Assignment_1

CMPT459 Fall 2020

Data Mining

Martin Ester

Assignment 1

Due 11th October 2020 23:59 PM

[Total Marks: 100]

Assignment 1.1 [20 marks for a), 20 marks for b)]

In class, we have covered the most common type of decision trees, namely decision trees that predict a categorical variable. These decision trees are also called classification trees. In this assignment, we will work on another type of decision trees, called regression trees. We consider the special case of a regression tree for numerical data (both the features, x, and the variable to be predicted, y, are numerical).

a) The metrics for choosing a split attribute for a classification tree (information gain or gini-index) are not meaningful if y is a numerical variable. Propose a metric for the quality of a split attribute for a regression tree.

b) In the simplest approach, the regression tree labels a leaf node (and all test examples that fall into this leaf node) by the mean of the y-values of all training examples whose x-valued correspond to this leaf node. For the case that the features (attributes x) are all numerical, propose a more accurate approach to predict the y-value for a test example. What is the drawback of this approach?

Assignment 1.2 [25 marks for a), 35 marks for b)]

a) Present the pseudo-code for a simple Random Forest for categorical data (no numerical attributes). The method for training and testing a Random Forest should have the following input parameters

TrainAndTestRandomForest(trainingdata, numberOfTrees, percentageOfAttributes, testdata)

and return a file with predictions for the test data.

The method should do the following:

- Train numberOfTrees different trees. [5 marks]
- The trees are grown deep, i.e. they are grown until all training examples corresponding to a leaf node belong to the same class. [5 marks]
- Each tree is trained on the entire training dataset. [5 marks]
- For each tree, a different randomly selected percentageOfAttributes is considered as potential split attributes. [5 marks]

- The information gain (or entropy) is used as a split criterion. [5 marks]

b) Implement the Random Forest according to your pseudo-code.

The dataset contains data in CSV format and has a first row with the attribute names. The last attribute is the label, the other attributes are the features for the classifier. You can download these files from Coursys here *(https://coursys.sfu.ca/2020fa-cmpt-459-d1/pages/Assignment_1_Datasets_)* .

The method returns a file (with one row per test example) containing the predicted classes for the test dataset.

Follow these instructions for your implementation:

- Implement the Decision Tree algorithm.
- Implementing the Decision Tree algorithm correctly would be the first step. We have provided a small dataset interviewee.csv. This dataset is simple enough so you can figure out the rules that your Decision Tree should have.
- Implement your own function to calculate Information Gain to determine the best Split. [7 marks]
- The tree is grown so that the leaf nodes are pure, i.e. they are grown until all training examples corresponding to a leaf node have the same label. [7 marks]
- Implement a method to predict the label using your trained Decision Tree. [7 marks]
- Implement the Random Forest algorithm with two input parameters numberOfTrees and percentageOfAttributes.
- Grow numberOfTrees Decision Trees. For each Decision Tree randomly sample a percentage of the Attributes to determine the best Split. [7 marks]
- Implement a method to save the prediction labels for the test dataset in a csv file predictions.csv. Please print the numberOfTrees, percentageOfAttributes, and Accuracy on the test dataset. Use the majority vote to predict labels. [7 marks]
- You might want to use a larger dataset to test your Random Forest implementation. There is another dataset banks.csv (for training) and banks-test.csv (for testing) attached for this. You should try out different values for numberOfTrees and percentageOfAttributes to determine the optimal values.

You can use libraries including math,numpy, scipy, random, etc.

You MUST provide YOUR OWN implementation for the Information Gain calculation, Decision Tree algorithm, Random Forest algorithm and the method to make Predictions. These MUST be implemented from scratch i.e. not using scikit-learn libraries.

You will be marked on the correctness of your implementation.

Updated Tue Oct. 06 2020, 13:47 by ester.