Machine Learning Algorithms in Spark

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

November 4, 2015

1 Background

Spark has a machine learning library called MLib which has a missing Neural networks functionality which serves as our main inspiration to implement additional ML algorithms in Spark.

2 Benchmarking Data

We intend to use the (small dataset) for initial testing which we would later scale up to (large dataset) to benchmark spark implementations against scikit-learn.

3 Objectives - Functionality and Performance

4 Models to be implemented

4.1 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.

4.2 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.

5 Design Overview

One of the critical tasks in this project is using both an efficient and parallizable 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.

6 Verification

7 Schedule, Milestones, and Division of Work