

## Handout 0: Learning problem: Definitions, notation, and formulation –A recap

Lecturer &amp; author: Georgios P. Karagiannis

georgios.karagiannis@durham.ac.uk

---

**Aim.** To get some definitions and set-up about the learning procedure; essentially to formalize what introduced in term 1.

---

### Reading list & references:

- Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer.
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

### 1. INTRODUCTIONS AND LOOSE DEFINITIONS

**Pattern recognition** is the automated discovery of patterns and regularities in data  $z \in \mathcal{Z}$ . **Machine learning (ML)** are statistical procedures for building and understanding probabilistic methods that 'learn'. **ML algorithms**  $\mathfrak{A}$  build a (probabilistic/deterministic) model able to make predictions or decisions with minimum human interference and can be used for pattern recognition. **Learning** (or training, estimation) is called the procedure where the ML model is tuned. **Training data** (or observations, sample data set, exemplars) is a set of observables  $\{z_i \in \mathcal{Z}\}$  used to tune the parameters of the ML model. **Test set** is a set of available examples/observables  $\{z'_i\}$  (different than the training data) used to verify the performance of the ML model for a given a measure of success. **Measure of success** (or performance) is a quantity that indicates how bad the corresponding ML model or Algorithm performs (eg quantifies the failure/error), and can also be used for comparisons among different ML models; eg, **Risk function** or **Empirical Risk Function**. Two main problems in ML are the supervised learning (we focus here) and the unsupervised learning.

**Supervised learning** problems involve applications where the training data  $z \in \mathcal{Z}$  comprises examples of the input vectors  $x \in \mathcal{X}$  along with their corresponding target vectors  $y \in \mathcal{Y}$ ; i.e.  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ . **Classification problems** are those which aim to assign each input vector  $x$  to one of a finite number of discrete categories of  $y$ . **Regression problems** are those where the output  $y$  consists of one or more continuous variables. All in all, the learner wishes to recover an unknown pattern (i.e. functional relationship) between components  $x \in \mathcal{X}$  that serves as inputs and components  $y \in \mathcal{Y}$  that act as outputs; i.e.  $x \mapsto y$ . Hence,  $\mathcal{X}$  is the input domain, and  $\mathcal{Y}$  is the output domain. The goal of learning is to discover a function which predicts  $y \in \mathcal{Y}$  from  $x \in \mathcal{X}$ .

**Unsupervised learning** problems involve applications where the training data  $z \in \mathcal{Z}$  consists of a set of input vectors  $x \in \mathcal{X}$  without any corresponding target values ; i.e.  $\mathcal{Z} = \mathcal{X}$ . In clustering the goal is to discover groups of similar examples within the data of it is to discover groups of similar examples within the data.

## 2. (LOOSE) NOTATION IN LEARNING

**Definition 1.** The learner's output is a function,  $h : \mathcal{X} \rightarrow \mathcal{Y}$  which predicts  $y \in \mathcal{Y}$  from  $x \in \mathcal{X}$ . It is also called hypothesis, prediction rule, predictor, or classifier.

*Notation 2.* We often denote the set of hypothesis as  $\mathcal{H}$  ; i.e.  $h \in \mathcal{H}$ .

**Definition 3.** Training data set  $\mathcal{S}$  of size  $m$  is any finite sequence of pairs  $((x_i, y_i) ; i = 1, \dots, m)$  in  $\mathcal{X} \times \mathcal{Y}$ . This is the information that the learner has assess.

**Definition 4.** Data generation model  $g(\cdot)$  is the probability distribution over  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ , unknown to the learner that has generated the data.

**Definition 5.** We denote as  $\mathfrak{A}(\mathcal{S})$  the hypothesis (outcome) that a learning algorithm  $\mathfrak{A}$  returns given training sample  $\mathcal{S}$ .

**Definition 6.** (Loss function) Given any set of hypothesis  $\mathcal{H}$  and some domain  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ , a loss function  $\ell(\cdot)$  is any function  $\ell : \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}_+$ . The purpose of loss function  $\ell(h, z)$  is to quantify the “error” for a given hypothesis  $h$  and example  $z$  –the greater the error the greater its value of the loss.

