Machine Learning: notes

Niccolò Simonato

October 13, 2022

Contents

Chapter 1

Introduction

1.1 Different kinds of learning

There are two main kinds of machine learning:

- **Supervised Learning**, where the training set is composed of labeled tuples (x_i, y_i), where y_i is the label.
- **Unsupervised Learning**, where the training set is composed just of unlabeled elements x_i.

Also, basing on the type of y_i, we have **Regression** (y_i is continuous) and **Classification** (y_i is discrete).

1.2 Recalls to Theory of Probability for Discrete Random Variables

1.2.1 Distributions and Probability

A probability space is a tuple (Z, F, D), where:

Definition 1. *Z is the **sample space**, the set of all possible outcomes of the random process modeled by the probability space.*

Definition 2. *F is the **family of allowable events**, a set of events $A \subseteq F$*

Definition 3. *D is the **Probability distribution**, a function $D : F \to [0,1]$ that:*

- $D[Z] = 1$
- *Given $E_1, E_2, \ldots$ the sequence of pairwise mutually disjoint events: $D[\bigcup_{i\geq 1} E_i] = \sum_{i\geq 1} D[E_i]$*

1.2.2 Notation

- We say that $z \sim D$ when $z \in Z$ is sampled according to D.
- Given a function $f : Z \to \{true, false\}$, the **probability of** $f(z)$ is

  Definition 4. $\mathbb{P}_{z\sim D}[f(z)] = D(\{z \in Z : f(z) = true\})$

- The **probability can be expressed with an event** $A \subseteq Z$.

  Definition 5. $A = \{z \in Z : \pi_A(z) = true\}$, *where* $\pi_a : Z \to \{true, false\}$.

- Then, $\mathbb{P}[A] = \mathbb{P}_{z\sim D}[\pi_A(z)] = D(A)$.

1.2.3 Independent Events

Definition 6. *Two events* A, B *are said to be independent when*
$\mathbb{P}[A \cap B] = \mathbb{P}[A] \cdot \mathbb{P}[B]$

1.2.4 Random Variables

Definition 7. *A random variable* $X(z)$ *is a function* $X : Z \to \mathbb{R}$.
A random variable can also be vectorial, when $X \in \mathbb{R}^n$.

Each random variable has a Probability Mass Function (PMF) and a Cumulative Distribution Function (CDF) associated:

Definition 8. *The Probability Mass Function of the r.v.* X *is defined as*
$p_X(x) = \mathbb{P}[X = x]$

Definition 9. *The Cumulative Distribution Function of the r.v.* X *is defined as*
$F_X(x) = \mathbb{P}[X \leq x]$
$= \sum_{k \leq x} p_X(k)$

Two random variables, such as two events, can be independent:

Definition 10. *Two random variables* X, Y *are said to be independent when*
$\mathbb{P}[(X = x) \cap (Y = y)] = \mathbb{P}[X = x] \cdot \mathbb{P}[Y = y]$, *for all the values of* x *and* y.

1.2.5 Expected Value and Moments

Definition 11. *The **Expected Value** of a random variable X is defined as* $E[X] = \sum_x x \cdot p_X(x)$.

Definition 12. *The **k-th Moment** of a random variable X is defined as* $E[X^k] = \sum_x x^k \cdot p_X(x)$.

Definition 13. *The **Variance** of a random variable X is defined as* $Var[X] = E[(X - \mu_x)^2] = E[X^2] - E[X]^2$.

Definition 14. *The **Covariance** of two random variables X, Y is defined as* $Cov(X, Y) = E[(X - \mu_x)(Y - \mu_y)]$.

Properties

- The Expected Value of a random variable is a linear operator: $E[aX + bY + c] = aE[X] + bE[Y] + c$.
- The Variance, though, is not linear: $Var[aX + b] = a^2 Var[X]$.
- Also, $Var[X + Y] = Var[X] + Var[Y] + 2Cov(X, Y)$.

1.2.6 Conditional Probability

Definition 15. *The probability of the event A given the event B is defined as* $\mathbb{P}[A|B] = \frac{\mathbb{P}[A \cap B]}{\mathbb{P}[B]}$.

