Outline of the Course

1. The Learning Problem (*April 3*)
2. Is Learning Feasible? (*April 5*)
3. The Linear Model I (*April 10*)
4. Error and Noise (*April 12*)
5. Training versus Testing (*April 17*)
6. Theory of Generalization (*April 19*)
7. The VC Dimension (*April 24*)
8. Bias-Variance Tradeoff (*April 26*)
9. The Linear Model II (*May 1*)
10. Neural Networks (*May 3*)
11. Overfitting (*May 8*)
12. Regularization (*May 10*)
13. Validation (*May 15*)
14. Support Vector Machines (*May 17*)
15. Kernel Methods (*May 22*)
16. Radial Basis Functions (*May 24*)
17. Three Learning Principles (*May 29*)
18. Epilogue (*May 31*)

- theory; mathematical
- technique; practical
- analysis; conceptual

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

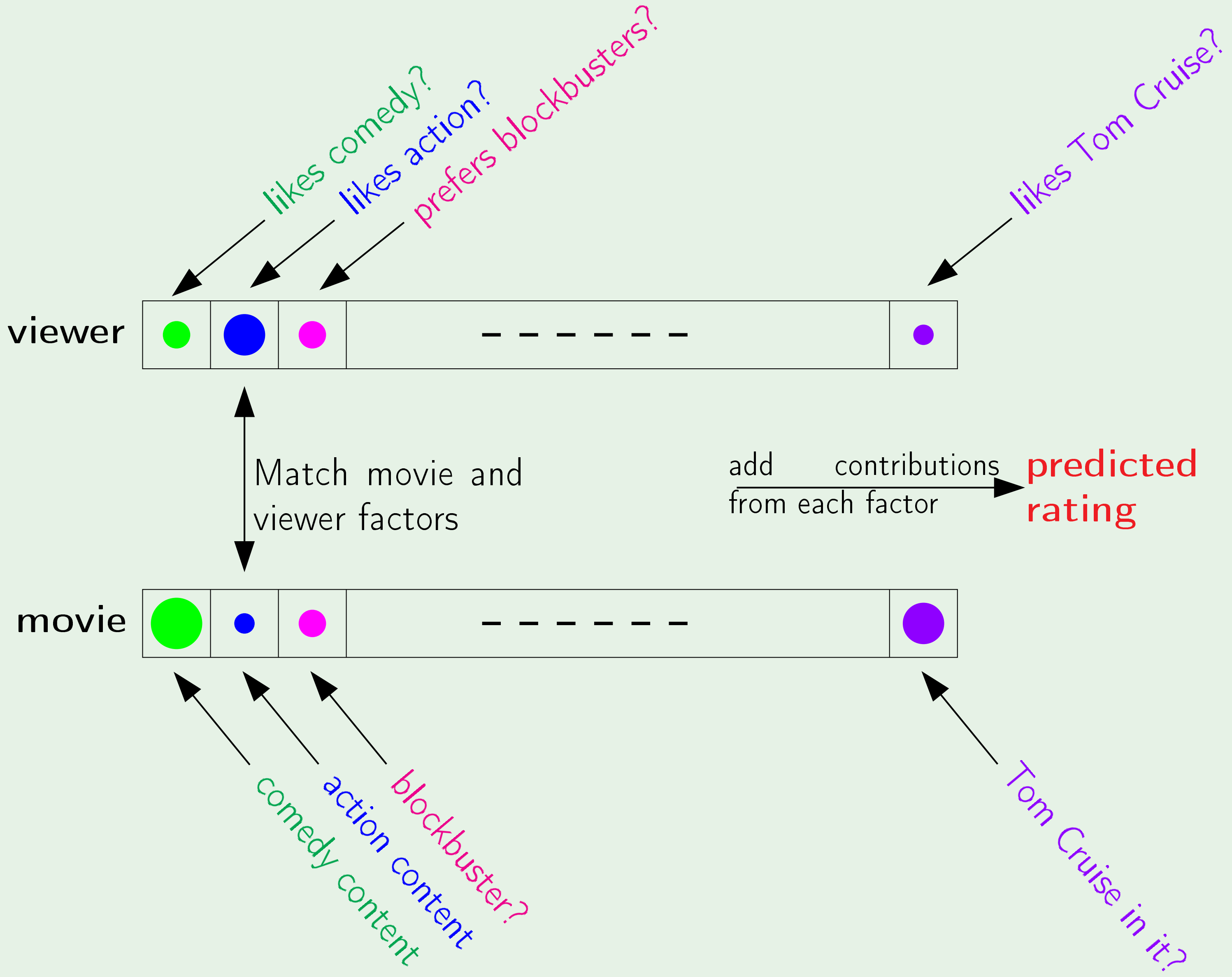
Example: Predicting how a viewer will rate a movie

