Predictive Analytics and Big Data with MATLAB

Ian McKenna, Ph.D.
Agenda

Introduction

- Predictive Modeling
  - Supervised Machine Learning
  - Time Series Modeling

- Big Data Analysis
  - Load, Analyze, Discard workflows
  - Scale computations with parallel computing
  - Distributed processing of large data sets

- Moving to Production with MATLAB
Financial Modeling Workflow

Small/Big Data

- Access
  - Files
    - .xls
  - Databases
  - Datafeeds
    - BLZ
    - NG
    - CL
    - 71.92
    - 5.332
    - 81.

Predictive Modeling

- Explore and Prototype
  - Data Analysis & Visualization
- Financial Modeling
  - S=31; K=30
  - C=blsprice
  - P=C-S+K*ex
- Application Development
  - Option 1
  - Option 2
  - NEXT

Scale

Deploy

- Share
  - Reporting
    - PDF
    - .doc
    - .html
- Applications
  - .dll
  - C/C++
  - Java
  - .NET
- Production
Financial Modeling Workflow

Explore and Prototype

Data Analysis & Visualization

Financial Modeling

Application Development

Predictive Modeling

\[ S = 31; K = 30 \]
\[ C = \text{blsprice} \]
\[ P = C - S + K \times e^x \]
Agenda

- Introduction

Predictive Modeling
- Supervised Machine Learning
- Time Series Modeling

Big Data Analysis
- Load, Analyze, Discard workflows
- Scale computations with parallel computing
- Distributed processing of large data sets

Moving to Production with MATLAB
What is Predictive Modeling?

- Use of mathematical language to make predictions about the future

Examples

Trading strategies

Electricity Demand

\[ EL = f(T, t, DP, ...) \]
Why develop predictive models?

- Forecast prices/returns
- Price complex instruments
- Analyze impact of predictors (sensitivity analysis)
- Stress testing
- Gain economic/market insight
- And many more reasons
Challenges

- Significant technical expertise required
- No “one size fits all” solution
- Locked into Black Box solutions
- Time required to conduct the analysis
Predictive Modeling Workflow

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

**Predict:** Integrate trained models into applications

- **LOAD DATA**
- **PREPROCESS DATA**
  - FILTERS
  - PCA
  - SUMMARY STATISTICS
  - CLUSTER ANALYSIS
- **SUPERVISED LEARNING**
  - CLASSIFICATION
  - REGRESSION
- **MODEL**
- **NEW DATA**
- **PREDICTION**
Classes of Response Variables

Structure
- Sequential
- Non-Sequential

Type
- Continuous
- Categorical
Examples

- Classification Learner App

- Predicting Customer Response
  - Classification techniques
  - Measure accuracy and compare models

- Predicting S&P 500
  - ARIMA modeling
  - GARCH modeling
Getting Started with Predictive Modeling

- Perform common tasks interactively
  - Classification Learner App
  - Neural Net App
Example – Bank Marketing Campaign

- **Goal:**
  - Predict if customer would subscribe to bank term deposit based on different attributes

- **Approach:**
  - Train a classifier using different models
  - Measure accuracy and compare models
  - Reduce model complexity
  - Use classifier for prediction

Data set downloaded from UCI Machine Learning repository
http://archive.ics.uci.edu/ml/datasets/Bank+Marketing
Classification Techniques

Regression

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

Classification

Support Vector Machines
Discriminant Analysis
Naive Bayes
Nearest Neighbor
Example – Bank Marketing Campaign

- Numerous predictive models with rich documentation
- Interactive visualizations and apps to aid discovery
- Built-in parallel computing support
- Quick prototyping; Focus on modeling not programming
Example – Time Series Modeling and Forecasting for the S&P 500 Index

- **Goal:**
  - Model S&P 500 time series as a combined ARIMA/GARCH process and forecast on test data

