

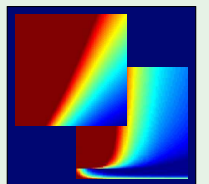
# Learning From Data

Yaser S. Abu-Mostafa  
*California Institute of Technology*

## Lecture 1: **The Learning Problem**



Sponsored by Caltech's Provost Office, E&AS Division, and IST • Tuesday, April 3, 2012



# The learning problem - Outline

- Example of machine learning
- Components of Learning
- A simple model
- Types of learning
- Puzzle

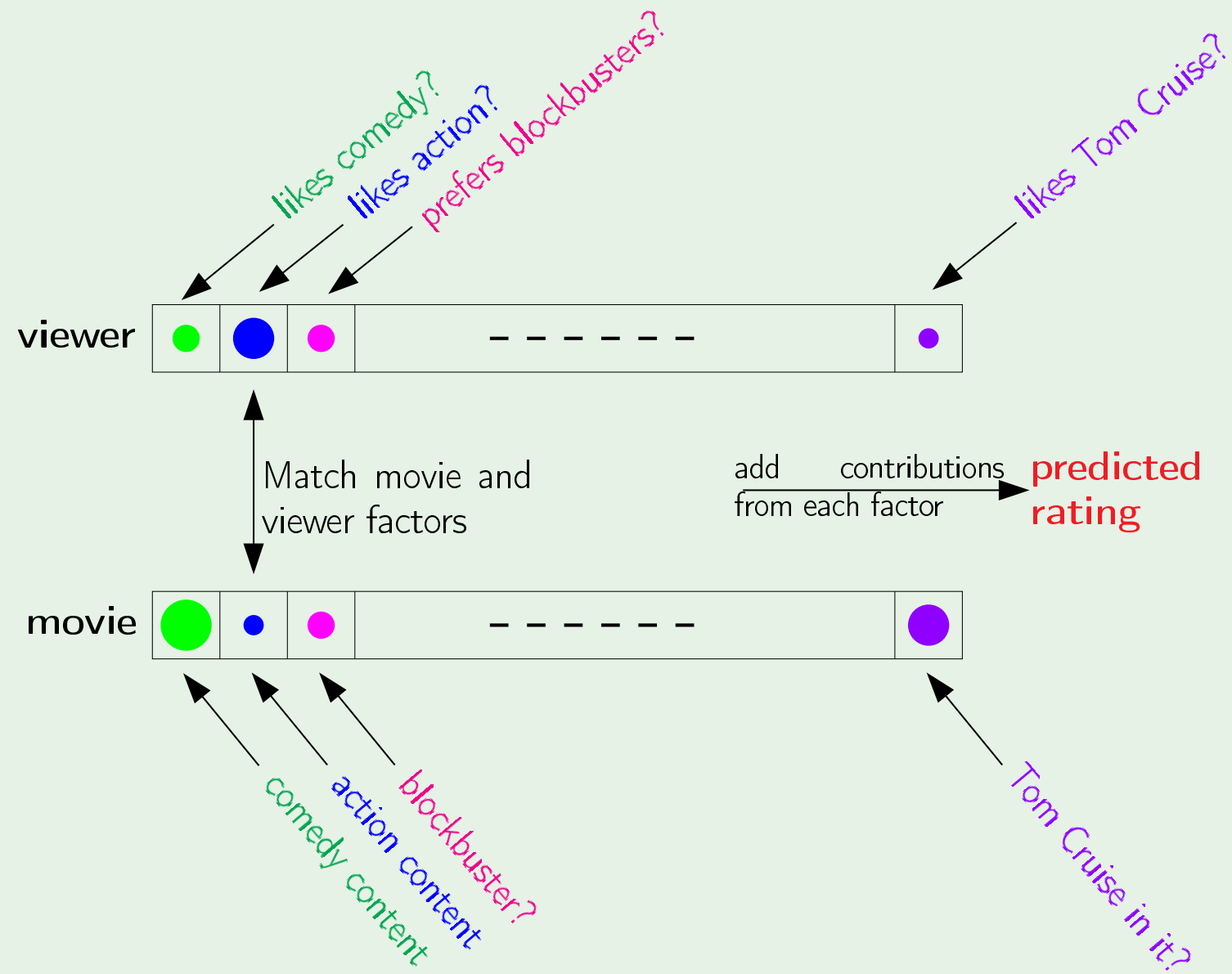
**Example:** Predicting how a viewer will rate a movie