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

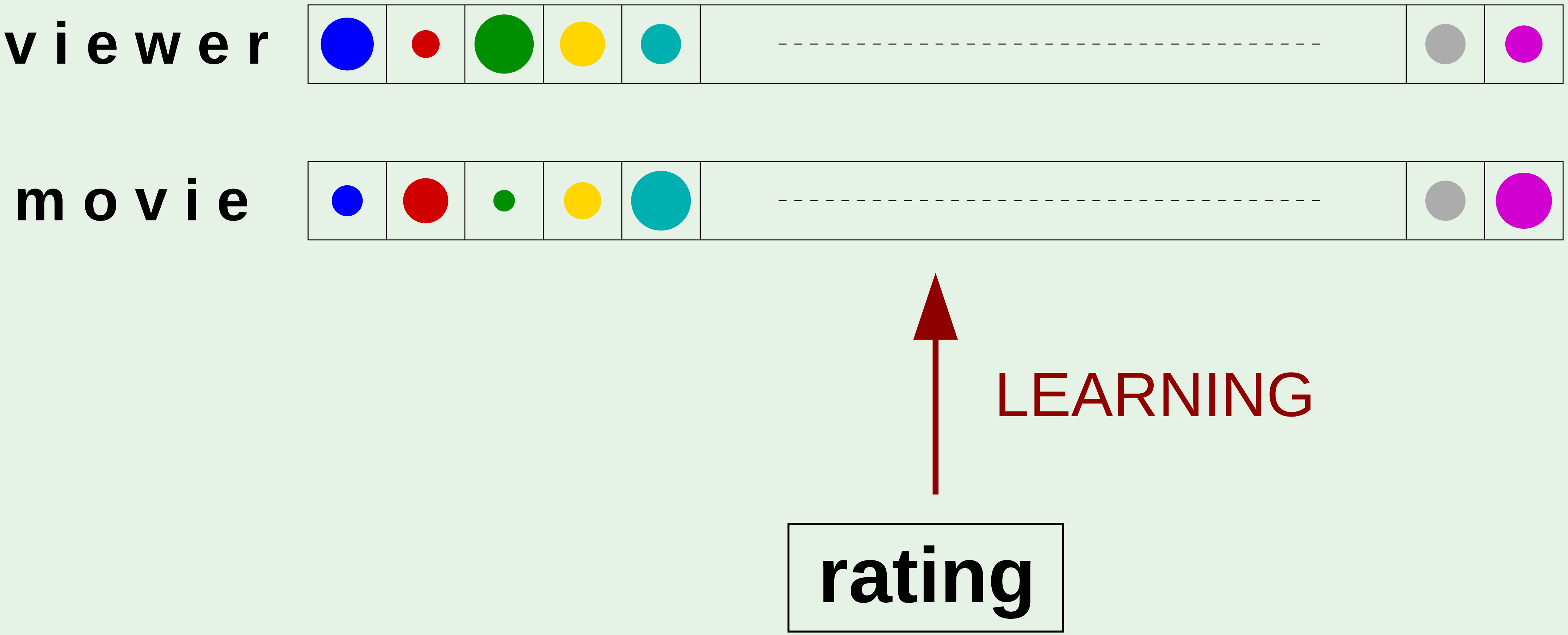
The essence of machine learning:

- 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
$\cdots$	$\cdots$

Approve credit?

