R4ML: An R Based Scalable Machine Learning Framework
About Presenter

Alok Singh

Principal Engineer at the IBM Spark Technology Center. He is leading the R4ML team at IBM. He has built/architected analytical frameworks and implemented various machine learning algorithms.

IBM Spark Technology Center

Founded in 2015.
Location:
Physical: 505 Howard St., San Francisco CA
Web: [http://spark.tc](http://spark.tc)  Twitter: [@apachespark_tc](https://twitter.com/apachespark_tc)

Mission:
Contribute intellectual and technical capital to the Apache Spark community.
Make the core technology enterprise- and cloud-ready.
Build [data science skills](http://bigdatauniversity.com) to drive intelligence into business applications — [http://bigdatauniversity.com](http://bigdatauniversity.com)

Key statistics:
About 50 developers, co-located with 25 IBM designers.
Major contributions to Apache Spark [http://jiras.spark.tc](http://jiras.spark.tc)
Apache SystemML is now an Apache Incubator project.
Founding member of UC Berkeley AMPLab and RISE Lab
Member of R Consortium and Scala Center
Agenda

- Introducing R4ML
  - What is SparkR?
  - What is Apache SystemML?
  - R4ML (Architecture and motivation)
- How R4ML helps in Machine Learning
- Demo
- Future Plans
- Q/A
Introducing R4ML ...
R4ML Stack Diagram (More details later…)
What is SparkR?
SparkR is a Thin Wrapper Over Apache Spark

Brings in the powerful statistics from R to Spark

Source: SparkR talk by Shivaram Venkatraman in UseR! 2016
What is Apache SystemML?
What is Apache SystemML

Apache SystemML™

• In a nutshell
  • Provides a language for data scientists to implement machine learning algorithms
  • Also comes with approx. 20+ algorithms pre-implemented
  • Compiles execution plans ranging from single node (scale up) to clusters (scale out on data parallel platforms MapReduce, Spark)
  • Runs in embeddable, standalone, and cluster (hybrid)
  • Supports various APIs

• Where is SystemML today?
  • Apache SystemML Incubator project
  • [http://systemml.apache.org](http://systemml.apache.org)
  • Ongoing research effort at IBM Almaden Research Center
Linear Algebra is the Language of Machine Learning.

Linear algebra is powerful, precise, and high-level. Express complex transformations over large arrays of data... in a clear and unambiguous way... using a small number of operations.
Apache SystemML (incubating)

Goal: Take the linear algebra formulation of an algorithm and make it scale automatically.

IBM Research technology, now an open-source Apache Incubating project.
Try SystemML online on the IBM Data Science Experience: http://datascience.ibm.com
L₂ regularized linear regression by the conjugate gradient method in DML:

\[ \text{X} = \text{read.csv(...)} \quad \# \ n \times m \text{ feature matrix} \]
\[ \text{y} = \text{read.csv(...)} \quad \# \ n \times 1 \text{ feature vector} \]
\[ \text{maxi} = 50; \lambda = 0.001; \ldots \]
\[ r = -(t(\text{X}) \times y) \]
\[ \text{norm}_r^2 = \text{sum}(r \times r); \ p = -r \quad \# \text{initial gradient} \]
\[ \text{w} = \text{matrix}(0, \text{ncol(X)}, 1); \ i = 0 \]
\[ \text{while}(i < \text{maxi} \& \text{norm}_r^2 > \text{norm}_r^2 \text{_trgt}) \{ \]
\[ q = ((t(\text{X}) \times (\text{X} \times p)) + \lambda \times p); \quad \# \text{compute conjugate gradient} \]
\[ \text{alpha} = \text{norm}_r^2 / \text{sum}(p \times q) \quad \# \text{compute step size} \]
\[ \# \text{update model and residuals} \]
\[ \text{w} = \text{w} + \text{alpha} \times p; \ r = r + \text{alpha} \times q \]
\[ \text{old}_r^2 = \text{norm}_r^2 \]
\[ \text{norm}_r^2 = \text{sum}(r^2); \ i = i + 1 \]
\[ \text{p} = -r + \text{norm}_r^2 / \text{old}_r^2 \times p \]
\[ \} \]
\[ \text{write.csv(w, ...);} \]
R4ML Details...
- Motivation
- Data Flow (simplified)
- Architecture
R4ML Goals

• R4ML is an open-source scalable machine learning R package for doing end to end ML flow.
• Primarily focus on R users (API support, functionality)
• Support all the major features of SparkR (since it is built on top of SparkR)
• Support all the major features of Apache SystemML
• Ability to let user write highly performant custom ML algorithms
• User can switch back and forth between SparkR and R4ML
• In Summary, it is bridging the gap in SparkR and uses best of Apache SystemML and Apache Spark to make data scientists and analysts get the most out of big data analytics
R4ML Data Flow Diagram (Simplified)
How R4ML helps with scalable Machine Learning
A Typical ML Flow