10% improvement = **1 million dollar prize**

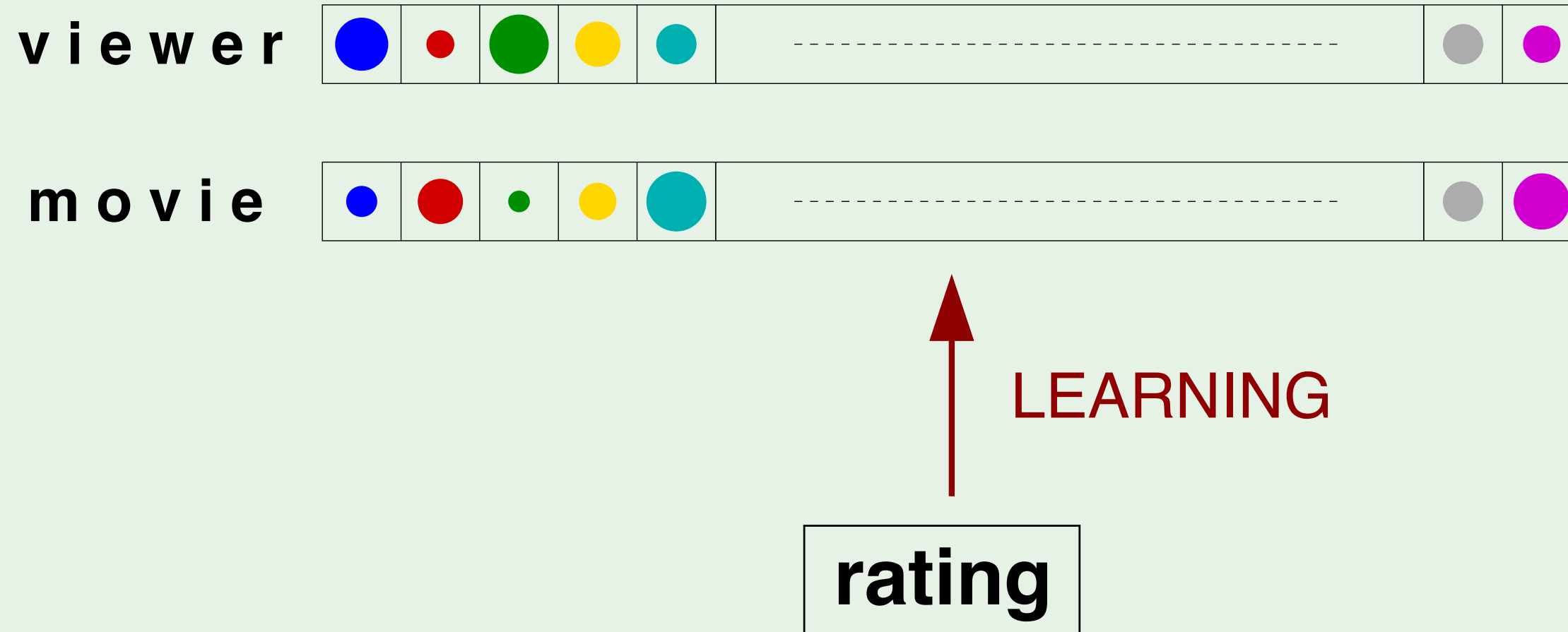
The essence of machine learning:

- A pattern exists.
- We cannot pin it down mathematically.
- We have data on it.

# Movie rating - a solution



# The learning approach



# Components of learning

**Metaphor:** Credit approval

Applicant information:

