CS60050: Machine Learning

Autumn 2024

Sudeshna Sarkar

Introduction to ML

24 July 2024

Course Website

https://machine-learning-2024-cs60050.netlify.app/

Course Timings

- Wed 11-12 PM, Thu 12-1 PM, Fri 8-9 AM
- **Section 1 (Sudeshna Sarkar, NC 243)**
  - All PG students. UG students of EC, EE, MA
- Section 2 (Dr. Somak Aditya, NC442)
  - All UG students except EC, EE, MA

Contact

Email: sudeshna@cse.iitkgp.ac.in Subject “CS60050 ML24”

Course Evaluation Plan

Tentative

- Mid Term - 25%
- Final Exam - 40%
- Assignments (3-4) - 20%
- Class-Tests (Two) - 15%

Attendance Policy

Students must attend all classes.
Enter the class on time.

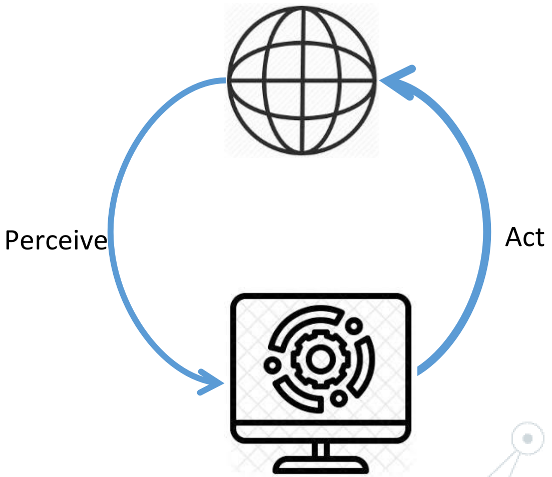
Attendance Policy: Attendance record will be maintained and uploaded on the website.

All students must maintain 75% attendance to continue the course.

What is Machine Learning?

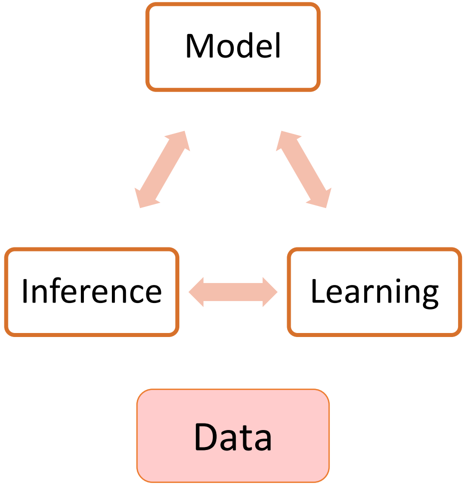
Artificial Intelligence

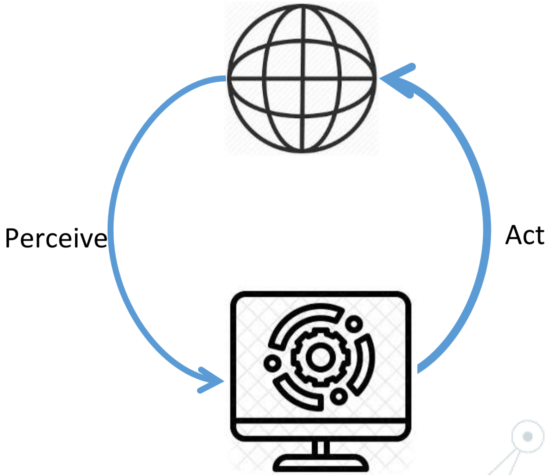
- The quest for building Intelligent Agents that can think and act rationally (or humanlike)
- Aspects of intelligent behaviour
  - **Learn (from experience)**
  - Adapt to new situations
  - Reason
  - Act intelligently



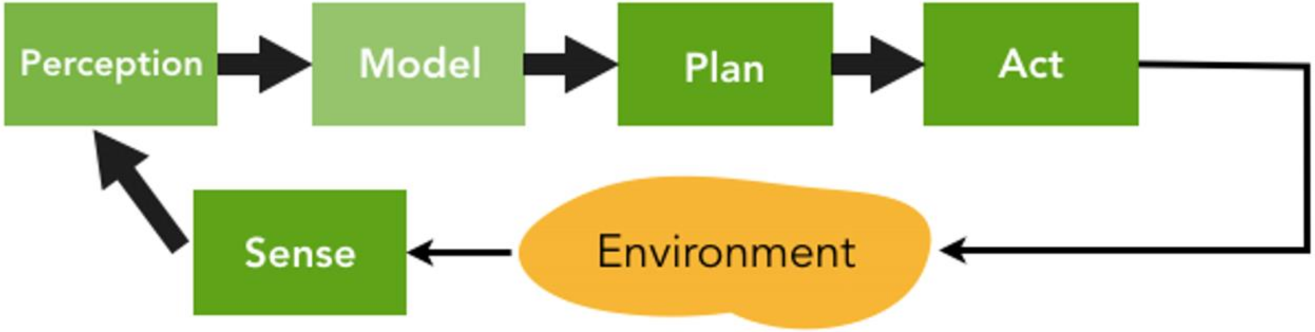
Artificial Intelligence

- The quest for building Intelligent Agents that can think and act rationally (or humanlike)

