Empirical/Programming Assignment 2

Due date: Wednesday, October 15 (electronically by 12:00 noon and in hardcopy in class)

1 Introduction

In this assignment you will implement several variants of the Perceptron algorithm, compare their performance, and compare them to (weka’s implementation of) decision trees and nearest neighbors.

2 Data

For this assignment we use 3 datasets: diabetes.arff, sonar.arff and sport700.arff. The first 2 are from the weka data distribution. The last one is a text classification task from the well known “20-newsgroups” dataset which we have preprocessed into an arff file. In particular the task is to distinguish between posts to the newsgroups rec.sport.baseball and rec.sport.hockey. Our preprocessing uses words as features and picked the top 500 words in terms of information gain as features. To make this task more challenging we include only 700 examples in this file. The data files are accessible through the course web page.

3 Your Tasks

3.1 Processing arff files

Write code to read data in arff format and produce data for training and testing in k-fold cross validation. In addition, make sure that the folds are stratified, that is, up to rounding to nearest integer, each class splits evenly between the folds. For instance, if your dataset has 100 positive examples and 70 negative examples and $k = 10$ then each fold should have 10 positive and 7 negative examples. Make sure to randomize the folds and to seed the random number generator so that the results are easy to reproduce.

3.2 The Perceptron Algorithm and Variants

We recap here the code for Perceptron (P), Perceptron with Margin (PwM) and the Average Perceptron (AP) discussed in class. The algorithms (with parameters $\eta, \tau, I$) are as follows where examples $\vec{x}_i$ and the weight vector $\vec{w}$ are in $\mathbb{R}^m$, the labels $(y_i)$ are in $\{−1, +1\}$, the sign function gives a value in $\{−1, +1\}$, and the number of examples in the training set is $N$:

1. Initialize $v_k = w_k = 0$ for all $k \in \{1, \ldots, m\}$.

2. Repeat for $I$ Iterations
   
   (a) For each example $(\vec{x}_i, y_i)$ in training set do:
      
      • AP: $\vec{v} = \vec{v} + \vec{w}$
      • (Classify): $O = sign(\vec{w} \cdot \vec{x}_i)$.
      • (Update):
        P and AP: if $O \neq y_i$:
        PwM: if $y_i(\vec{w} \cdot \vec{x}_i) < \tau$
        For all $k \in \{1 \ldots m\}$, $w_k = w_k + \eta y_i x_k^i$

3. P and PwM: Output the last weight vector $\vec{w}$ as the final hypothesis.
   AP: Output $\frac{1}{\sqrt{I}} \vec{v}$ as the final hypothesis.
Thus, although the algorithms are on-line algorithms, in this assignment we treat them as batch algorithms and output a fixed vector for use in testing. Given a test set we can evaluate the error rate on this set in the usual manner.

Your implementation of these algorithms should add a feature with constant value 1 to all examples to account for the threshold. This should be done inside the algorithm so that the dataset files would not require changing. For parameters, use $\eta = 0.1$ and $\tau = 0.1 \ast \bar{A}$ where $\bar{A} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\|x^i\|^2}$ is the average norm of training examples. This needs to be calculated before training starts.

3.3 Experiments and Results to Report

Run the 3 perceptron variants, as well as the default versions of J48 and IBk in weka (5 algorithms in total) on the 3 datasets as follows. For J48 and IBk simply run the algorithms on the stratified folds produced by your code and record the average and standard deviation of the accuracy. For the perceptron variants do the same, evaluating the performance after 1, 5, 10, 15, 20, 25 iterations.

For each dataset, plot the accuracy of the perceptron variants as a function of iterations (please include average and $\pm$ std in the plots). In addition, include two flat lines (with $\pm$ std) representing the performance of J48 and IBk on the same dataset. Thus each plot should have 3 curves and two flat lines capturing the 5 algorithms.

Write a short report on any trends that you do see, or expect and do not see, in the results.

4 Submitting your assignment

- You should submit the following items both electronically and in hardcopy:
  (1) All your code for data processing, algorithms, and the experiments (please write clear code and document it as needed),
  (2) The plots of results and the report on your observations.

- Please submit electronically using provide by 12:00 noon: 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 a2 myfile.zip.

- Please submit a hardcopy in class (i.e., 4:30 pm).