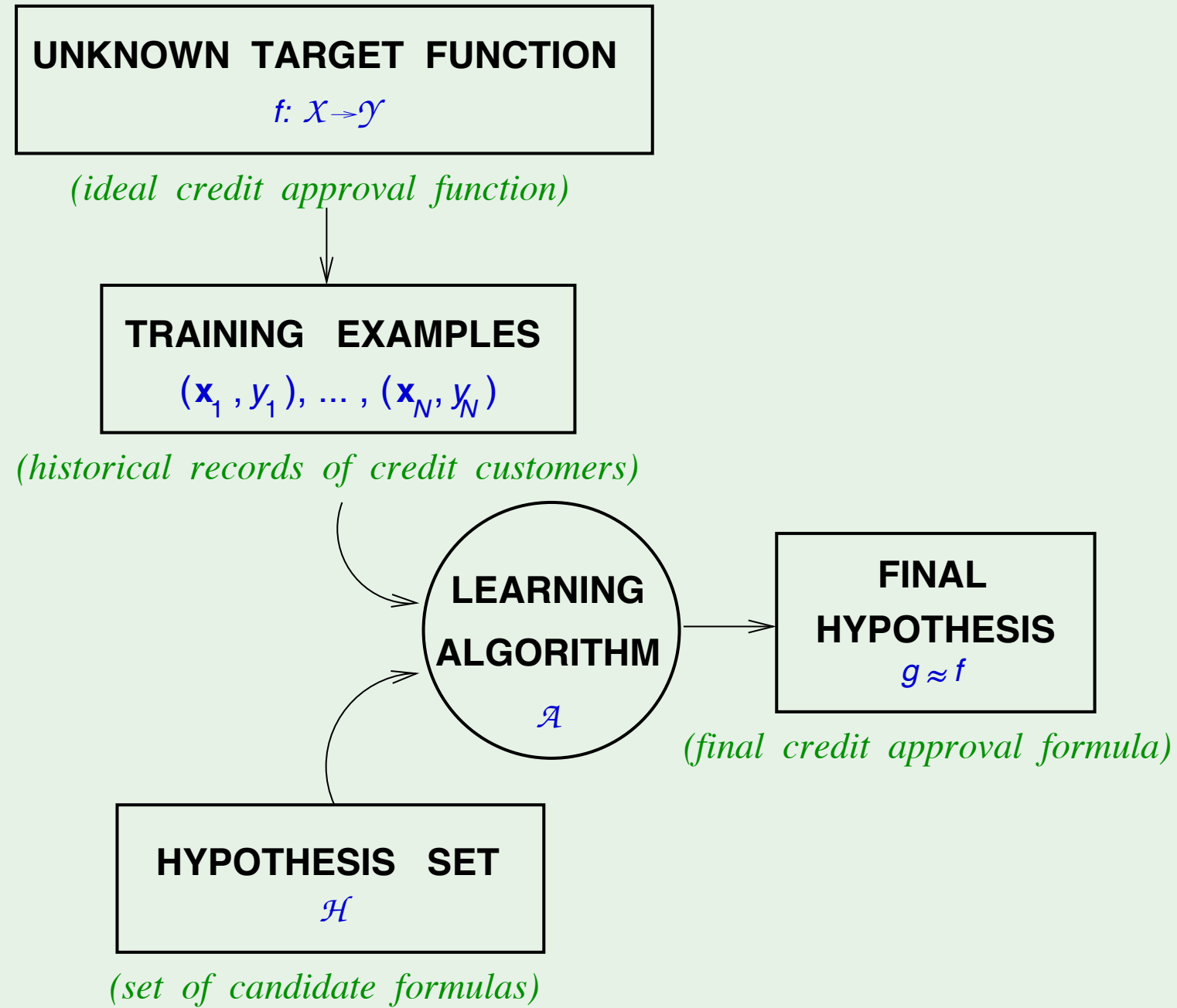
age	23 years
gender	male
annual salary	\$30,000
years in residence	1 year
years in job	1 year
current debt	\$15,000
...	...

Approve credit?

# Components of learning

## Formalization:

- Input:  $\mathbf{x}$  (*customer application*)
  - Output:  $y$  (*good/bad customer?*)
  - Target function:  $f : \mathcal{X} \rightarrow \mathcal{Y}$  (*ideal credit approval formula*)
  - Data:  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$  (*historical records*)
- ↓   ↓   ↓
- Hypothesis:  $g : \mathcal{X} \rightarrow \mathcal{Y}$  (*formula to be used*)