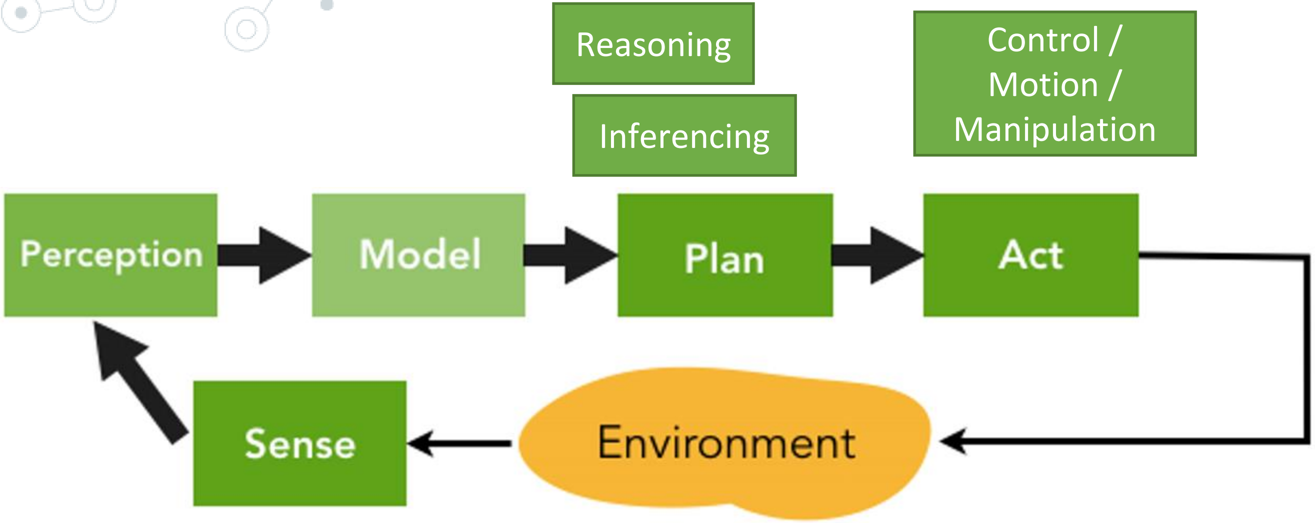




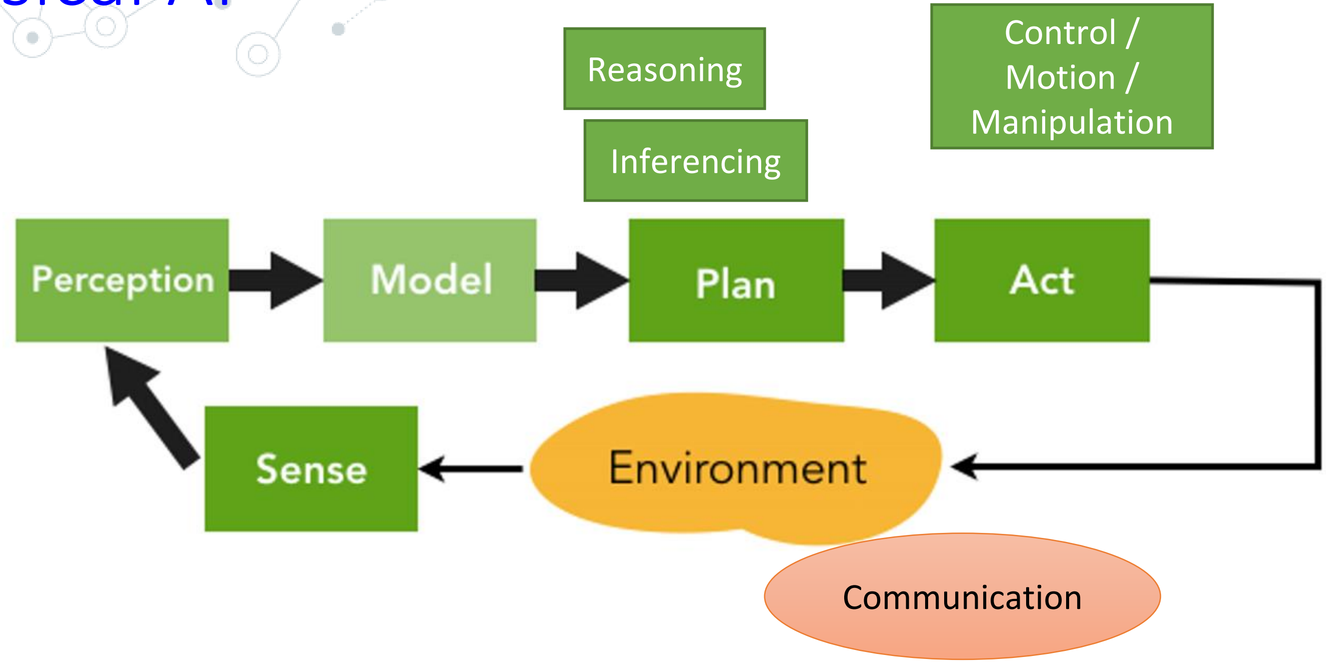
Classical AI



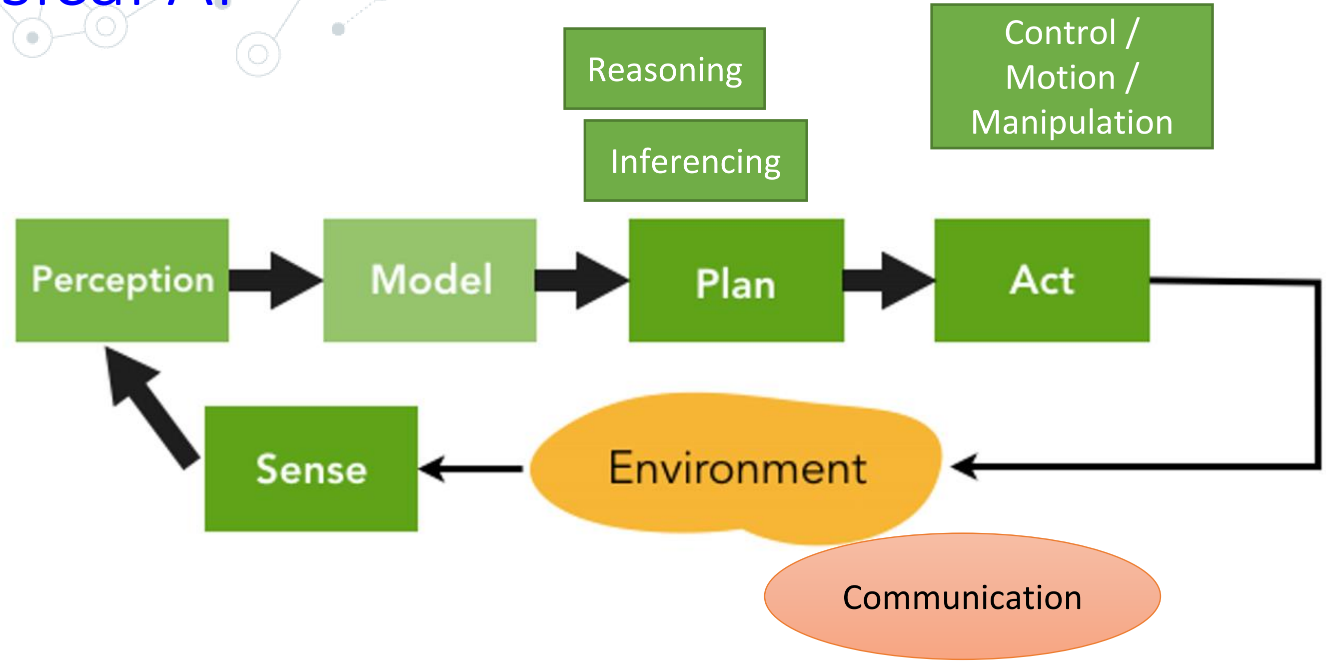
Classical AI



Classical AI



Classical AI



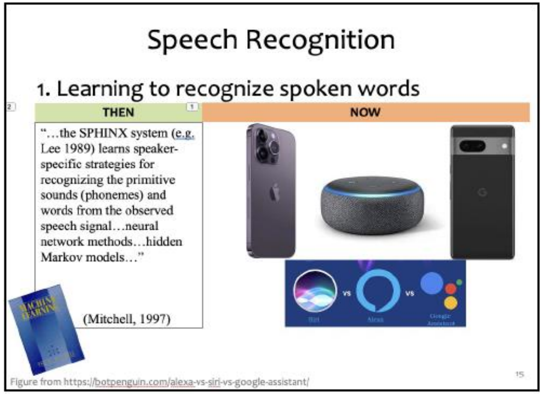
Learning

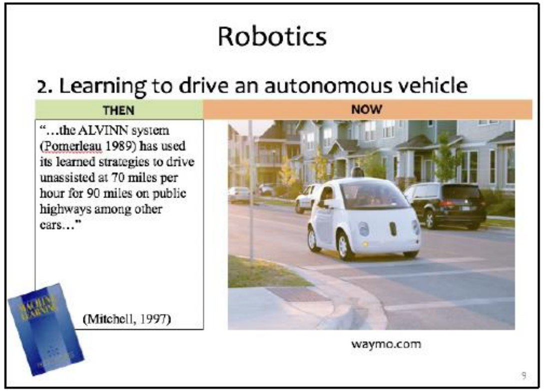
Data

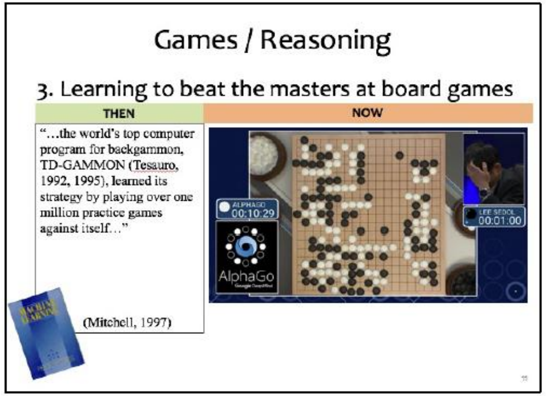
Machine Learning Toolbox

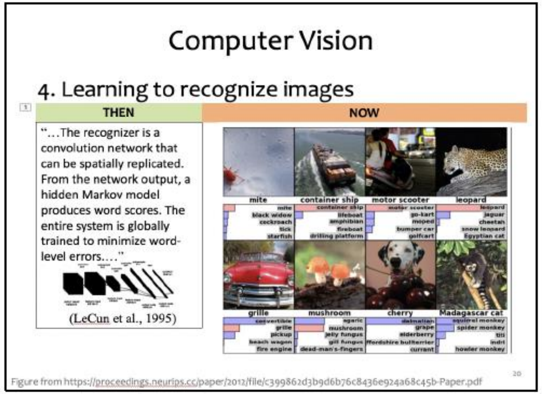
- Machine Learning
- Statistics
- Probability
- Computer Science
- Optimization

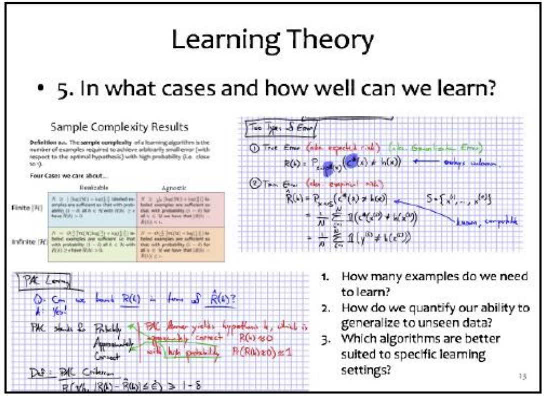
What is ML?











Slides from Dr. Matt Gormley (CMU)

Learning

Data → Model

- Requires a leap of faith: generalization

ML Big Picture

Learning Paradigms:
What data is available and when? What form of prediction?

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Problem Formulation:
What is the structure of our output prediction?

boolean	Binary Classification
categorical	Multiclass Classification
ordinal	Ordinal Classification
real	Regression
ordering	Ranking
multiple discrete	Structured Prediction
multiple continuous	(e.g. dynamical systems)
both discrete & cont.	(e.g. mixed graphical models)

Application Areas
Key challenges?
NLP, Speech, Computer Vision, Robotics, Medicine, Search

Theoretical Foundations:
What principles guide learning?

- ❑ probabilistic
- ❑ information theoretic
- ❑ evolutionary search
- ❑ ML as optimization

