Timing Comparisons for Supervised Learning of a Classifier

Author One, Udacity, Machine Learning Nanodegree
Author Two, Udacity, Machine Learning Nanodegree
Author Three, Udacity, Machine Learning Nanodegree
Author Four, Udacity, Machine Learning Nanodegree

Abstract

This paper describes n methods for fitting a binary supervised learning classifier on a single large dataset with multiple-typed features. Timing and accuracy metrics are presented for each method, with analysis on the results in terms of the structure of the data set. Fits were performed using the popular open-source machine learning library `scikit-learn`. Additionally, a code repository including all necessary infrastructure has been developed and shared for reproducibility of results.

System Design

For portability and reproducibility of results, we have elected to use the Docker system and its `Dockerfile` syntax to prepare. As this work is done using Python and its `scikit-learn` libraries we have elected to use a system built via the Anaconda package manager. Furthermore, leveraging images designed by and for using the Jupyter system, which is built via Anaconda, allows a single container to be used both for running the analysis script and for interactive analysis of the data via Jupyter. The following `Dockerfile` completely describes the system used for this work. Note that it inherits from a Docker image designed and maintained by the Jupyter team.

`mlnd/tcsl Dockerfile`

```
FROM jupyter/scipy-notebook
VOLUMES .:/home/jovyan/work
```

Via the above, fit analysis can be run on a single classifier,

```
$ docker run -e CLASSIFIER='decision tree' mlnd/tcsl python project.py
```

all classifiers,

`$ docker run mlnd/tcsl python project.py`

or via an interactive notebook server

`$ docker run mlnd/tcsl`

Note that the last leverages a built-in launch script inherited from the original notebook definition, in that no explicit command was passed to the container.

Data Set

Select a dataset Proposed requirements: - Large but not too large i.e. can fit on a single system running Docker - lends itself to binary classification - many different types of feature parameters - from UCI Machine Learning Dataset Library

Data Visualization

Feature Engineering

one-hot encode classification parameters convert all booleans to numeric values

Split Data Set

- training
- test
- use seed for reproducibility

Models

For each model complete the following: Copy and paste this template to add a new model. **PUT YOUR NAME NEXT TO ONE YOU WOULD LIKE TO IMPLEMENT**

name

> brief description
> time complexity, training
> time complexity, prediction

strengths
Weaknesses

Support Vector Machines (Matt)

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Decision Trees

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Naive Bayes

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Ridge Regression

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Stochastic Gradient Descent (Joshua)

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Adaptive Moment Estimation (ADAM)

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Linear/Logistic Regression

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

K-nearest Neighbors (Matt)

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Random Forests

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

XGBoost (may require additional lib) (Matt)

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Linear Discriminant Analysis

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Quadratic Discriminant Analysis (Joshua)

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Gaussian Processes

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Elastic Lasso

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

AdaBoost

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Gradient Tree Boost (Joshua)

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

Perceptron

brief description
time complexity, training
time complexity, prediction
strengths
Weaknesses

List of Supervised Learning Models:

http://scikit-learn.org/stable/supervised_learning.html

Metrics

What metrics should be used for timing, for accuracy, others?

Pipeline

1. raw fit of classifier
2. raw prediction of classifier
3. gridsearchCV fit
4. prediction on tuned model

Analysis

Highest performing model What this says about the data set chosen