Integrating Advanced Analytics with Big Data

Ian McKenna, Ph.D.
Senior Financial Engineer
The Goal

Predicting Apple's Stock Price
Exemplar workflow showing how to scale MATLAB analytics to big data using tall tables. This script takes approximately 7.5 minutes to execute on a T470s laptop.

Clean up any previous calculations
```matlab
clear allData
```

Connect to local files, databases, spreadsheets, HDFS, and more
```matlab
% Auto-generated code to import data from disk
importData
% Perform standard MATLAB calculations directly on tall data
allTickers = unique(t.Ticker);
```

Unstack close prices from stock data
```matlab
% Determine appropriate table column names based on stock tickers
stockNames = matlab.lang.makeValidName(cellstr(allTickers));
% Combine close data for each ticker into a single table
numCols = length(allTickers);
for i = 1:numCols
    filter = (t.Ticker == allTickers(i));
    allPrices(i).Close = t.Close(filter);
end
unstackedTable = arraytable(struct2array(allPrices), 'VariableNames', stockNames);
% Easily shift around and append columns
unstackedTable.Time = t.Time(filter);
unstackedTable = unstackedTable(:, [end 1:end-1]);
```

Examine unstacked data values and formatting
```matlab
% Perform operations directly on out-of-memory and notice missing data
figure;
```
The Solution

tall
Agenda

- Introduction to tall data

  - Case Study: Predicting Analytics
  - Scaling with PCT/MDCS
  - Scaling with Spark/Hadoop
    - Interactive Mode using MDCS
    - Deployment using MATLAB Compiler
  - Summary
Datastore - Accessing Big Data Sources

- Easily access large sets of data
- Works with various data formats
  - Databases
  - CSV files
  - Excel files
  - Images
- Select & preview columns/formats easily
- Use with parallel computing tools
- Easily use local and remote data sources
  - HDFS (hdfs://)
  - Amazon S3 (s3://)
  - Azure Blob Storage (wasbs://)
  - Databases
Big Data Frameworks in MATLAB

- **Tall** R2016b
  - Local, PCT, MDCS
  - MDCS + Spark
  - Compiler + Spark

- **MapReduce** R2014b
  - Deploy to Hadoop or run with MDCS

- **MATLAB API for Spark** R2016b
  - Access Spark functions (flatMap, aggregate, etc.)
  - Access Spark RDD API and create standalone apps
Tall Data

- New data type for data that doesn’t fit into memory
- Designed for mathematical/statistical operations
- Looks like a normal MATLAB array
  - Supports numeric types, tables, datetimes, strings, etc…
  - 300+ tall enabled functions supported in MATLAB
- Process big data on your desktop, compute clusters, and Hadoop/Spark systems
Big Data Without Big Changes

1 File

Access Data
stockData = readtable('SP100.csv');

Preprocess Data
Get only the weekdays
weekdays = ~isweekend(stockData.Time);
stockData = stockData(weekdays,:);

Clean data
stockData = fillmissing(stockData(Returns, 'linear'));

Calculate metrics/statistics
correlations = corr(stockData.Returns);
tickers = unique(stockData.Ticker);

1000+ Files

Access Data
stockData = datastore('SP100*.csv');
stockData = tall(stockData);

Preprocess Data
Get only the weekdays
weekdays = ~isweekend(stockData.Time);
stockData = stockData(weekdays,:);

Clean data
stockData = fillmissing(stockData(Returns, 'linear'));

Calculate metrics/statistics
correlations = corr(stockData.Returns);
tickers = unique(stockData.Ticker);
[tickers, correlations] = gather(tickers, correlations);
Analytics With Tall Include…

- **Machine Learning**
  - `fitlm` (linear regression)
  - `fitglm` (logistic & generalized linear)
  - `fitckernel` (Gaussian kernel classification)
  - `fitrlinear` (SVM regression)
  - `fitclinear` (SVM classification)
  - `fitctree` (classification tree)
  - `fitcnb` (naïve bayes)
  - `fitcdiscr` (discriminant analysis)
  - `TreeBagger` (random forest)
  - `lasso` (lasso regression)
  - `pca` (principal component analysis)
  - `kmeans` (clustering)