Facets of Building ML Systems:
How to build systems that are robust, efficient, adaptive, effective?

1. *Data prep*
2. *Model selection*
3. *Training (optimization / search)*
4. *Hyperparameter tuning on validation data*
5. *(Blind) Assessment on test data*

Big Ideas in ML:
Which are the ideas driving development of the field?

- *inductive bias*
- *generalization / overfitting*
- *bias-variance decomposition*
- *generative vs. discriminative*
- *deep nets, graphical models*
- *PAC learning*
- *distant rewards*

Slides from Dr. Matt Gormley (CMU)

CS60050: Machine Learning

Autumn 2024

Sudeshna Sarkar

Introduction to ML

25 July 2024

Well-Posed Learning Problem

Three components *<T,P,E>:*

1. *Task, T*
2. *Experience, E*
3. Performance measure, P

Definition of learning*:*

A computer program **learns** if its performance at task T, as measured by P, improves with experience E.

Well-Posed Learning Problem

Three components *<T,P,E>:*

1. *Task, T* The behaviour or task being improved
2. *Experience, E* *The experiences that are being used to improve performance in the task*
3. Performance measure, P For example: increasing accuracy in prediction, acquiring new, improved speed and efficiency

Definition of learning:

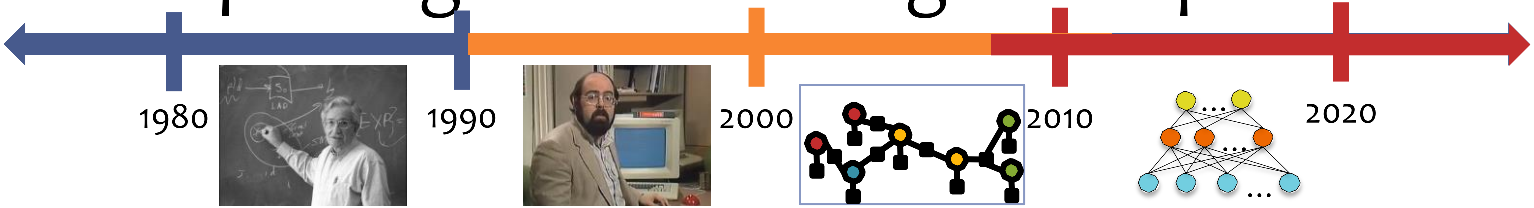
A computer program **learns** if its performance at task T, as measured by P, improves with experience E.

