Lecture notes 1: Machine learning -A recap on: definitions, notation, and formalism

Lecturer & 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.
  - Ch. 1 Introduction
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
  - Ch. 1 Introduction

1. GENERAL INTRODUCTIONS AND 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, fitting) is called the procedure where the ML model is tuned. **Training data** (or observations, sample data set, examples) is a set of observables $\{z_i \in \mathcal{Z}\}$ used to tune the parameters of the ML model. By $\mathcal{Z}$ we denote the examples (or observables) domain. **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 will focus on this here) and the unsupervised learning.

Supervised learning problems involve applications where the training data $z \in \mathcal{Z}$ comprise examples of the input vectors $x \in \mathcal{X}$ along with their corresponding target vectors $y \in \mathcal{Y}$; i.e. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. By $\mathcal{X}$ we denote the inputs (or instances) domain, and by $\mathcal{Y}$ we denote the target domain. **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 discover 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. $x \longmapsto y$. Hence, $\mathcal{X}$ is the input domain, and $\mathcal{Y}$ is the output (or target) domain. The goal of learning is to discover a function which predicts (or help us make decisions about) $y \in \mathcal{Y}$ from $x \in \mathcal{X}$.

Unsupervised learning problems involve applications where the training data $z \in \mathcal{Z}$ consist 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. NOTATION & DEFINITIONS IN LEARNING

Note 1. The aim is to learn the mapping $x \to y$ where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ with purpose to predict $y \in \mathcal{Y}$ from $x \in \mathcal{X}$. Denote $z = (x, y)^{\top}$ and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.

Definition 2. The learner's output is a function, $h : \mathcal{X} \to \mathcal{Y}$ which predicts $y \in \mathcal{Y}$ from $x \in \mathcal{X}$. It is also called hypothesis, prediction rule, predictor, or classifier.

Notation 3. We often denote the set of hypothesis as $\mathcal{H}$; i.e. $h \in \mathcal{H}$.

Example 4. (Linear Regression)[1] Consider the regression problem where the goal is to learn the mapping $x \to y$ where $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y \in \mathcal{Y} \subseteq \mathbb{R}$. A hypothesis is a linear function $h : \mathcal{X} \to \mathcal{Y}$ (that learner wishes to learn) with $h(x) = \langle w, x \rangle$ approximating the mapping $x \to y$. The hypothesis set is $\mathcal{H} = \{x \to \langle w, x \rangle : w \in \mathbb{R}^d\}$.

Example 5. (Binary Classification) Consider the classification problem where the goal is to learn the mapping $x \to y$ where $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y \in \mathcal{Y} = \{-1, +1\}$. A hypothesis can be a function $h : \mathcal{X} \to \mathcal{Y}$ with $h(x) = \text{sign}(\langle w, x \rangle)$ approximating the mapping $x \to y$. The hypothesis set $\mathcal{H} = \{x \to \text{sign}(\langle w, x \rangle) : w \in \mathbb{R}^d\}$.

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 $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$. Loss function $\ell(h, z)$ for $h \in \mathcal{H}$ and $z \in \mathcal{Z}$ is specified according to the purpose the machine learning algorithm. It reflects how the "error" is quantified for a given hypothesis h and a given example z. The rule is "the greater the error the greater the value of the loss".

Example 7. (Cont. Example 4) In regression problems $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $\mathcal{Y} \subset \mathbb{R}$ is uncountable, a potential loss function is
$$\ell_{\text{sq}}(h, (x, y)) = (h(x) - y)^2$$

Example 8. (Cont. Example 5) In binary classification problems with hypothesis $h : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y} = \{0, 1\}$ is discrete, a loss function can be
$$\ell_{0-1}(h, (x, y)) = 1(h(x) \neq y),$$

Definition 9. A learning problem with hypothesis class $\mathcal{H}$, examples domain $\mathcal{Z}$, and loss function ℓ may be denoted with a triplet $(\mathcal{H}, \mathcal{Z}, \ell)$.

