Chapter 1

Machine Learning framework

1.1 A formal model

We begin with the description of a formal model in order to capture what could be the learning tasks. The fundamental points are:

- **The learner's input**:
  - A domain set $\mathcal{X}$, whose points are the istances we want to label.
  - A label set $\mathcal{Y}$.
  - The training dataset $S = \mathcal{X} \times \mathcal{Y}$. It is a finite sequence of label domain points.
- **The learner's output**:
  - A prediction rule $h : \mathcal{X} \to \mathcal{Y}$ (also called predictor or hyothesis or classifier). It is used to predict the label of new domain points. Therefore $A(S)$ is the hypothesis, where A represents the algorithm.
  - **Simple data-generation model**:
    Assume that the training data are generated by a probability distribution $\mathbb{D}$ (over $\mathcal{X}$). Moreover, suppose that the learner doesn't know anything about the distribution and that there exists some "correct" labeling function $f \ : \ \mathcal{X} \to \mathcal{Y}$ such that $y_i = f(x_i) \ \forall i$ (and it is unknown to the learner as well). So, in summary each S is generated by first sampling a point x_i according to $\mathcal{D}$ and then labeling it with the function f.
- **Measures of success**:
  A definition of "**Error of the classifier**" sohuld be introduced: it is the probability to draw a random instance x, according to D, such that $h(x) \neq f(x)$. Formally: $A \subset \mathcal{X} \implies \mathcal{D}(A)$ determines how likely it is to observe a $x \in A$, where:

  $$A = \{x \in \mathcal{X} \ : \ \pi(x) = 1\} \quad \text{with} \quad \pi \ : \ \mathcal{X} \to \{0, 1\}$$

  We refer to A as an event and therefore:

  $$\mathcal{D}(A) \equiv \mathbb{P}_{x \sim \mathcal{D}}[\pi(x)]$$

  The error of $h \ : \ \mathcal{X} \to \mathcal{Y}$ is finally:

  $$L_{\mathcal{D},f}(h) \equiv \mathbb{P}_{x \sim \mathcal{D}}[h(x) \neq f(x)] \equiv \mathcal{D}(x \ : \ h(x) \neq f(x))$$

  Note that $(\mathcal{D}, f)$ means that the error is computed with respect to the probability distribution $\mathcal{D}$ and the correct labeling function f. Moreover, remind that the learner is blind to the $\mathcal{D}$ and f.

1.2 Empirical Risk Minimization (ERM)

As mentioned before, a learning algorithm receives as input a training set S, sampled from an unknown distriibution D and labeled by some target function f, and should output a predictor $h_S : \mathcal{X} \to \mathcal{Y}$. The goal of the algorithm is to find h_S that minimizes the error with respect to the unknown D and f. Since the learner doesn't know what D and f are, the true error is not directly available to the learner. However, we can define the useful notion of **training error**, i.e. the error the classifier incurs over the training sample:

$$L_S(h) := \frac{|\{i \in [m] \ : \ h(x_i) \neq y_i\}|}{m} \tag{1.1}$$

where $[m] = \{1, \dots, m\}$. Note that the terms *empirical error* and *empirical risk* are often used with the same meaning to indicate Eq. 1.1.

1.2.1 Something may go wrong: Overfitting

This approach can fail miserably if naively used and lead to this phenomenon. Our aim is to find some conditions that guarantee the absence of overfitting.
Intuitively, overfitting occurs when our hypotesis fits the training data "too well" and so our algorithm won't have good performances on a new never-seen dataset.

1.2.2 Empirical Risk Minimization with inductive bias

The ERM rule might lead to overfitting, so we have to find a way to rectify it. In other words, we have to find some conditions that guarantee no overfitting. It means that if the algorithm has good performances with respect to the training data, it is also highly likely to perform well over the underlying data distribution.
A common solution is to apply the ERM learning rule over a restricted search space. Formally, the learner should choose a set of predictors $\mathcal{H}$ before seeing the data. This is a **hypotesis class**. Then, given a training sample S, the $\text{ERM}_{\mathcal{H}}$ learner uses the ERM rule to choose a predictor $h \in \mathcal{H}$ with the lowest possible error over S, which means mathematically:

$$\text{ERM}_{\mathcal{H}}(S) \in \underset{h \in \mathcal{H}}{\text{argmin}} \ L_S(h) \tag{1.2}$$

Such restrictions are often denoted as **inductive bias**.

Finite hypotesis classes

Now we consider a finite hypothesis class $\mathcal{H}$ and denote with h_S the result obtained applying $\text{ERM}_{\mathcal{H}}$ to S, namely Eq. 1.2. We also make the following simplifying assumption:

Definition 1.2.1 (The realizability assumption)**.** There exists $h^* \in \mathcal{H}$ such that $L_{(\mathcal{D},f)}(h^*) = 0$.

Note that this assumption implies that with probability 1 over random samples S we have $L_S(h^*) = 0$, where the istances of S are sampled according to $\mathcal{D}$ and are labeled by f.
Another assumption has to be done, presented in the following definition.

Definition 1.2.2 (Independently and identically distributed)**.** The examples in the training set are independently and identically distributed (i.i.d.) according to the distribution $\mathcal{D}$ if every x_i in S is freshly sampled according to $\mathcal{D}$ and then labeled

according to the labeling function f. We denote this assumption by $S \sim \mathcal{D}^m$, where m is the size of S and $\mathcal{D}^m$ denotes the probability over m-tuples induced by applying $\mathcal{D}$ to pick each element of the tuple independently of the other members of the tuple.

