Computational Learning Theory

Agam Goyal

January 28, 2023

Lec 01: Introduction

Computational learning theory studies machine learning from a theoretical computer science point of view. **Machine Learning:** Programs that enable automatic extraction of useful information from "raw" data, and which improve their performance via interaction with this data. Examples: Classification, Clustering, Prediction, etc.

Overview of Computational Learning Theory

A Learning Model should specify the following:

(1) "Who" is learning?

- A computer program, usually restricted
  - (i) Polynomial run-time
  - (ii) Small sample complexity
- Particular format for output hypothesis (for the interpretability of the model

(2) "What" are we learning?

- Skills and environment
- "Classification" rule: Boolean function

(3) "How" does the learner get information?

- Learner us given examples as point-label pairs: $(x, f(x))$
  - (i) "Passive" learning - only observes the information provided by the environment without influencing it
  - (ii) Assumptions about x:
    - Randomly chosen
    - Maliciously chosen (Adversary models)
    - Provided by teacher (Adversarial teacher model – worst-case scenario)
- Learner makes queries
  - (i) "Active" learning - interacts with the environment at training time, by posing queries or performing experiments

Is the data ever noisy or incomplete?

The data is usually almost always noisy and/or incomplete, and it is useful to study the robustness of learning models to these datasets.

What "prior rule" does the learner have?

The typical assumption is that there is an apriori representational scheme for the function being learned known as the **target concept**.

Performance Criterion

- How to measure accuracy?
- Online vs Offline/Batch Learning Methods

Lecture 02: Online Mistake-Bound learning Models

Setup

$\mathbf{X}$: instance space
c : concept (a Boolean function on X) — $c : \mathbf{X} \longrightarrow \{0, 1\}$
$\mathcal{C}$: Concept Class (set of all Boolean functions) — $c \in \mathcal{C}$

Typically, $\mathbf{X} = \{0, 1\}^n$ or $\mathbf{X} = \mathbb{R}^n$ and the learner knows $\mathcal{C}$ and $\mathbf{X}$, but doesn't know the individual concept target c which is a specific function.

Example #1

$\mathbf{X} = \{0, 1\}^n$

$\mathcal{C} =$ All monotone conjunctions over $x_1, \cdots, x_n$
A monotone conjunction is an AND of variables such that no variable in the conjunction is negated.
$|\mathcal{C}| = 2^n \longrightarrow$ All possible subsets

Example of a concept: $c(x) = x_3 \wedge x_5 \wedge x_6$

Example #2

$\mathbf{X} = \{0, 1\}^n$

$\mathcal{C} =$ All conjunctions over $x_1, \cdots, x_n$
$|\mathcal{C}| = 3^n \longrightarrow$ Each literal can exist as itself, its conjugate, or not exist

Example of a concept: $c(x) = \bar{x}_2 \wedge \bar{x}_4 \wedge x_7 \wedge \bar{x}_8$

Example #3

$\mathbf{X} = \{0, 1\}^n$

$\mathcal{C} =$ All DNFs over $x_1, \cdots, x_n$ with $\leq n^2$ terms
A Disjunctive Normal Form (DNF) is an OR of ANDs of literals.
$|\mathcal{C}| =$ Number of terms in the DNF formula

Example of a concept: $c(x) = (x_2 x_3) \vee (x_1 \bar{x}_2 x_4) \vee (\bar{x}_2 \bar{x}_4 \bar{x}_5 x_7) \vee x_7 \bar{x}_9$

We can extend this idea to a k–DNF: a DNF with $\leq k$ variables.

Example #4

$\mathbf{X} = \mathbb{R}^n$

$\mathcal{C} =$ All Linear Threshold Functions (LTFs / Halfspaces)
Example of a concept: $c : \mathbb{R}^n \longrightarrow \{0, 1\}$

$\exists\ n$ real weights $w_1, w_2, \cdots, w_n \in \mathbb{R}$ and $\theta \in \mathbb{R},\ s.t.\ \forall x \in \mathbb{R}^n,\ c(x) = \begin{cases} 1, & \text{if } \langle w, x \rangle \geq \theta \\ 0, & \text{otherwise} \end{cases}$

where w is the vector of weights $w_1, w_2, \cdots, w_n$, and $x \in \mathbf{X}$.

Online Mistake-Bound Learning Models

Algorithm 1 Online Mistake-Bound Learning Model

```
A learning session proceeds in a sequence of trials
Throughout the session, learner maintains hypothesis h : X ⟶ {0, 1}
while In a trial do:
    Learner is given some x ∈ X
    Learner outputs h(x) as prediction
    Learner is told the true c(x)
    if h(x) ≠ c(x) then
        Learner is charged for the mistake
    Learner may update h
```

Performance Measure = Number of Mistakes

Definition: Mistake Bound (M)

A learning algorithm A has a Mistake Bound M for a concept class $\mathcal{C}$ if

$$(\forall c \in \mathcal{C})\ (\forall \text{ sequence of examples from } \mathbf{X}) \quad A \text{ makes } \leq M \text{ mistakes.}$$

Observations:

- If $\mathbf{X}$ is finite, the learner can always achieve $M = |\mathbf{X}|$.
- If $\mathcal{C}$ is finite, the learner can always achieve $M = |\mathcal{C}| - 1$.

Example #1

$\mathbf{X} = \{0, 1, \cdots, 2^n - 1\}$
$\mathcal{C}$ = All initial intervals $[0, \alpha)$ where $\alpha \in \mathbb{Z}^+ \cup 0$
$|\mathcal{C}| = 2^n + 1 \longrightarrow$ Includes every possible integer from 0 to $2^n - 1$ and also the interval that even excludes 0.

Consider the scenario where we want to split the points in $\mathbf{X}$ into 0s and 1s or $\mathbf{X} \longrightarrow \{0, 1\}$. Everything part of the open interval is assigned a 0 and everything else is assigned a 1. Our goal is to find an initial interval which clearly separates these two classes.
Note that we can use binary search to learn $\mathcal{C}$ with $M = \log_2(2^n) \implies \boxed{M = n}$. Our initial hypothesis h in this case would just be the initial half interval $\{0, 1, \cdots, 2^{n-1}\}$. And then if the learner makes a mistake for some input x, we can update the interval accordingly. So essentially, each mistake eliminates $\geq$ Half of the remaining possibilities.

Example #2

$\mathbf{X} = [0, 1]$
$\mathcal{C}$ = All initial intervals $[0, \alpha)$ where $\alpha \in [0, 1]$

In this case, the mistake bound $\boxed{M = \infty}$ because of examples arbitrarily close to the boundaries we choose. So we may never really get to the *exact* answer we desire. However, in reality we would often just want to achieve a result that would be a *good enough* approximation.

Lecture 03: Online Learning of Disjunctions and Decision Lists