Example 10. The standard multiple linear regression problem with regressors $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and response $y \in \mathcal{Y} \subseteq \mathbb{R}$, is a learning problem with examples domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \mathbb{R}^{d+1}$, hypothesis class $\mathcal{H} = \{x \to \langle w, x \rangle : w \in \mathbb{R}^d\}$, and loss function $\ell_{\text{sq}}(h, (x, y)) = (h(x) - y)^2$.

---

[1] $\langle w, x \rangle = w^{\top} x$

Example 11. The binary classification regression problem with regressors $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and response $y \in \mathcal{Y} = \{-1, +1\}$, is a learning problem with examples domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, hypothesis class $\mathcal{H} = \{x \to \text{sign} (\langle w, x \rangle) : w \in \mathbb{R}^d\}$, and loss function $\ell_{0-1} (h, (x, y)) = 1 (h(x) \neq y) = 1 (\langle w, x \rangle y < 0)$.

Definition 12. Data generation model $g(\cdot)$ is the probability distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, unknown to the learner which describes the probabilistic law that realizations $z = (x, y) \in \mathcal{Z}$ are realized.

Conventions...

Note 13. If the Hypothesis set $\mathcal{H}$ is a known parametric family of functions; i.e. $\mathcal{H} = \{h_w (\cdot) ; w \in \mathcal{W}\}$ parameterized by unknown $w \in \mathcal{W}$, then we can equivalently consider the convention $\mathcal{H} = \{w \in \mathcal{W}\} = \mathcal{W}$ keeping in mind that the learner's output is restricted to formula $h_w (\cdot)$.

Example 14. Because it involves only linear functions as predictors $h_w (x) = \langle w, x \rangle$, we could consider a hypothesis class $\mathcal{H} = \{w \in \mathbb{R}^d\} = \mathbb{R}^d$ and loss function loss $\ell (w, (x, y)) = (\langle w, x \rangle - y)^2$ for simplicity.

Example 15. Because it involves only linear functions as predictors $h_w (x) = \text{sign} (\langle w, x \rangle)$, we could consider a hypothesis class $\mathcal{H} = \{w \in \mathbb{R}^d\} = \mathbb{R}^d$ and loss function loss $\ell (w, (x, y)) = 1 (\langle x, w \rangle y < 0)$ for simplicity.

2.1. **Learning...**

Definition 16. (Risk function) The risk function $R_g (h)$ of h is the expected loss of the hypothesis $h \in \mathcal{H}$, w.r.t. the data generation model (which is a probability distribution) g over domain Z; i.e.

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

Definition 17. (Risk minimization learning) The Risk minimization (RM) learning paradigm computes the optimal predictor h^* as the minimizer of the risk $R_g (h)$ of h; i.e.

$$(2.2) \qquad h^* = \arg \min_h (R_g (h))$$

Example 18. (Cont. Ex. 7) The risk function is $R_g (h_w) = \text{E}_{z \sim g} (h_w(x) - y)^2 = \text{E}_{z \sim g} (\langle w, x \rangle - y)^2$, and it measures the quality of the hypothesis function $h : \mathcal{X} \to \mathcal{Y}$, (or equiv. the validity of the class of hypotheses $\mathcal{H}$) against the data generating model g, as the expected square difference between the predicted values form h and the true target values y at every x.

Example 19. (Cont. Ex.) The risk function is $R_g (h_w) = \text{E}_{z \sim g} (1 (h_w(x) \neq y)) = \text{Pr}_{z \sim g} (\langle x, w \rangle y < 0)$.

Note 20. 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). Sub-optimally, instead of the Risk function in (2.2), one use the Empirical risk function as a Monte Carlo approximation of it.

Definition 21. Training data set $\mathcal{S}$ of size m is any finite sequence of pairs $(z_i = (x_i, y_i) ; i = 1, ..., m)$, called examples, in $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$; i.e. $\mathcal{S} = \{(x_i, y_i) ; i = 1, ..., m\}$. This is the information that the learner has assess.

Definition 22. (Empirical risk function) The Empirical Risk Function (ERF) $\hat{R}_S(h)$ of h is the expectation of loss of h over a given sample $S = (z_1, ..., z_m) \in \mathcal{Z}^m$; i.e.

