MATLAB for Data Analytics and Machine Learning

Amit Doshi
Application Engineer-Technical Computing
Amit.Doshi@mathworks.in
Data Analytics and Technical Computing Workflow

- **Data Exploration**:
  - Gain Insights
  - Filter Data
  - Build Intuition
  - Hypothesize

- **Analytics Development**:
  - Create prototype
  - Machine Learning
  - Optimization
  - App Development

- **Analytics Integration**:
  - Version Control
  - Testing Code
  - Validation
  - Deploy & Share

---

**SERVER**

- HDFS

---

**Desktop**

- MATLAB Production Server(s)

---

**Web Application**

- Web Server(s)

---

**Data Exploration** and **Analytics Development** are connected to **Analytics Integration** through **Version Control** and **Testing Code**, followed by **Deploy & Share**.

**Data Exploration** also connects to **Analytics Development** through separate paths, with **Hypothesize** and **Build Intuition** highlighted.

**Analytics Integration** connects back to **Analytics Development** through **Validation** and **Deploy & Share**.
“Science of examining raw data with the purpose of drawing conclusions to allow make better business decisions” - wiki
Challenges in Data Analytics

“Any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications.” (Wikipedia)

- Getting started when data volume exceeds memory limits
- Significant technical expertise required
- No “one size fits all” solution
- Locked into Black Box solutions
- Time required to conduct the analysis
MATLAB Solutions for Data Analytics

Solution 1: Getting Insights into Data
Solution 2: Data Volume Exceeds Memory Limits
Solution 3: Analytics Development - Machine Learning
Solution 1: Getting Insights into Data

- Import and explore data
- Visualize fuel consumption pattern on the map
- Build intuition

Source: relayrides.com
Solution 1: Setup at MathWorks

Let’s see how to-
• Import data from the EC2 server
• Carry out data analysis

Note: server is Amazon EC2
MATLAB Solutions for Data Analytics

Solution 1: Getting Insights into Data
Solution 2: Data Volume Exceeds Memory Limits
Solution 3: Analytics Development - Machine Learning
Big Data Capabilities in MATLAB

Memory and Data Access
- 64-bit processors
- Memory Mapped Variables
- Disk Variables
- Databases
- Datastores

Programming Constructs
- Streaming
- Block Processing
- Parallel-for loops
- GPU Arrays
- SPMD and Distributed Arrays
- MapReduce

Platforms
- Desktop (Multicore, GPU)
- Clusters
- Cloud Computing (MDCS on EC2)
- Hadoop
Solution 2: Data Volume Exceeds Memory Limits

Goal → Examine hybrid car adoption

- **Data**
  - Massachusetts Vehicle Registration Data from 2008-2011
  - 16M records, 45 fields

- **Analysis**
  - Calculate % of hybrids registered
    - By Quarter
    - By Regional Area
  - Create map of results
When to Use datastore

- **Data Characteristics**
  - Text data in files, databases or stored in the Hadoop Distributed File System (HDFS)

- **Compute Platform**
  - Desktop

- **Analysis Characteristics**
  - Supports Load, Analyze, Discard workflows
  - Incrementally read chunks of data, process within a `while` loop
Reading in Part of a Dataset from Files

- Text file, ASCII file
  - datastore

- MAT file
  - Load and save part of a variable using the matfile

- Binary file
  - Read and write directly to/from file using memmapfile
  - Maps address space to file

- Databases
  - ODBC and JDBC-compliant (e.g. Oracle, MySQL, Microsoft, SQL Server)
To use MapReduce in MATLAB:

- datastore
- mapreduce
- mapreducer
mapreduce

Data Store

<table>
<thead>
<tr>
<th>Veh_typ</th>
<th>Q3_08</th>
<th>Q4_08</th>
<th>Q1_09</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SUV</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SUV</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Car</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SUV</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Key: Q3_08

<table>
<thead>
<tr>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

Key: Q4_08

<table>
<thead>
<tr>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

Key: Q1_09

<table>
<thead>
<tr>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

Key: % Hybrid (Value)

<table>
<thead>
<tr>
<th>Key</th>
<th>% Hybrid (Value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q3_08</td>
<td>0.4</td>
</tr>
<tr>
<td>Q4_08</td>
<td>0.67</td>
</tr>
<tr>
<td>Q1_09</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Parallel Computing with MATLAB

- Parallel Computing Toolbox
- MATLAB Distributed Computing Server

Client
- MATLAB Desktop (Client)

Cluster
- MATLAB Distributed Computing Server
Analyze Data on Hadoop Interactively