Example Learning Problem(s)

Learning to respond to voice commands (Siri)

- *1. Task, T:*
- *2. Performance measure, P:*
- *3. Experience, E:*

Capturing the Knowledge of Experts

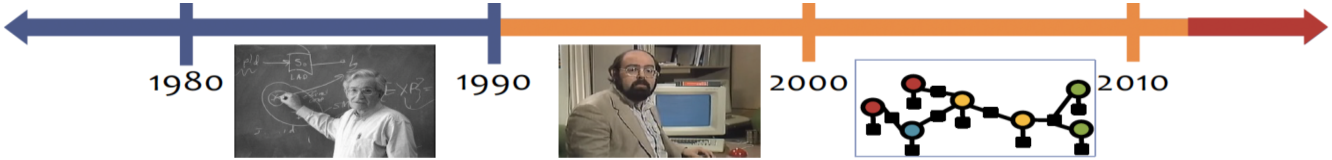


Solution #1: Expert Systems

- Over 20 years ago, we had rule-based systems:
  1. Put a bunch of linguists in a room
  2. Have them think about the structure of their native language and write down the rules they devise

Introspection…	**Rules…**
x = "Give me directions to Starbucks"	`if x matches "give me directions to Z": cmd = DIRECTIONS(here, nearest(Z))`
x = "How do I get to Starbucks?"	`if x matches "how do i get to Z": cmd = DIRECTIONS(here, nearest(Z))`
x = "Where is the nearest Starbucks?"	`if x matches "where is the nearest Z": cmd = DIRECTIONS(here, nearest(Z))`
x = "I need directions to Starbucks"	`if x matches "I need directions to Z": cmd = DIRECTIONS(here, nearest(Z))`
x = "Is there a Starbucks nearby?	`if x matches "Is there a Z nearby": cmd = DIRECTIONS(here, nearest(Z))`
x = "Starbucks now!"	`if x matches "Z now!": cmd = DIRECTIONS(here, nearest(Z))`

Capturing the Knowledge of Experts



Solution #2: Annotate Data and Learn

- Experts:
  - **Very good at** answering questions about specific cases
  - **Not very good at** telling **HOW** they do it
- 1990s: So why not just have them tell you what they do on **SPECIFIC CASES** and then let **MACHINE LEARNING** tell you how to come to the same decisions that they did

34

Slides from Dr. Matt Gormley (CMU)

Capturing the Knowledge of Experts

Solution #2: Annotate Data and Learn

1. Collect raw sentences $\{x^{(1)}, \ldots, x^{(n)}\}$
2. Experts annotate their meaning $\{y^{(1)}, \ldots, y^{(n)}\}$

$x^{(1)}$: How do I get to Starbucks?	
$y^{(1)}$: `directions(here, nearest(Starbucks))`	

$x^{(2)}$: Show me the closest Starbucks	
$y^{(2)}$: `map(nearest(Starbucks))`	

$x^{(3)}$: Send a text to John that I'll be late	
$y^{(3)}$: `txtmsg(John, I'll be late)`	

$x^{(4)}$: Set an alarm for seven in the morning	
$y^{(4)}$: `setalarm(7:00AM)`	

Slides from Dr. Matt Gormley (CMU)

Learning to respond to voice commands

1. Task, T: predicting action from speech

2. Performance measure, P: percent of correct actions taken in user pilot study

3. Experience, E: examples of (speech, action) pairs

Learning to approve loans

1. Task, T:

2. Performance measure, P:

3. Experience, E:

Slides from Dr. Matt Gormley (CMU)

Learning to approve loans

1. Task, T: decide whether to extend a loan to someone

2. Performance measure, P: minimize number of defaulters

3. Experience, E: Data from past loans, interview with loan officers

Slides from Dr. Matt Gormley (CMU)

Learning to approve loans

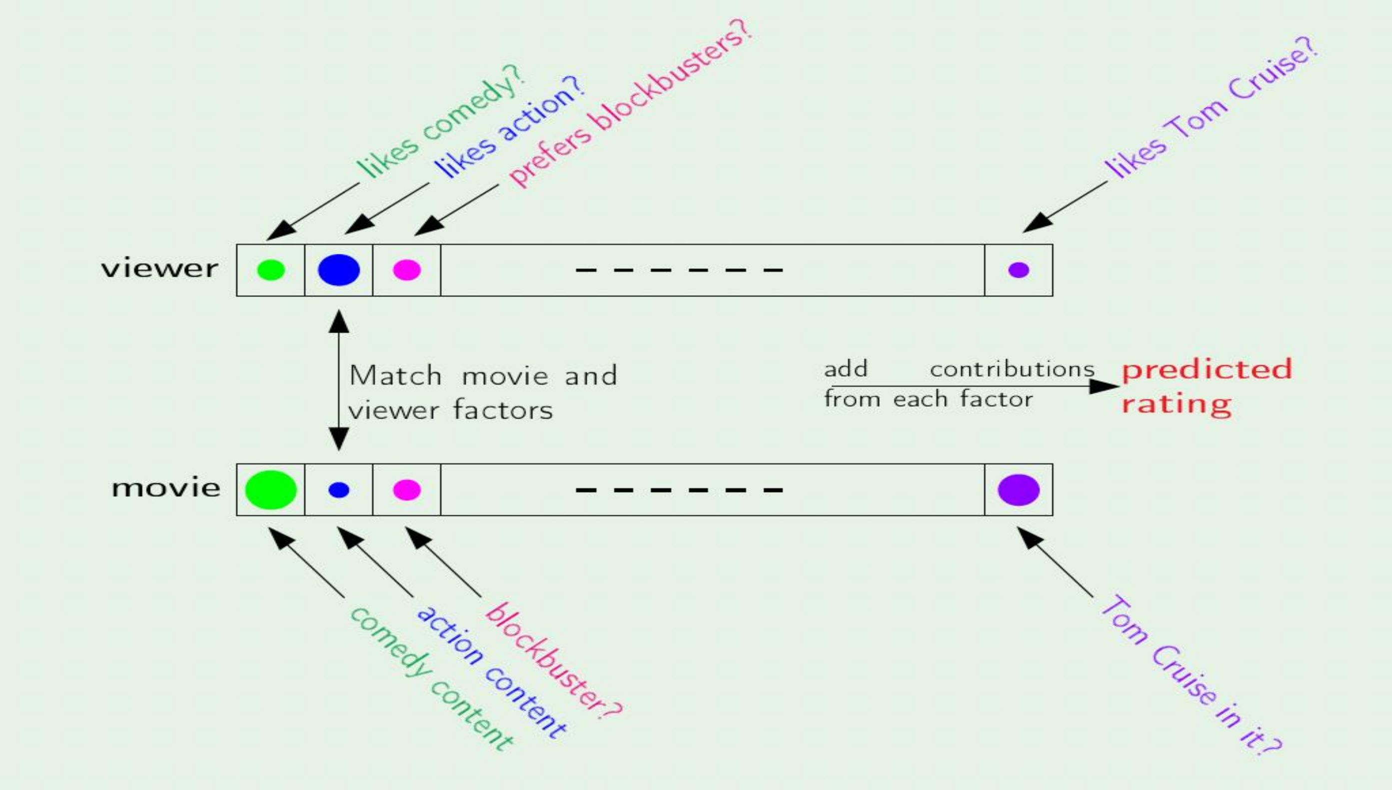
1. Task, T: predict the probability of someone defaulting on a loan

