Homework 1, Machine Learning

Decision tree learning

1 Task

Your task in this homework is to implement decision tree learning algorithm. You can either:

1. Implement the model and learning algorithm by your self using your favorite programming language or
2. Use some toolkit or library and implement only the code to execute experiments.

You can do both of course if you like to.

2 Toolkits

For python the recommended toolkit is `scikit-learn`: `http://scikit-learn.org`. Look for examples in `http://scikit-learn.org/stable/modules/tree.html` and reference in `http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier`.

You can also use any other toolkit that is available for your favorite programming language.

3 Data

Use the wine dataset from `http://archive.ics.uci.edu/ml/datasets/Wine`. This dataset has 13 features and 3 classes. Class labels are in the first column in each row.

4 Handling continuous features

This sections applies to people who want to implement the learning algorithm by themselves.

Note that the wine dataset contains real-valued features. One of the most common method for finding the cost of real-valued features is to make it binary with some

threshold value t. For finding the suitable threshold value several splitting points should be considered. There are several methods for doing that[1], but one of the simplest thing to do is:

1. Sort the data according to the feature value;
2. Determine all pairs of consecutive points for which the labels are different;
3. Choose the potential split point as the middle (feature) value between those points;
4. Apply the cost (or gain) function to all of the proposed splits to find the one with minimum (maximum) value.

5 Experiments

Experiment with different options:

- cost or score function;
- maximum depth of the tree;
- minimum number of items needed for a split;
- minimum number of samples required in a leaf;
- any other options provided by the toolkit used or you think might prove useful.

If you are using a library tool then the work should contain more experiments in terms of varying different options and hyperparameters. If you are implementing learning algorithm by yourself then the experiments can be less thorough.

6 Evaluation

Evaluate the results of different experiments using k-fold cross-validation. Choose either 5 or 10 folds (or try with both). In `scikit-learn` cross-validation is implemented in `http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cross_validation`. You can also make your own implementation.

7 Report

The homeword submission must include the program code and a short write-up (preferably in LaTeX). The write-up should include a self-contained description of the task and the solution:

- What is the problem;

[1] `http://research.microsoft.com/en-us/um/people/dmax/publications/pdf/splits.pdf`

- Short description of the used method;
- Details of how to use your program;
- Descriptions of the experiments;
- Results of the experiments.

Preferably include figures that plot the cross-validation accuracy describing the dependence on hyperparameter values. In python the module `matplotlib` can be used to generate plots. These can be saved as pdf and imported easily into LaTeX.