# Machine Learning

December 15, 2017

# The Essence of Machine Learning

## Learning from data

- A pattern exists.
- There seems to be no easy mathematical relationship.
- We have data on it.

This set of slides are based on Professor Yaser Abu-Mostafa's course *Learning from Data*. See `http://work.caltech.edu/telecourse.html`

# Examples of Machine Learning

**Credit Approval**

Applicant Information:

- age
- gender
- salary
- current debt
- years in job . . .

# Examples of Machine Learning

**Credit Approval**

Applicant Information:

- age
- gender
- salary
- current debt
- years in job . . .

Should we approve credit?

**Data**

- Data on previous applications and their credit history.

# Examples of Machine Learning

**Movie Ratings**

Predict how a user would rate a movie.

# Examples of Machine Learning

## Movie Ratings

Predict how a user would rate a movie.

Each user is modelled as a vector of attributes:

- likes comedy?
- likes block-busters?
- likes sci-fi?
- likes a specific actor?
- ...

# Examples of Machine Learning

## Movie Ratings

Predict how a user would rate a movie.

Each user is modelled as a vector of attributes:

- likes comedy?
- likes block-busters?
- likes sci-fi?
- likes a specific actor?
- ...

How would the user rate a given movie on a scale from $1$ to $10$?

### Data

- Data on how other users rated the given movie.

**Formalization**

- **Input:** $\mathbf{x}$ (applicant information)

# Components of the Learning Problem

**Formalization**

- **Input:** $\mathbf{x}$ (applicant information)
- **Output:** $y$ (good/bad customer)

# Components of the Learning Problem

**Formalization**

- **Input:** $\mathbf{x}$ (applicant information)
- **Output:** $y$ (good/bad customer)
- **Target function:** $f : \mathcal{X} \to \mathcal{Y}$ (ideal credit approval formula)

# Components of the Learning Problem

**Formalization**

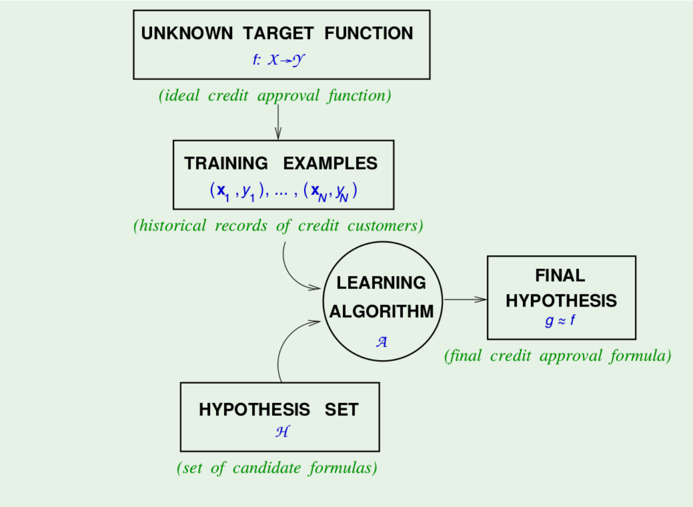
- **Input:** $\mathbf{x}$ (applicant information)
- **Output:** $y$ (good/bad customer)
- **Target function:** $f\colon \mathcal{X} \to \mathcal{Y}$ (ideal credit approval formula)
- **Data:** $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})$ (historical records)

# Components of the Learning Problem

**Formalization**

- **Input:** $\mathbf{x}$ (applicant information)
- **Output:** $y$ (good/bad customer)
- **Target function:** $f\colon \mathcal{X} \to \mathcal{Y}$ (ideal credit approval formula)
- **Data:** $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})$ (historical records)
- **Hypothesis:** $h\colon \mathcal{X} \to \mathcal{Y}$ (formula to be used)

# The Learning Problem



# Solution Components

Two solution components:

- The Hypothesis Set $\mathcal{H}$
- The Learning Algorithm

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

# Solution Components

Two solution components:

- The Hypothesis Set $\mathcal{H}$
- The Learning Algorithm

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

Why specify a hypothesis set?