2. 2. Performance measure, P: Amount of interest earned over 10 years

3. Experience, E: Data from past loan applications and defaults

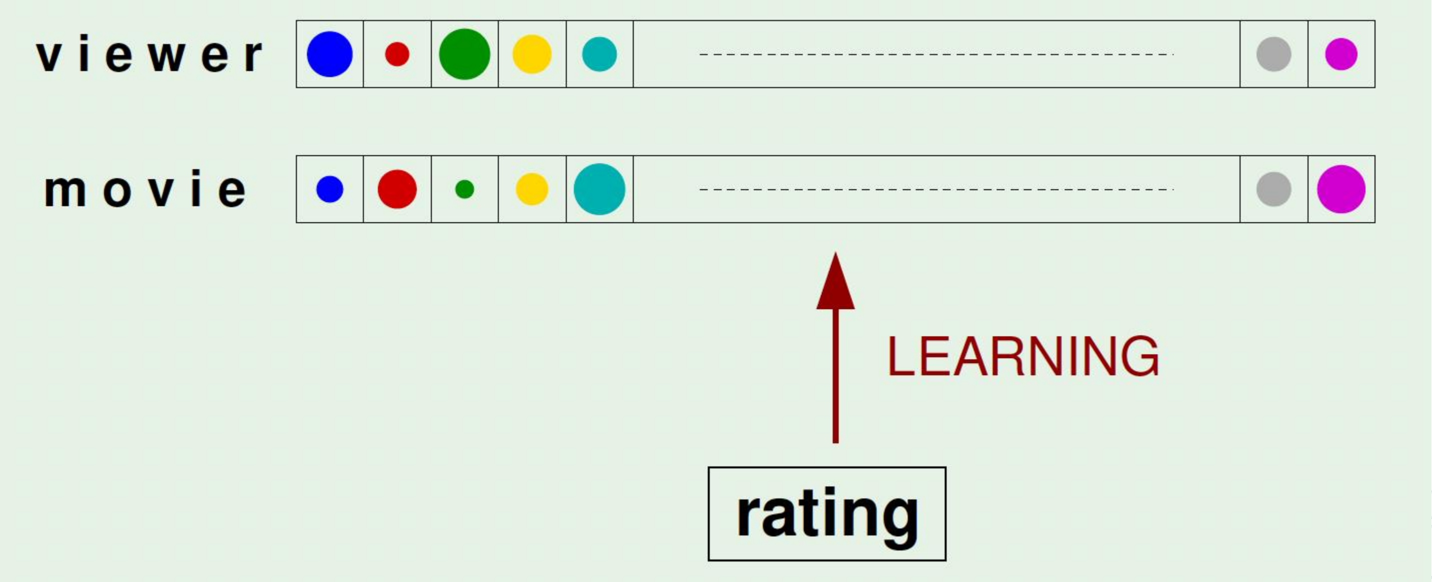
Slides from Dr. Matt Gormley (CMU)

Movie Rating – A Solution



Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

The Learning approach



Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

Components of a ML application

Representation

- Features: Data specification
- Function class: Model form

Optimization

- Model Training

Evaluation

- Performance measure

A. Representation of Data

How is the data specified?

A. Features

- Feature vector of n features

$$\bar{x} = (x_1, x_2, \ldots, x_n)$$

B. Convert input to a vector of basis functions

$$\left(\phi_0(\bar{x}), \phi_1(\bar{x}), \ldots, \phi_p(\bar{x})\right)$$

A microwave

Attributes:

- Volume: 17 l, 23 l, …
- Functions: Micro, Convection, ..
- Power level
- Accessories
- Type of dial
- Brand
- Warranty
- Price

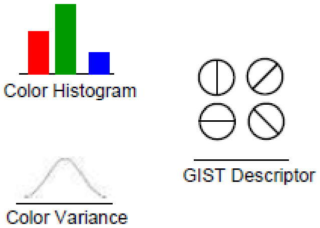
Image of shirt

- Collar style
- Sleeve type
- Colour
- …

Features

Image classification

- Raw pixels
- Histograms
- GIST descriptors



Product Rating (Webcam)

- Frame rate
- Resolution
- Autofocus
- Microphone
- Lens
- Brand

Bank Marketing Dataset

http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

Predict if the client will subscribe (yes/no) a term deposit (variable y).

Input variables:

 # bank client data:

1. age
2. type of job
3. marital status
4. education
5. has credit in default?
6. has housing loan?
7. has personal loan?

 # related with the last contact of the current campaign:

8. contact communication type ('cellular','telephone')
9. last contact date and duration

 # other attributes:

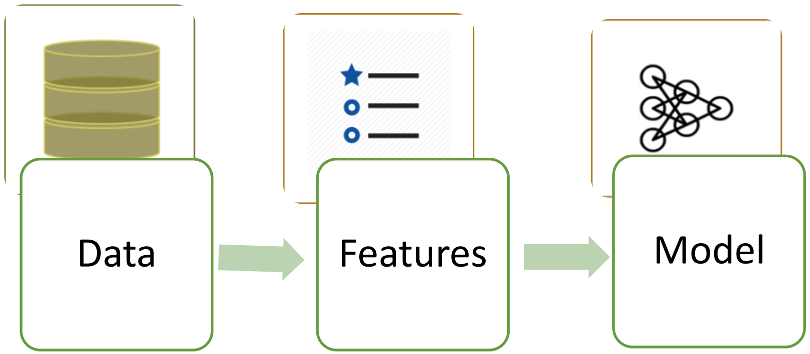
12. No of contacts performed for this client
13. No of days after client last contacted
14. No of contacts performed before this campaign and for this client
15. outcome of prev marketing campaign

 # social and economic context attributes

16. employment variation rate - quarterly indicator
17. consumer price index - monthly indicator
18. consumer confidence index - monthly indicator
19. euribor 3 month rate - daily indicator
20. number of employees - quarterly indicator