# Solution components

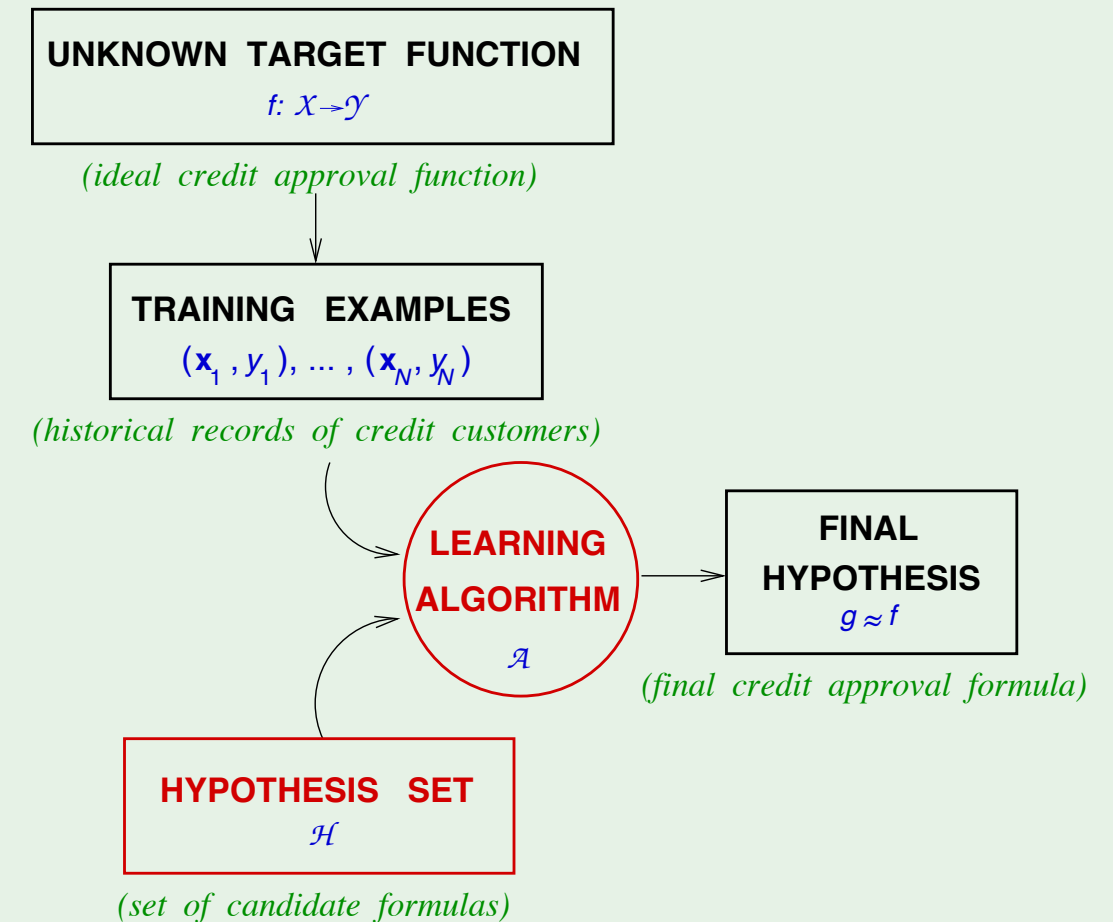
The 2 solution components of the learning problem:

- The Hypothesis Set

$$\mathcal{H} = \{h\} \quad g \in \mathcal{H}$$

- The Learning Algorithm

Together, they are referred to as the *learning model*.



## A simple hypothesis set - the 'perceptron'

For input  $\mathbf{x} = (x_1, \dots, x_d)$  'attributes of a customer'

Approve credit if  $\sum_{i=1}^d w_i x_i > \text{threshold}$ ,

Deny credit if  $\sum_{i=1}^d w_i x_i < \text{threshold}$ .

This linear formula  $h \in \mathcal{H}$  can be written as

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

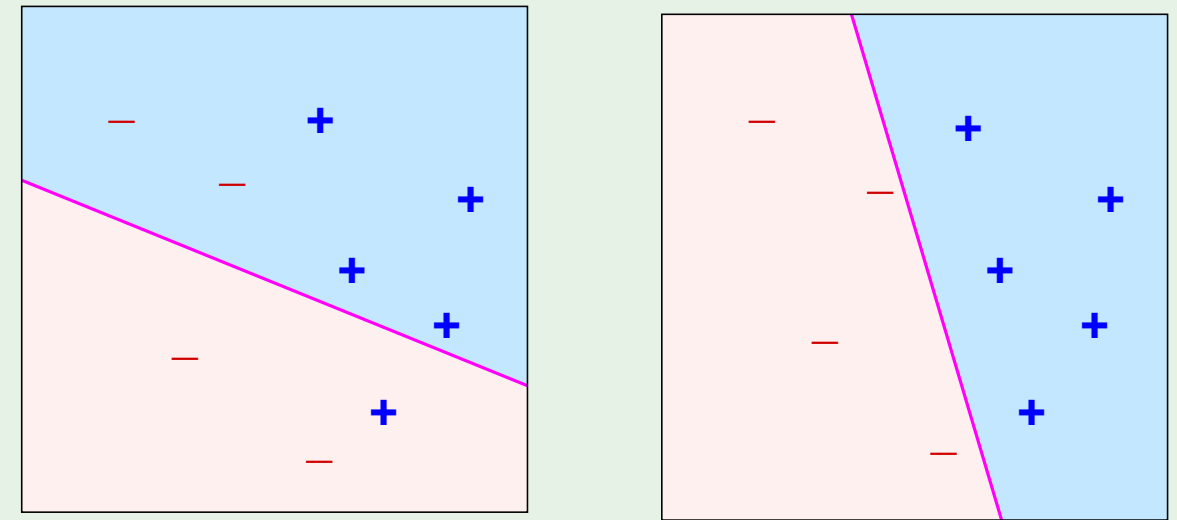
$$h(\mathbf{x}) = \text{sign} \left( \left( \sum_{i=1}^d w_i x_i \right) + w_0 \right)$$