We want to give an upper bound of the probability to sample a m-tuple of istances that will lead to the failure of the learner. Therefore we consider the probability δ of getting a non representative sample S of the distribution. Through δ we define the **confidence parameter** $1-\delta$. We also introduce an **accuracy parameter** ε, with this meaning:

- $L_{(\mathcal{D},f)}(h_S) > \varepsilon \implies$ failure of the learner;
- $L_{(\mathcal{D},f)}(h_S) \leq \varepsilon \implies$ success of the learner.

So we would like to upper bound:

$$\mathcal{D}^m(\{S|_x \ : \ L_{(\mathcal{D},f)}(h_S) > \varepsilon\}) \tag{1.3}$$

For this reason we consider the set of "bad" hypotheses $\mathcal{H}_B$:

$$\mathcal{H} = \{h \in \mathcal{H} \ : \ L_{(\mathcal{D},f)}(h_S) > \varepsilon\} \tag{1.4}$$

In addition, we consider the set of misleading samples:

$$M = \{S|_x \ : \ \exists h \in \mathcal{H}_B, \ L_S(h) = 0\} \tag{1.5}$$

namely, $\forall S|_x \in M \ \exists h \in \mathcal{H}_B$ that looks like a "good" hypothesis on $S|_x$, which means:

$$\{S|_x \ : \ L_{(\mathcal{D},f)}(h_S) > \varepsilon\} \subseteq M \implies M = \bigcup_{h \in \mathcal{H}_B} \{S|_x \ : \ L_S(h) = 0\} \tag{1.6}$$

Hence:

$$\mathcal{D}^m(\{S_x \ : \ L_{(\mathcal{D},f)}(h_S) > \varepsilon\}) \leq \mathcal{D}^m(M) = \mathcal{D}^m\left(\bigcup_{h \in \mathcal{H}_B} \{S|_x \ : \ L_S(h) = 0\}\right) \tag{1.7}$$

Next, we upper bound the right-hand side of Eq. 1.7 using the following lemma, derived from basic properties of probability.

Lemma 1.2.1 (Union Bound)**.** *For any two sets A, B and a distribution $\mathcal{D}$ we have:*

$$\mathcal{D}(A \cup B) \leq \mathcal{D}(A) + \mathcal{D}(B) \tag{1.8}$$

Applying Lemma 1.2.1 to the right-hand side of Eq. 1.7 yields:

$$\mathcal{D}^m(\{S_x \ : \ L_{(\mathcal{D},f)}(h_S) > \varepsilon\}) \leq \sum_{h \in \mathcal{H}_B} \mathcal{D}^m(\{S|_x \ : \ L_S(h) = 0\}) \tag{1.9}$$

The next step is to bound each summand of the right-hand side of Ineq. 1.9. Let's fix some "bad" hypothesis $h \in \mathcal{H}_B$. The event $L_S(h) = 0$ is equivalent to the event $\forall i, \ h(x_i) = f(x_i)$. Since the examples in the training dataset are sampled i.i.d. we get:

$$\mathcal{D}^m(\{S|_x \ : \ L_S(h) = 0\}) = \mathcal{D}^m(\{S|_x \ : \ \forall i, \ h(x_i) = f(x_i)\}) \tag{1.10}$$

$$= \prod_{i=1}^{m} \mathcal{D}(\{x_i \ : \ h(x_i) = f(x_i)\}) \tag{1.11}$$

Lastly, for each individual sampling of an element of the training set we have:

$$\mathcal{D}(\{x_i \ : \ h(x_i) = y_i\}) = 1 - L_{(\mathcal{D},f)}(h) \leq 1 - \varepsilon \tag{1.12}$$

The last inequality follows from the fact that $h \in \mathcal{H}_B$. So we combine Eq. 1.12 with Eq. 1.11 and we use the inequality $1 - \varepsilon \leq e^{-\varepsilon}$ to get $\forall h \in \mathcal{H}_B$:

$$\mathcal{D}^m(\{S|_x \ : \ L_S(h) = 0\}) \leq (1 - \varepsilon)^m \leq e^{-\varepsilon m} \tag{1.13}$$

Combining Eq. 1.13 with Eq. 1.9 we conclude that:

$$\mathcal{D}^m(\{S_x \ : \ L_{(\mathcal{D},f)}(h_S) > \varepsilon\}) \leq |\mathcal{H}_B| e^{-\varepsilon m} \leq |\mathcal{H}| e^{-\varepsilon m} \tag{1.14}$$

Finally, we get the following useful corollary.

Corollary 1.2.1.1. *Let $\mathcal{H}$ be a finite hypothesis class. Let $\delta \in (0,1)$ and $\varepsilon > 0$ and let m be an integer that satisfies:*

$$m \geq \frac{\log(|\mathcal{H}|/\delta)}{\varepsilon}$$

Then, for any labeling function, f, and for any distribution, $\mathcal{D}$, for which the realizability assumption holds (for some $h \in \mathcal{H}$, $L_{(\mathcal{H},f)}(h) = 0$), with probability of at least $1 - \delta$ over the choice of an i.i.d. sample S of size m, we have that for every ERM hypothesis, h_S, it holds that:

$$L_{(\mathcal{H},f)}(h_S) \leq \varepsilon$$

The Corollary 1.2.1.1 tells us that for a sufficiently large m, the $\text{ERM}_{\mathcal{H}}$ rule over a finite hypotesis class will be *probably* (with confidence $1 - \delta$) *approximately* (up to an error of ε) correct.