Experiment 6: Decision Tree

November 16, 2018

1 Description

In this exercise, you need to implement Decision Tree.

2 Data

To begin, download ex6Data.zip and extract the file from the zip file.

The file is a comma separated file (csv), which is actually a wine dataset. This dataset is often used for evaluating classification algorithms, where the classification task is to determine whether a wine quality is over 7.

We have mapped the wine quality scores for you to binary classes of 0 and 1. Wine scores from 0 to 6 (inclusive) are mapped to 0, wine scores of 7 and above are mapped to 1. You will be performing binary classification on the dataset.

Each line describes a wine, using 12 columns: the first 11 describe the wine's characteristics (details), and the last column is a ground truth label for the quality of the wine (0/1). You must not use the last column as an input feature when you classify the data.

3 Decision Tree

In this task, you will implement a well-known decision tree classifier. The performance of the classifier will be evaluated by 10-fold cross validation on a provided dataset. Decision trees and cross validation were covered in class. You will implement a decision tree classifier from scratch using either Python or Matlab.

You should not use existing machine learning or decision tree libraries.

3.1 Implementing Decision Tree

In your decision tree implementation, you may apply any variations that you like (e.g., using entropy, Gini index, or other measures binary split or multi-way split). Besides, you should explain your approaches and their effects on the classification performance in the experimental report.

We provide skeleton code (here) written in Python. It helps you set up the environment (loading the data and evaluating your model). You may choose to use this skeleton, write your own code from scratch in Python or Matlab.

3.2 Evaluation using Cross Validation

You will evaluate your decision tree using 10-fold cross validation. Please see the lecture slides for details. In a nutshell, you will first make a split of the provided data into 10 parts. Then hold out 1 part as the test set and use the remaining 9 parts for training. Train your decision tree using the training set and use the trained decision tree to classify entries in the test set. Repeat this process for all 10 parts, so that each entry will be used as the test set exactly once. To get the final accuracy value, take the average of the 10 folds' accuracies.

With correct implementation of both parts (decision tree and cross validation), your classification accuracy should be around 0.78 or higher.

3.3 Visualizing your decision tree

You'd better use the visualization library in Python or Matlab to visualize your decision tree, but you can also draw your decision tree flowchart with existing drawing tools. Once you get the decision tree flowchart, look over the structure and put it in the experimental report.