Introduce an artificial coordinate  $x_0 = 1$ :

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

In vector form, the perceptron implements

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



'linearly separable' data

# A simple learning algorithm - PLA

The perceptron implements

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

Given the training set:

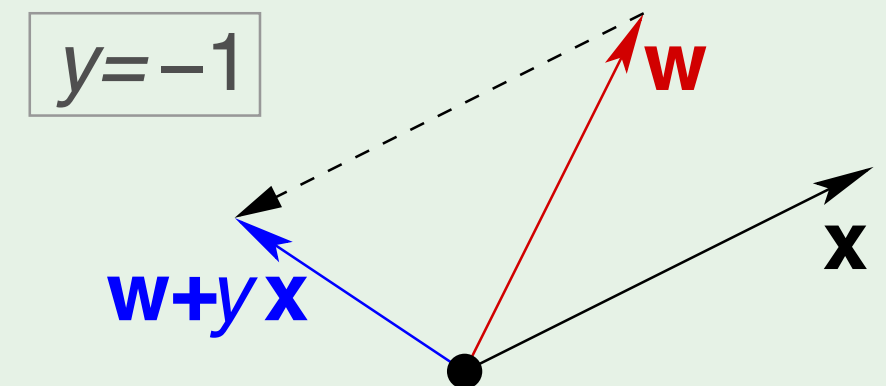
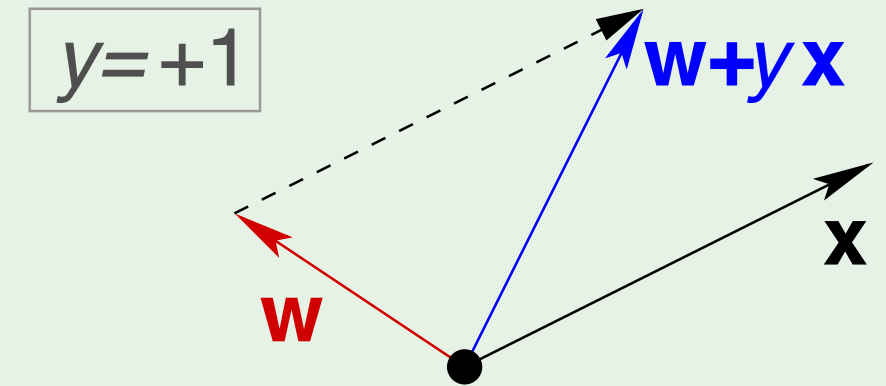
$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$$

pick a **misclassified** point:

$$\text{sign}(\mathbf{w}^T \mathbf{x}_n) \neq y_n$$

and update the weight vector:

$$\mathbf{w} \leftarrow \mathbf{w} + y_n \mathbf{x}_n$$



# Iterations of PLA

- One iteration of the PLA:

$$\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$$

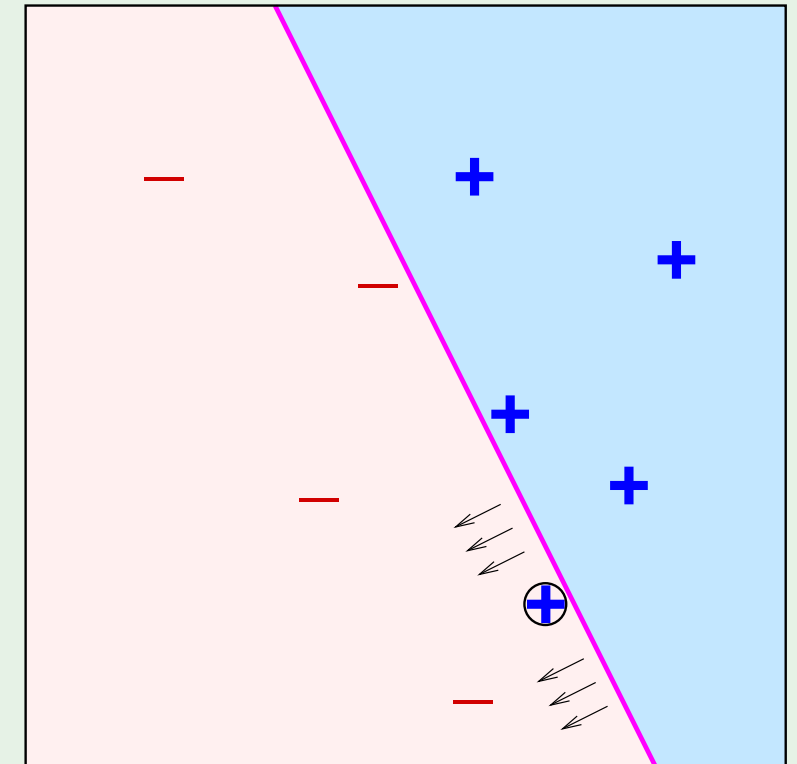
where  $(\mathbf{x}, y)$  is a misclassified training point.

