Assignment 3: Classification

Ricco Amezcua
CS422 Data Mining
Department of Computer Science
Illinois Institute of Technology

March 26, 2013

Abstract

This is a report for the third assignment of CS422 Data Mining. In this report, two data sets are looked at: one about spam emails and another about breast cancer. Then various classifiers are tested on these data sets to check the accuracy of the classifiers.

1 Problem Statement

In this report various classifying algorithms are tested. They include: support vector machines, neural networks, naive bayes classifier, and logistic regression. My own interpretation of the perceptron algorithm is also tested. The data sets that looked at include a data set involving spam email and the other involving breast cancer. The spam email data set has various attributes about the emails including he frequency of certain characters and phrases. The emails are classed into spam or not spam. The breast cancer data set contains various medical measurements about patients. The patients are then classified as being benign or malignant. The major difference between these two data sets is size and number of attributes. The spam data set contains more than 4000 examples with 58 attributes, where as the breast cancer data set has less 1000 examples with only 10 attributes. Therefore, the results will show each classifier's adaptability to large and small data sets.

2 Proposed Solution

The data sets were first split up in to training and testing sets. 80% of the data went to training and 20% went to testing. The examples for each set were randomly chosen without replacement.

The support vector machine, neural network, naive bayes classifier, and logistic regression were all from different R packages. All attributes for the classifiers were kept at default except for the neural network. The neural network was given 4 passes to create the network.

For my perceptron algorithm, first the values had to be normalized. This was done by first finding the mean and standard deviation of each attribute. Then each value was subtracted by the mean and then divided by the standard deviation. After the data had been normalized, the perceptron algorithm was run. First an empty set of weights was created and set to 0. Then, a vector was created from one example in the data, and an extra attribute was added and set to 1. This would create the bias. The weight vector and example was then multiplied together and the *sign* function was used to determine if the multiplication return 1 or -1. If the weights correctly classified the class, they were not changed. However, if they incorrectly classified the class, the weights had to be changed. First an error value was created using the actual class minus the classification. This was multiplied by the learning rate (which was set to 0.1) and each value in the example vector. This was then added to the old weight vector to create a new vector. This process was run on every example in the data set. To correctly form the weights, the algorithm was ran multiple times using the same weights. Using the weights from the perceptron, the testing set was classified.

For every classifier, a confusion matrix was built. From each confusiton matrix, the accuracy and error was found, which is detailed in table 1 and table 2.

3 Implementation Details

The R scripts can be found inside the *code* folder. The scripts are separated in to two scripts: question 1, question 2 . They are ran by running all of the commands in order. The code must be in the same file as the data files. The data matrices can be found in the *data* folder. The data files will contain the confusion matric for a certain classifier, the classifier's accuracy and the error.

My logistric regression algorithm was not completed in time for the report deadline.

4 Results and Discussion

Looking at the spam base data in table 1, the support vector machine classifier was the most accurate. This classifier was also the second most accurate for the breast cancer data set. In table 2, the naive bayes classifier was 96% accurate. However, interestingly, the naive bayes classifier was only 71% accurate for the spam base data set. This may be a reflection on the size of the data set. Since the spam base data set contained many more examples, it is possible that the naive bayes classifier had a difficult time classifiying all of the examples. It may also be that the spam data set had attributes that depended on each other, which the naive bayes classifier cannot take in to account. It is interesting to note that the naive bayes classifier is the only package algorithm to have less than 90% in both tables.

My perceptron algorithm scored at most 66% in both tables 1 and 2. In table 1, having more runs made the classifier less accurate. This lower accuracy is not seen in table 2, which may be a reflection on the data set size. It is uncertain as to what caused the 66% accuracy rating. It may either be that the perceptron algorithm is not suited to this type of classification or whether there was a user error when implementing the algorithm.

Table 1: Spam Base Classifiers

Classifier	Accuracy	Error
SVM	0.931	0.068
Neural Network	0.928	0.071
Naive Bayes	0.712	0.287
Logistic Regression	0.923	0.076
My Perceptron (10 runs)	0.661	0.338
My Perceptron (25 runs)	0.424	0.565

Table 2: Breast Cancer Classifiers

Classifier	Accuracy	Error
SVM	0.954	0.045
Neural Network	0.941	0.058
Naive Bayes	0.964	0.035
Logistic Regression	0.941	0.058
My Perceptron (10 runs)	0.661	0.338
My Perceptron (25 runs)	0.661	0.338

5 References

1. http://cran.r-project.org/manuals.html
2. http://en.wikipedia.org/wiki/Perceptron