MATLAB Distributed Computing Server

Datastore

HDFS

Node
Data
Map
Reduce

Node
Data
Map
Reduce

Node
Data
Map
Reduce

MATLAB MapReduce Code
Deployed Applications with Hadoop

MATLAB runtime

Node

Datastore

HDFS

Map

Reduce

Node

Data

Map

Reduce

Node

Data

Map

Reduce

MATLAB MapReduce Code
When to Use mapreduce

- **Data Characteristics**
  - Text data in files, databases or stored in the Hadoop Distributed File System (HDFS)
  - Dataset will not fit into memory

- **Compute Platform**
  - Desktop
  - Scales to run within Hadoop MapReduce on data in HDFS

- **Analysis Characteristics**
  - Must be able to be Partitioned into two phases
    1. Map: filter or process sub-segments of data
    2. Reduce: aggregate interim results and calculate final answer
Techniques for Big Data in MATLAB

Load, Analyze, Discard
parfor, datastore,

MapReduce

Distributed Memory
SPMD and distributed arrays

out-of-memory
in-memory

Embarrassingly Parallel
Non-Partitionable

Complexity

Webinar: Tackling Big Data with MATLAB
MATLAB Solutions for Data Analytics

Solution 1: Getting Insights into Data
Solution 2: Data Volume Exceeds Memory Limits
Solution 3: Analytics Development - Machine Learning
Machine Learning
Characteristics and Examples

- Characteristics
  - Lots of data (many variables)
  - System too complex to know the governing equation (e.g., black-box modeling)

Biology

- Tumor Detection, Drug Discovery

Financial Services

- Credit Scoring, Algorithm Trading, Bond Classification

Image & Video Processing

- Pattern Recognition

Audio Processing

- Speech Recognition

Energy

- Load, Price Forecasting, Trading
Overview – Machine Learning

Type of Learning

- Unsupervised Learning
  - No known grouping pattern
- Supervised Learning
  - Known responses and predictors

Categories of Algorithms

- Clustering
- Classification
- Regression
Unsupervised Learning

Clustering

- k-Means, Fuzzy C-Means
- Hierarchical
- Neural Networks
- Gaussian Mixture
- Hidden Markov Model
Supervised Learning

Regression

- Neural Networks
- Decision Trees
- Ensemble Methods
- Non-linear Reg. (GLM, Logistic)
- Linear Regression

Classification

- Support Vector Machines
- Discriminant Analysis
- Naive Bayes
- Nearest Neighbor
Train: Iterate till you find the best model using historical data

Predict: Integrate trained models into applications
Solution 3: Analytics Development - Machine Learning

Machine learning uses **data** and produces a **program** to perform a **task**

**Task:** Classify human activity from sensor data

<table>
<thead>
<tr>
<th>Predictors</th>
<th>3-axial Accelerometer and Gyroscope data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Response</strong></td>
<td><strong>Activity:</strong></td>
</tr>
<tr>
<td><img src="activity_icon.png" alt="Activity Icon" /></td>
<td><img src="activity_icon.png" alt="Activity Icon" /></td>
</tr>
</tbody>
</table>

**Standard Approach**

- Write Program
  - If $X_{acc} > 0.5$
    - then “SITTING”
  - If $Y_{acc} < 4$ and $Z_{acc} > 5$
    - then “STANDING”
  - ...

- Have an Equation
  - $Y_{activity} = \beta_{1}X_{acc} + \beta_{2}Y_{acc} + \beta_{3}Z_{acc} + ...$

**Machine Learning Approach**

- **Model**: Inputs $\rightarrow$ Outputs
  - $model = \langle Machine\ Learning\ Algorithm > (sensor\_data, activity)$

**Approach:**
- Extract features from raw sensor signals
- Train and compare classifiers
- Test results on new sensor data
Choose a Classifier

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Predictive Accuracy</th>
<th>Fitting Speed</th>
<th>Prediction Speed</th>
<th>Memory Usage</th>
<th>Easy to Interpret</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Trees</td>
<td>Medium</td>
<td>Fast</td>
<td>Fast</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td>High</td>
<td>Medium</td>
<td>Fast for few support vectors. Slow for many support vectors.</td>
<td>Fast for few support vectors. Slow for many support vectors.</td>
<td>Yes only for Linear SVM. No for all other kernel types.</td>
</tr>
<tr>
<td>Nearest Neighbor Classifiers</td>
<td>High only in low dimensions. Low for high dimensions.</td>
<td>Fast</td>
<td>Fast for low dimensions (&lt;10), slow for high dimensions (&gt;20)</td>
<td>High</td>
<td>No</td>
</tr>
<tr>
<td>Ensemble Classifiers</td>
<td>High</td>
<td>Slow</td>
<td>Qualities depend on choice of algorithm.</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
Solution 3: Machine Learning Summary

