STATISTICAL MACHINE LEARNING (SML2019)
5. COMPUTER LAB
Ensembling

Jan Drchal

1 Overview

The goal of this laboratory lab is to experiment with ensemble methods. Given an implementation of regression tree, your task will be to implement Random Forest and Gradient Boosted Trees algorithms. In both cases you will fit regression models to a) a simple one-dimensional *sin* dataset (see Figure 1), b) more intricate Boston housing data[1].

2 Download

The Python 3 sources and data can be downloaded from this link:

`https://cw.fel.cvut.cz/wiki/_media/courses/be4m33ssu/ensembling_src.zip`

The zip file contains a single commented source file `ensembling.py` and Boston housing dataset stored in `housing.csv`. Your task is to fill in or modify the code. You are free to modify any parts of the source with an exception of methods generating/importing the datasets. Unless otherwise stated, run the methods with default parameters as indicated in the corresponding class constructors.

3 Task assignment

Assignment 1 (1 points)

1. *Plot train and test RMSEs for regression trees of maximum depth* $d \in \{0, 1, \ldots, 15\}$ *fitting them on the sin dataset.*
2. *Plot the same for the Boston housing data.*
3. *Discuss results in both cases.*

Hint: Functions `experiment_tree_sin` *and* `experiment_tree_housing` *should get you started. You can employ the* `generate_plot` *for all assignments.*

Assignment 2 (3 points)

1. *Implement Random Forest by completing the* `RandomForest` *class.*

[1]The dataset is described here: `https://www.kaggle.com/c/boston-housing`. We use only the *train* part here.

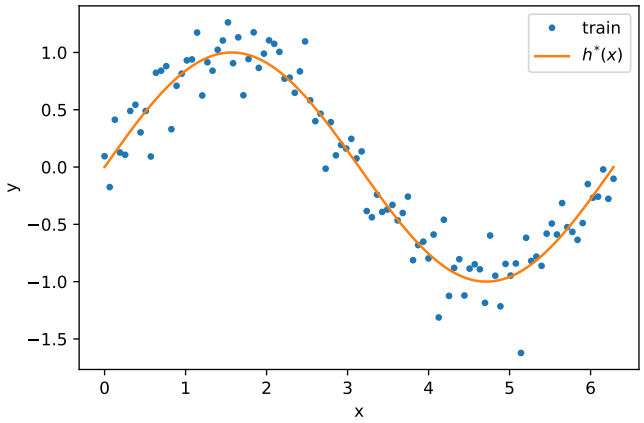

Figure 1: **Sin dataset.** The training set consists of 100 samples $(x_i, y_i) \in \mathcal{T}^{100}$, $i = 1, \ldots, 20$, where $x_i \sim \mathcal{U}(0, 2\pi)$ and $y_i \sim \sin(x_i) + \mathcal{N}(\mu = 0, \sigma = 0.2)$. The test set contains 1000 samples generated in exactly the same way. Note that $h^*(x) = \sin(x)$ is the optimal predictor.

2. *Plot train and test RMSEs for the Random Forests of* $K \in \{1, 2, 5, 10, 20, 50, 100, 200\}$ *trees trained on the sin dataset. Do not limit tree depths.*
3. *Plot the same for the Boston housing data.*
4. *Plot train and test RMSEs for Random Forests of* $K = 200$ *trees comparing results when a) two, b) half or c) all attributes*[2] *are considered when splitting a node in* `RegressionTree.build_tree()`*. Run for Boston housing dataset, only.*
5. *Discuss results in all cases.*

Assignment 3 (4 points)

1. *Implement Gradient Boosted Trees algorithm by completing the* `GradientBoostedTrees` *class. Use squared loss and a fixed learning rate* β *(see lecture slides).*
2. *Plot train and test RMSEs for the GBTs of* $K \in \{1, 2, 5, 10, 50, 100, 200, 500, 1000\}$ *trees trained on the sin dataset. Set maximum depth of trees to* $d = 1$ *(decision stumps). Give four figures for the following values of the learning rate* $\beta \in \{0.1, 0.2, 0.5, 1.0\}$*.*
3. *Plot the same for the Boston housing data.*
4. *Discuss results in all cases.*

Hint: You can use a regression tree with maximum depth $d = 0$ *as an initial model* $f_0(x)$*.*

Assignment 4 (2 bonus points)

1. *Parallelize your implementation of the Random Forest from Assignment 2.*
2. *Measure speedup on Boston housing for* $K = 1000$ *trees.*

[2]This is the default setting, see `max_features` in both `RegressionTree` and `RandomForest` constructors.