- **Cleaning data**
  - `fillmissing`
  - `rmmissing`
  - `synchronize`
  - `retime`
  - `splitapply`
  - `datasample`
  - `cvpartition`

- **Visualizing data**
  - `plot`
  - `scatter`
  - `binscatter`
  - `histogram`
  - `histogram2`
  - `pie`
Example Analytics Use Case

- **Objective:** Predict Apple Stock Price

- **Inputs:**
  - Price series for all constituents of S&P100
  - Scale to billions of rows (20 years of minutely data)

- **Approach:**
  - Preprocess and explore data
  - Work with subset of data for prototyping
  - Fit regression models
  - Predict price and validate model
  - Scale to full data set on HDFS
Scaling Analytics With Tall

- Non-scaled Desktop Application
- Tall (Local)
- Tall + Parallel Computing (Local)
- Tall + MDCS (MATLAB Distributed Computing Server)
- Tall + MDCS + Spark
- Tall + MATLAB Compiler + Spark
What Is Spark/Hadoop?

- Cluster management and computing software for big data
- Hadoop:
  - HDFS (File System)
  - YARN (Scheduler)
  - MapReduce (Programming Model)
- Spark: Computational engine
- MATLAB is certified for HDP and Cloudera
MATLAB workers must be installed or accessible to all worker nodes
- MDCS workers (working from MATLAB)
- MATLAB Runtime (deployed)
Running On Spark + Hadoop (MDCS)

- Desktop

```matlab
% Define the Execution Environment.
mapreducer(gcp);

% Access the data.
d = datastore('~/home/data/SP100/*.csv');
t = tall(d);
```

- Spark

```matlab
%% Define the Execution Environment
setenv('HADOOP_HOME', '~/usr/hdp/2.6.2.0/hadoop');
setenv('SPARK_HOME', '~/usr/hdp/2.6.2.0/spark');

cluster = parallel.cluster.Hadoop;
cluster.SparkProperties('spark.executor.instances') = '16';
mapreducer(cluster);

% Access the data
d = datastore('hdfs://hadoop01:8020/user/data/SP100/*.csv');
t = tall(d);
```
Running On Spark + Hadoop (MDCS)

tall Arrays for Big Data in MATLAB

Predict Cost of Taxi Ride in New York City

Analyze data from .csv files containing taxi trip information, separated by month. The data set is available from the City of New York.

Set up execution environment:

```matlab
global numWorkers = 16;
setenv('HADOOP_HOME', '/mathworks/test/hadoop');
setenv('SPARK_HOME', '/mathworks/test/spark');
cluster = parallel.cluster.Hadoop;
cluster.SparkProperties('spark.executor.instances') = num2str(numWorkers);
```
Deploying Applications to Spark

MATLAB

Toolboxes

MATLAB Application

Edge Node

MATLAB Compiler

Worker Nodes

MATLAB Runtime

1

2

3

.sh
Big Data for New Users

- Desktop
  - Datastore & tall
  - Run in parallel
  - Prototype code locally

- Compute Clusters
  - Scale parallel applications to grid, cluster, & cloud

- Spark + Hadoop
  - Run in parallel on Spark cluster
  - Deploy as standalone applications
Multiple Choices, Many Benefits

- **Benefits of Spark/Hadoop**
  - Scalability and robustness
  - Fault-tolerant distributed data storage
  - Move compute to the data

- **Benefits of MDCS**
  - Interactive connection
  - Easy prototyping and debugging

- **Benefits of Compiler**
  - Easily invoke from outside MATLAB
  - Royalty free deployment
  - No licensing necessary on cluster
Easy Scaling with Tall

- Designed for visualization, data cleansing, statistics, and machine learning
- Deferred evaluation optimizes big data analytics
- Perform visualizations directly on big data
- Easily convert between in-memory and out-of-memory
- No need to rewrite code, just call ‘tall’
- Support production and prototype using ‘isdeployed’
Summary

- Get started scaling right away on your local machine with tall
- Don’t need Spark/HDFS cluster to scale, can use MDCS
- MATLAB scales from desktop to production
  - Transition from desktop to cluster with minimal changes
  - Using Spark/HDFS is simple with MATLAB
MATLAB Central Community

Every month, over 2 million MATLAB & Simulink users visit MATLAB Central to get questions answered, download code and improve programming skills.

