ELEN0062 - Introduction to Machine Learning
Project 2 - Bias and variance analysis

29 octobre 2022

The goal of this second assignment is to help you to better understand the important notions of bias and variance. The first part is purely theoretical, while the second part requires to perform some experiments with scikit-learn. You should hand in a *brief* report giving your developments, observations and conclusions along with the scripts you have implemented to answer the questions of the second part. The project must be carried out by groups of at most *three students* and submitted on Gradescope[1] before *November 21, 23:59 GMT+2*. There will be two projects to submit to: one for your python scripts and one for your report.

1 Analytical derivations

Let us consider a regression problem, where each example $(\mathbf{x}^i, y^i) \in \mathbb{R}^p \times \mathbb{R}$ is drawn i.i.d. from a distribution characterized as follows:

- The input $\mathbf{x}^i$ is drawn from a density $p(\mathbf{x})$;
- $y^i = h(\mathbf{x}^i) + \epsilon^i$, with h a function from $\mathbb{R}^p$ to $\mathbb{R}$;
- ϵ^i is drawn from a normal distribution $\mathcal{N}(0, \sigma^2)$.

Given a learning sample $LS = \{(x^1, y^1), \ldots, (x^N, y^N)\}$ of N pairs, let us denote by $LS_x = \{x^1, \ldots, x^N\}$ the inputs of the learning sample examples and by $LS_y = \{y^1, \ldots, y^N\}$ their outputs. We consider in this question estimators $\hat{f}_{LS}$ that are linear in the y_i, i.e. such that:

$$\hat{f}_{LS}(\mathbf{x}) = \sum_{i=1}^{N} w_i(\mathbf{x}; LS_x) y^i, \tag{1}$$

where the weights $w_i(\mathbf{x}; LS_x)$ only depend on $\mathbf{x}$ and the inputs in LS_x.

(1.1) Show that both linear regression and the k-nearest-neighbors method fit into this class of estimators.

(1.2) Explain why regression trees do not fit into this class, despite the fact that all their predictions are averages of outputs y^i of learning sample examples.

(1.3) Decompose the following expected conditional mean square error into a residual error, a bias and a variance term at a fixed point $\mathbf{x}_0$:

$$E_{LS_y|LS_x}\{E_{y|\mathbf{x}_0}\{(y - \hat{f}_{LS}(\mathbf{x}_0))^2\}\}. \tag{2}$$

This decomposition takes thus into account only the variability coming from the outputs of the learning sample examples.
hint: adapt the decomposition from the course to the fact that expectations over E_{LS} are replaced by expectations over $LS_y|LS_x$ and then plug (1) into the resulting terms. In the end, the different terms should be expressed using the problem and method parameters only and not contain any expectation.

[1] `https://www.gradescope.com`, Entry code: N8VE5V.

(1.4) Simplify further this decomposition in the case of the k-NN model and use this result to discuss the effect of k on each term of the bias-variance decomposition.

(1.5) Let us now consider the regular bias-variance decomposition of the following expected error:

$$E_{LS_x, LS_y}\{E_{y|\mathbf{x}_0}\{(y - \hat{f}_{LS}(\mathbf{x}_0))^2\}\}. \tag{3}$$

Briefly discuss intuitively how the terms of the decomposition of this error relate to the corresponding terms of the decomposition of (2) averaged over LS_x (are they equal/smaller/greater relative to each other and why?).

2 Empirical analysis

In this section, we assume that we have access to a (large) dataset (or pool) of N_S pairs $P = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^{N_S}, y^{N_S})\}$ from which we would like to design an experiment to estimate the bias and variance of several learning algorithms. We assume that N_S is large with respect to the size N of learning samples for which we want to estimate bias and variance. In the questions below, the different terms are to be understood as their mean over the input space ($E_{\mathbf{x}}$), not at a particular point $\mathbf{x}_0$ as in the previous section.

Data. For the experiments below, we propose to use the California housing dataset available in scikit-learn[2]. This dataset contains $N_S = 20640$ samples, described by 8 inputs. The goal is to predict house prices, as they were in 1990, in various California districts.

Note: in the following questions, we do not tell you precisely how to set all parameter values or ranges. It is your responsability to choose them wisely to illustrate the expected behaviors.

(2.1) Explain why estimating the residual error term is very difficult in this setting.

(2.2) Describe a protocol to nevertheless estimate variance, the expected error, as well as the sum of the bias and the residual error from a pool P. Since the residual error is constant, this protocol is sufficient to assess how method hyper-parameters affect biases and variances.

(2.3) Implement and use this protocol on the given dataset to estimate the expected error, variance, and the sum of bias and residual error, for kNN, ridge regression, and regression trees. For all three methods, plot the evolution of the three quantities as a function of its main complexity parameter (respectively, k, λ[3], and tree depth) on bias and variance. You can fix the learning sample size N to 500 for this experiment. Briefly discuss the different curves with respect to the theory.

(2.4) For the same three methods, show the impact of the learning sample size on bias and variance. In the case of kNN and ridge regression, choose one particular value of k and λ respectively. In the case of regression trees, compare fully grown trees with trees of fixed depth.

(2.5) One generic method to reduce variance is bagging (for "bootstrap aggregating"), which consists in growing several models from bootstrap samples drawn from the original LS and then to average their predictions. Apply this bagging[4] idea on both linear regression (for a fixed λ) and regression trees (fully grown) and evaluate its impact on bias and variance. Discuss the results. In particular, discuss the interest of bagging when combined with both methods.

(2.6) From all these experiments, what can you say about the value of the residual error for this problem?

[2] A python script is provided on the project website to retrieve it.
[3] It is denoted λ in the lecture slides. In scikit-learn, this parameter is denoted α.
[4] You can use its implementation in scikit-learn.