Advanced Analytics: Using Spark and R to explore data and leverage advanced and predictive analytics

Anthony F Maresco
January 17, 2018
Safe Harbor Notice

This presentation describes features that are under development by MicroStrategy. The objective of this presentation is to provide insight into MicroStrategy’s technology direction. The functionalities described herein may or may not be released as shown.

This presentation contains statements that may constitute “forward-looking statements” for purposes of the safe harbor provisions under the Private Securities Litigation Reform Act of 1995, including estimates of future technology releases. Forward-looking statements inherently involve risks and uncertainties that could cause actual results of MicroStrategy Incorporated and its subsidiaries (collectively, the “Company”) to differ materially from the forward-looking statements.

Factors that could contribute to such differences include: the Company’s ability to develop, market and deliver on a timely and cost-effective basis new or enhanced offerings that respond to technological change or new customer requirements; delays in the Company’s ability to develop or ship new products; the extent and timing of market acceptance of MicroStrategy’s new offerings; continued acceptance of the Company’s other products in the marketplace; competitive factors; general economic conditions; and other risks detailed in the Company’s registration statements and periodic reports filed with the Securities and Exchange Commission. By making these forward-looking statements, the Company undertakes no obligation to update these statements for revisions or changes after the date of this presentation.
Topics

• Objective
• The Basic Architecture with SparkR
• Start Simple
• Characterizing New Data
• SparklyR
• Machine Learning and Artificial Intelligence
• Summary
• Q&A
Objective
Objectives…

- The data about the data…
- Use the infrastructure for all types of cluster-based advanced analytics
Basic Architecture
Spark Architecture

Spark SQL
Spark Streaming
MLlib (machine learning)
GraphX (graph)

Apache Spark

http://spark.apache.org/
Resilient Distributed Data Structures

Collections distributed across nodes to support distributed job.
Spark 2.X adds unified DataFrames and Datasets
Unified Data Structures Unifies Features

Unified Apache Spark 2.0 API

- DataFrame = Dataset[Row]
- Alias
- Dataset[T]

Spark SQL Optimization and Execution

https://databricks.com/blog/2015/04/13/deep-dive-into-spark-sqls-catalyst-optimizer.html
Dual Role of SparkSQL

- Internally SQL is used to simplify set processing
- SQL Enabled data structures evolved from RDDSchema, to DataFrames, to Datasets which adds names, columnar formats, type safety, and optimization
- SparkSQL thru ODBC/JDBC is yet another way to get tables persisted in Hive and uses the Spark Thrift Server
- A Spark script or program needs to persist into Hive to make data available thru SparkSQL ODBC
- There is also Hive on Spark which uses Hive SQL and executes over Spark
SparkR

• SparkR is an R package that allows you to access a spark cluster and Spark API’s from within R. You can execute Analytics mixed with Spark data pipeline processing in the cluster.

• It is part of the Spark project and installed with Spark.

• This functionality is then available from the R Integration Package in MicroStrategy.
SparkR Execution Framework
Integration of R with Spark using SparkR or SparklyR

- R integrates to Spark with SparkR or SparklyR
- MicroStrategy can execute RScript that includes packages using a metric
- You can separate model generation from model scoring
- These scripts can be used for algorithm training as well as to generate statistics for exploratory data analysis

Sources: unstructured, semi-structured, structured

Storage: HDFS, RDBMS, Hive, HBase, Cassandra, MongoDB, SOLR, Elastic, Other NoSQL

Leveraging Spark

- R integrates to Spark with SparkR or SparklyR
- MicroStrategy can execute RScript that includes packages using a metric
- You can separate model generation from model scoring
- These scripts can be used for algorithm training as well as to generate statistics for exploratory data analysis
Preliminaries

- Install Hadoop/Spark client software on MicroStrategy Intelligence Server
- R and R Integration Package installed on Intelligent Server
- RStudio installed for development. If not on I-Server then this node has to be a Hadoop/Spark client as well
- R needs to be installed on every node in the Hadoop cluster
Get Spark Summary Statistics in RStudio
Execute Script with RIP
Review the Output Table
View the Output in a Dashboard
View the Spark Execution Details
SparkDataFrame vs. R data frame
100 Billion Rows in Fact Table? Use Sample!