- **Approach:**
  - Fit ARIMA model with S&P 500 returns and estimate parameters
  - Fit GARCH model for S&P 500 volatility
  - Perform statistical tests for time series attributes e.g. stationarity
<table>
<thead>
<tr>
<th>Conditional Mean Models</th>
<th>Conditional Variance Models</th>
<th>Non-Linear Models</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR – Autoregressive</td>
<td>ARCH</td>
<td>NAR Neural Network</td>
<td>Regression with ARIMA errors</td>
</tr>
<tr>
<td>MA – Moving Average</td>
<td>GARCH</td>
<td>NARX Neural Network</td>
<td></td>
</tr>
<tr>
<td>ARIMA – Integrated</td>
<td>EGARCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARIMAX – eXogenous inputs</td>
<td>GJR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARMA – Vector ARMA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARMAX – eXogenous inputs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VEC – Vector Error Correcting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State Space Models</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time Varying</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time Invariant</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Models for Time Series Data**
Example – Time Series Modeling and Forecasting for the S&P 500 Index

- Numerous ARIMAX and GARCH modeling techniques with rich documentation
- Interactive visualizations
- Code parallelization to maximize computing resources
- Rapid exploration & development
Agenda

- Introduction

- Predictive Modeling
  - Supervised Machine Learning
  - Time Series Modeling

Big Data Analysis
- Load, Analyze, Discard workflows
- Scale computations with parallel computing
- Distributed processing of large data sets

- Moving to Production with MATLAB
Financial Modeling Workflow

Small/Big Data
- Access
  - Files
  - Databases
  - Datafeeds

Predictive Modeling
- Explore and Prototype
  - Data Analysis & Visualization
  - Financial Modeling
  - Application Development

Deploy
- Share
  - Reporting
  - Applications
  - Production

Scale

MathWorks

Financial Modeling Application Development

Report

Small/Big Data

Predictive Modeling

Deploy
Financial Modeling Workflow
Challenges of Big Data

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

(Wikipedia)

- Volume
  - The amount of data

- Velocity
  - The speed data is generated/analyzed

- Variety
  - Range of data types and sources

- Value
  - What business intelligence can be obtained from the data?
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
Techniques for Big Data in MATLAB

- datastore
- MapReduce
- parfor
- SPMD, Distributed Memory
- 64bit Workstation
- Consulting

Scale

Hard drive

RAM

Embarrassingly Parallel

Non-Partitionable

Complexity
Techniques for Big Data in MATLAB

- `datatstore`
- `MapReduce`
- `parfor`
- `SPMD, Distributed Memory`
- `64bit Workstation`

Complexity:

- Embarrassingly Parallel
- Non-Partitionable

Scale:

- Hard drive
- RAM
Memory Usage Best Practices

- Expand Workspace: 64bit MATLAB

- Use the appropriate data storage
  - Categorical Arrays
  - Be aware of overhead of cells and structures
  - Use only the precision your need
  - Sparse Matrices

- Minimize Data Copies
  - In place operations, if possible
  - Use nested functions
  - Inherit data using object handles
Techniques for Big Data in MATLAB

- **Complexity**
  - Embarrassingly Parallel
  - Non-Partitionable

- **Scale**
  - Hard drive
  - RAM

- **Techniques**
  - datastore
  - MapReduce
  - Consulting
  - parfor
  - SPMD, Distributed Memory
  - 64bit Workstation
Parallel Computing with MATLAB
Example: Analyzing an Investment Strategy

- Optimize portfolios against target benchmark
- Analyze and report performance over time
- Backtest over 20-year period, parallelize 3-month rebalance
When to Use `parfor`

- **Data Characteristics**
  - The data for each iteration must fit in memory
  - Loop iterations must be independent

- **Transition from desktop to cluster with minimal code changes**

- **Speed up analysis on big data**
Techniques for Big Data in MATLAB

- Datastore
- MapReduce
- Parfor
- SPMD, Distributed Memory
- 64-bit Workstation

Scale

- Hard drive
- RAM

Complexity

- Embarrassingly Parallel
- Non-Partitionable
Parallel Computing – Distributed Memory

Using More Computers (RAM)
spmd blocks

spmd
    \% single program across workers
end

- Mix parallel and serial code in the same function
- Single Program runs simultaneously across workers
- Multiple Data spread across multiple workers
Example: Airline Delay Analysis

- **Data**
  - Airline On-Time Statistics
  - 123.5M records, 29 fields

- **Analysis**
  - Calculate delay patterns
  - Visualize summaries
  - Estimate & evaluate predictive models
When to Use Distributed Memory

- **Data Characteristics**
  - Data must be fit in collective memory across machines

