# Learning From Data
# Lecture 1
# The Learning Problem



**M. Magdon-Ismail**
CSCI 4100/6100

## The Storyline

1. What is Learning?
2. Can We do it?
3. How to do it?
4. How to do it well?
5. General principles?
6. Advanced techniques.
7. Other Learning Paradigms.

concepts
theory
practice

our language will be mathematics . . .
. . . our sword will be computer algorithms

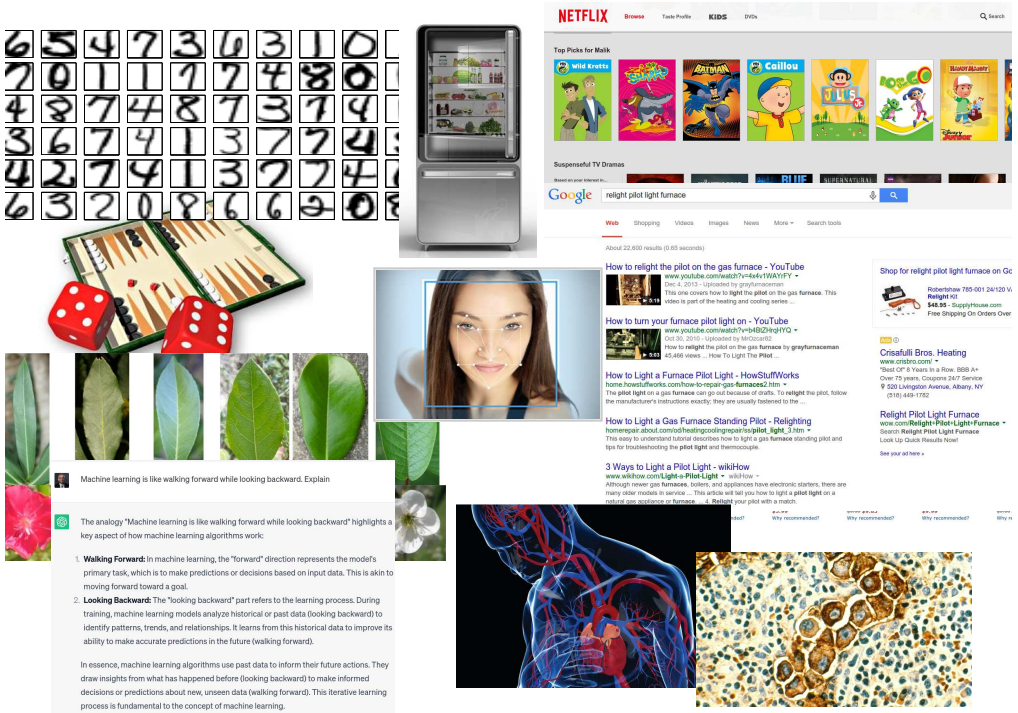


## Resources

1. Web Page: www.cs.rpi.edu/∼magdon/courses/learn.php
   - course info: www.cs.rpi.edu/∼magdon/courses/learn/info.pdf
   - slides: www.cs.rpi.edu/∼magdon/courses/learn/slides.html
   - assignments: www.cs.rpi.edu/∼magdon/courses/learn/assign.html
2. Text Book: *Learning From Data*
   Abu-Mostafa, Magdon-Ismail, Lin


3. Piazza
4. TAs.
5. Professor.
6. Prerequisites? assignment #0





## Let's *Define* a Tree?

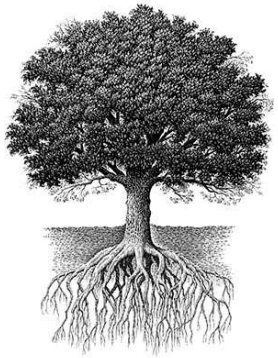


## Let's *Define* a Tree?

A brown *trunk* moving upwards and *branching* with *leaves* . . .