1.2.7 Chebyshev's Inequality

Definition 16. *The Chebyshev's inequality is defined as follows:*
Given $E[X] = \mu, Var[X] = \sigma$, *then*
$\mathbb{P}[|X - \mu| > \epsilon] \leq \frac{\sigma^2}{\epsilon^2}$

1.2.8 Law of Large Numbers

Definition 17. $\lim_{n \to +\infty} \mathbb{P}[|\frac{1}{n} \sum (X_i - \mu)| > \epsilon] = 0$

1.2.9 Law of Total Probability

Definition 18. $\mathbb{P}[A] = \sum_{i=1}^{n} \mathbb{P}[A|C_i] \cdot \mathbb{P}[C_i]$,
Given $C_1, C_2, \ldots, C_n$ *a partition of* Ω.

Chapter 2

Learning Model

2.1 Definition

This section will define what kind of objects out *learner* as access to:

Definition 19. *The **Domain Set** X is the set of all possible objects to make predictions about.*
Usually, each object $x \in X$ is represented as a vector, in which each component is a feature.

Definition 20. *The **Label Set** Y is the finite set of possible labels.*

Definition 21. *The **Training Set** S is the finite set of tuples, that represent a subset of the labeled domain points.*
$S = ((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m))$

Definition 22. *The **Learner's output, or hypothesis** $h : X \to Y$ is the prediction rule. Given an element $x \in X$ it produces a label $y \in Y$.*

Definition 23. *The **Data-Generation model**:*
It is assumed that the data in X are produced by some unknown *probability distribution D.*
Also, it is assumed that exists an unknown ***labeling function*** *$f : X \to Y$ that predicts correctly the labels.*

Definition 24. *The **Measure of success** is the probability that the Learner does not predict the label correctly.*

2.2 Loss

Let's consider the set $A \subset X$, where $D(A)$ is the probability of observing a point $x \in A$.
Let $\pi : X \to \{0, 1\}$.

Let $A = \{x \in X : \pi(x) = 1\}$.
Then, $\mathbb{P}_{x\sim D}[\pi(x)] = D(A)$.

Definition 25. *The **Error of the prediction rule, or generalization error** is defined as:*
$L_{D,f}(h) = \mathbb{P}_{x\sim D}[h(x) \neq f(x)] = D(\{x : h(x) \neq f(x)\})$

To indicate a learner's prediction rule based on a training set S we'll use $h_S : X \rightarrow Y$.
It is now clear that, in order to succeed, it is necessary to find an h_S that minimizes $L_{D,f}(h)$.

2.3 Empirical Risk Minimization

As we have previously seen, we can minimize the generalization error in order to improve our learner.
There's just one problem: $L_{D,f}(h)$ is unknown. Therefore, we have to find a different measure.
Let's consider the **Training error**:

Definition 26. *The **Training error** is defined as follows:*
$L_S(h) = \frac{|\{i:h(x_i)\neq y_i, 1\leq i\leq m\}|}{m}$

The **Empirical Risk Minimization** aims to minimize L_S, in order to improve h.
The problem with this approach is that this can lead to **overfitting**: since the *generalization error* and the *empirical error* are not the same, we could obtain a model where the former is high and the latter is low.

2.4 Hypothesis Class

In order to prevent overfitting, we can apply ERM with a restricted set of **hypothesis**.
This means that we choose h from a defined set H.
In particular, we choose an h such that:
$ERM_H \in \arg\min_{h\in H} L_S(h)$.
Also, let H be a finite set, and h_S be the output of ERM_S, that is:
$h_S \in \arg\min_{h\in H} L_S(h)$.
In order for this to work, we have to make two assumptions:

- **Realizability**: there must be some $h^* \in H$ such that $L_{D,f}(h^*) = 0$
- **i.i.d**[1]: the examples in the training set must be independently and identically distributed (i.i.d) according to D.

[1] identically, indipendently distributed

2.5 PAC learning

If we consider the previously made assumptions, it is clear that will be very hard to create the perfect model.
This is the reason why this method is called **Probably Approximately Correct (PAC)** learning:

- The model can only be *approximately* correct;
- The model can only be *probably* correct.

This two degrees of correctness are defined by two parameters:

- The parameter ϵ is the **accuracy parameter**. $L_{D,f}(h_s) \leq \epsilon$
- The parameter δ is the **confidence parameter**. This represents the probability that h_S is not a good hypothesis, or that h_S is a good hypothesis with probability $1-\delta$.

Theorem 1. *Let H be a finite hypothesis class.*
Let $\epsilon, \delta \in (0,1)$ and $m \in \mathbb{N}$ such that:
$m \geq \frac{log(|H|/\delta)}{\epsilon}$
Then, for any f and any D for which the realizability assumption holds, with probability $\geq 1-\delta$ we have that, for every ERM hypothesis h_S, is true that:
$L_{D,f}(h_S) \leq \epsilon$.

Proof. □