- **Compute Platform**
  - Prototype (subset of data) on desktop
  - Run on a cluster or cloud

- **Analysis Characteristics**
  - Distributed arrays support a subset of functions
Techniques for Big Data in MATLAB

- datastore
- MapReduce
- parfor
- SPMD, Distributed Memory
- 64bit Workstation

Scale:
- Hard drive
- RAM

Complexity:
- Embarrassingly Parallel
- Non-Partitionable
Access Big Data

datastore

- Easily specify data set
  - Single text file (or collection of text files)

- Preview data structure and format

- Select data to import using column names

- Incrementally read subsets of the data

```
airdata = datastore('*\.csv');
airdata.SelectedVariables = {'Distance', 'ArrDelay'};
data = read(airdata);
```
Example: Determine unique tickers

- 15 years of daily S&P 500 data
- Data in multiple files of different sizes
- Many irrelevant columns in dataset
When to Use datastore

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

- **Analysis Characteristics**
  - 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
  - Read part of a collection of files using `datastore`

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

- Binary file
  - Read and write directly to/from file using `memmapfile`

- Databases
  - ODBC and JDBC-compliant (e.g. Oracle, MySQL, Microsoft SQL Server)
Techniques for Big Data in MATLAB

- datastore
- parfor
- SPMD, Distributed Memory
- 64bit Workstation
- MapReduce

Complexity

- Embarrassingly Parallel
- Non-Partitionable
Analyze Big Data

mapreduce

- MapReduce programming technique to analyze big data
  - mapreduce uses a datastore to process data in small chunks that individually fit into memory

- mapreduce on the desktop
  - Access data on HDFS
  - Integrates with Parallel Computing Toolbox

- mapreduce with Hadoop
  - Run on Hadoop using MATLAB Distributed Computing Server
  - Deploy to Hadoop using MATLAB Compiler
MapReduce

Data Store → Map → Reduce

<table>
<thead>
<tr>
<th>Date</th>
<th>Ticker</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-Jan</td>
<td>AIG</td>
<td>-0.051</td>
</tr>
<tr>
<td>3-Jan</td>
<td>AMZN</td>
<td>NaN</td>
</tr>
<tr>
<td>3-Jan</td>
<td>GE</td>
<td>-0.040</td>
</tr>
<tr>
<td>3-Jan</td>
<td>INTC</td>
<td>NaN</td>
</tr>
<tr>
<td>3-Jan</td>
<td>AIG</td>
<td>-0.051</td>
</tr>
<tr>
<td>4-Jan</td>
<td>YHOO</td>
<td>-0.067</td>
</tr>
<tr>
<td>4-Jan</td>
<td>INTC</td>
<td>-0.046</td>
</tr>
<tr>
<td>5-Jan</td>
<td>GE</td>
<td>0.025</td>
</tr>
<tr>
<td>5-Jan</td>
<td>AIG</td>
<td>NaN</td>
</tr>
<tr>
<td>5-Jan</td>
<td>AMZN</td>
<td>0.078</td>
</tr>
<tr>
<td>5-Jan</td>
<td>GE</td>
<td>0.025</td>
</tr>
<tr>
<td>5-Jan</td>
<td>YHOO</td>
<td>-0.039</td>
</tr>
</tbody>
</table>
Example: Calculate covariance of S&P500

*Using MapReduce*

- 15 years of daily S&P500 returns stored in multiple files
- Use all the data to calculate the mean and covariance
- Computation must scale to 1-minute bars for 30 years of data
Challenges

- Multiple files of differing sizes
Challenges

- How do we read/partition this dataset if it doesn’t fit in memory?

