Overview of today's lecture:
- introductions (me, you, topic)
- admin aspects of course (webpage)
- high level overview of topic/course
- some technical content: key basic notions/terminology.

Intro to CLT: "machine learning from a CS theory perspective."
ML: goal is to develop programs that self-improve from experience/data.
- program is derived from data, not explicitly programmed.
Useful when
- we don’t know how to write program (complex software)
  - program must handle new/adaptive environments.
Applications of ML:
- clustering/classification: text categ, spam filtering, etc
- prediction:
  - complex software: self-driving cars, robot navigation, etc.
Back to CLT (course overview):
2 aspects to CLT:
A) developing/defining comp. models of learning
  (rules of game) 1% exploring
B) proving results for these models - “how the model works”
  (getting good at game) 99%

A learning model specifies the learning framework.
Specifies:
1) Supervised vs unsupervised
   labeled data \( (x, f(x)) \)
   \( \uparrow \) label
   \( \leftarrow \) no labels; clustering

This course: supervised.

2) What’s being learned?
   For us: binary classification rules
   \( f: X \rightarrow \{0,1\} \) “concept learning”
3) How does learner get its info?
   Passive: learner gets \( (x^{(1)}, f(x^{(1)})) \), \( (x^{(2)}, f(x^{(2)})) \), ...
   Active: learner chooses \( x \), given \( f(x) \)

Teacher: noisy info possible; incomplete/partial; malicious

4) What prior knowledge does learner have?
   Need to assume unknown concept not too complex.
   Our typical assumption: \( \mathcal{C} \) has a known and not too complex syntactic structure

5) Constraints on/resources of learner?
   For us: algorithms that have bounded (polynomial) runtime, data, etc.

6) Performance criteria: success?
   - online vs offline (batch)
   - hypothesis intelligibility
   - accuracy: error rate on test set
We'll discuss these learning models:

- Online mistake bound model
- Probably Approximately Correct (PAC) learning from random examples
- "Statistical Query" model
- Exact learning from queries.

We'll:

- describe & analyze particular learning algos in various models
- study general techniques for designing learning algos
- how much data do you need?
- computation barriers to eff. learning (P vs NP, crypto)
- learning w/ noise
- boosting accuracy of "weak" learners
- compare diff. learning models

**Basic Notions**

Notation: \( X = \text{domain for functions to be learned} \)

\( x, f(x) \)

\( x \in X : \text{an instance} \)
$X = \{ \text{all cars in the world} \}$

We consider encodings of objects.

Our $X$: typically

- $X = \{0,1\}^n$ (set of all $n$-bit strings)
  - $x_1 = 0/1$: red/not red
  - $x_2 = 0/1$: foreign/domestic etc.

or

- $X = \mathbb{R}^n$ (numerical)

A **concept** is a subset $c \subseteq X$.

- $c = \{ \text{all midsize cars} \}$

A **concept class** $\mathcal{C}$ is a set of concepts.

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

Basic idea of our learning models:

- We have a known concept class $\mathcal{C}$. 

...
There's an unknown target concept \( c \in C \).

- We have some source of info about how \( c \) labels instances \( x \in X \) (details: model-specific). Goal is to find/approximate \( c \).

Some examples of the kind of \( C \)'s we'll consider:

1. \( X = \{0,1\}^n \)  \( O = F \), \( I = T \)
   - \( C = \text{all Bool. conjunctions} \)
   - \( c(x) = \bigwedge \neg x_i \)
   - \( c(1,0,0,0,0) = 0 \).
   - \( |C| = 2^n \) (can't brute force...)

2. \( X = \{0,1\}^n \)  \( C = \text{all monotone conjunctions} \)
   - \( \text{mon conj: an AND of un-negated literals} \)
   - \( c(x) = x_2 \land x_4 \land x_5 \)
   - \( |C| = 2^n \) (empty conj: id. TRUE for \( I \).)
3. $X = \{0, 1\}^n$: $C = \{\text{all $s$-term DNF formulas}\}$
   DNF = disjunctive normal form; disj. of conj.

There are $\leq (3^n)^s$ many $s$-term ONFs.

4. $X = \{0, 1\}^*$: $C = \{\text{"$k$"-DNFs}\}$
   a $k$-DNF is an OR of any # of ANDs, AND
   but each AND has $\leq k$ vars.

$k = 2$:

Next time: one more ex. $C$
first learning model:
online nist. bound learning.