Components of learning

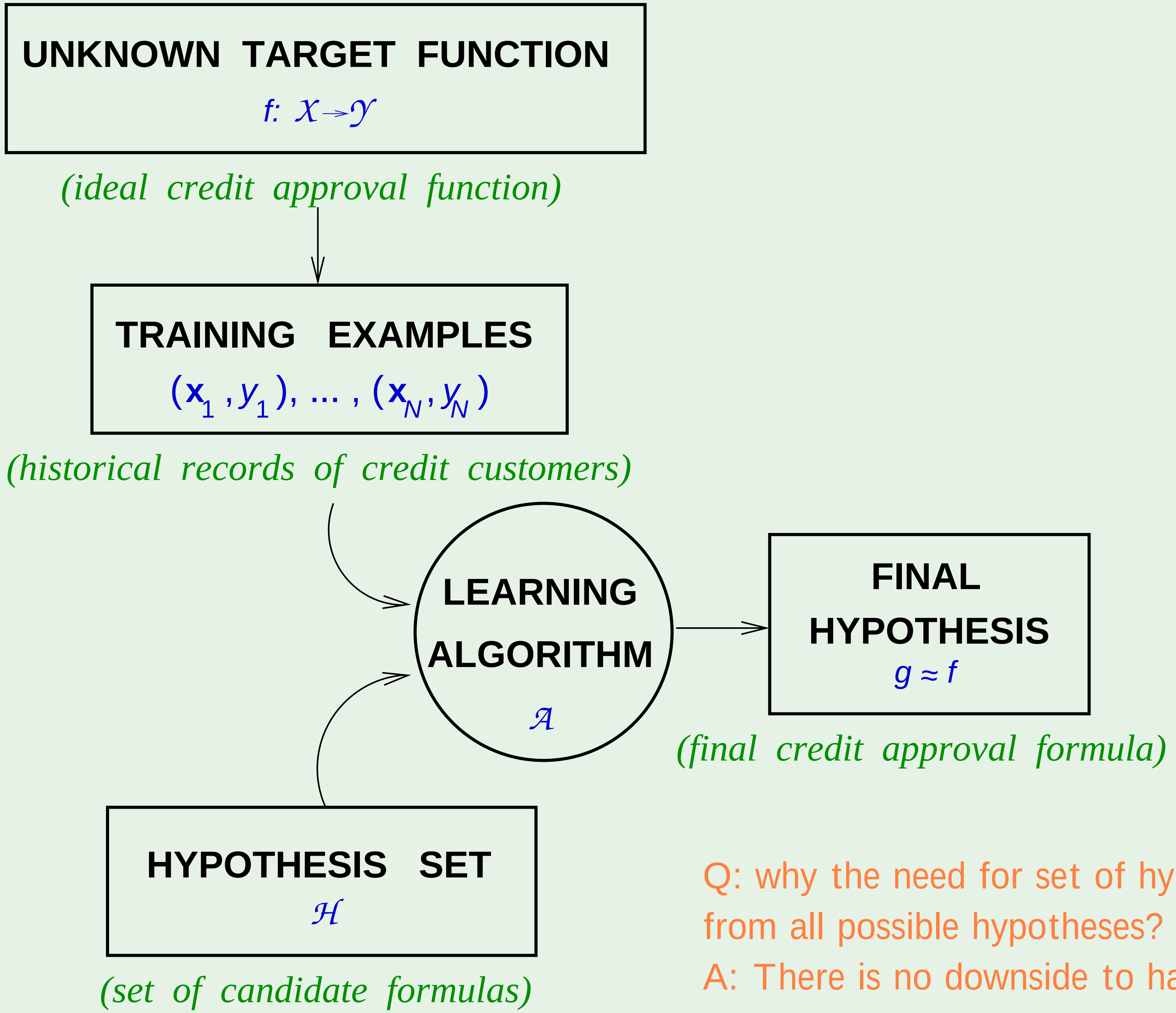
Formalization:

