CSCI447: Analysis of Traditional Machine-Learning Algorithms on Real-World Data.

Christopher R. Barbour, Brandon Fenton, John Sherrill

September 9, 2015

Abstract

Hello, this is our abstract.

Contents

1 Introduction

The purpose of this study is to assess the classification performance of 10 machine-learning algorithms on 10 datasets from the UCI repository of machine-learning datasets. The algorithms, listed in section 2.1, cover a broad spectrum of classification techniques with varying degrees of assumptions, computational intensity, and inductive biases. The latter of these will be the focus on our performance hypothesis of each dataset and interpretation of results.

2 Experimental Design

For each of the selected datasets and algorithms, we will randomly partition the examples into a training dataset which will be used to construct the classifier and a test dataset which will be used to test the classifiers ability to generalize, or correctly classify similar examples from the same population of interest. The training set will be approximately 2/3 of the dataset and the training set will be approximately 1/3 of the dataset. 100 repetitions will be performed to obtain distributions for the three loss functions chosen for algorithm performance analysis. Each algorithm will be heuristically tuned to each dataset prior to the experiment using a single run 10-fold Cross-validation. The measures used to quantify this classification ability will be one minus the misclassification rate, the weighted average F-measure, and the weighted average Area under the ROC curve. These measures were chosen to assess different capabilities of each algorithm. Misclassification rate measures.... . Weighted Average F-measure related to .... . Weighted Average Area under the ROC assess.... .

2.1 Datasets

The datasets selected were chosen to contain a classification problem, be free of missing attributes or classes, and be of somewhat interest to the authors. Prior to the experiment, variables with each example having a unique value (i.e. a sample identification, or name of an animal) was removed. Table 1 displays the name of our datasets with brief descriptions, the types of attributes present, the number of classes for prediction, and the number of total examples.

2.2 Algorithm Descriptions

The algorithms implemented in the experiment are displayed below in Table 2. As stated earlier, individual tuning to each dataset was done prior to the experiment, and these results are displayed in the supplementary information. Some options remained fixed during all of the experiment, and these are discussed below along with certain examples of how tuning was performed.

2.2.1 Simple nearest neighbor (IB1)

These algorithms classify instances by way of referencing a set X of classified instances. For a given instance x of unknown class, the algorithm searches X for an element y that most closely resembles x in attribute. Various metrics may be used for defining "resemblance". The class of x is then deteremined to be the class of y.

Inductive biases: assuming that points close together are alike.

2.2.2 K-nearest neighbor (IBk)

This is a generalization of the simple nearest neighbor algorithms. For instance x to be classified, the K-nearest neighbor algorithm searches a reference set X for the K elements most closely resembling x (again, this is dependent upon the metric choice). The class of x is given by the most frequent class of the K elements found.

Inductive biases: assuming that points close together are alike.

First developed in 1951 by Fix and Hodges.

Nearest Neighbor Search:: A Database Perspective. Apostolos N. Papadopoulos. 2005.

2.2.3 Naive Bayes

2.2.4 Logistic regression

2.2.5 Decision tree (J48)

These algorithms classify instances by forming a cascading decision making mechanism in the form of a "descision tree". The tree is constructed such that attributes found to carry more information about the class of the instances are used earlier in the mechanism than those attributes found to carry less information.

J48 is an open source Java implementation of Quinlan's C4.5 algorithm. The reference for that particular flavor of Decision tree algorithms is Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

Inductive biases may include: a preference for shorter trees, trees that place high information gain attributes close to the root are preferred over those that do not, selection of the first functioning tree (the learning algorithm is a greedy algorithm). Linear separability of domain (reference for this statement?)

2.2.6 Ripper (JRip)

2.2.7 Support vector machine (LibSVM or SMO)

2.2.8 Feedforward neural networks (Multilayer Perceptron)

These algorithms classify instances by forming a directed graph that contains no cycles, that is, the information passes through the constructed network in a forward only direction. The graph consists of connected nodes (neurons) that either engage or don't based upon signals presented to them from other neurons upstream. The signals are added in a weighted manner and the neuron fires if a specified threshold is met.

Back-propagation is the most common learning method for multi-layer perceptrons. Neurons feed other neurons by manner of weights and ideal weights are found by utilizing gradient descent on a specified loss function where the loss function is minimized over the space of possible weights.

Inductive biases include: starting point for gradient descent optimization, require linear separability of domain.

Pattern Recognition and Image Preprocessing. Sing T. Bow. 2002.

2.2.9 Kernel neural network (RBFNetwork)

An RBF Network is essentially a neural network with that uses radial basis functions as the activation function for neurons.

Broomhead, D. S.; Lowe, David (1988). Radial basis functions, multi-variable functional interpolation and adaptive networks (Technical report). RSRE. 4148.

2.2.10 Ensemble (Adaboost)

2.3 Hypothesis

Which algorithm do we think will work best with each dataset. There are natural coparisons between certain algorithms (e.g. JRip and C4.5).

2.3.1 Abalone

I would think the nearest neighboor algorithms would perform well, as instances with predictors 'close together' (in a metric sense) would likely be related. SVM I would think would perform well for the same reasons.

2.3.2 Car Evaluation

This is faked data, 'derived from a simple hierarchical decision model' and I would think algorithms that model hierarchies in some regard would perform well (e.g. hierarchical decision trees, feed-forward neural networks, Ripper). I would think that learners are 'metric' oriented would perform poorly as closely spaced instances could more likely be classified differencely.

2.3.3 Contraceptive Method

I believe decision trees will work well because I imagine there are a few features that are far more important than the others (wife's education, number of children ever born, religion). I don't think SVMs will work well because intuitively it doesn't seem like there would be very distict groups (three distict groups for this data set in particular) to classify.

2.3.4 Ecoli

I have absolutely no idea what type of learner would perform well with this data. I'd put all my money on either a naive bayes learner or the boosting methods.

2.3.5 Flag

I would think that a naive bayes learner would not perform well because conditional independence seems to be fairly strongly violated with some of the color features. Maybe SVM or nearest neighbor will work well simply because we're trying to classify the continent feature.

2.3.6 Tic Tac Toe

Again, one feature leads to another so I would go with decision trees.

2.3.7 Wine

It looks like some of the features are missing so perhaps SVMs and tree oriented learning will not perform well. Mabybe naive bayes/logistic regression will work well. Independence may be an acceptible assumption for this data.

2.3.8 Wine Quality

Since quality is determined by human taste, I think that neural networks will work well simply because those learners seek to model rudementary cognative function. I do not think SVMs, naive bayes, or logistic regression will work well. Linear seperability will be an issue, there will be much noise, and classes will not be sepereated by wide margins... I imagine.

2.3.9 Yeast

No idea so I'm going with Adaboost.

2.3.10 Zoo

Decision trees will mostly likely perform the best here (J48 but JRip as well) because heirarchical modeling is the definition of what is being classed. As long as all the relevant features are included in the dataset, which appears to be the case, then these will perform well. I don't think naive bayes will work well as independence is clearly violated (it's a safe bet that if you have wings, you don't have gills).

3 Results

4 Discussion

5 References

Supplementary Information

Tuning Parameter Information

Supplemental Plots of Results