Ensimag – Grenoble INP – UGA Year 2024-2025
Introduction to Statistical Learning and Applications
Pedro L. C. Rodrigues pedro.rodrigues@inria.fr
Alexandre Wendling alexandre.wendling@univ-grenoble-alpes.fr

TD 5: Gradient boosting

▶ Exercise 1: Decision trees

Suppose we have a training dataset with N samples $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$.

We would like to learn a regression model on this dataset using a decision stump, i.e. a decision tree with just one level that splits the data into just two regions as per:

$$f(\mathbf{x}) = c_1 \mathbf{I}(\mathbf{x} \in \mathcal{R}_1) + c_2 \mathbf{I}(\mathbf{x} \in \mathcal{R}_2)$$

Note that regions $\mathcal{R}_1$ and $\mathcal{R}_2$ are defined based on a choice of splitting value $s \in \mathbb{R}$ and predictor $j \in \{1, \ldots, p\}$,

$$\mathcal{R}_1 = \{\mathbf{x}_i \mid x_{ij} \leq s\} \quad \text{and} \quad \mathcal{R}_2 = \{\mathbf{x}_i \mid x_{ij} > s\} \ .$$

Our criterion for choosing c_1, c_2, s, j is based on the minimization of the variance in each of the regions, i.e.

$$\min_{j,s,c_1,c_2} \left(\sum_{\mathbf{x}_i \in \mathcal{R}_1} (y_i - c_1)^2 + \sum_{\mathbf{x}_i \in \mathcal{R}_2} (y_i - c_2)^2 \right)$$

(a) Show that for a fixed choice of j and s, we can minimize the above problem with

$$c_1 = \frac{1}{|\mathcal{R}_1|} \sum_{\mathbf{x}_i \in \mathcal{R}_1} y_i \quad \text{and} \quad c_2 = \frac{1}{|\mathcal{R}_2|} \sum_{\mathbf{x}_i \in \mathcal{R}_2} y_i$$

For the following items, choose the only sentence which is true about decision trees:

(b) A given split node in a decision tree classifier makes:

- a binary decision considering a single feature at a time
- a binary decision considering a combination of all the input features
- multiple binary decisions considering a single feature
- a binary decision considering a non-linear combination of all input features

(c) A decision tree split is built:

- using a random threshold
- using the median value of a single feature as a threshold
- using a threshold that minimizes an error

(d) Decision tree regressors can predict:

- any values, including values larger or smaller than the y_i observed in the training dataset.
- only values in the range from $\min(y_i)$ to $\max(y_i)$.

▶ Exercise 2

Recall from class that in the first round of gradient boosting we look for a predictor f_1 such that

$$f_1 = \underset{h \in \mathbb{H}}{\operatorname{argmin}} \ \langle \nabla \mathcal{L}(f_0, \mathcal{D}), h \rangle$$

where for a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ we have that

$$\mathcal{L}(f, \mathcal{D}) = \sum_{i=1}^{N} \ell\Big(f(\mathbf{x}_i), y_i\Big)$$

where ℓ is a function used to measure the error of an estimator, e.g. MSE in regression or classification error.

In what follows, we will assume that

$$\forall h \in \mathbb{H}, \sum_i h(\mathbf{x}_i)^2 = 1 \ ,$$

which is easy to ensure by making the weak learners normalized on the dataset.

(a) Show that

$$f_1 = \underset{h \in \mathbb{H}}{\operatorname{argmin}} \ \sum_{i=1}^{N} \Big(h(\mathbf{x}_i) - r_i^{(1)}\Big)^2 \quad \text{where} \quad r_i^{(1)} = -\frac{\partial \ell(f_0(\mathbf{x}_i), y_i)}{\partial f_0(\mathbf{x}_i)}$$

Interpret this result.

(b) Consider that we are in a regression setting for which

$$\ell(f(\mathbf{x}), y) = \frac{1}{2}(f(\mathbf{x}) - y)^2 \ .$$

What would be the expression for the optimization problem from item (a) in this case?

(c) The most common version of gradient boosting is to assume that the weak learners are regression trees, such as the decision stumps from Exercise 1. Describe how the parameters c_1, c_2, s, j are estimated in this case for a general choice of ℓ.

▶ Exercise 3

In this exercise, we will see how the AdaBoost algorithm can be obtained using the gradient boosting framework. We will consider $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{-1, +1\}$ and the goal is to fit a set of weak learners $f_0, f_1, \ldots, f_T$ such that the classification of a data point $\mathbf{x}$ can be done as per

$$c(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=0}^{T} \alpha_t f_t(\mathbf{x})\right)$$

(a) Since we're in a classification setting, we need to choose suitable loss function ℓ. At first glimpse, we could be tempted to choose the 0-1 loss:

$$\ell_{0-1}(f(\mathbf{x}), y) = \mathbf{1}(f(\mathbf{x}) \neq y) \ .$$

Explain why this would not be a good choice for gradient boosting?

(b) In AdaBoost we actually consider the exponential loss given as

$$\ell_{\exp}(f(\mathbf{x}), y) = e^{-yf(\mathbf{x})}$$

Compare ℓ_{0-1} and $\ell_{\exp}$ and explain why using $\ell_{\exp}$ should be fine.

(c) Using the notation from class, calculate the expression for the $t_i^{(1)}$ using $\ell_{\exp}$ as loss.

(d) Give an interpretation in terms of a weighted classification error to the optimization problem

$$f_1 = \underset{h \in \mathbb{H}}{\operatorname{argmin}} \ \sum_{i=1}^{N} t_i^{(1)} h(\mathbf{x}_i)$$