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

$$\downarrow \quad \downarrow \quad \downarrow$$

- Hypothesis: $g : \mathcal{X} \to \mathcal{Y}$ *(formula to be used)*

We modify g to attempt to approximate f



Q: why the need for set of hypotheses H, why not let A pick from all possible hypotheses?

A: There is no downside to having H since from a practical POV, you pick a Linear Model/NN/SVM at the beginning. The upside to H is that it is pivotal to if we can learn and how well we learn from the data.

Solution components

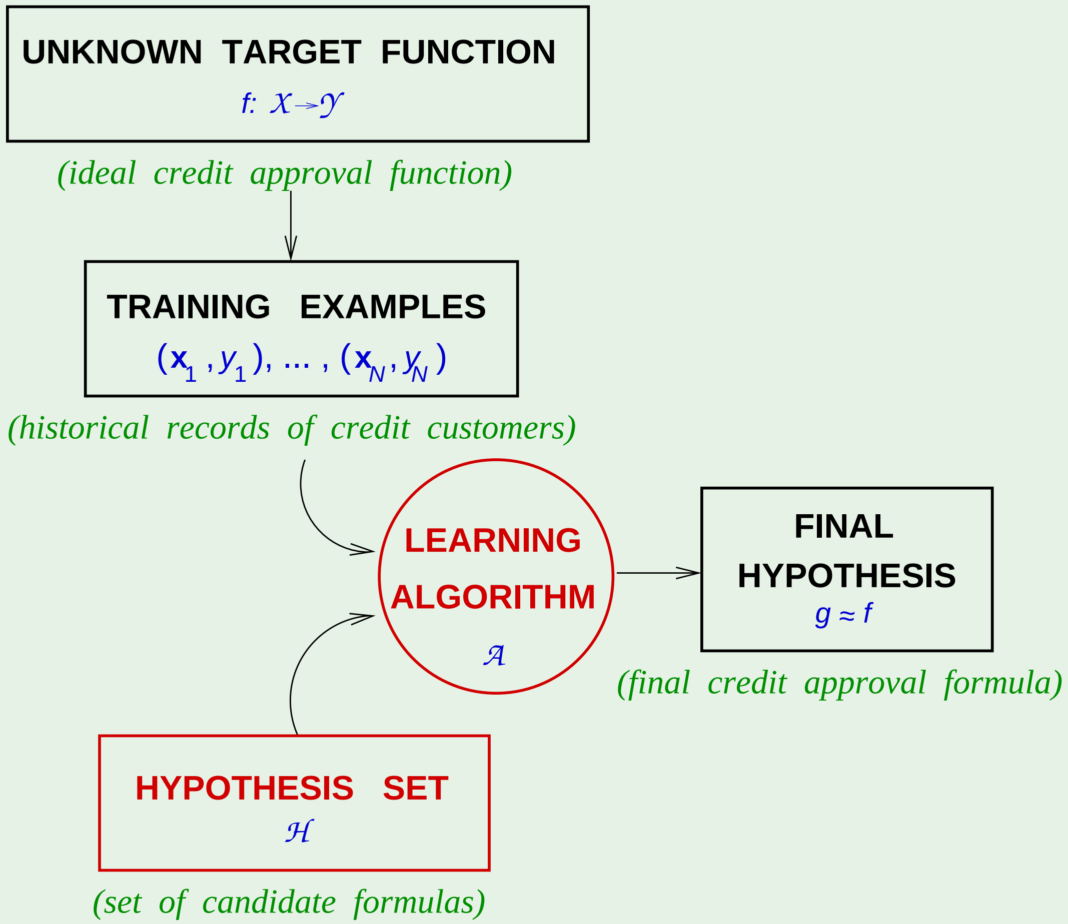
The 2 solution components of the learning problem:

- The Hypothesis Set (e.g. perceptron, NN, SVM)

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

- The Learning Algorithm (e.g. perceptron learning model, backprop., quadratic programming)

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



A simple hypothesis set - the 'perceptron'

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

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

The linear addition of w*x makes this a perceptron

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

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

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

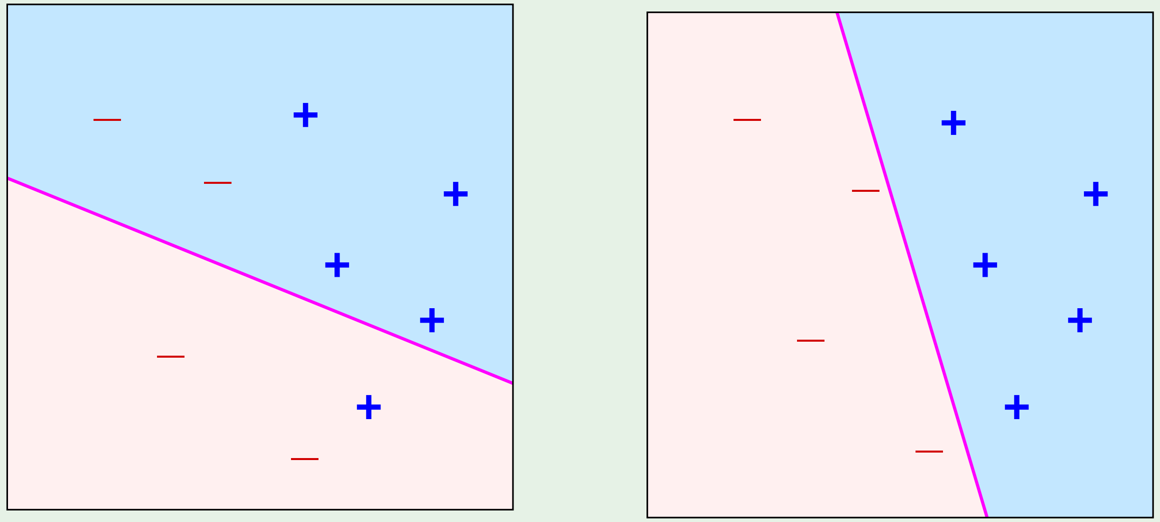
Red parameters determine the final hypothesis - these are the aspects we change in the learning algorithm

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

A choice of wi determines the location of the purple separation line below. (Note: change threshold to w0 and let x0 = 1 to simplify the expression)

Introduce an artificial coordinate $x_0 = 1$:

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



'linearly separable' data

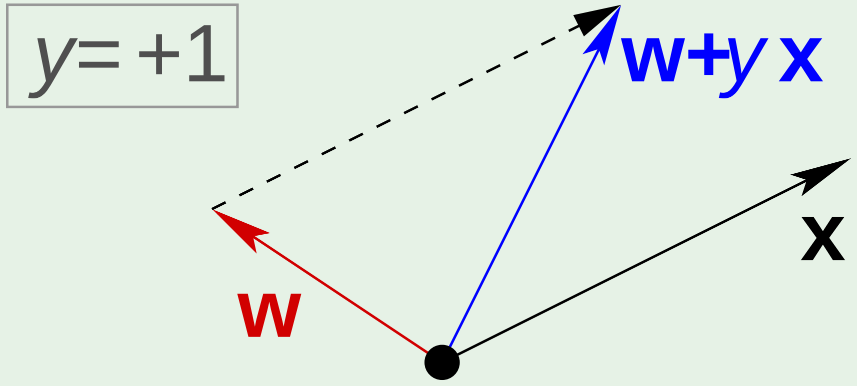
In vector form, the perceptron implements