**MATLAB Answers**: Q&A forum; most questions get answered in only 60 minutes

**File Exchange**: Download code from a huge repository of free code including tens of thousands of open source community files

**Cody**: Sharpen programming skills while having fun

**Blogs**: Get the inside view from Engineers who build and support MATLAB & Simulink

**ThingSpeak**: Explore IoT Data

And more for you to explore…
Get Training

Accelerate your learning curve:
- Customized curriculum
- Learn best practices
- Practice on real-world examples

Options to fit your needs:
- Self-paced (online)
- Instructor led (online and in-person)
- Customized curriculum (on-site)
Consulting

- Engineering expertise and deep product knowledge, specializing in:
  - Application development using MATLAB
  - Model-Based Design using Simulink and Stateflow
  - Embedded systems development
  - Enterprise-wide integration of MathWorks products into engineering process and systems
  - Jumpstart services for a fast, smooth transition to MathWorks products

- Project-based services for a growing number of industries, including aerospace and defense, automotive, communications, power and marine, and financial services
Contact us to learn more!

- Senior Financial Engineers
  - Ian McKenna ([ian.McKenna@mathworks.com](mailto:ian.McKenna@mathworks.com)) [Chicago]
  - Marshall Alphonso ([Marshall.Alphonso@mathworks.com](mailto:Marshall.Alphonso@mathworks.com)) [NYC]

- Senior Account Managers
  - Chuck Castricone ([Chuck.Castricone@mathworks.com](mailto:Chuck.Castricone@mathworks.com))
  - Mike DeLucia ([Mike.DeLucia@mathworks.com](mailto:Mike.DeLucia@mathworks.com))
  - Jim Coughlin ([Jim.Coughlin@mathworks.com](mailto:Jim.Coughlin@mathworks.com))
  - Mark DeMaio ([Mark.DeMaio@mathworks.com](mailto:Mark.DeMaio@mathworks.com))
  - David Habeeb ([David.Habeeb@mathworks.com](mailto:David.Habeeb@mathworks.com))
Appendix
Requirements

- **MDCS**
  - Windows, Linux, Mac

- **Spark**
  - Linux & Mac (on Cluster)
    - 17b: MDCS method – can use tall arrays on Spark cluster supporting all architectures for the client, while supporting Linux & Mac architectures for the cluster (includes cross-platform support)
  - MDCS: Spark 1.x or 2.x (Spark enabled Hadoop system only)
  - Compiler: Spark 1.x or 2.x (Spark enabled Hadoop system only)
  - Hadoop 2.x or higher
Tips

- Use head/tail to pull portion of data into memory (also faster!)
- Work with unevaluated array as much as possible
  - Gives MATLAB ability to further optimize execution
- ‘Gathering’ more is faster!
  - \([a,b,c] = \text{gather}(a,b,c)\)
- Use ‘dot’ notation or array2table, cell2mat, etc. to index data types
- Make sure indices are in sorted order (e.g. \(T([2\ 5\ 7],:)\))
  - Use ‘sort’ on the indices
MATLAB Integrates With Many Systems

- Built-in support for interoperability with various analytics platforms:
  - HDFS, Hadoop/MapReduce, YARN, Spark 1.X
  - Cloudera, Hortonworks
  - MongoDB
  - Cloud and local databases using ODBC/JDBC
  - AWS S3 and Azure Blob

- Applications our service teams can assist with include:
  - Running of MathWorks products onto cloud platforms (AWS, Azure, Google, etc.)
  - Read/write from: AWS S3, Azure Blob, Azure Data Lake
  - Streaming data: Kafka, Azure IoT Hub, Azure Event Hub, and AWS services
  - Tableau, Qlikview, Spotfire
  - Hive, Cassandra, Impala, Parquet, and AVRO
  - Netezza, Teradata