COMS 4252: Intro to Computational Learning Theory (CLT)

Overview of today's lecture:

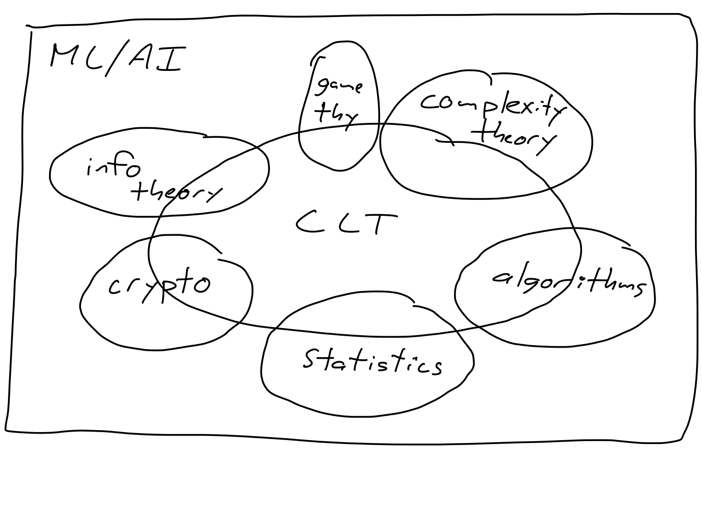
- introductions (me, you, topic)
- administrative overview (web page)
- high level overview of course → learning models
- some technical material: key basic notions + terminology

CLT: "machine learning from TCS point of view"

→ ML: developing programs that improve own performance; extracting info "automatically" from raw data; we don't do explicit programming.

→
- Complex software
- adaptive environments
- classification
- clustering
- prediction

:)



Admin: web page.

Communication:

- Tech. stuff : Ed Discussion
- Admin, only to course staff:
  coms4252columbiaf23@gmail.com

Course Overview

2 aspects to our CLT life:

1) defining/motivating (computation) learning models → rules of chess 1%

2) proving things about our learning models → learning how to play 99%
   (giving algs, hardness results for specific problems; developing general theory of "how the model works".)

What goes into def. of a learning model?

Learning model = the framework/rules of learning.

Specifies:

- Supervised / unsupervised?

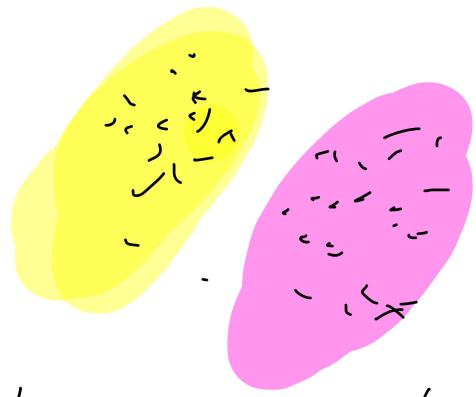
Supervised → labeled data;

$(x^1, f(x^1))$, $(x^2, f(x^2))$, ⋮ } given to learner, who tries to infer f

Unsupervised → no labels

$x^1, x^2, \ldots$ } clustering

This course:
supervised learning.



- What's being learned: skill, language, environment

Our focus: learning a (binary) classification rule, a.k.a. a Boolean function:

$$f : X \rightarrow \{0, 1\}$$

concepts → X; 0 → not elephant; 1 → elephant

"concept learning"

- How is info available to learner?
  - passive vs active models
    - passive: learner receives examples but doesn't select them
    - active: learner can actively make queries
      - "membership query"
      - "equivalence query"

teacher (helpful, slightly misinformed, adversarial)

Noisy/partial information?

- What prior knowledge does learner have?
  ↳ (assumptions)

↳ Should be only trying to learn "simple" things.

Our POV: we generally assume the unknown target concept has a prespecified simple syntactic representation (eg an AND of input variables).

- Constraints on learner? ← eff. algs.
  Us: polynomial-time algorithms that use poly(n) amount of data

- Performance criteria: what's success?
  - online vs offline learning
  - form of hypothesis
  - accuracy: statistical, # mistakes, etc.
  - comput. efficiency
  - data efficiency

Some models we'll analyze:

- Online mistake bound learning model (OLMB);
- Probably Approximately Correct (PAC);
- Statistical Query (SQ) model;
- Exact learning from queries

Things we'll do:

- Look at specific algs for specific problems
- General tech. for designing learning algs
- How much data do we need?
- Computational barriers: P vs NP, crypto, etc.
- Noise?
- Boosting accuracy?
- Compare / relate models

⋮

Church-Turing Thesis

Basic Notions + Terminology

X = domain for functions we're learning.

