Post 2 - SparkR in Machine Learning

Stanley Ho

December 2, 2017

SparkR in Machine Learning

Introduction

Developed in 2009 at UC Berkeley, Spark became a popular among data scientists and engineers for being faster than past big data software like Hadoop as well as being easy to use.Used in companies like Netflix, Yahoo, and Tencent (to name a few), Spark allows for companies to process petabytes of data much more efficiently.

Understanding Spark

Spark is a library of code used to parallelize and process code on a cluster. Parallelism allows for a speedup in code run time. This allows for large chunks of code that may take seconds to run, to run in milliseconds, or even from minutes to seconds. In Spark, data is broken down to pieces, sent to different computers for processing, then returned and consolidated to return the final result.

At the base of Spark is a data abstraction concept called RDD or Resilient Distributed Dataset. In Spark, RDD is a basic data structure that allows for data to distributed among clusters of nodes. There are RDD transformations and RDD actions that can be done on given data. [More on RDD transformations and RDD actions.] (https://rpubs.com/wendyu/sparkr)

So What is Special about SparkR?

In addition to the importance of Spark as discussed above, let us look at how Spark manifest in R (SparkR.) As we know, even though R is a simple syntax language that can run complex alogrithms, R struggles with memory on large datasets. This is where SparkR shines. Extra Intro

How to set up SparkR - It can be a pain!

After all this introductory material on spark, let's work with SparkR!

Downloading and Installation

First download Spark @ Link

1. Set up SparkR by setting SPARK_HOME to where you extracted the file.

```
Sys.setenv(SPARK_HOME = "C:/Users/Stanley Ho/spark-2.2.0-bin-hadoop2.7")

.libPaths(c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib"), .libPaths()))
```

2. Load Library

```
library(SparkR)
```

```
##
## Attaching package: 'SparkR'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, filter, lag, na.omit, predict, sd, var, window
```

```
## The following objects are masked from 'package:base':
##
##     as.data.frame, colnames, colnames<-, drop, endsWith,
##     intersect, rank, rbind, sample, startsWith, subset, summary,
##     transform, union
```

```
library(rJava)
```

```
## Warning: package 'rJava' was built under R version 3.4.2
```

3. Optional - Initialize Spark Context given by official documentation. Then we will use SQL. Source

```
## To initialize and specify specific Spark driver properities.
# sc <- sparkR.session(master = "local",sparkEnvir = list(spark.driver.memory="2g"))


## You may get an error loading it, stating JVM is not ready after 10 seconds. It may be due to the submit2.cmd no
t processed.
```

Spark in Machine Learning

Spark provides a simplist general machine learning library called MLlib. The break through of MLlib enables data scientists to focus on problems and models rather than worrying about data complexities. This is through an open-sourced and wide library built for most cases that data scientists encounter.

Some of the *Algorithms* supported by Spark for Machine Learning are: * Classification * Regression * Tree * Clustering * Collaborative Filtering * Frequent Pattern Mining * Statistics

Classification

To describe some algorithms supported by Spark, I'll aim to describe Classification types. Source

Problem Type	***Supported Methods***
Binary Classification	linear SVMS, logistic regression, decision trees, naive Bayes
Multiclass Classification	decision trees, naive Bayes
Regression	linear least squares, ridge regression, decision trees

Binary Classification

As the name suggest, **Binary Classification** splits items into two categories, positive and negative. Positive is represented by 1 and negative is respresented as 0 in MLlib, but in mathematic formulas, it is represented as -1.

1. *Linear Support Vector Machine (SVM)* Known as a standard method for large-scale classification tasks. Linear SVM uses a supervised learning model. This means that there is a training model that changes the weight of the nodes so that classifications are made.

   Real World Application
   Linear SVMs are used for image classifications. For example, Google Images allows for users to submit a picture and can tell the user what it is an image of.

2. *Logistic Regression* Classifciation algorithm that treats probablity as belonging to an exponential distribution. Fit coefficients for the long probability. This means that the binary logistic model is used to estimate the probability of binary response based on predictors (independent varialbes). This can be used to answer questions like, what is the probability of passing an exam versus amount of hours spent studying. The passing or failing of an exam is represented by the binary 1, and 0.

Sum Up

Overall, Spark is very useful addition to R. By adding a machine learning library, SparkR allows for data scientist to tackle huge problems without having to worry about creating methods to solve problems. I only covered a small portion of the machine learning capabilities of SparkR, but I hope there is some interest created through this post.

Take Home Messages

1. SparkR when dealing with large datasets!
2. SparkR uses parallelism to distribute tasks to help with runtime.
3. Used a lot in Machine Learning
4. Different problem types in Classification Machine Learning supported by Spark

References.

1. https://databricks.com/blog/2016/06/22/apache-spark-key-terms-explained.html
2. https://rpubs.com/wendyu/sparkr
3. http://spark.apache.org/docs/latest/sparkr.html#starting-up-sparksession
4. https://www.analyticsvidhya.com/blog/2016/06/learning-path-step-step-guide-beginners-learn-sparkr/
5. https://www.youtube.com/watch?v=5o-9ozwQgMw
6. http://spark.apache.org/downloads.html
7. https://www.codementor.io/jadianes/spark-r-data-frame-operations-sql-du1080rl5
8. https://spark.apache.org/docs/1.1.0/mllib-linear-methods.html#binary-classification