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

A simple learning algorithm - PLA

The perceptron implements

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

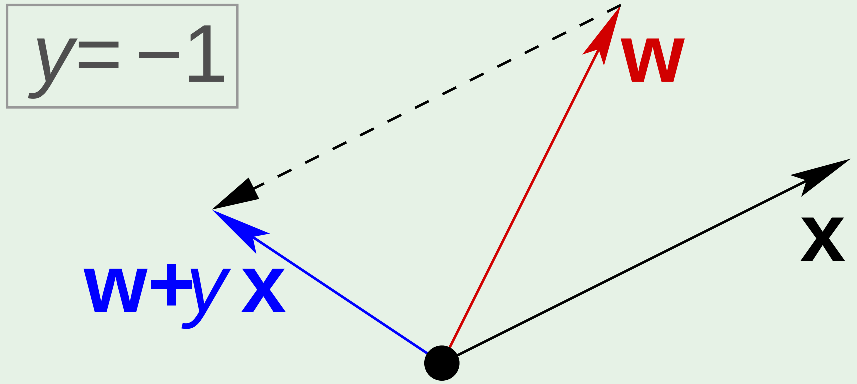


Given the training set:

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

pick a misclassified point:

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



and update the weight vector:

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

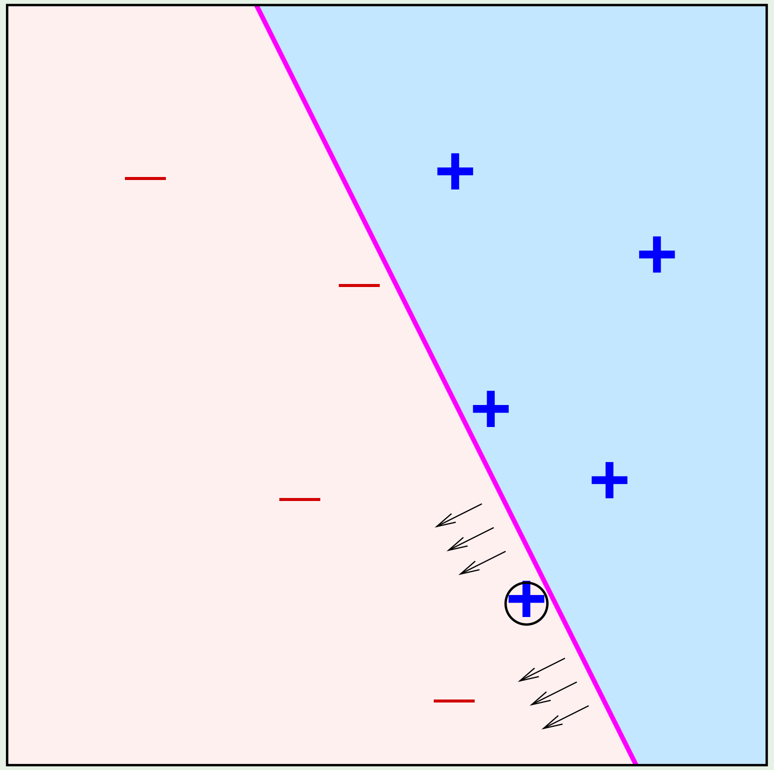
So: since yn = + or - 1, add or subtract x to w,
so now the hypothesis will correctly classify this point

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 $t = 1, 2, 3, \cdots$, pick a misclassified point from

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

and run a PLA iteration on it.

- That's it!