- At iteration  $t = 1, 2, 3, \dots$ , pick a misclassified point from

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$$

and run a PLA iteration on it.

- That's it!



# Basic premise of learning

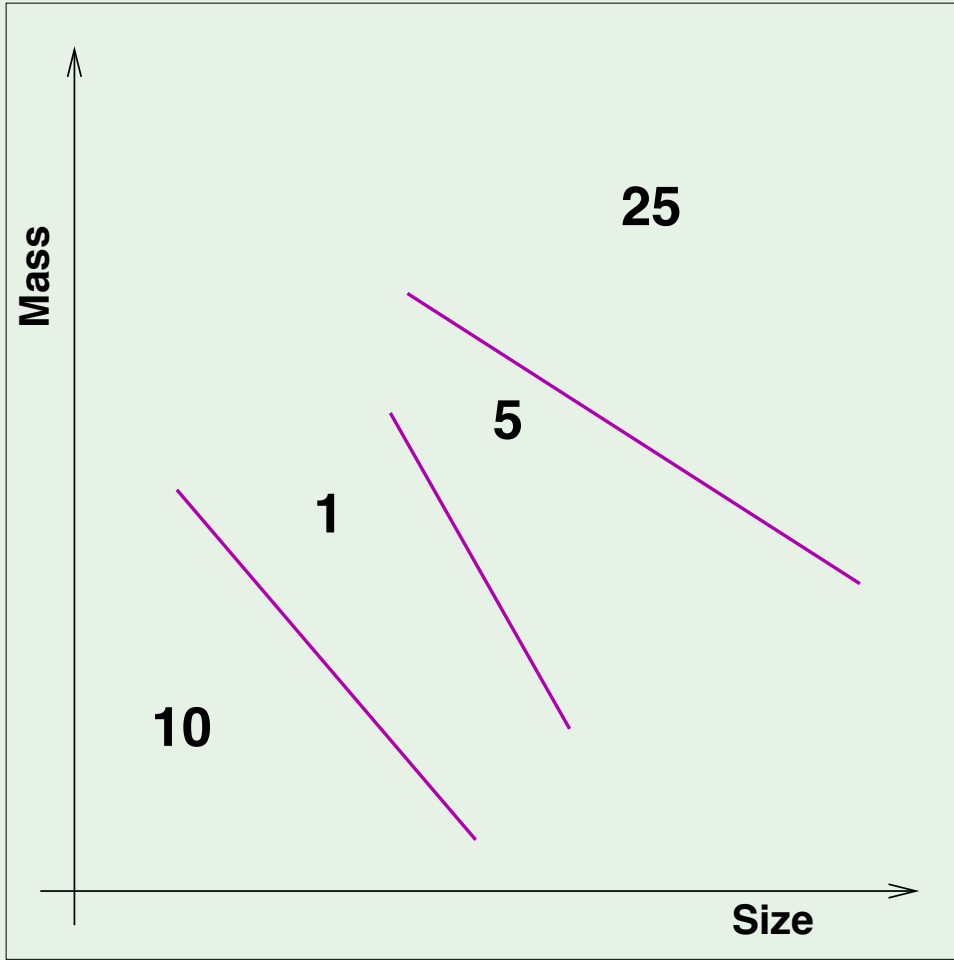
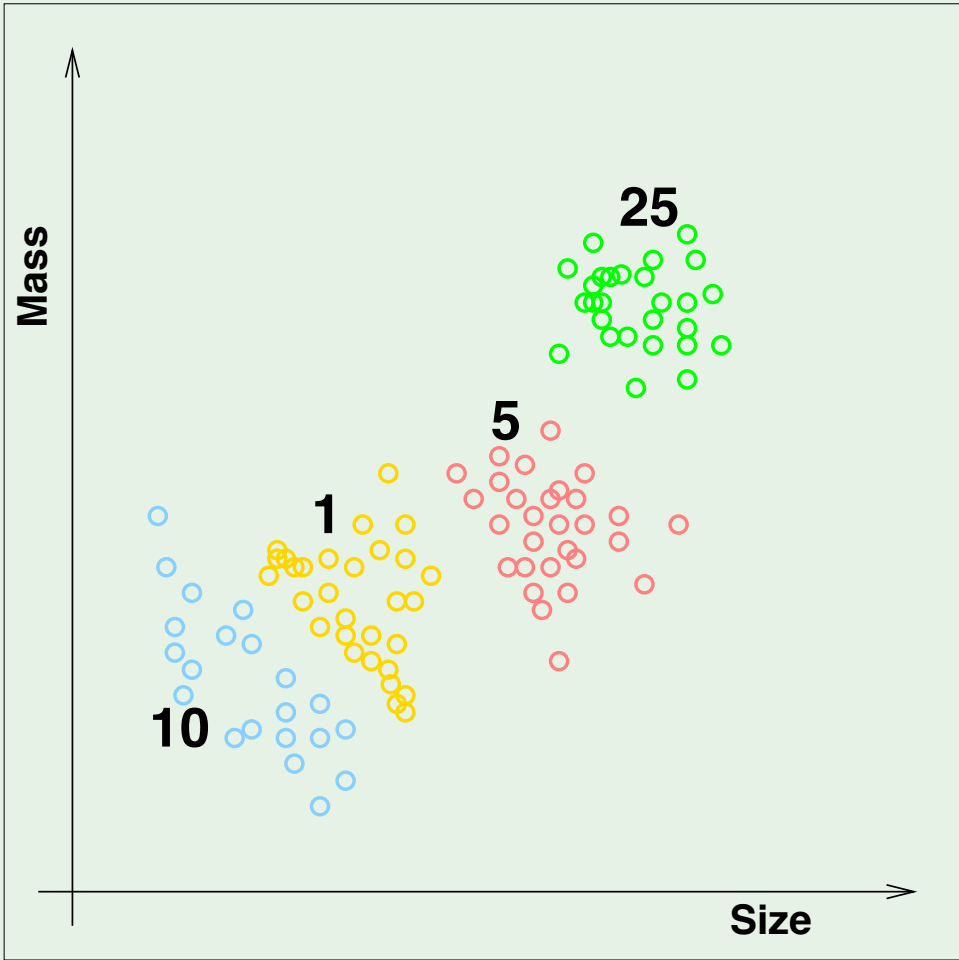
*“using a set of observations to uncover an underlying process”*