- This is what is generally done: you choose a linear model, or an SVM or a neural network
- Important for developing a theory of learning

# Hypotheses Sets and Learning Algorithms

**Examples**

| *Hypothesis Set* | *Learning Algorithm* |
|---|---|
| Linear Regression | Gradient Descent |
| Neural Networks | Back Propagation |
| SVM | Quadratic Programming |
| Mixture of Gaussians Model | EM Algorithm |

# The Perceptron: A Simple Hypothesis Set

For input $\mathbf{x} = (x_1, \ldots, x_n)$, the customer attributes,

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

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

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

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

# The Perceptron: A Simple Hypothesis Set

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

# The Perceptron: A Simple Hypothesis Set

$$g(\mathbf{x}) = \mathrm{sign}\left(\sum_{i=1}^{n} w_i x_i + w_0\right)$$

# The Perceptron: A Simple Hypothesis Set

$$g(\mathbf{x}) = \mathrm{sign}\left(\sum_{i=1}^{n} w_i x_i + w_0\right)$$

Introduce an artificial coordinate $x_0 = 1$:

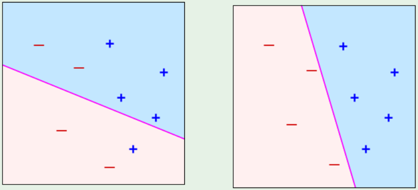
$$g(\mathbf{x}) = \mathrm{sign} \sum_{i=0}^{n} w_i x_i = \mathrm{sign}(\mathbf{w}^{\mathsf{T}} \cdot \mathbf{x})$$

# The Perceptron: A Simple Hypothesis Set

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

Introduce an artificial coordinate $x_0 = 1$:

$$g(\mathbf{x}) = \text{sign} \sum_{i=0}^{n} w_i x_i = \text{sign}(\mathbf{w}^\mathsf{T} \cdot \mathbf{x})$$




'linearly separable' data

# The Perceptron Learning Algorithm (PLA)

The perceptron implements

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

Given a training set:

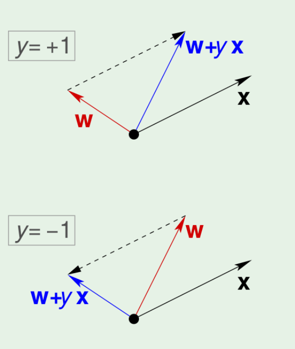
$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})$$

pick a **misclassified** point:

$$\text{sign}(\mathbf{w}^{\mathsf{T}} \mathbf{x}^{(k)}) \neq y^{(k)}$$

and update the weight vector:

$$\mathbf{w}_{\text{new}} \leftarrow \mathbf{w}_{\text{old}} + y^{(k)} \mathbf{x}^{(k)}$$



# Iterations of PLA

- One iteration of the PLA:

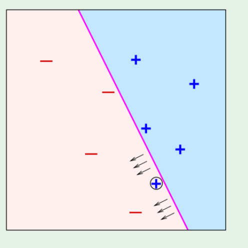
$$\mathbf{w}_{\text{new}} \leftarrow \mathbf{w}_{\text{old}} + y\mathbf{x}$$

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

- On iteration $i = 1, 2, 3, \ldots$, pick a misclassified point from

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})$$

and run a PLA iteration on it.



Theorem (Convergence)

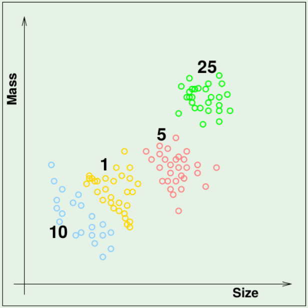
*If the data is linearly separable then the PLA will find a set of weights* $\mathbf{w}$ *that correctly classifies the training examples in a finite number of steps.*

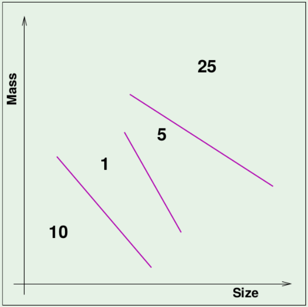
# Types of Learning

- Supervised Learning
- Unsupervised Learning
- Reinforced Learning

# Supervised Learning

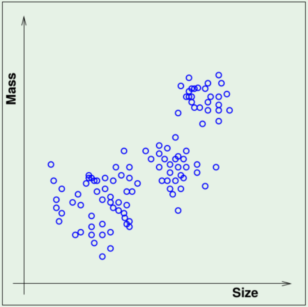
Example from vending machines – **coin recognition**



# Unsupervised Learning

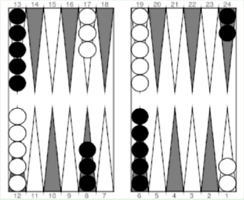
Instead of (input,correct output), we get (input, ? )



# Reinforced Learning

Instead of (input,correct output),

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




The world champion was a neural network!

# The Shape of Things to Come . . .

- 1 hr of theory + 1 hr of practicals each week
- Next week we start with linear regression.