

# Pattern Recognition and Machine Learning:

## Homework 4

Qingru Hu 2020012996

March 23, 2023

### Problem 1

(1)

Use the linear property of expectation and expand the square of  $E_{COM}$ :

$$E_{COM} = \frac{1}{M^2} \left( \sum_{m=1}^M \mathbb{E}_x[\epsilon(x)]^2 + 2 \sum_{m \neq l}^M \mathbb{E}_x[\epsilon_m(x)\epsilon_l(x)] \right)$$

All prediction model errors are zero-mean and uncorrelated, so the latter part disappears:

$$E_{COM} = \frac{1}{M^2} \sum_{m=1}^M \mathbb{E}_x[\epsilon(x)]^2$$

We notice:

$$E_{AV} = \frac{1}{M} \sum_{m=1}^M \mathbb{E}_x[\epsilon(x)]^2$$

Therefore:

$$E_{COM} = \frac{1}{M} E_{AV}$$

### Problem 2

(1)

See the decision\_tree.ipynb.

(2)

`make_split(variable, value, data, is_numeric)`

**Input:**

**variable**, which is a str, the feature used to split the node;

**value**, which is either a number or str, the decision value for split, can be a quantitative value or a categorical feature;

**data**, which is a pandas dataframe, the subdataset at the split node. Each item of data represents whether the person is obese (1) or not (0).

`is_numeric`, which is a bool, whether the split feature is numeric or categorical.

**Return:**

`data_1`, which is a pandas dataframe, one child node dataset after split;

`data_2`, which is a pandas dataframe, the other child node dataset after split.

`get_best_split(y, data)`

**Input:**

`y`, which is a str, the label, that is 'obese' in this data;

`data`, which is a pandas dataframe, the dataset at the node, constaining the features and labels;

**Return:**

`split_variable`, which is a str, the feature that has the maximum IG at this node;

`split_value`, the decision value for the split feature;

`split_ig`, the value of the maximum IG;

`split_numeric`, which is a bool, whether the split feature is numeric or categorical.

(3)

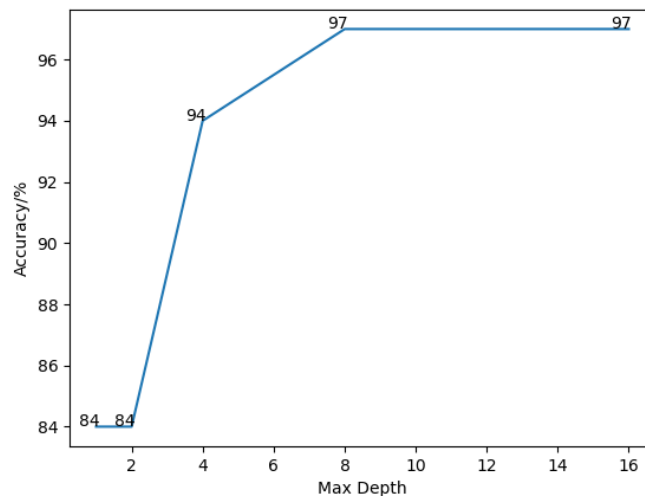


Figure 1: The validation accuracy

The validation accuracy increases with the max depth of the decision from 1 to 8, but after 8 the validation accuracy remains the same when the max depth equals 8 and 16. Since the validation doesn't decrease, overfitting does not occurred.

The two other hyperparameters `min_samples_split` and `min_information_gain` have succesfully prevented the overfitting.