broad premise  $\implies$  many variations

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

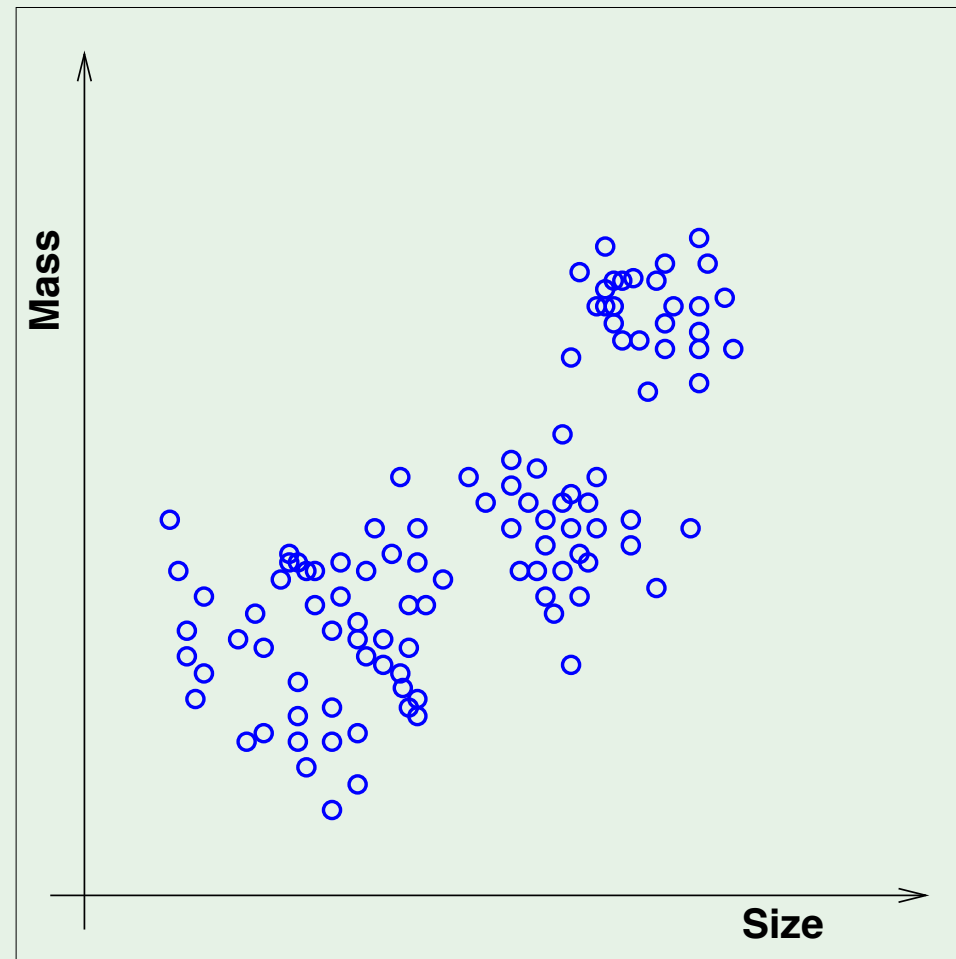
# Supervised learning

Example from vending machines – coin recognition



# Unsupervised learning

Instead of (input, correct output), we get (input, ? )