**Example 7.** In binary classification problems where  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$  and  $\mathcal{Y} = \{0, 1\}$  is discrete, a loss function can be

$$\ell_{0-1}(h, (x, y)) = 1(h(x) \neq y),$$

**Example 8.** In regression problems  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$  and  $\mathcal{Y}$  is uncountable, a loss function can be

$$\ell_{\text{sq}}(h, (x, y)) = (h(x) - y)^2$$

**Definition 9.** (Risk function) The risk function  $R_g(h)$  of  $h$  is the expected loss of the hypothesis  $h \in \mathcal{H}$ , w.r.t. probability distribution  $g$  over domain  $\mathcal{Z}$ ; i.e.

$$(2.1) \quad R_g(h) = \mathbb{E}_{z \sim g}(\ell(h, z))$$

*Remark 10.* In learning, an ideal way to obtain an optimal predictor  $h^*$  is to compute the risk minimizer

$$h^* = \arg \min_{\mathcal{H}} (R_g(h))$$

**Example 11.** (Cont. Ex. 8) The risk function is  $R_g(h) = \mathbb{E}_{z \sim g} (h(x) - y)^2$ , and it measures the quality of the hypothesis function  $h : \mathcal{X} \rightarrow \mathcal{Y}$  as the expected square difference between the predicted values  $h$  and the true target values  $y$  at every  $x$ .

*Remark 12.* Computing the risk minimizer may be practically challenging due to the integration w.r.t. the unknown data generation model  $g$  involved in the expectation (2.1). Suboptimally, one may resort to the Empirical risk function.

**Definition 13.** (Empirical risk function) The empirical risk function  $\hat{R}_S(h)$  of  $h$  is the expectation of loss of  $h$  over a given sample  $S = (z_1, \dots, z_m) \in \mathcal{Z}^m$ ; i.e.

$$\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^m \ell(h, z_i).$$

**Example 14.** (Cont. Example 11) Given given sample  $S = \{(x_i, y_i); i = 1, \dots, m\}$  the empirical risk function is  $\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^m (h(x_i) - y_i)^2$ .

**Example 15.** Consider a learning problem where the true data generation distribution (unknown to the learner) is  $g(z)$ , the statistical model (known to the learner) is given by a sampling distribution  $f(y|\theta)$  where the parameter  $\theta$  is unknown. The goal is to learn  $\theta$ . If we assume loss function

$$\ell(\theta, z) = \log \left( \frac{g(z)}{f_\theta(z)} \right)$$

then the risk is

$$(2.2) \quad R_g(\theta) = \mathbb{E}_{z \sim g} \left( \log \left( \frac{g(z)}{f_\theta(z)} \right) \right) = \mathbb{E}_{z \sim g} (\log(g(z))) - \mathbb{E}_{z \sim g} (\log(f_\theta(z)))$$

whose minimizer is

$$\theta^* = \arg \min_{\forall \theta} (R_g(\theta)) = \arg \min_{\forall \theta} (\mathbb{E}_{z \sim g} (-\log(f_\theta(z))))$$

as the first term in (2.2) is constant. Note that in the Maximum Likelihood Estimation technique the MLE  $\theta_{\text{MLE}}$  is the minimizer of

$$\theta_{\text{MLE}} = \arg \min_{\forall \theta} \left( \frac{1}{m} \sum_{i=1}^m (-\log(f_\theta(z_i))) \right)$$

where  $S = \{y_1, \dots, y_m\}$  is an IID sample from  $g$ . Hence, MLE  $\theta_{\text{MLE}}$  can be considered as the minimizer of the empirical risk  $R_S(\theta) = \frac{1}{m} \sum_{i=1}^m (-\log(f_\theta(z_i)))$ .

**Example 16.** (Linear Regression) Consider the multiple linear regression problem  $x \mapsto y$  where  $x \in \mathcal{X} \subseteq \mathbb{R}^d$  and  $y \in \mathcal{Y} \subseteq \mathbb{R}^d$ .

- The hypothesis is a linear function  $h : \mathcal{X} \rightarrow \mathcal{Y}$  (that learner wishes to learn) to approximate mapping  $x \mapsto y$ . The hypothesis set  $\mathcal{H} = \{ \langle w, x \rangle \mapsto y : w \in \mathbb{R}^d \}$ . We can use the loss  $\ell(h, (x, y)) = (h(x) - y)^2$ .
- Equivalently, learning problem can be set differently because the predictor (linear function) is parametrized by  $w \in \mathbb{R}^d$  as  $\langle w, x \rangle \mapsto y$ . Set  $\mathcal{H} = \{w \in \mathbb{R}^d\}$ . The set of examples is  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ . The loss is  $\ell(w, (x, y)) = (\langle w, x \rangle - y)^2$ .