alteryx

Linking R to the MLlib Machine Learning Library

Dan Putler, Chief Scientist

UseR! 2015

What is MLlib?

- Spark's machine learning library, and one of Spark's five underlying libraries
  - SparkSQL is another of those libraries, and helps to provide the R data frame like capabilities of SparkR
- Consistent with its lineage, MLlib has the "vibe" of computer science rather than applied statistics

alteryx

What methods (algorithms) does it provide? Lots…

- Classification and regression models
  - Linear regression
  - Logistic regression
  - Elasticnet regularized regression
  - Support vector machines
  - Decision trees
  - Random forest model
  - Gradient boosted tree models

What methods (algorithms) does it provide? Lots…

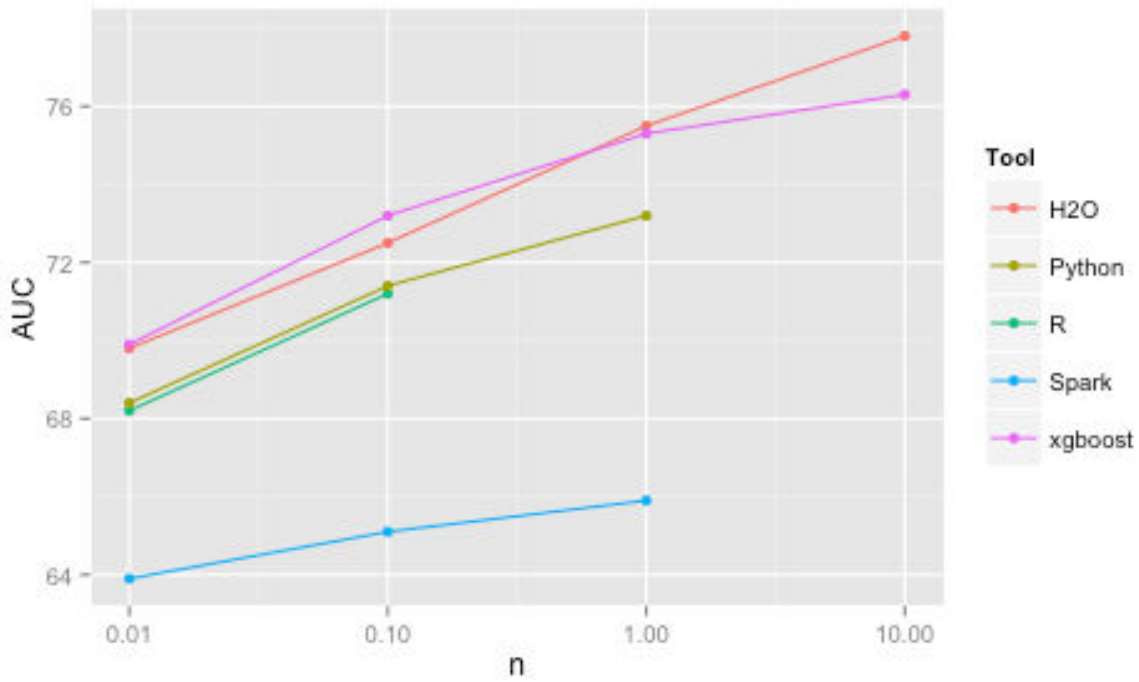
- Unsupervised learning models
  - Principal components
  - Singular value decomposition
  - K-means clustering
  - Alternating least squares
  - Cosine distance
- Descriptive statistics
  - Mean, standard deviation, min/max, etc.
- Linear algebra
  - Basic arithmetic operators
  - Transposes

Linking R to MLlib is not hard…

…but whether it is worth the effort is another question

- The regression algorithms in MLlib use either limited memory BFGS (the good ones) or stochastic gradient descent (the not so good ones)
  - The problem is that if the original start values are too far outside of the neighborhood of the optimal values, the algorithms won't converge to the solution
  - The methods used in R's lm and glm functions are ***much*** more robust
- In addition, model summary and in-sample diagnostic information is nearly nonexistent
- While we have not worked with them extensively, the decision tree based algorithms in MLlib also seem to be problematic

Szilard Pafka's benchmark of random forest implementations



Is it even worth pursuing Spark as a compute platform for R?

Yes!!!

Why? The fundamentals of Spark itself

- Spark is a general-purpose, *in-memory*, *cluster computing* system that can scale to allow for distributed computation on large volumes of data
- Cluster computing
  - Distributed data
  - Distributed, parallel computations
  - Spark handles the distributed part for you, behind the scenes
- In-memory
  - Spark allows you to persist data in-memory when performing costly computations, no disk I/O means much faster processing time
- Close integration between R data frames and Spark DataFrames, coupled with SparkR's use of the really excellent SparkSQL Catalyst query optimizer for doing data wrangling

alteryx

Making Spark a Useful Compute Environment for R

Dan Putler, Chief Scientist

UseR! 2015

So how do we do that?

- Provide parallelized linear model and generalized linear model capabilities based on those provided by R's lm and glm functions
  - Solving for the coefficients of linear regression models using least squares (normal equations) and weighted least squares
  - Solving for the coefficients of generalized linear models using Fisher scoring (iteratively re-weighted least squares)
- Include as many of the standard set of model summary and diagnostic information as is possible
- Provide capabilities similar to model.matrix to address categorical variables, and gracefully handle records with missing data in the relevant fields
- We are currently engaged in addressing these issues by developing the ***sparkGLM*** package: https://github.com/AlteryxLabs/sparkGLM

Some core technologies underpinning sparkGLM

- Spark and SparkSQL
- The ml-matrix parallelized linear algebra package for Spark: https://github.com/amplab/ml-matrix
- The Breeze linear algebra library for Scala
- The Apache Commons math library to obtain capabilities related to probability distributions

What is our progress?

- Linear models
  - The normal equations coefficient estimation is done
  - The predict method is done
  - The summary method is nearly feature complete (missing *p*-value calculations)
  - Full R bindings are close to being committed
- Generalized linear models
  - A flexible architecture for a generalized linear model method (as opposed to one only for the binomial family using a logit link) is in place
  - A specific implementation to the binomial family (but with logit, probit, and complementary log-log link functions) is in progress
- Other
  - A limited version of a model.matrix like function is completed
  - R packaging and other bindings are in the works

Why sparkGLM rather than just contributing to MLlib?

- Our goal is to eventually have much of the work we are doing rolled into MLlib, but there are institutional issues
  - MLlib is part of Spark itself, and Spark is a very large project. There are typically over 300+ open pull requests on the Spark GitHub request at anyone time
- MLlib is a “work in progress”, and to show a fairly complete proof of concept (PoC) for a possible direction change in MLlib is difficult if commits are made to of a branch of MLlib
- There are other project taking a similar PoC approach at this point in time
  - ml-matrix: https://github.com/amplab/mlmatrix
  - KeystoneML: https://github.com/amplab/keystone

Questions?

https://github.com/AlteryxLabs/sparkGLM

http://www.alteryx.com