coins