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

Definition 23. (Empirical risk minimization learning) The Empirical risk minimization (ERM) learning paradigm computes the optimal predictor h^* as the minimizer of the ERF $\hat{R}_S(h)$ of h; i.e.

$$(2.3) \qquad h^* = \arg\min_h \left(\hat{R}_S(h) \right)$$

Example 24. (Cont. Example 18) Given given sample $S = \{(x_i, y_i); i = 1, ..., m\}$ the empirical risk function is $\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} (h(x_i) - y_i)^2 = \frac{1}{m} \sum_{i=1}^{m} \left(x_i^\top w - y_i \right)^2$.

Example 25. (Cont. Example 8) Given given sample $S = \{(x_i, y_i); i = 1, ..., m\}$ the empirical risk function is $\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} 1\left(h(x_i) \neq y_i \right) = \frac{1}{m} \sum_{i=1}^{m} 1\left(\langle x_i, w \rangle y_i < 0 \right)$.

Definition 26. We denote as $\mathfrak{A}(\mathcal{S})$ the hypothesis (outcome) that a learning algorithm $\mathfrak{A}$ returns given training sample S.

3. REGULARIZED LOSS MINIMIZATION (RLM) LEARNING

Note 27. Consider a learning problem $(\mathcal{H}, \mathcal{Z}, \ell)$ with training data set $\mathcal{S}$.

Definition 28. (Regularized loss minimization learning) Regularized loss minimization learning paradigm computes the optimal rule $h^* \in \mathcal{H}$ by jointly minimizing the Risk function $R_g(h)$ (or Empirial Risk Function $\hat{R}_\mathcal{S}(h)$) and a regularization function $J : \mathcal{H} \to \mathbb{R}$ where $J(h)$ increases in value with the complexity of the hypothesis $h \in \mathcal{H}$. Formally, the Regularized loss minimization rule h^* is

$$h^* = \arg\min_{h \in \mathcal{H}} \left(R_g(h) + J(h) \right), \text{ given the Risk function}$$

$$h^* = \arg\min_{h \in \mathcal{H}} \left(\hat{R}_\mathcal{S}(w) + J(h) \right), \text{ given the Empirical risk function}$$

3.1. ℓ_0, ℓ_1, and ℓ_2 **norm regularization problems.**

Note 29. Consider a parameterized hypothesis $h_w(\cdot)$ (e.g. $h_w(x) = \eta(\langle x, w \rangle)$) with $\eta(\cdot)$ a function and $w \in \mathbb{R}^d$. Consider the hypothesis class in the simplified form $\mathcal{H} = \mathcal{W} = \left\{ w \in \mathbb{R}^d \right\} = \mathbb{R}^d$ of Note 13. Assume we want the vector w to be sparse, in the sense that "sparser" means more zero elements.

3.1.1. ℓ_0 *norm regularization.*

Note 30. The problem of minimizing the empirical risk subject to a budget of k features can be written as

$$\underset{w \in \mathcal{H}}{\text{minimize}} R(w)$$

$$\text{subject to } \|w\|_0 \leq k_0$$

where $\|w\|_0 = \# \{j : w_j \neq 0\}$. By Lagrange multiplies we can re-write it as

$$w^* = \arg\min_{w \in \mathcal{H}} (R(w) + \lambda_0 \|w\|_0)$$

for some $\lambda_0 \geq 0$ (k_0 dependent). Hence, $h^*(x) = h_{w^*}(x)$.

Example 31. (Cont. Example 14) The ℓ_0-RLM rule in Example 14 becomes

$$w^* = \arg\min_w \left(\mathrm{E}_{z \sim g} (\langle w, x_i \rangle - y)^2 + \lambda_0 \|w\|_0 \right)$$

using the Risk function, and

$$w^* = \arg\min_w \left(\frac{1}{m} \sum_{i=1}^m (\langle w, x_i \rangle - y_i)^2 + \lambda_0 \|w\|_0 \right)$$

using the Empirical risk function.

