Comparison of Supervised Learning Algorithms

Abstract

Throughout the course of COGS 118A, we have learned many algorithms for supervised learning problems. These methods vary in terms of complexity, speed and ease of use. This report will benchmark Support Vector Machines, Random Forest, and Logistic Regression Classifier in an empirical experimentation. For each classifier, we train it on 3 datasets with varying partition schemes (20-80, 50-50, 80-20) and using 3 trials to reduce variance and randomness. Each time, we also do cross validation to find the best hyper-parameters corresponding to the classifier being used. The only performance metric being used here is accuracy.

Introduction

Based on the empirical study by the department of Computer Science at Cornell (Caranua, 2006), this report will evaluate the performance of 3 popular machine learning algorithms for supervised learning problems.

Methodology

Learning Algorithms

We used GridSearchCV to explore different hyperparameters of each algorithm to reach the best performance, while keeping the study computationally feasible. This section will summarize the algorithm and the parameters used.

**Support Vector Machine:** Using the classification implementation of SVM from sklearn [REFERENCE]. It is a C-Support Vector Classification machine. The implementation is based on libsvm [REFERENCE], which is a well known open source library for SVM written in C++ by developers at the National Taiwan University. We use linear and polynomial kernels, with degree 2 and 3 for the latter. The regularization parameters is varied by factors of ten from 10-7 to 103 with each kernel. The gamma it set to 1/[number of features].

**Logistic Regression Classifier:** This class implements the regularized logistic regression using the ‘linear’ library, with ‘newton-cg’, ‘sag’, ‘saga’, and ‘lbfgs’ solvers. We use L1 and L2 norm for penalization parameter. On each penalty parameter we vary the regularization strength by factors of ten from 10-8 to 105.

**Random Forest Classifier:** This is a meta estimator that fits decision tree classifiers on sub-samples of a dataset and combines the average to improve the accuracy and reduce overfitting. Number of trees in the forest is set to 1024. Max depth is set to 10. For the number of features to consider (max features), we use [1,2,4,6,8,12,16,20].

Performance Metric

The only performance metric we are using is accuracy since it is the most important metric learned in COGS 118A. Each training we report the best accuracy for GridSearchCV and the chosen hyperparameters. The accuracy is averaged in 3 trials to rank the classifiers. The accuracy is averaged out to remove randomness in the data or training process.

Calibration Methods

Since some of the learning algorithms do not work well with imbalanced data or skewed data, we introduce a scaler to transform the data. The MinMaxScaler transforms features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one. [REFERENCE] I chose this scaler as it works better for cases in which the distribution is not Gaussian, or the standard deviation is very small.

**Data Sets**

The 3 dataset I used for this experiment are DOTA\_2\_GAME\_RESULTS, CONNECT\_4, and ADULT from the UCI Repository [REFERENCE]. Due to limitation of time and computing power, I only use the first 10000 entries of each dataset. 3 fold cross validation is used with GridSearchCV in conjunction with 3 partition schemes (20%-80%, 50%-50%, 80%-20%).

Dota2 Games Results Data Set: Dota 2 is a popular computer game with two teams of 5 players. At the start of the game each player chooses a unique hero with different strengths and weaknesses. [REFERENCE] Each row consist of a label to indicate which team won the game, the Cluster ID (location), game mode, game type, and 113 columns of one hot encoded heroes chosen in the match.

Connect-4 Data Set: The data set contains legal 8-ply positions in the game of connect-4 in which neither player has won yet, and in which the next move is not forced [REFERENCE]. The label indicates win, loss, draw for the first player. I modified the dataset to a binary classification by removing all drawn games, so one player must win.

Adult Data Set: This dataset predicts the income of an adult. The label corresponds to a person making over $50k/year or not.