- Read input data
- Pre-process
- Run built-in ML algorithm
- Run a custom ML algorithm
- Scoring
Reading input files

- Support for csv and other standard formats.
- APIs `r4ml.read.csv` same as `read.csv`. 
library(R4ML) # load the library
r4ml.session() # init the session

hdfs_or_local_file = "sample_file.csv"  # local or hdfs file
r4f <- r4ml.read.csv(hdfs_or_local_file, header = TRUE, sep = ",")
Pre-processing

• Typical machine learning use cases require many pre-processing steps

• R4ML provides and enhances these commonly used pre-processing options:
  • NA removal
  • Binning
  • Normalizing
  • Recoding
  • One-hot encoding

• R4ML also supports all the standard SparkR pre-processors
Pre-processing Example

```r
r4f_tx <- r4ml.ml.preprocess(# run pre-processing
  r4f,
  dummycodeAttrs = "Species", # one hot encoding
  recodeAttrs=c("Species"), # recoding
  scalingAttrs=c("Petal_Length"), # scaling and normalization
  binningAttr = c("Sepal_Length", "Sepal_Width"), numBins=4, # binning
  missingAttr = c("Petal_Length", "Sepal_Width"), # na handling
  imputationMethod = c("global_mean", "constant"),
  imputationValues = list("Sepal_Width" = 40), omit.na="Petal_Width")

r4_mat <- as.r4ml.matrix(r4f_tx$data) # the matrix after transform
r4_mat_info <- r4f_tx$data$metadata # contains the meta info about transform
```
R4ML’s Built-in ML Algorithms

- **Regression**
  - Linear Model (LM)
  - Generalized Linear Model (GLM)
  - Step Linear Model (Step.LM)
- **Classification**
  - Multi logistic Regression (MLOGIT)
  - Support Vector Machine (SVM)
- **Factorization**
  - Principal Component Analysis (PCA)
  - Alternating Least Square (ALS)
- **Survival analysis**
  - Kaplan Meier (KM) Survival model
  - Cox Proportional Hazard model
Training & Scoring GLM Example

```r
# split the data into training and test (continue from previous examples)
split_r4_mat <- r4ml.sample(r4_mat, perc=c(0.8,0.2))
train <- split_r4_mat[[1]]
test <- split_r4_mat[[2]]

glm_m <- r4ml.glm (Petal_Length ~ ., data = train, lambda = 1.1)

glm_pred <- predict(glm_m, test)

# statistics
glm_pred$statistics # user can see R-squared, dispersion etc
r4ml.glm$predictions # prediction columns
```
Custom ML algorithms

- Allows user to create the custom machine learning algorithms for POC.
- Works via the Apache SystemML DML connector.
- User need to attach the input and output variables from R/R4ML to the DML variables.
- User can create the library using the R package management.
# Let's create a simple onehot custom utility in R4ML
# What is One Hot encoding
# Step 1:
# One hot encoding for the input matrix X

dml <- \texttt{`Y = table(seq(1, nrow(X), 1), X`)}

# Step 2:
# Attach the input and output via call to sysml.execute

X <- as.r4ml.matrix(as.r4ml.frame(iris$Species))
sysml.outs <- Sysml.execute(dml = dml, X = X, \texttt{`Y`)}

# Step 3:
# Collect the output

Y = sysml.outs[,\texttt{`Y`}]
DEMO
Airline Flight Delay Prediction Using SVM

- Do exploratory analysis and manual feature selection for the real dataset
- Pre-process the inputs and do proper na removal, recoding etc
- PCA examples:
- SVM example:
  - See the results
  - Talks about wider table support in R4ML and systemML
- Illustrate the cross validation and model tuning
SVM in a Slide
SVM Formulation

\[
\begin{align*}
\min_{w, b, \xi} & \quad \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\
\text{subject to} & \quad y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i, \\
& \quad \xi_i \geq 0.
\end{align*}
\]
Gradient Descent
Go to RStudio/PDF docs for Demo
Future Plans
Future work

• More ML algorithm support.

• We will have many real world use cases, examples, and PDF demos.

• Many more new work items are in the pipeline. Please see the latest changes at https://github.com/SparkTC/r4ml
Summary
Invitation to contribute

• R4ML is open source and we invite all to start using it and contributing to it
  • https://github.com/sparkTC/r4ml

• Other useful links
  • SparkR: https://spark.apache.org/docs/latest/sparkr.html
R4ML is an open-source scalable machine learning package for implementing end to end ML flows.

It bridges important gaps in SparkR and uses the best of Apache SystemML and SparkR to make data scientists and analysts get the most out of big data analytics.

R4ML advantages:
- More ML algorithms
- Ability to write custom algorithms
- Wide table data set (images, etc.)
- Commonly used pre-processors
Thank You!

Thanks to the R4ML Team!

https://github.com/sparkTC/r4ml