Feature Choice

- Input Data comprise features
  - Structured features (numerical or categorical values)
  - Unstructured (text, speech, image, video, etc)
- Use only relevant features
- Too many features?
  - Select feature subset (reduction)
  - Extract features: Transform features



B. Model Representation

- The richer the representation, the more useful it is for subsequent problem solving.
- The richer the representation, the more difficult it is to learn.

$$y = f(\bar{x})$$

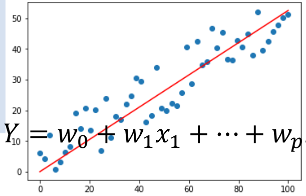
$$y = g(\bar{\phi}(\bar{x}))$$

- Linear function
- Decision Tree
- Graphical Model
- Neural Network

B. Model Representation
Hypothesis space

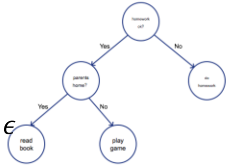
$$y = f(\bar{x})$$

Linear Function

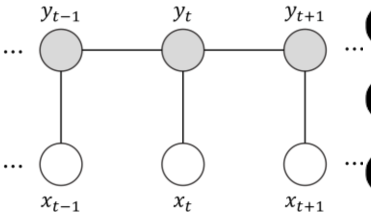


$$Y = w_0 + w_1 x_1 + \cdots + w_p x_p + \epsilon$$

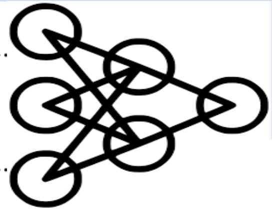
Decision Tree



Graphical Model



Neural Net



2. Evaluation

1. Accuracy $= \dfrac{\#\text{ correctly classified}}{\#\text{ all test examples}}$
2. Logarithmic Loss:

$$L_i = -\log(P(Y = y_i | X = x_i))$$

$$L = \sum_{c=1}^{M} y_{oc} \log(p_{oc})$$

3. Mean Squared error

$$MSE = \frac{1}{m}\sum\left(y_{pred} - y_{true}\right)^2$$

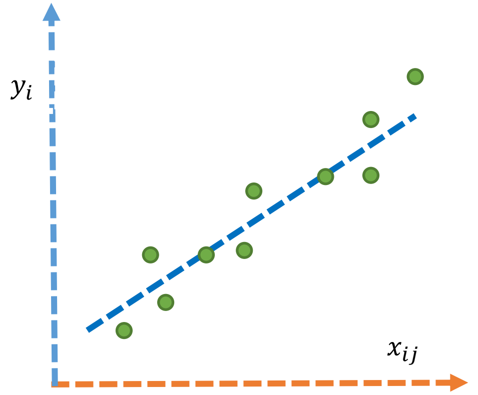
3. Optimization

- Define loss function
- Optimize loss function

- Stochastic Gradient Descent (Convex functions)
- Combinatorial optimization
  - E.g.: Greedy search
- Constrained optimization
  - E.g.: Linear programming

Elements of Optimization

1. Variables
2. Constraints
3. Objective Function



Simple Linear Regression

1. Variables: $w_0, w_1, \dots, w_n$
2. Constraints: none
3. Objective Function: Minimize

$$\sum_{i=1}^{m} \left(y_i \left(w_0 + \sum_{j=1}^{n} w_j x_{ij}\right)\right)^2$$

- m data points, n features
  - x_{ij}: jth attribute of ith instance
  - y_i: output of ith instance

Find coefficients $w_0, w_1, \dots, w_n$ to best fit data

- Task: Credit approval
- Applicant information:

Age	23 years
Gender	male
Annual salary	$30000
Years in residence	1 year
Years in job	1 year
Current debt	$15000
…	…

- Approve Credit?

Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

Components of Learning

Formalization

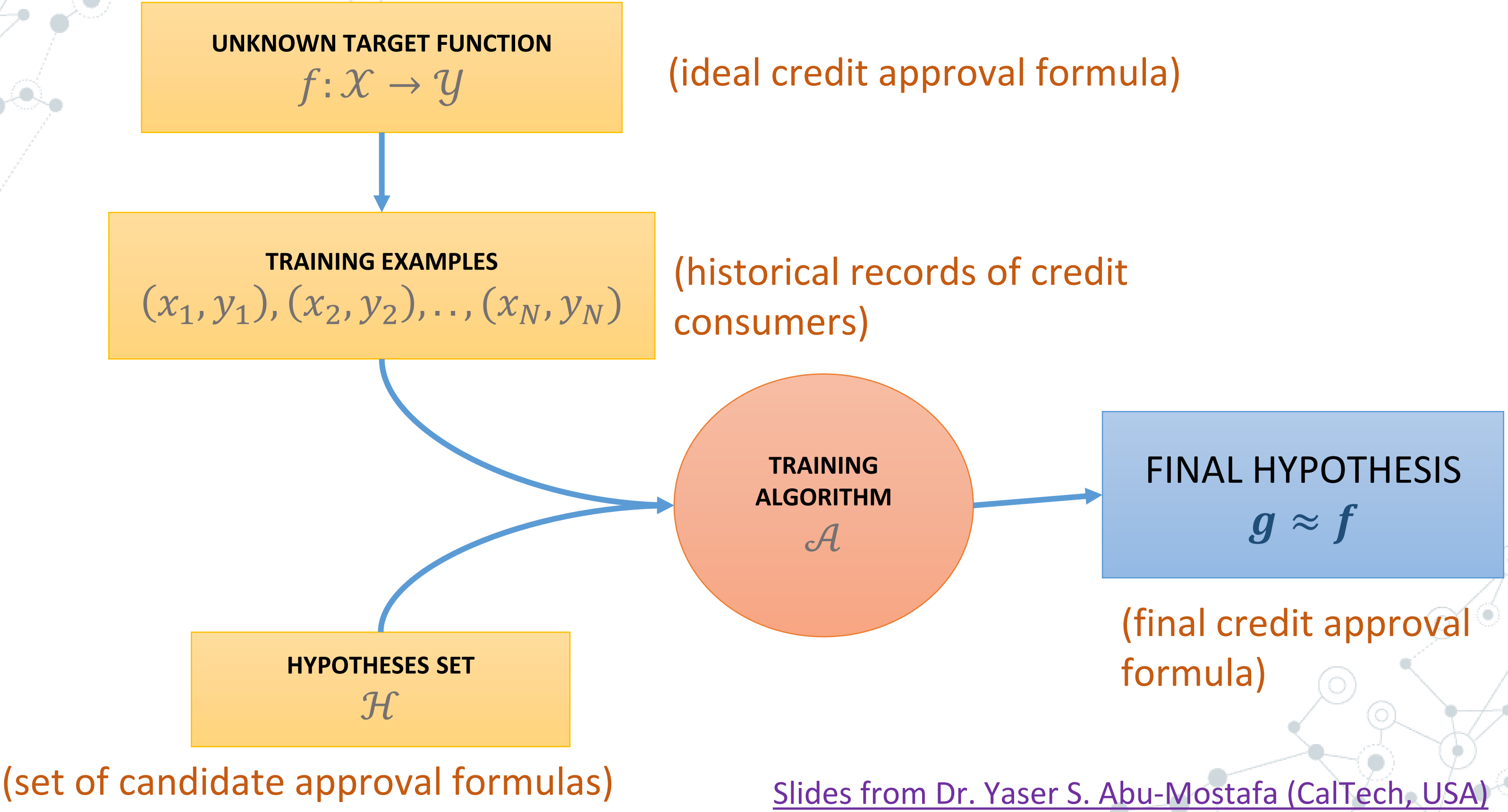
- Input x (customer application)
- Output y (good/bad customer?)
- Target function $f: \mathcal{X} \to \mathcal{Y}$ (ideal credit approval formula)
- Data $(x_1, y_1), (x_2, y_2), .., (x_N, y_N)$ (historical records)

