CSCI4230 Computatioal Learning Theory Spring 2019

Lecturer: Siu On Chan

Notes 1: Introduction; Learning Models

Textbook: *An Introduction to Computational Learning Theory*, Michael J. Kearns and Umesh V. Vazirani

This course (and notes) will mostly follow Servedio's, Diakonikalas', and Kanade's

Theory course — homeworks & exams about **proofs**; no programming

Pre-requiste: Discrete Math, Probability, math maturity

---

1. Introduction

This course focuses on (binary) classification problem in supervised learning

> Input: training samples $(x^1, y^1), \dots, (x^n, y^n)$
> Output: hypothesis $h \subseteq X$

x^i: an instance/features; $y^i \in \{0, 1\}$: category (instance result)

e.g. x^i are emails; $y^i \in \{\text{spam}, \text{not spam}\}$

e.g. x^i are documents; $y^i \in \{\text{English}, \text{not English}\}$

samples x^i belong to **instance space** X (typically $X = \{0, 1\}^n$ or $\mathbb{R}^n$)

assume samples are classified according to unknown **concept** $c \subseteq X$ i.e. $y^i = \mathbb{1}(x^i \in c)$

c belongs to known **concept class** $\mathcal{C}$ (some collection of subsets of X)

Want output hypothesis h to be close to unknown concept c

Will also think of c and h as $X \to \{0, 1\}$ (indicator functions)

---

2. Examples of problems (concept classes)

2.1. k-DNF (disjunctive normal form) formulae.

boolean variables $x_1, x_2, \dots, x_n$

literals $x_1, \overline{x}_1, x_2, \overline{x}_2, \dots, x_n, \overline{x}_n$ (a variable or its negation)

k-DNF formula: disjunction of terms, each term being conjuction of at most k literals

3-DNF e.g. $(x_1 \wedge \overline{x}_5 \wedge \overline{x}_9) \vee (\overline{x}_4 \wedge x_7 \wedge x_8)$

1-DNF (also called **disjunction**) e.g. $x_1 \vee \overline{x}_8 \vee \overline{x}_4 \vee x_2$

2.2. k-term DNF formulae.

k-term DNF formula: disjunction of k terms, each being conjuction of (any number of) literals

2-term DNF e.g. $(x_1 \wedge x_2 \wedge \overline{x}_3 \wedge x_4) \vee (\overline{x}_2 \wedge x_6 \wedge \overline{x}_7)$

2.3. k-CNF (conjuctive normal form) formulae.

k-CNF formula: conjunction of terms, each term being disjuction of at most k literals

1-CNF (also called **conjunction**) e.g. $x_1 \wedge \overline{x}_8 \wedge \overline{x}_4 \wedge x_2$

3-CNF e.g. $(x_1 \vee \overline{x}_5 \vee \overline{x}_9) \wedge (\overline{x}_4 \vee x_7 \vee x_8)$

Every k-term DNF is equivalent to k-CNF, because $\vee$ distributes over $\wedge$, i.e.

$$(u \wedge v) \vee (x \wedge y) = (u \vee x) \wedge (u \vee y) \wedge (v \vee x) \wedge (v \vee y)$$

But some k-CNF has no equivalent k-term DNF when $k \geqslant 2$

---

3. Overview of some models

3.1. Probably Approximately Correct (PAC) model.

Valiant'84 seminal paper *"A Theory of the Learnable"*

Assume instances x drawn from an unknown but fixed distribution D over X

Random instances, hence more realistic than worst case instances

3.2. PAC model with random noise.

Random classification noise: each sample's label y^i is corrupted independently with probability η, for some fixed $\eta > 0$

3.3. **Online model.**
Examples arrive online; classify each example before the next arrives
Sequence of examples may be worst case or random

3.4. **Active learning.**
Learning algorithm can choose example x and query $c(x)$

Questions we will ask:

(1) Given concept class $\mathcal{C}$, how many samples suffice to learn $c \in \mathcal{C}$?
    e.g. $\mathcal{C} = \{\text{conjunctions}\}$
(2) How many samples are needed?
(3) Given random samples, how to efficiently learn $c \in \mathcal{C}$?
    Even with enough samples to information-theoretically learn $c \in \mathcal{C}$, there may not be efficient algorithm