Introduction to Machine Learning

Alessandro Rudi, Umut Simsekli
Notes by Antoine Groudiev

6th March 2024

Contents

Introduction

1 An overview of Machine Learning

1.1 What is ML?

Considering a problem, such as image classification: given an input image of a dog or a cat, the program is asked to determine whether the image is a dog or a cat. Conventional programming would hardcode the solution to this problem. But this process takes time and is not easily generalisable. Instead, an ML model is trained on a dataset to produce a program to solve the problem.

Many successfull applications of Machine Learning are:

- Face recognition
- Spam filtering
- Speech recognition
- Self-driving systems; pedestrian detection

1.2 Topics in Machine Learning

1.2.1 Supervised Learning

Example (Classification)**.** *Features* $x \in \mathbb{R}^d$*, labels* $y \in \{1, \dots, k\}$

Definition (Regression)**.** Features $x \in \mathbb{R}^d$, labels $y \in \mathbb{R}$. To tackle such problem, we look for a parametrized function $f_\theta(x_i) \simeq y_i$ for some f_θ in a function space

$$\mathcal{F} = \{f_\theta : \theta \in \Theta\}$$

Our goal is therefore to find the best function in $\mathcal{F}$ such that f "fits" the training data. For example, we can say that f "fits" the training data when

$$\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2$$

is "small". Such a function is not interesting in general, like for classification.

Definition (Loss function)**.** Assums that the features are in $\mathcal{X}$ and the labels are in $\mathcal{Y}$. We introduce the more general *loss function* notion:

$$l : \mathcal{Y}^2 \to \mathbb{R}_+$$

For a regression task, we can use $l(\hat{y}, y) = (\hat{y} - y)^2$. For a classification task, $l(\hat{y}, y) = \mathbb{1}_{\hat{y}=y}$.

Therefore, for a regression problem, we might choose:

$$f^\star = \operatorname*{argmin}_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i)$$

In the parametric case, when $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$, we might minimize with respect to θ:

$$\theta^\star = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i)$$

1.2.2 Probabilistic approach

Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ be the feature space. Let D be a distribution on $\mathcal{Z}$; we make the assumption that the training data is iid from D:

$$(x_i, y_i) \sim D$$

and the same thing hold for the test data:

$$(\tilde{x_i}, \tilde{y_i}) \sim D$$

According to the Strong Law of Large Numbers, the test loss converges almost surely:

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} l(f_\theta(\tilde{x_i}), \tilde{y_i}) = \mathbb{E}_{(x,y)\sim D}[l(f_\theta(x), y)] =: R(\theta) = R(f_\theta)$$

where $R(\theta)$ is the *population risk.*

Definition (Risk minimization)**.**

1.2.3 Unsupervised Learning

Example (Clustering)**.**

Example (Dimensionnality reduction)**.** *We are given features $x \in \mathbb{R}^d$ and labels $y \in \{0, 1\}$ which form a "training" dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. We assume that $d >> 1$; our goal is to find $d' << d$ such that $(x_1, y_1, \ldots,)$*

2 Linear Least Squares Regression

Consider an input space X and an output space Y. We consider a function $f : X \to Y$ unknown to us, that we want to recover. We are given samples $D_N = [(x_1, y_1), \dots, (x_N, y_N)]$. Our goal is to produce $\hat{f}_D$ such that $\hat{f}_D$ "converges" to f when $|D| \to +\infty$.