**Train:** Iterate till you find the best model

- **LOAD DATA**
- **PREPROCESS DATA**
  1. Mean
  2. Standard deviation
  3. PCA
- **SUPERVISED LEARNING**
  Classification Learner
- **MODEL**

**Predict:** Integrate trained models into applications

- **TEST DATA**
- **PREPROCESS DATA**
  1. Mean
  2. Standard deviation
  3. PCA
- **MODEL**
- **PREDICTION**
MATLAB for Machine Learning

Classification
Build models to classify data into different categories.

Algorithms: support vector machine (SVM), boosted and bagged decision trees, k-nearest neighbor, Naïve Bayes, discriminant analysis, neural networks, and more

» Get started with introductory examples

Applications: credit scoring, tumor detection, image recognition

Regression
Build models to predict continuous data.

Algorithms: linear model, nonlinear model, regularization, stepwise regression, boosted and bagged decision trees, neural networks, adaptive neuro-fuzzy learning, and more

» Get started with introductory examples

Applications: electricity load forecasting, algorithmic trading

Clustering
Find natural groupings and patterns in data.

Algorithms: k-means, hierarchical clustering, Gaussian mixture models, hidden Markov models, self-organizing maps, fuzzy c-means clustering, subtractive clustering, and more

» Get started with introductory examples

Applications: pattern mining, medical imaging, object recognition

Go to MATLAB Help → • Functions • Classes • Examples and How-To • Concepts
Taking MATLAB to Production

- Royalty-free deployment
- Point-and-click workflow
- Unified process for desktop and server apps
# Summary - MATLAB for Data Analytics

<table>
<thead>
<tr>
<th>Challenges</th>
<th>MATLAB Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘..Volume exceeds memory limits.’</td>
<td>• Functions - Datastore, MapReduce- to handle big data, create prototypes&lt;br&gt;• Facility to scale it to HADOOP without any extra efforts</td>
</tr>
<tr>
<td>‘Technical expertise needed ’</td>
<td>• Built-in High-quality, robust and tested algorithms&lt;br&gt;• Apps – Import Wizard, Plot tools to increase productivity for data preparation, interactive exploration and visualization</td>
</tr>
<tr>
<td>‘No “one size fits all” solution’</td>
<td>• Various built-in algorithms for various different problems&lt;br&gt;• Recommendation - “Choose a Classifier” for choosing the best algorithm</td>
</tr>
<tr>
<td>‘Locked into Black-box solutions’</td>
<td>• Algorithms are not hidden and can be further customized&lt;br&gt;• Structured documentation- syntax, description, examples etc.</td>
</tr>
<tr>
<td>‘Time required to conduct the analysis’</td>
<td>• Parallel computation, Optimized libraries, support to GPU&lt;br&gt;• Automatic code generation from many apps</td>
</tr>
</tbody>
</table>
Key Takeaway

You can Import-
- Historical data
- Live HW data
- From Databases
- From HADOOP

You can Explore-Discover
- Using built-in apps
- Robust algorithms
- Parallel computing
- Neat documentation

You can Share
- Automatic reports
- Standalone exe
- Interface with other frameworks
Additional Resources

Website:
www.mathworks.in
Webinars, User Stories etc.

Customer Service for non-technical questions: info@mathworks.in

Product Training:
www.mathworks.in/training

Technical Support India:
support@mathworks.in

MATLAB Central
www.mathworks.in/matlabcentral
Training Services

Exploit the full potential of MathWorks products

Flexible delivery options:

- Public training available in several cities
- Onsite training with standard or customized courses
- Web-based training with live, interactive instructor-led courses

More than 30 course offerings:

- Introductory and intermediate training on MATLAB, Simulink, Stateflow, code generation, and Polyspace products
- Specialized courses in control design, signal processing, parallel computing, code generation, communications, financial analysis and other areas.

www.mathworks.in/training
Training Certification

- Accelerate professional growth
- Validate proficiency with MATLAB
- Increase productivity and project success

MathWorks Certified MATLAB Associate Examination

**Bangalore**
29th July & 25th Nov

**Pune**
3rd June

Recommended Courses
MATLAB Fundamentals (MLBE)

Email: training@mathworks.in
URL: http://in.mathworks.com/#training
Phone: 080-6632-6000
Thank You