Lecture Notes

MATHEMATICS FOR MACHINE LEARNING

This version: October 14, 2025

Latest version: `github.com/felipe-tobar/Maths-for-ML`

Felipe Tobar
Department of Mathematics
Imperial College London

`f.tobar@imperial.ac.uk`
`www.ma.ic.ac.uk/~ft410`

Contents

1 Introduction

MISSING

2 Optimisation

NB: in this chapter, we follow (Murphy, 2022).

Optimisation is central to ML, since models are *trained* by minimising a loss function (or optimising a reward function). In general, model design involves the definition of a training objective, that is, a function that denotes how good a model is. This training objective is a function of the training data and a model, the latter usually represented by its parameters. The best model is is the chosen by optimising this function.

Example: Linear regression (LR)

In the LR setting, we aim to determine the function

$$\begin{aligned} f\colon \mathbb{R}^M &\to \mathbb{R} \\ x &\mapsto f(x) = a^\top x + b, \quad a \in \mathbb{R}^M, b \in \mathbb{R} \end{aligned} \tag{2.1}$$

conditional to a set of observations

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^M \times \mathbb{R}. \tag{2.2}$$

Using least squares, the function f is chosen via minimisation of the sum of the square differences between observations $\{y_i\}_{i=1}^N$ and predictions $\{f(x_i)\}_{i=1}^N$. That is, we aim to minimise he loss:

$$J(\mathcal{D}, f) = \sum_{i=1}^N (y_i - f(x_i))^2 = \sum_{i=1}^N (y_i - a^\top x_i - b)^2. \tag{2.3}$$

[FT: Camilo, por favor generar figura aqui. Mira la fig 1 del apunte del curso de AM]

Example: Logistic regression

Here, we aim to determine the function

$$\begin{aligned} f\colon \mathbb{R}^M &\to \mathbb{R} \\ x &\mapsto f(x) = \frac{1}{1 + e^{-\theta^\top x + b}}, \quad \theta \in \mathbb{R}^M, b \in \mathbb{R} \end{aligned} \tag{2.4}$$

conditional to the observations

$$\mathcal{D} = \{(x_i, c_i)\}_{i=1}^N \subset \mathbb{R}^M \times \{0, 1\}. \tag{2.5}$$

The standard loss function for the classification problem is the cross entropy, given by:

$$\begin{aligned} J(\mathcal{D}, f) &= -\frac{1}{N} \sum_{i=1}^N (c_i \log f(x_i) + (1 - c_i) \log(1 - f(x_i))) && (2.6) \\ &= \frac{1}{N} \sum_{i=1}^N \left(\log(1 + e^{-\theta^\top x + b}) - y_i(-\theta^\top x + b) \right) && (2.7) \end{aligned}$$

[FT: Camilo, por favor generar figura aqui]

Example: Clustering (K-means)

Given a set of observations

$$\mathcal{D} = \{x_i\}_{i=1}^{N} \subset \mathbb{R}^M, \tag{2.8}$$

we aim to find cluster centres (or prototypes) $\mu_1, \mu_2, \dots, \mu_K$ and *assignment variables* $\{r_{ik}\}_{i,k=1}^{N,K}$, to minimise the following loss

$$J(\mathcal{D}, f) = \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}||x_i - \mu_k||^2 \tag{2.9}$$

$$\tag{2.10}$$

[FT: Camilo, por favor generar figura aqui]

2.1 Terminology

We denote an optimisation problem as follows:

$$\min_{x\in\mathcal{X}} f(x) \;\text{ s.t. }\; g_i(x) \leq 0,\; h_j(x) = 0,\; i = 1,\dots,I,\; j = 1,\dots,J. \tag{2.11}$$

We describe the components of this statement in detail:

- **Objective function:** The function $f : \mathcal{X} \to \mathbb{R}$ is the quantity to be minimised, with respect to x.
- **Optimisation variable:** Minimising f requires fining the value of x such that $f(x)$ is minimum. This is also written as

$$x_\star = \arg\min_{x\in\mathcal{X}} f(x) \;\text{ s.t. }\; g_i(x) \leq 0,\; h_j(x) = 0. \tag{2.12}$$

- **Restrictions:** These are denoted by the functions g_i and h_i above, which describe the requirements for the optimiser in the form of equalities and inequalities, respectively.
- **Feasible region:** This is the subset of the domain that complies with the restrictions, that is

$$C = \{x \in \mathcal{X},\; \text{ s.t. }\; g_i(x) \leq 0,\; h_j(x) = 0,\; i = 1,\dots,I,\; j = 1,\dots,J\} \tag{2.13}$$

- **Local / global optima.** Values for the optimisation variable that solve the optimisation problem wither locally or globally. More formally:

$$x_\star \text{ is a local optima } \iff \exists\lambda > 0 \;\text{ s.t. }\; x_\star = \underset{x\in\mathcal{X}\;\text{ s.t. }\;||x-x_\star||\leq\lambda}{\arg\min} f(x). \tag{2.14}$$

$$x_\star \text{ is a global optima } \iff x_\star = \arg\min_{x\in\mathcal{X}} f(x). \tag{2.15}$$

Interplay between constrains and local/global optima

[FT: Camilo, por favor una ilustracion de como las differentes restriccines cambian la cantidad y tipo de minimos]

Example: XXX

[FT: Camilo: Present a few parametric functions and indicate their (closed-form) minima]

3 Continuous unconstrained optimisation

We will ignore constrains in this section, and we will focus on problems of the form

$$\theta \in \arg\min_{\theta \in \Theta} L(\theta). \tag{3.1}$$

We emphasise that if $\theta_\star$ satisfies the above, then

$$\forall \theta \in \Theta,\ L(\theta_\star) \leq L(\theta), \tag{3.2}$$

meaning that it is a **global** optimum. However, as this might be very hard to find, we are also interested in local optima, that is, $\theta_\star$ such that

$$\exists \delta > 0\ \ \forall \theta \in \Theta\ \text{ s.t. }\ \|\theta - \theta_\star\| < \delta\ \Rightarrow\ L(\theta_\star) \leq L(\theta). \tag{3.3}$$

3.1 Optimality Conditions

Assumption 3.1. The loss function L is twice differentiable.

Denoting $g(\theta) = \nabla_\theta L(\theta)$ and $H(\theta) = \nabla^2_\theta L(\theta)$, we can state the following optimality conditions.

References

Murphy, K. P. (2022). *Probabilistic machine learning: An introduction.* MIT Press. Retrieved from `probml.ai`