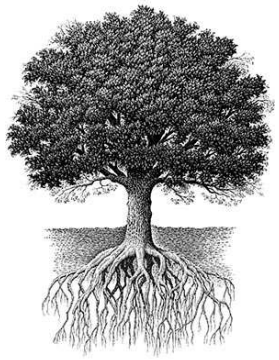
## Are These Trees?





## *Learning* "What are Trees" is 'Easy'

## Defining is Hard; Recognizing is Easy



Hard to give a complete mathematical definition of a tree.

Even a 3 year old can tell a tree from a non-tree.

The 3 year old has learned from data.

(Other tasks like graphics or GAN?)

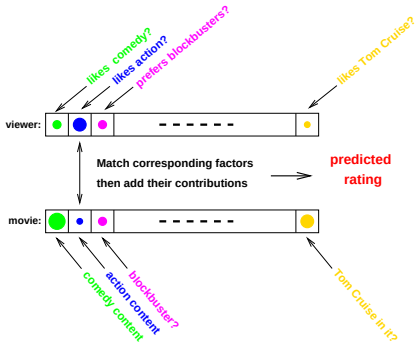


## Learning to Rate Movies

- Can we predict how a viewer would rate a movie?
- Why? So that Netflix can make better movie recommendations, and get more rentals.
- **$1 million** prize for a mere 10% improvement in their *recommendation system*.



## Previous Ratings Reflect Future Ratings




- Viewer taste & movie content imply viewer rating.
- No magical formula to predict viewer rating.
- Netflix has data. We can **learn** to identify movie "categories" as well as viewer "preferences"

Class Motto:

**A pattern exists. We don't know it. We have data to learn it.**



## Credit Approval

Let's use a conceptual example to crystallize the issues.

| | |
|---|---|
| age | 32 years |
| gender | male |
| salary | 40,000 |
| debt | 26,000 |
| years in job | 1 year |
| years at home | 3 years |
| ... | ... |

Approve for credit?

## Credit Approval

Let's use a conceptual example to crystallize the issues.

- Using salary, debt, years in residence, etc., approve for credit or not.
- No magic credit approval formula.
- Banks have lots of data.
  - customer information: salary, debt, etc.
  - whether or not they defaulted on their credit.

| | |
|---|---|
| age | 32 years |
| gender | male |
| salary | 40,000 |
| debt | 26,000 |
| years in job | 1 year |
| years at home | 3 years |
| ... | ... |

Approve for credit?

**A pattern exists. We don't know it. We have data to learn it.**



## The Key Players

- Salary, debt, years in residence, ... — *input* $\mathbf{x} \in \mathbb{R}^d = \mathcal{X}$.
- Approve credit or not — *output* $y \in \{-1, +1\} = \mathcal{Y}$.
- True relationship between $\mathbf{x}$ and $y$ — *target function* $f : \mathcal{X} \mapsto \mathcal{Y}$. **(The target $f$ is *unknown*.)**
- Data on customers — *data set* $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$. **($y_n = f(\mathbf{x}_n)$.)**

$\mathcal{X}$ $\mathcal{Y}$ and $\mathcal{D}$ are *given* by the learning problem;
The target $f$ is fixed but unknown.

We learn the function $f$ from the data $\mathcal{D}$.



## Learning

- Start with a set of candidate hypotheses $\mathcal{H}$ which you think are likely to represent $f$.
$$\mathcal{H} = \{h_1, h_2, \ldots, \}$$
is called the hypothesis set or *model*.
- Select a hypothesis $g$ from $\mathcal{H}$. The way we do this is called a *learning algorithm*.
- Use $g$ for new customers. We hope $g \approx f$.

$\mathcal{X}$ $\mathcal{Y}$ and $\mathcal{D}$ are *given* by the learning problem;
The target $f$ is **fixed but unknown**.

*We choose $\mathcal{H}$ and the learning algorithm*

This is a very general setup (eg. choose $\mathcal{H}$ to be all possible hypotheses)



## Summary of the Learning Setup