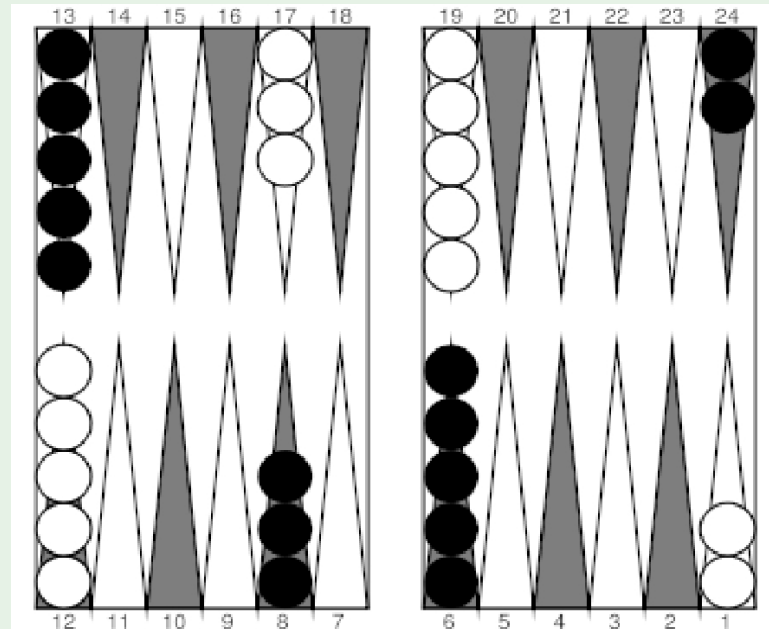




# Reinforcement learning

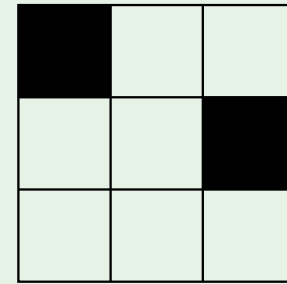
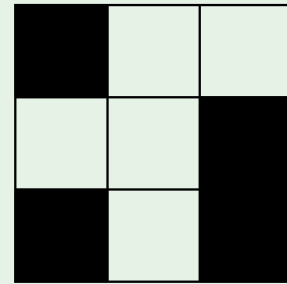
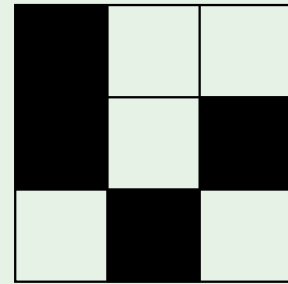
Instead of (input, correct output),

we get (input, *some* output, grade for this output)

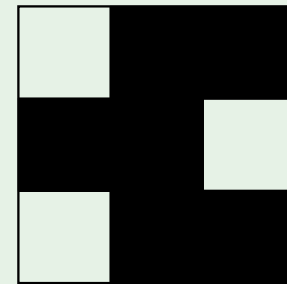
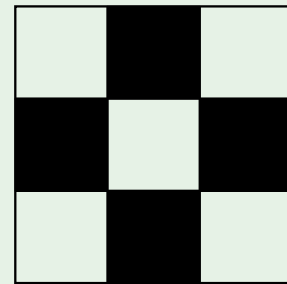
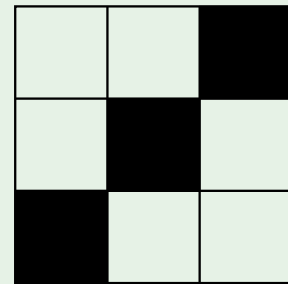


The world champion was a neural network!

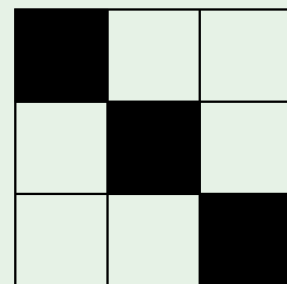
# A Learning puzzle



$$f = -1$$



$$f = +1$$



$$f = ?$$