The learning problem - Outline

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

Basic premise of learning

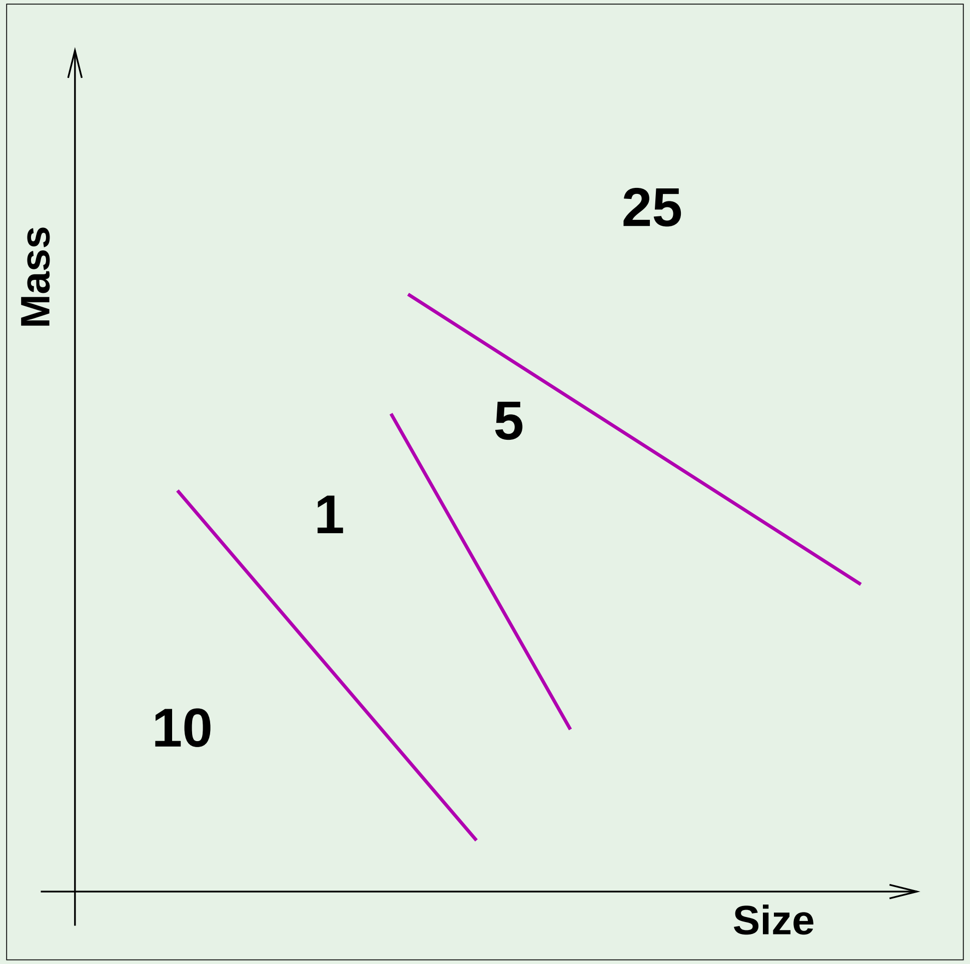
"using a set of observations to uncover an underlying process"

broad premise $\Longrightarrow$ many variations

- **Supervised Learning**
- **Unsupervised Learning**
- **Reinforcement Learning**

