Machine Learning Algorithms in Spark

Abhishek Malali (abhishekmalali@g.harvard.edu)
Neil Chainani (chainani@g.harvard.edu)
Leonhard Spiegelberg (spiegelberg@g.harvard.edu)

November 9, 2015

1 Background

Spark's machine learning library MLlib has vast functionality but unfortunately lacks certain classifiers. Ordinal logistic regression is one of those missing classifiers, and is particularly useful for ordinal dependent variables, such as rating systems and surveys. We propose to design an ordinal logit library built on top of Spark that is optimized to take advantage of Spark's parallelism.

2 Objectives - Functionality and Performance

To implement the machine learning algorithms in Spark while benchmarking the code against Scikit-learn and MLlib which is the standard Machine learning library for Spark. A part of the project will also focus on fitting the data to the problem in the best possible way while ensuring all standard ML rules are followed.

3 Models to be implemented

3.1 Ordinal regression

Ordinal regression is a variation of a linear regression model which allows for the prediction variable to take ordinal values only. In a first version, we want to implement a standard ordinal regression model and parallelize the learning procedure for it. Other components involve implementing parallelized versions of functions to address the root mean squared error, p-values and the f-statistic.

3.2 Bayesian ordinal regression

As the standard ordinal regression model might overfit data one usual approach is to introduce a prior. In our project the second step is to use different priors and provide an overall framework which allows the user a convenient process of model selection.

3.3 Random Forests

Random forests will be a classifier we would like to implement since the tree formation process can be split across multiple cores hence speeding up the process.

4 Design Overview

One of the critical tasks in this project is using both an efficient and parallelizable algorithm for the involved optimization problem. Therefore, we will first start with an implementation of stochastic gradient descent which we will base on http://martin.zinkevich.org/publications/nips2010.pdf.

Another target of our implementation is to provide the user a covenient framework allowing to test different (Bayesian) ordinal regression models (i.e. using different priors) with the parallel processing power of Spark.

For our implementation we will use pyspark or Scala.

5 Milestones

- 11/13 - Research and benchmark implementations(mord/scikit-learn)
- 11/20 - Preliminary implementations in Spark for ordinal regressions
- 11/27 - Parallelizing regression code and work on bayesian ordinal regressions
- 12/4 - Final analysis of current work and start work on project webpage

6 Division of work

- Leonhard - Design of algorithms for Ordinal regressions
- Neil - Implementation of benchmark code and implementation of algorithms in Spark
- Abhishek - Implementing algorithms in Spark and Parallelizing strategies

7 Verification

Once a data set is determined, we will use the existing ordinal logit libraries in Mort or scikit-learn to establish a baseline. We will evaluate performance as compared against other Spark implementations of ordinal logit found online (if they exist) and against MLlib's linear regression, by treating the ordinal data as continuous.