CMPUT 654 Fa 23: Theoretical Foundations of Machine Learning **Fall 2023**

Lecture 1: September 5

Lecturer: Csaba Szepesvári *Scribes: Zixin Zhong*

Note: *LaTeX template courtesy of UC Berkeley EECS dept. (link to directory)*
Disclaimer: *These notes have **not** been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

1.1 About this class

1. Textbook: 'Mathematical Analysis of Machine Learning Algorithms' by Tong Zhang (https://tongzhang-ml.org/lt-book/lt-book.pdf).
   This class will cover about one chapter per week up to the 'sequential learning' part.
2. The focus of this class will be **the statistical approach to supervised learning**.
3. Recordings of lectures: https://www.youtube.com/watch?v=arbGdCqn2Io

1.2 Supervised learning: a statistical approach

We start with a few definitions.

Definition 1.1 (Data set)**.** Let the data set $D_n = ((X_1, Y_1), \dots, (X_n, Y_n))$, where (X_i, Y_i)'s are i.i.d. random variables and $X_i \in \mathcal{X}$, $Y_i \in \mathcal{Y}$.

Remark 1.2 (Sets?)**.**

(a) D_n is actually a list of data, but we just say it is a 'set' for no good reason; the order of pairs can be important and there can be multiple identical pairs.

(b) An example of $\mathcal{X}$ and $\mathcal{Y}$ is $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$.

Definition 1.3 (Predictor, loss function, expected loss)**.** Fix a function $f : \mathcal{X} \to \mathcal{Y}$. The expected loss of f is

$$L(f) = \mathbb{E}[\ell(f(X), Y)]\,,$$

where (X, Y) has the same distribution as (X_1, Y_1).

Remark 1.4.

(a) We usually have loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$.

(b) The quality of a predictor is going to be measured by its expected loss.

(c) Data (X, Y) is identically distributed to the data we have during training. This can be thought of as the "test data".

Remark 1.5 (Probability spaces hiding in plain sight)**.** We should have started by saying: Let us fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. In what follows, all random variables, unless otherwise specified, will be above this space. We also use random variable in a liberal fashion: We actually mean random elements. In what follows, the expectation $\mathbb{E}[\cdot]$ will correspond to the probability measure $\mathbb{P}$.

Definition 1.6 (Pushforward distributions)**.** For a random variable $Z : \Omega \to \mathcal{Z}$, we use P_Z denote the push-forward of Z: If $\mathcal{F}_Z \subset 2^{\mathcal{Z}}$ is the sigma-algebra that makes $\mathcal{Z}$ a measurable space, for $A \in \mathcal{Z}$, $P_Z(A)$ is defined by

$$P_Z(A) = \mathbb{P}(Z^{-1}(A))(= \mathbb{P}(Z \in A)) \,, .$$

Definition 1.7. Let $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}}$ denote a learning algorithm. Let $f_n = \mathcal{A}(D_n)$ and $L(f_n) : \Omega \to \mathbb{R}$.

Remark 1.8.

(a) $\mathcal{A}$ takes a data set as input and returns predictions as output.

(b) $\mathcal{A}$ has to be a measurable function. We are going to assume that all functions are measurable.

(c) Since D_n is random, f_n is random.

Proposition 1.9. *Assume that (X, Y) is independent of D_n (Notation: $(X, Y) \perp D_n$). Then,*

$$\mathbb{E}[\ell\left(f_n(X), Y\right) | D_n] = L(f_n) \text{ w.p.1 and } \quad \mathbb{E}[\ell\left(f_n(X), Y\right)] = \mathbb{E}[L(f_n)]$$

Proof. (Homework #1) □

Let P denote $P_{X,Y}$ for brevity. We will call P a learning **instance**: P is what is unknown and it summarizes everything about the data.

Note that $L(f)$ depends on P. To denote this dependence, by abusing notation, we redefine L to mark this dependence:

$$L(P, f) = \mathbb{E}[\ell(f(X), Y)] \,.$$

Note that

$$L(P, f) = \int \ell(f(x), y)\, P(dx, dy) \,.$$

Definition 1.10 (Expected loss of an algorithm)**.** The expected loss of $\mathcal{A}$ on instance P is

$$L(P, \mathcal{A}) = \mathbb{E}[L(P, f_n)] \,.$$

The expected loss $L(\mathcal{A}, P) = \mathbb{E}[L_P(f_n)]$. Our **goal** is to find an algorithm $\mathcal{A}$ such that $L(\mathcal{A}, P)$ is small no matter what P is.

For every P, we try to find a predictor such that the loss is small. How closely can an algorithm approaches the minimum loss?

Homework #2: Is there a single algorithm $\mathcal{A}$ that minimizes $L(\mathcal{A}, P)$ for all P?

Discussion Why randomness?