Comparing R and Spark implementation of Lasso

Jakob Gerstenlauer and German Navarro
Cloud Computing, FIB, UPC, Barcelona, Spain

Some Background

Lasso regression:

- Regularization is used to avoid **overfitting**.
- L1-regularization as opposed to L2 ridge regression.
- Leads to **sparse** solution (implicit feature selection).
- Regularization parameter: λ.
- λ has to be tuned using cross-validation.

We compare *cv.glmnet()* (*glmnet*) with *LassoWithSGD* in MLlib.

Apache Spark:

- A general purpose engine for large-scale data processing.
- **MLlib**: distributed machine learning algorithms.
- Can process data from local file system, HDFS, distributed data bases, or streams.

Methodology

Instance generator:

- Number of observations N.
- Number of features d.
- Ratio $\frac{N}{d}$.
- Signal-to-noise ratio $n \in [0, 1]$.
- Polynomial degree p.
- Generative mechanism for $p = 2$:
  $y = N(c_{x11}x_1 + c_{x12}x_1^2 + \ldots + c_{xd1}x_d + c_{xd2}x_d^2, 1 - n)$

Latin Hypercube sampling:

- n: signal-to-noise ratio,
- N: number of observations,
- $\frac{N}{d}$: observations / features,
- p: polynomial degree.

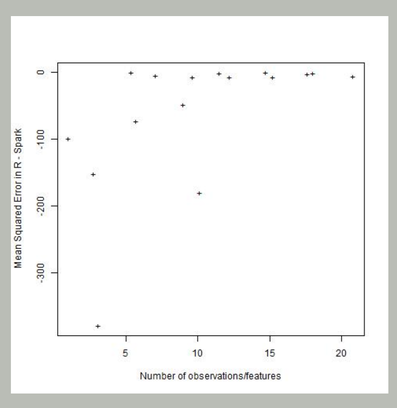
Methodological objective:

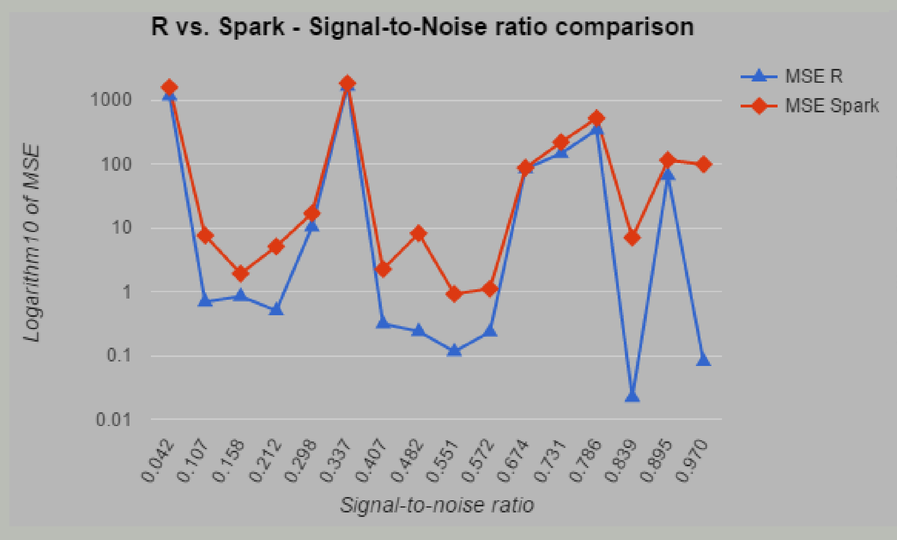
- Quantify complexity of data sets.

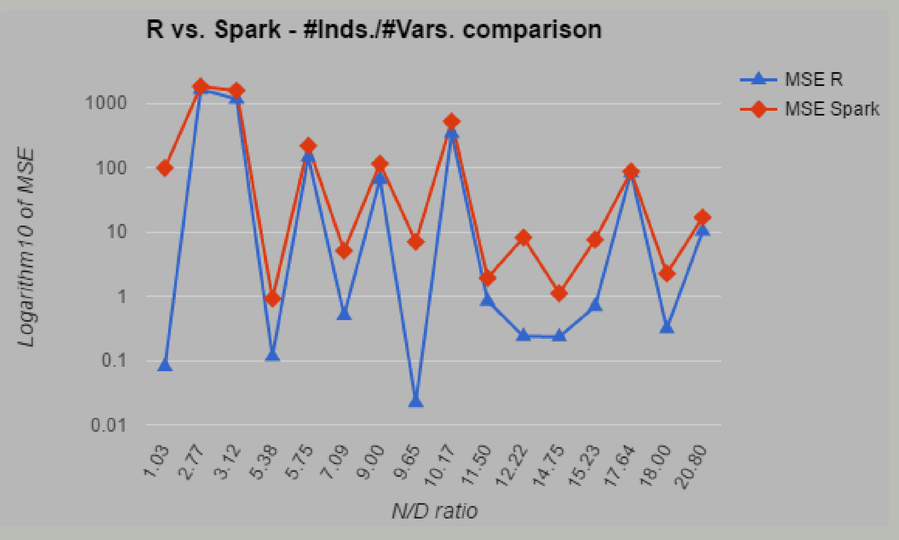
Results

Index	$p = 1$	$p = 2$
$\bar{x}$	-16.14	-105.55
SD(x)	33.61	130.08

Table: Mean and SD of Δ MSE.







Source: https://docs.google.com/spreadsheets/d/1bIL8jMmi2DNqOOnhHZfQ2aiWy5nRFdDQjJCC − 7E/edit?usp = sharing

Motivation

There is a hype about “big data analytics”:

- **Hadoop** (Mahout),
- **Spark** (MLlib),
- **Flink** (FlinkML).

But, how **accurate** and **performant** are machine learning algorithms based on distributed programming models? How does their **usability** compare with standard, centralised algorithms available e.g. in *R*?

Motivation

In this project we compared the Lasso (L1) regression model between R and MLlib (Spark) in terms of:

- accuracy,
- performance,
- usability.

Conclusions: Usability

Compared to *R*, in Spark MLlib:

- limited range of machine learning methods.
- only token implementations of some methods (i.e. SVM).
- manipulating data interactively in shell is clumsy.
- parameter-tuning requires sophisticated code.
- debugging is more difficult.

Conclusions: Accuracy and Performance

- MSE of MLlib was **always** higher for a given data set!
- The error of MLlib increased with $\frac{N}{d}$ and p.
- MLlib can not deal with difficult data sets!

Acknowledgment

This poster was reviewed by Santiago Rodrigo Muñoz and Hao Wu.

https://github.com/jakobgerstenlauer/CompareMLLibWithR {jakob.gerstenlauer,german.navarro}@est.fib.upc.edu