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

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

Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

Components of Learning



Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

CS60050: Machine Learning

Autumn 2024

Sudeshna Sarkar

Introduction to ML

26 July 2024

Course Website

https://machine-learning-2024-cs60050.netlify.app/

1st Lab Session

Monday , 29 July

Tentative time: 6:30 pm

Exact Venue and Time To be announced

Contact

Email: sudeshna@cse.iitkgp.ac.in Subject “CS60050 ML24”

- Task: Credit approval
- Applicant information:

Age	23 years
Gender	male
Annual salary	$30000
Years in residence	1 year
Years in job	1 year
Current debt	$15000
…	…

- Approve Credit?

Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

Components of Learning

Formalization

- Input x (customer application)
- Output y (good/bad customer?)
- Target function $f: \mathcal{X} \to \mathcal{Y}$ (ideal credit approval formula)
- Data $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$ (historical records)

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

Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

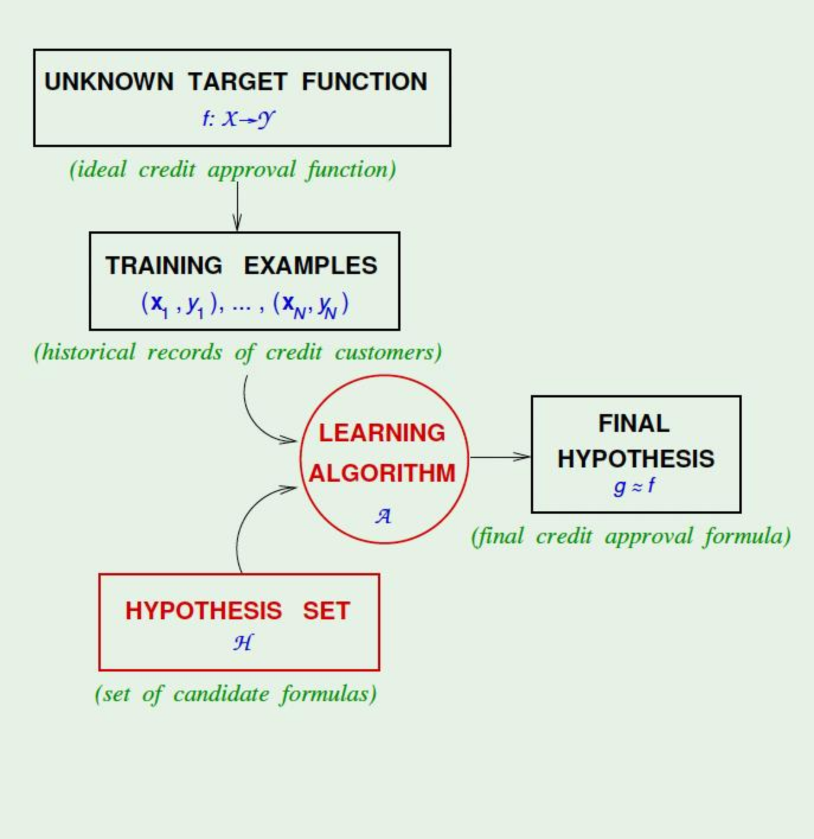
Solution Components

2 solution components of the learning problem

- The Hypothesis Set

  $\mathcal{H} = \{h\}, g \in \mathcal{H}$
- The Learning Algorithm

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



Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

A simple hypothesis set – the ‘perceptron’

For input $x = (x_1, \ldots, x_d)$ ‘attributes of a customer’

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

Deny credit if $\Sigma_{i=1}^{d} w_i x_i <$ threshold,

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

$$h(x) = sign(\Sigma_{i=1}^{d} w_i x_i \ - threshold)$$

Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

Separability

The perceptron implements

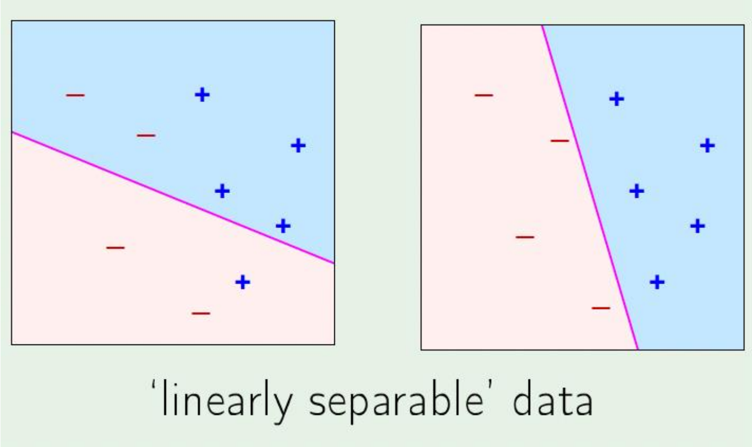
$$h(x) = sign(\Sigma_{i=1}^{d} w_i x_i + \quad w_0)$$

Introducing an artificial coordinate $x_0 = 1$

$$h(x) = sign(\Sigma_{i=0}^{d} w_i x_i)$$

In vector form, the perceptron implements

$$h(x) = sign(w^T x)$$



Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

A simple Learning Algo - PLA

The perceptron implements

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

Given the training set:

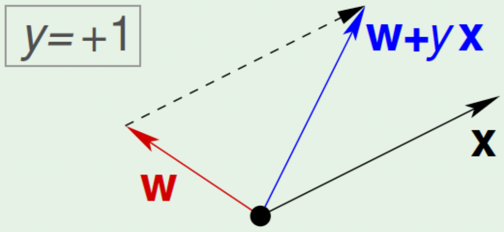
$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$$

