CS639: Introduction to computational learning theory	**Scribe:** Changho Shin
Lecture 1: Introduction to computational learning theory	**Lecturer:** Ilias Diakonikolas

1.1 What is Computational Learning Theory

Machine Learning from theory point of view. What is ML?

- Automatic extraction of useful information from "raw" data
- Programs which improve performance via interaction with data
- Examples
  - Classification/clustering: text categorization, fraud detection, web search
  - Prediction: weather, financial market, earthquakes
  - Writing complex software: driving a car, etc.

1.2 Overview of Computational Learning Theory

- Specifying learning models
- Proving results in these models
- A learning model should specify
  - Who is learning? $\rightarrow$ computer program, usually restricted:
    - polynomial learning time; $O(n^2), O(n \log n)$
    - small sample complexity, particular format for output hypothesis
  - What are we learning?
    - Skill (e.g. curve ball)
    - Environment (e.g. NYC)
    - In this class, we will focus on classification rules/Boolean functions
  - How does learner get information
    - Learner is given examples $(x, f(x))$
      - x: feature vector
      - $f(x)$: label
    - Passive learning (main focus of this lecture)
      - x can be chosen randomly from some distribution, or
      - some data $(x, f(x))$ can be chosen maliciously (corrupted data), or
      - x can be chosen by a helpful teacher
    - Active learning: learner can make guess

        - · Learner chooses x, get $f(x)$ (Membership query)
        - · Learner poses hypothesis, get counterexamples
        - · Others (subset query, etc)
  - Is information/data every noisy/incomplete?
    - A few bits of $(x, f(x))$
    - (x, y), where y might have some noise
  - What prior info does learner have?
    - Typical assumption: there is a prior representation scheme for the function being learned. (e.g. ”target concept” is known/assumed to be an ”And” function of features)
    - Performance criteria (Batch/offline vs online)
    - How do we measure accuracy of learner's hypothesis

- **The main focus in this class** is learning boolean functions (binary classification rules/concepts) "concept learning"

1.3 Topics in this class

1.3.1 Our models

- Online mistake-bound model
- Probably Approximately Correct (PAC) model - which is the standard model for statistical learning
- Query models (Statistical Query, Membership Query, Equivalance Query, etc.)

1.3.2 Topics in these models

- Describe and analyze specific learning algorithms
- General theory
  - Necessary, sufficient conditions for learning
  - Sample complexity: how much data is need to get a "pre-specified" accuracy (e.g. PAC learning: exact characterization)
- Computational issues in learning (e.g. computational hardness)
- Learning from noisy data
- Boosting
- Compare learning models (e.g. Statistical Query learning $\rightarrow$ PAC learning)

1.4 Basic terms and concepts

1.4.1 Basics

- X="instance space"=domain of the functions we've learning (e.g. $X = \{\text{all cars in the world}\}$)
- We want to learn an unknown "target function" (or concept)
  - $c : X \to \{0, 1\}$ (e.g. $c(x)$=1 if x is a mid-sized car or 0 if not)
- C : a "concept class", which is a set of concepts (=set of subset of X)
  - e.g. $C = \{c_1 = \{red\ cars\}, c_2 = \{convertibles\}, ...\}$

1.4.2 Basic idea of our learning models

- X and C are known to the learner
- There is a unknown target concept $c \in C$.
- Learner has some source of information about c, wants to identify/approximate c.
- Typically $X = \{0, 1\}^n$ or $X = \mathbb{R}^n$
- An $x \in X$ is $x = (x_1, ..., x_n)$, where each x_k are features
- Learner "knows" C and X, but does not know target concept $c \in C$
- Examples
  1. Monotone conjunctions
     - $X = \{0, 1\}^n$ Boolean hypercube
     - C is all monotone (no negation) conjunctions (AND) over $x_1, ..., x_n$ (e.g., $c(x) = x_3 \wedge x_5 \wedge x_6$, then if $n = 7$ and $x = 1100111$, $c(x) = 0$ (False))
     - $|C| = 2^n$
  2. Conjunctions
     - $X = \{0, 1\}^n$ Boolean hypercube
     - C is all conjunctions (AND) over $x_1, ..., x_n$ (e.g., $\bar{x_2} \wedge \bar{x_4} \wedge \bar{x_7} \wedge \bar{x_8}$)
     - $|C| = 3^n$
  3. Disjunctive Normal Form
     - $X = \{0, 1\}^n$ Boolean hypercube
     - C=all DNF formulas with $\leq n^2$ terms
     - A DNF = an OR of ANDs (e.g. $c(x) = (x_2 \wedge x_3) \vee (x_1 \wedge \bar{x_2} \wedge x_6) \vee (x_4 \wedge \bar{x_5} \wedge x_7 \wedge \bar{x_9})$)
  4. k-DNF
     - $X = \{0, 1\}^n$ Boolean hypercube

- C=all k-DNF formulas
  - A k-DNF is a DNF in which each term (=AND of literals has $\leq k$ variables
  - e.g. $c(x) = x_1\bar{x_2} \vee x_2\bar{x_6} \vee x_3\bar{x_5}$ is a 2-DNF
- A DNF = an OR of ANDs (e.g. $c(x) = (x_2 \wedge x_3) \vee (x_1 \wedge \bar{x_2} \wedge x_6) \vee (x_4 \wedge \bar{x_5} \wedge x_7 \wedge \bar{x_9}))$

5. Linear Threshold Functions (LTF)
   - $X = \mathbb{R}^n$
   - C=all LTF over $\mathbb{R}^n$ (LTF is also called halfspace)
   - A function $c : \mathbb{R}^n \rightarrow \{0, 1\}$ is LTF if there are n real weights $w_1, ..., w_n$ and a threshold θ s.t. for all $x = (x_1, ..., x_n) \in \mathbb{R}^n$

$$c(x) = \begin{cases} 1, \text{ if } w_1x_1 + ... + w_nx_n \geq \theta \\ 0, \text{ otherwise} \end{cases}$$

   - We also can study LTFs over $X = \{0, 1\}^n$, e.g.

$$MAJ(x) = \begin{cases} 1, , \text{ if } \sum_{i=1}^n x_i \geq \frac{n}{2} \\ 0, , \text{ otherwise} \end{cases}$$