"instance space"

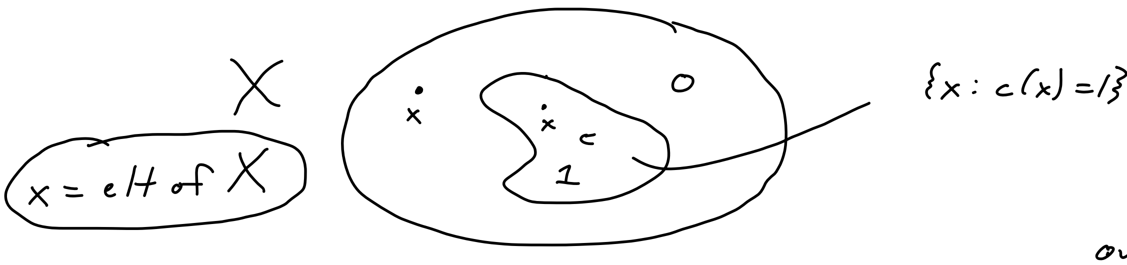
Ex: X = all cars in the world

For us, X usually $= \{0,1\}^n$

or $= \mathbb{R}^n$

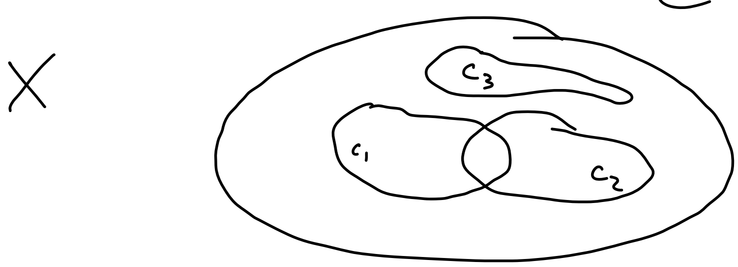
A concept is a subset $c \subseteq X$

or $\equiv$ly, a Bool fn $c : X \to \{0,1\}$.



A concept class $\mathcal{C}$ is a set of concepts over a fixed X.

$\mathcal{C} = \{c_1, c_2, c_3\}$



Basic idea of our learning models:

- Have a "known" X
  + a "known" $\mathcal{C}$
- There's an unknown target function $c \in \mathcal{C}$.

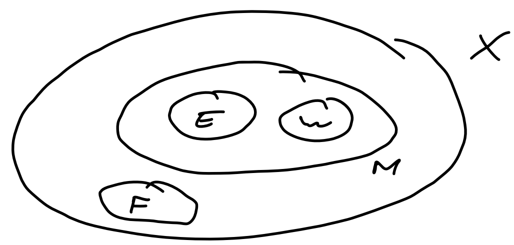
- We (learner) get some source of info about how c labels various x's in X.

From this info, learner's goal is to find (exactly or approximately) what c is.

Examples of c.c.'s $\mathcal{C}$:

- X = all animals

  $\mathcal{C}$ = {elephants, mammals, frogs, whales}



$x = (x_1, \ldots, x_n)$

- $X = \{0,1\}^n$ $\quad$ 0 = False, 1 = true

  disjunction

  $\mathcal{C}$ = all Boolean conjunctions over X.

A literal is x_i or $\overline{x_i}$ (negation)

disjunction $\quad$ OR $\quad$ $\vee$

A conjunction is an AND ($\wedge$) of literals, e.g.

$$c(x) = x_1 \wedge \overline{x_3} \wedge x_5 \wedge x_6 \wedge \overline{x_{11}}.$$

$$c(x) = x_2 \vee \overline{x_3} \vee x_4$$

$|\mathcal{C}| = 3^n$ $\qquad -, x_i, \overline{x_i}$.

- $X = \{0,1\}^n$, $\mathcal{C}$ = all monotone conj. (disj.)

no negations allowed

$c(x) = x_1 \wedge x_3 \wedge x_4 \wedge x_5$

$c(x) = x_2 \vee x_4 \vee x_5$

$|\mathcal{C}| = 2^n$

Next time:
- a few more example $\mathcal{C}$'s
- OLMB learning model
- algs for particular $\mathcal{C}$'s in (OLMB).

$x \in X$

$x = (x_1, \ldots, x_n)$