3 Logic regression and convex analysis

Recap of important notions and notations

We are given an input space X and an output space Y. We want to learn the relationship between input and output, modelised by a probability distribution $\rho \in \mathbb{P}(X \times Y)$. Thus, we try to find the best function $f_\star : X \to Y$, given a loss function $l : Y \times Y \to \mathbb{R}$. Therefore, $f_\star$ is often defined by:

$$f_\star = \underset{f:X\to Y}{\operatorname{argmin}} \mathbb{E}_{X,Y}[l(f(X), Y)]$$

where

$$\mathbb{E}_{X,Y}[g(X,Y)] = \int_{\mathbb{R}^2} g(x,y) \cdot \mathrm{d}\rho(x,y)$$

In practice, you only know some samples $D_N = [(x_1, y_1), \dots, (x_N, y_N)]$ with $(x_i, y_i) \sim \rho$, making it impossible to choose such an $f_\star$. Therefore, we try to find a good model $\hat{f}_{D_N}$, such that

$$\lim_{N\to+\infty} \mathcal{E}(\hat{f}_{D_N}) - \mathcal{E}(f) = 0$$

Such a result will often be given by a *learning rate function* $c(N)$, with

$$\mathbb{E}_{D_N}[\mathcal{E}(\hat{f}_{D_N}) - \mathcal{E}(f)] \ \leqslant c(N) = o(1)$$

The function $\hat{f}_{D_N}$ can be chosen such that it minimizes the empirical error:

$$\hat{f}_{D_N} = \underset{f\in\mathcal{H}}{\operatorname{argmin}} \hat{\mathcal{E}}(f) = \underset{f\in\mathcal{H}}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} l(f(x_i), y_i)$$

3.1

We consider the case where $X = \mathbb{R}^d$ and $Y = \mathbb{R}$. We define the loss l to be the least squares, $l(y, y') = (y - y')^2$, and we choose our functions to be of the form of $f_\star = \theta_\star^T X$. In this case, ERM is OLS:

$$\hat{\theta}_N = \underset{\theta\in\mathbb{R}^d}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}(\theta^T x_i - y_i)^2$$

We can also define $\hat{\theta}_{N,\lambda}$ to be:

$$\hat{\theta}_{N,\lambda} = \underset{\theta\in\mathbb{R}^d}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}(\theta^T x_i - y_i)^2 + \lambda||\theta||^2$$

This allows to regulate the "complexity" of the function to avoid overfitting. This is called Tikhonov regularization. In this case, we have

$$\mathbb{E}_{\hat{Y}}[\hat{\theta}_{\mathcal{N}} - \mathcal{E}(\theta_\star)] = \frac{\sigma^2 d}{N}$$

and therefore the optimal function is

$$\hat{f}_{N,\lambda} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \hat{\mathcal{E}}(f) + \lambda R(f)$$

We define $X \in \mathbb{R}^{N \times d} := (x_1^T, \dots, x_N^T)$, and $\hat{Y} = (\hat{y}_1, \dots, \hat{y}_n)$. We this notation, we have

$$\hat{\theta}_{N,\lambda} = \frac{1}{N} ||X\theta - \hat{Y}||^2 + \lambda ||\theta||^2$$

Thus, we have

$$\begin{aligned} \nabla \mathcal{L}(\theta) := \frac{2}{N} X^T X \theta - 2\frac{X^T \hat{Y}}{N} + 2\lambda\theta = 0 \\ (\frac{X^T X}{N} + \lambda)\theta = X^T \hat{Y} \end{aligned}$$

therefore,

$$\hat{\theta}_{N,\lambda} = \left(\frac{X^T X}{N} + \lambda I\right)^{-1} \frac{X^T \hat{Y}}{N} = \left(X^T X + \lambda N I\right)^{-1} X^T \hat{Y}$$

We introduce the singular value decomposition of X:

$$X = U \Sigma V^T$$

where $U^T U = U U^T = I_N$, $V^T V = V V^T = I_d$, and Σ is diagonal with $\forall i$, $\Sigma_{i,i} \geqslant 0$. In this case,

$$\begin{aligned} X^T X + \lambda N I_d &= V \Sigma U^T U \Sigma V^T + \lambda N I_d \\ &= V \underbrace{(\Sigma^2 + \lambda N I)}_{\text{invertible}} V^T \end{aligned}$$