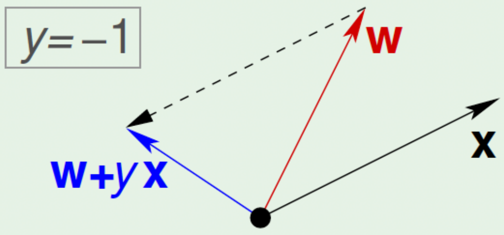
pick a misclassified point:

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

and update the weight vector:

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





Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

Iterations of PLA

- One iteration of the PLA:

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

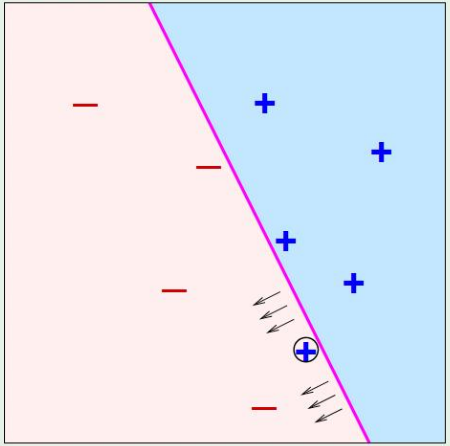
where $(\mathbf{x}, y)$ is a misclassified training point.

- 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!



Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

Broad types of machine learning

- Supervised Learning
  - Training Data with labels: X,y (pre-classified)
  - Given an observation x, what is the best label for y?
- Unsupervised learning
  - Training Data without labels: X
  - Given a set of x's, find hidden structure
- Semi-supervised Learning
  - Training Data + some Labels
- Reinforcement Learning
  - Given: observations and periodic rewards as the agent takes sequential action in an environment
  - Determine optimum policy

Supervised Learning

- Given data containing the inputs and outputs:

Training Data:

$$\{(\overline{x_1}, y_1), (\overline{x_2}, y_2), \ldots, (\overline{x_m}, y_m)\}$$

- Learn a function *f*(*x*) to predict *y* given *x*

$\bar{X}$	Y
$\overline{x_1}$	y_1
$\overline{x_2}$	y_2
…	..
$\overline{x_m}$	y_m

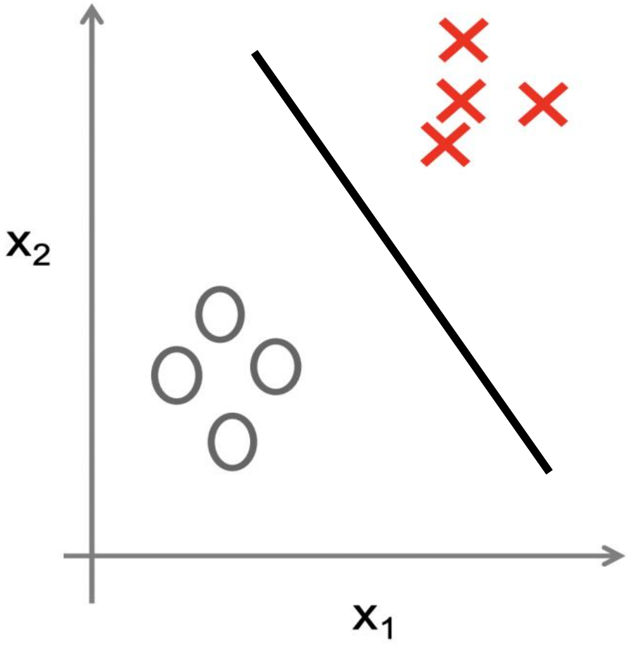
$\boxed{\text{Model}} \rightarrow f_\theta(\bar{x}) \rightarrow y$

Parameters

Training: Learn the model from the Training Data

Given Test instance $\overline{x'}$, predict $y' = f_\theta(\overline{x'})$

Supervised Learning



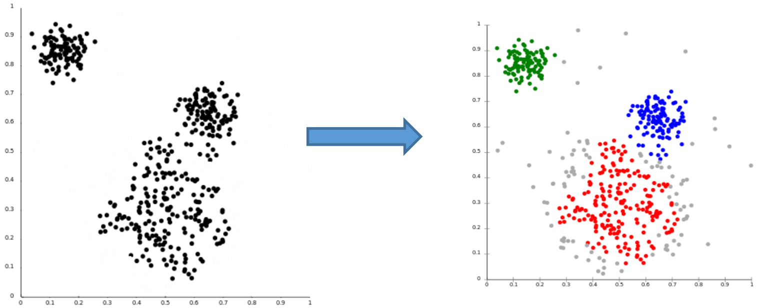
Classification (one example)

- Input: (input-features, correct output)
  - <size, #rooms>, <cheap/costly>
- Output of learning algorithm
  - Function maps features to output
  - F(<size, #rooms>) = cheap/costly

Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)

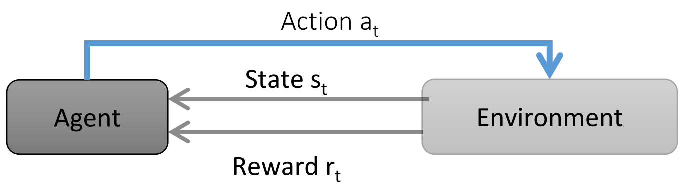
Unsupervised Learning (Clustering)

- Given $\{\overline{x_1}, \overline{x_2}, \ldots \overline{x_m}, \}$ without labels
- Find hidden structure in the data
  - Clustering
  - Dimensionality Reduction
- Clustering: Grouping similar objects



Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy.



  - Receive feedback in the form of rewards
  - Agent's utility is defined by the reward function
  - Must (learn to) act so as to maximize expected rewards

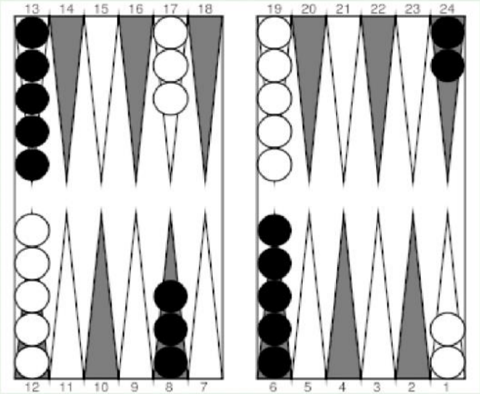
- Examples:
  - Game playing (Go)
  - Robot grasping
  - Controlling aircraft and robotic motion

Goal: Constantly learn to make 'optimal' predictions based on real-time feedback from past predictions

Reinforcement Learning

Instead of (input, correct output),

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



The world champion was a neural network!

Slides from Dr. Yaser S. Abu-Mostafa (CalTech, USA)