Empirical/Programming Assignment 1

Due date: Wednesday, September 27 (by the beginning of class, both paper and electronically)

You can write your code in any programming language as long as the submitted code runs on homework.eecs.tufts.edu so that it can be tested as needed.

1 Introduction

This is a simple assignment whose main goal is to introduce you to the process of experimenting with machine learning algorithms. We will use implementations of machine learning algorithms from the weka system. Your code will embed calls to weka in different configurations in order to investigate the performance of the kNN and decision tree learning algorithms.

2 Data

In this assignment we will use the EEG Eye State Data Set available from the UCI repository\(^1\) which we have subsampled and preprocessed. In particular the original dataset has 14 features for each example. We have made multiple versions of the data with 14, 24,34,44,54,64,74,84,94 features respectively by adding random irrelevant features beyond the original 14 features. Each dataset is split into a training portion and test portion for use in experiments. The data files are accessible through the course web page.

3 Your Tasks

3.1 Sensitivity to Irrelevant Features

Write a program to run the default versions of the IBk and J48 in weka on all datasets and plot the test set accuracy as a number of features (use the options `-t` and `-T` to specify train and test files respectively). What can you conclude regarding algorithm sensitivity in this case?

Note: (1) It is acceptable to print out the accuracies, collect them, and then manually feed the data to another program and get the plots. But, I highly recommend automating the data collection and plotting process as well. This will be useful in all your assignments. (2) In this part and the next, all code other than calling weka to run IBk and J48, must be your own.

3.2 Learning Curves

In this part you will investigate the performance as a function of training set size. We will focus on the original (14 features) and the 54 features datasets. Note that these datasets have 500 training examples. For this assignment please use the following simple pseudocode to produce the learning curves:

1. Let the initial data file be `train.arff` and `test.arff`.
2. Repeat 10 times:
   (a) Randomly reshuffle the order of the examples
   (b) for \(i \in \{50,100,\ldots,500\}\)
      i. Put the initial segment of \(i\) examples (in permuted order) from `train.arff` into `traini.arff`.
      ii. Run the learning algorithm on the corresponding train/test combination and record the accuracy on the test data.

\(^1\)See http://archive.ics.uci.edu/ml/datasets/EEG+Eye+State
3. For each algorithm and each training set size calculate the average and standard deviation in performance over the 10 trials.

In order to do this you will need to write a reader for arff data files and identify the header and example portions, so that you can permute and select subsets of examples for these runs. Once results are collected plot the performance (accuracy and standard deviations as error bars) as a function of training set size. This makes 4 curves, for two algorithms times 2 datasets, which you should put together in the same plot, because trends are more visible in this way. What can you conclude regarding algorithm sensitivity in this case?

4 Submitting your assignment

- You should submit the following items **both electronically and in hardcopy**:
  1. All your code for data processing, learning algorithms, test program, and the experiments. Please write clear code and document it as needed.
     As stated above please make sure that your code runs on `homework.eecs.tufts.edu`. Please include a README file with instructions how to compile and run your code to reproduce the results of experiments. If this is nontrivial please include a script to run your code.
  2. A short report with the results and plots as requested and a discussion with your observations from these plots. Please make sure to address the questions posed above in your discussion.

- **Please submit a hardcopy** in class.

- **Please submit electronically using provide** by 4:30 (class time). Put all the files from the previous item into a zip or tar archive (no RAR please). For example call it `myfile.zip`. Then submit using `provide comp135 p1 myfile.zip`.

Your assignment will be graded based on the code, its clarity, documentation and correctness, the presentation of the results/plots, and their discussion.