COMP 135 – Machine Learning – Fall 2015

Empirical/Programming Assignment 1

Due date: Tuesday, 9/29 (by the beginning of class, both paper and electronically)

1 Introduction

In this assignment you will experiment with the *k nearest neighbors algorithm* and the *decision tree learning algorithm*, and evaluate the Relief algorithm for feature weighting and selection. You will use the weka system for some parts, and will write your own code for others.

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 for the assignment. 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 is already normalized and does not require further preprocessing. The data files are accessible through the course web page.

3 Your Tasks

3.1 Evaluating Decision Trees

Write a script to run the default version of the J48 algorithm in weka on all datasets and plot the test set accuracy as a function of the number of features (use the options `-t` and `-T` to specify train and test files respectively). What can you conclude regarding sensitivity of J48 in this case?
Please mechanize as much of this process as possible. Your program (or script) should call weka multiple times, extract the corresponding test set accuracies from weka's output, and feed that collected data to a plotting software (for example, gnuplot), ideally automatically. This initial investment is well worth it because it will save time in multiple runs and future assignments.

3.2 Reading Data Files

Similarly, write code to read data in data in `arff` format for use with machine learning algorithms. In view of implementing kNN and Relief below, you should also build in facilities to calculate the Euclidean distance, and weighted Euclidean distance between examples, and obtain specific feature values or their differences.

3.3 Implementing and Evaluating kNN

Implement your own version of k nearest neighbors. Your procedure should take three parameters corresponding to train set, test set, and the value of k. The search for neighbors can be done using linear time search - i.e., you need not worry about the computational improvements discussed in class.
As with the decision trees, run kNN on all datasets and plot the test set accuracy as a function of the number of features. Please repeat this with $k = 1$ and $k = 5$. What can you conclude regarding sensitivity of kNN in this case? How does it compare with J48?

[1] See http://archive.ics.uci.edu/ml/datasets/EEG+Eye+State

3.4 Implementing Relief and Evaluating kNN with Relief

Implement a version of the Relief algorithm discussed in class, that can be used for feature weighting and selection. For concreteness, we describe the algorithm here.

- The algorithm maintains a weight w_i for each feature in the data set. For all i, w_i is initialized to zero.
- The algorithm repeats the following m times:
  - Pick a random example from the training set – call it x. Search for the nearest example in the training set that has the same label as x - call it x^{hit}. Search for the nearest example in the training set that has the opposite label to x - call it x^{miss}. These searches use the standard Euclidean distance.
  - For every feature i, update w_i as follows: $w_i \leftarrow w_i - \text{diff}(x_i^{hit}, x_i) + \text{diff}(x_i^{miss}, x_i)$, where x_i, x_i^{hit} and x_i^{miss} are the ith features in the corresponding examples, and where for scalars (numbers) a, b we define $\text{diff}(a, b) = |a - b|$ (the absolute value of the difference).

Once m updates on all features have been done, we can get two outcomes. (1) feature selection: we view the weights as ranks and pick the top ranking set of features. In this assignment we will always pick the top 14 features, the same as the number of features in the original dataset (do not expect to be able to recover the original set exactly). (2) we view the weights as a way to modify the distance function. Since negative weights do not make sense we first replace any negative weight with zero. We then replace the standard Euclidean distance: $d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$ with a weighted version $d_w(x, y) = \sqrt{\sum_i w_i (x_i - y_i)^2}$.

3.4.1 Evaluation with and Without Relief

For each of the two options (feature selection, weighted distance) run kNN using the outcome of Relief on all datasets. Use $m = 10000$ for Relief in this part of the assignment. Plot the performance as a function of the number of features comparing to the original kNN (without Relief). Repeat this for $k = 1$ and $k = 5$. Overall, we expect two plots (for $k = 1, 5$) each having 3 lines (for the original kNN and the two Relief improvements). What can you conclude regarding sensitivity of kNN in this case?

3.4.2 Evaluating the Effect of m

Consider the dataset with 94 features only, and repeat the above evaluation for $m = 100, 200, \ldots, 1000$. Now plot the performance of kNN with Relief (two lines for the two variants) as a function of m. What can you conclude regarding the effect of m in this experiment?

4 Submitting your assignment

- You should submit the following items both electronically and in hardcopy:
  (1) All your code for data processing, learning algorithms, and the experiments. Please write clear code and document it as needed.
  (2) A short report with the plots as requested and a discussion with your observations from these plots.
- **Please submit a hardcopy** in class.
- **Please submit electronically using provide** by 1:30 (class time). Put all the files from the previous item into a zip or tar archive, for example call it `myfile.zip`. Then submit using `provide comp135 a1 myfile.zip`.

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