- `sample(SparkDataFrame, withReplacement, fraction)`

  # Create a 0.1 sample of df, without replacement
  # Random seed will be used
  subsetDF <- sample(df, FALSE, 0.1)
Characterizing New Data
Candidate Numeric Statistics

- Measures of location:
  - Mean
  - Median
- Measures of dispersion:
  - Min and Max
  - Variance
  - Standard deviation
  - Quantiles
- Measures of distribution shape:
  - Skewness
  - Kurtosis
- Measures of Dependence:
  - Covariance
  - Correlation
Candidate Category Statistics

- Distinct Counts
- Frequency table
- Relative frequency table
- Contingency table
Adding a Workflow With MicroStrategy

• Prompt for database and table name
• Run scripts in RIP metric to generate tables
  • Numeric summaries
  • Category summaries
• Run long preparation reports to history list
• View dashboard of summary statistics in tables
Summary Statistics Workflow
What is SparklyR?

- From RStudio
- Definition
- [https://spark.rstudio.com/guides/distributed-r/](https://spark.rstudio.com/guides/distributed-r/)
  Try these operations in RStudio
SparkR vs. SparklyR

• SparkR is part of the Spark distribution
• SparklyR is developed by Rstudio
• SparklyR uses dplyr
• SparkR provides access to UDF functions
• People are gravitating towards SparklyR it seems
Available SparkR ML Algorithms

- AFTSurvivalRegressionModel
- ALSModel
- FPGrowthModel
- GBTClassificationModel
- GBTRegressionModel
- GuassianMixtureModel
- GeneralizedLinearRegressionModel
- GroupedData
- IsotonicRegressionModel
- KMeansModel
- KSTest
- LDAModel
- LinearSVCModel
- LogisticRegressionModel

- MultiLayerPerceptronClassificationModel
- NaiveBayesModel
- RandomForestClassificationModel
- RandomForestRegression

https://spark.apache.org/docs/2.2.1/api/R/
SparklyR Algorithms

- Logistic Regression
- Survival Regression
- Generalized Linear Regression
- Decision Trees
- Random Forests
- Gradient Boosted Trees
- Principal Components Analysis
- Naïve Bayes
- Multilayer Perceptron
- Latent Dirichlet Allocation
- One vs. Rest
- ensure function to wrap other SparkML algorithms

https://spark.rstudio.com/mlib/
SparklyR and H2O

• H2O is an open source platform from h2o.ai that makes it to deploy AI and deep learning to solve complex problems
• Sparkling Water allows users to combine the fast, scalable machine learning algorithms of H2O with the capabilities of Spark
• rsparkling extension package provides bindings to H2O’s distributed machine learning algorithms via sparkly
• access the machine learning routines provided by the Sparkling Water Spark package
• Create and tune H2O machine learning workflows on Spark, orchestrated entirely within R

https://spark.rstudio.com/guides/h2o/
library(rsparkling)
library(sparklyr)
library(h2o)
library(dplyr)

sc <- spark_connect("local", version = "2.1.0")

mtcars_tbl <- copy_to(sc, mtcars, "mtcars")

# transform our data set, and then partition into 'training', 'test'
partitions <- mtcars_tbl %>%
  filter(hp >= 100) %>%
  mutate(cyl8 = cyl == 8) %>%
  sdf_partition(training = 0.5, test = 0.5, seed = 1099)

training <- as_h2o_frame(sc, partitions$training, strict_version_check = FALSE)
test <- as_h2o_frame(sc, partitions$test, strict_version_check = FALSE)

# fit a linear model to the training dataset
glm_model <- h2o.glm(x = c("wt", "cyl"),
  y = "mpg",
  training_frame = training,
  lambda_search = TRUE)
What’s Next?

- EDA package w/ OOB visualizations and put in github
- Add scoring examples for common use cases and algorithm choices when trained externally
- Identify and add combined training/scoring examples
Summary

- Spark offers a broad range of data processing capabilities with a unified approach for data structuring and processing in a large-scale environment
- Use SparkR or SparklyR in the MicroStrategy R Integration Package to execute Spark code and R code
- For Data Discovery we can implement an automated Exploratory Data Analysis workflow to tell us “Data about our Data”
- With the same infrastructure we can add more advanced Machine Learning, Deep Learning and Artificial Intelligence