3.1.2. *ℓ_1 norm regularization (LASSO).*

Note 32. Solving ℓ_0 norm optimization problem is computationally hard. To simplify computations, we can replace $J(\cdot) = \|\cdot\|_0$ with $J(\cdot) = \|\cdot\|_1$ i.e.

$$\underset{w \in \mathcal{H}}{\mathrm{minimize}} R(w)$$

$$\text{subject to } \|w\|_1 \leq k_1$$

where $\|w\|_1 = \sum_j |w_j|$. By Lagrange multiplies we can re-write it as

$$w^* = \arg\min_w (R(w) + \lambda_1 \|w\|_1)$$

for some $\lambda_1 \geq 0$ (k_1 dependent). Hence, $h^*(x) = h_{w^*}(x)$.

Example 33. (Cont. Example 14) The ℓ_1-RLM rule in Example 14 becomes

$$w^* = \arg\min_w \left(\mathrm{E}_{z \sim g} (\langle w, x \rangle - y)^2 + \lambda_1 \sum_{j=1}^d |w_j| \right)$$

using the Risk function , and

$$w^* = \arg\min_w \left(\frac{1}{m} \sum_{i=1}^m (\langle w, x_i \rangle - y_i)^2 + \lambda_1 \sum_{j=1}^d |w_j| \right)$$

using the Empirical risk function.

3.1.3. *ℓ_2 norm regularization (Ridge).*

Note 34. To further simplify computations, we can consider $J(\cdot) = \|\cdot\|_2$ i.e.

$$\underset{w}{\mathrm{minimize}} R(w)$$

$$\text{subject to } \|w\|_2^2 \leq k_2$$

where $\|w\|_2^2 = \sum_j w_j^2$. By Lagrange multiplies we can re-write it as

$$w^* = \arg\min_w \left(R(w) + \lambda_2 \|w\|_2^2 \right)$$

for some $\lambda_1 \geq 0$ (k_1 dependent). Hence, $h^*(x) = h_{w^*}(x)$.

Example 35. (Cont. Example 14) The ℓ_2-RLM rule in Example 14 becomes

$$w^* = \arg\min_w \left(\mathrm{E}_{z \sim g}\left(\langle w, x \rangle - y \right)^2 + \lambda_2 \sum_{j=1}^d w_j^2 \right)$$

using the Risk function, and

$$w^* = \arg\min_w \left(\frac{1}{m} \sum_{i=1}^m \left(h(x_i) - y_i \right)^2 + \lambda_2 \sum_{j=1}^d w_j^2 \right)$$

using the Empirical risk function.

Example 36. (Cont. Example 15) The ℓ_2-RLM rule in Example 15 becomes

$$w^* = \arg\min_w \left(\Pr_{z \sim g}\left(\langle x, w \rangle\, y < 0 \right) + \lambda_2 \sum_{j=1}^d w_j^2 \right)$$

using the Risk function, and

$$w^* = \arg\min_w \left(\frac{1}{m} \sum_{i=1}^m 1\left(\langle x_i, w \rangle\, y_i < 0 \right) + \lambda_2 \sum_{j=1}^d w_j^2 \right)$$

using the Empirical risk function.

Created on 2025/03/21 at 11:26:41   by Georgios Karagiannis

Example 37. 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 $f_\theta(y) := f(y|\theta)$ labeled by an unknown parameter θ. The goal is to learn θ. If we assume loss function

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

then the risk is

$$(A.1) \qquad R_g(\theta) = E_{z \sim g}\left(\log\left(\frac{g(z)}{f_\theta(z)}\right)\right) = E_{z \sim g}\left(\log\left(g(z)\right)\right) - E_{z \sim g}\left(\log\left(f_\theta(z)\right)\right)$$

whose minimizer is

$$\theta^* = \arg\min_{\forall \theta}\left(R_g(\theta)\right) = \arg\min_{\forall \theta}\left(E_{z \sim g}\left(-\log\left(f_\theta(z)\right)\right)\right)$$

as the first term in (A.1) is constant. Note that in the Maximum Likelihood Estimation technique the MLE θ_{MLE} is the minimizer

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

where $S = \{z_1, ..., z_m\}$ is an IID sample from g. Hence, MLE θ_{MLE} can be considered as the minimizer of the empirical risk $R_S(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(-\log\left(f_\theta(z_i)\right)\right)$.