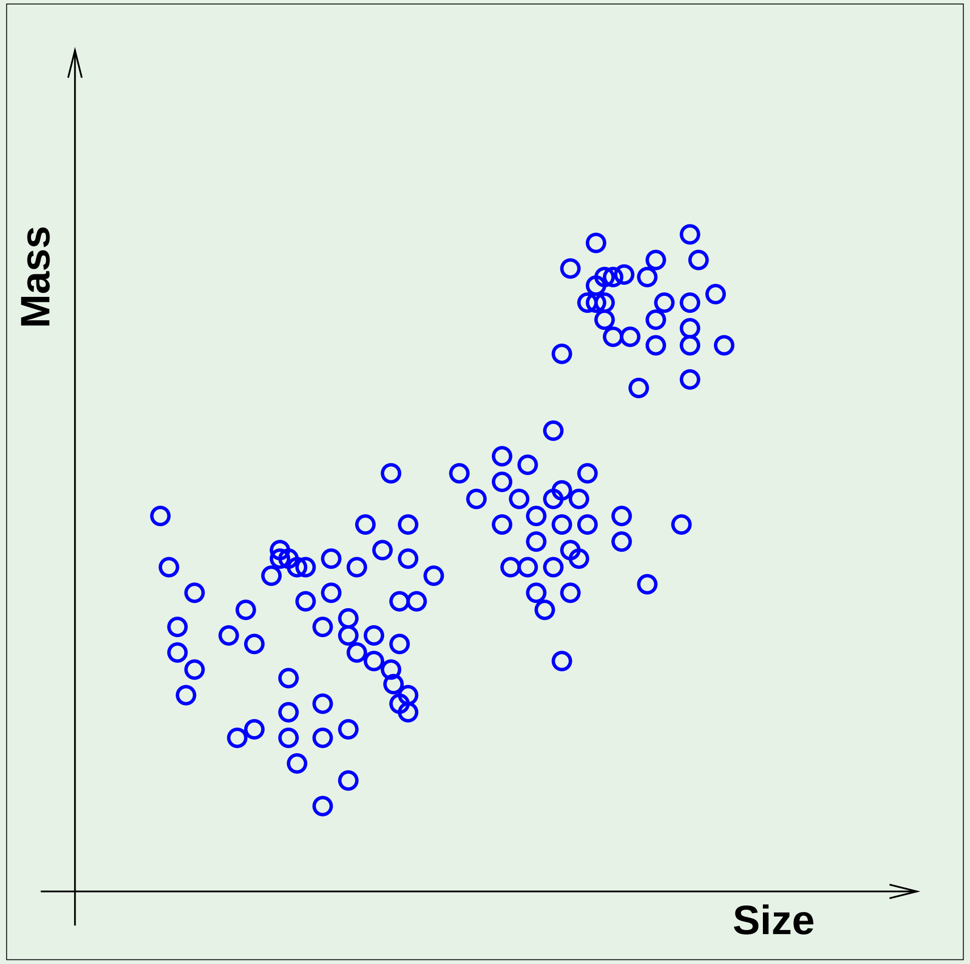
Supervised learning

Example from vending machines – **coin recognition**



Unsupervised learning

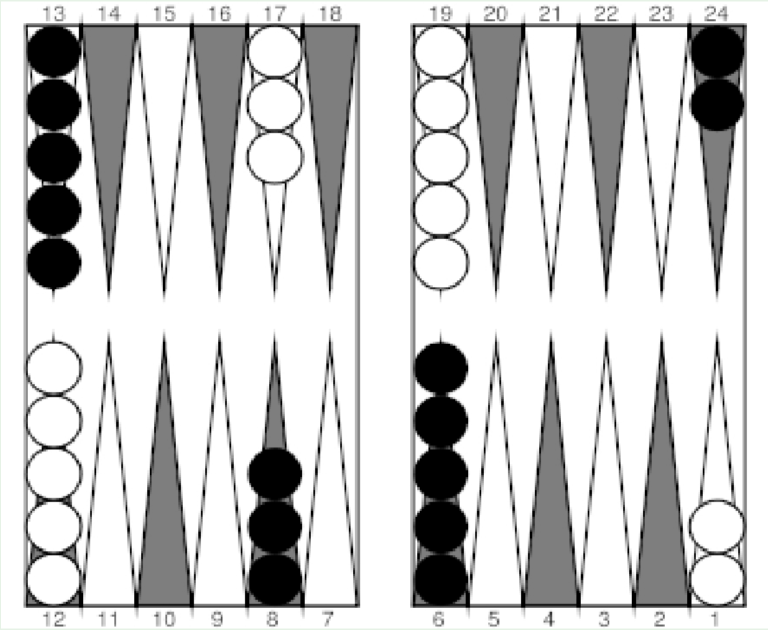
Instead of (input,correct output), we get (input, ?)



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