<table>
<thead>
<tr>
<th>Date</th>
<th>Ticker</th>
<th>Open</th>
<th>High</th>
<th>Low</th>
<th>Close</th>
<th>Volume</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-Jan-2000</td>
<td>AIG</td>
<td>107.13</td>
<td>107.44</td>
<td>103</td>
<td>103.94</td>
<td>166500</td>
<td>NaN</td>
</tr>
<tr>
<td>3-Jan-2000</td>
<td>AMZN</td>
<td>87.25</td>
<td>89.56</td>
<td>79.05</td>
<td>89.56</td>
<td>16117600</td>
<td>NaN</td>
</tr>
<tr>
<td>3-Jan-2000</td>
<td>GE</td>
<td>147.25</td>
<td>148</td>
<td>144</td>
<td>144</td>
<td>22121400</td>
<td>-0.040</td>
</tr>
<tr>
<td>8-Jan-2000</td>
<td>AMZN</td>
<td>81.5</td>
<td>89.56</td>
<td>79.05</td>
<td>89.38</td>
<td>16117600</td>
<td>NaN</td>
</tr>
<tr>
<td>4-Jan-2000</td>
<td>AIG</td>
<td>101.5</td>
<td>102.13</td>
<td>98.31</td>
<td>98.63</td>
<td>364000</td>
<td>-0.051</td>
</tr>
<tr>
<td>Jan 4,2000</td>
<td>YHOO</td>
<td>464.5</td>
<td>500.12</td>
<td>442</td>
<td>443</td>
<td>69868800</td>
<td>-0.067</td>
</tr>
<tr>
<td>4-Jan-2000</td>
<td>INTC</td>
<td>85.44</td>
<td>87.88</td>
<td>82.25</td>
<td>92.94</td>
<td>51019600</td>
<td>-0.046</td>
</tr>
<tr>
<td>4-Jan-2000</td>
<td>GE</td>
<td>147.25</td>
<td>148</td>
<td>144</td>
<td>144</td>
<td>22121400</td>
<td>-0.040</td>
</tr>
<tr>
<td>8-Jan-2000</td>
<td>GE</td>
<td>143.12</td>
<td>146.94</td>
<td>142.63</td>
<td>145.67</td>
<td>19873200</td>
<td>0.013</td>
</tr>
</tbody>
</table>

- Missing data (explicit/implicit)
Challenges

- **Mean**
  - Coupling between rows

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]

- **Covariance**
  - Coupling between rows
  - Coupling between columns

\[
cov(X, Y) = \sum_{i=1}^{N} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N}
\]
Approach

- Reading in chunks – do we have a full column of data?

<table>
<thead>
<tr>
<th>Date</th>
<th>Ticker</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-Jan-2000</td>
<td>AIG</td>
<td>-0.012</td>
</tr>
<tr>
<td>3-Jan-2000</td>
<td>AMZN</td>
<td>NaN</td>
</tr>
<tr>
<td>3-Jan-2000</td>
<td>GE</td>
<td>0.051</td>
</tr>
<tr>
<td>4-Jan-2000</td>
<td>AMZN</td>
<td>NaN</td>
</tr>
<tr>
<td>4-Jan-2000</td>
<td>AIG</td>
<td>0.097</td>
</tr>
<tr>
<td>4-Jan-2000</td>
<td>YHOO</td>
<td>-0.035</td>
</tr>
<tr>
<td>4-Jan-2000</td>
<td>GE</td>
<td>NaN</td>
</tr>
</tbody>
</table>

- Solution: convert to tabular form with all columns
- Further memory savings (ticker/date not repeated)
Goal: Calculate mean/covariance for big data sets

- Unique tickers
- Data Store
- Tabular conversion
- MapReduce
- Calculate mean/cov
- MapReduce
- Combine mean/cov
- Scale
- Hadoop
The Big Data Platform

- Fault-tolerant distributed data storage
- Take the computation to the data

- HDFS
- MapReduce
Deployed Applications with Hadoop

MATLAB runtime

Node

Datastore

HDFS

Node

Data

Node

Data

Map

Reduce

Map

Reduce

Map

Reduce

MATLAB runtime

MATLAB MapReduce Code
Solution

- **Datastore**
  - Treat multiple files as a pool of data
  - Parse data in chunks to determine unique values

- **Mapreduce**
  - Group, filter, and calculate summary statistics

- **Hadoop**
  - Algorithm is the same as the one developed on desktop
  - Easily deploy to Hadoop using interactive tools

- **MATLAB Interactive Environment**
  - Debugger and profiler
  - Validate algorithms using built-in functions for rapid prototyping
Big Data Summary

- Access portions of data with datastore

- Cluster-ready programming constructs
  - parfor
  - SPMD
  - MapReduce
  - Distributed arrays

- Prototype code for your cluster
  - Transition from desktop to cluster with no algorithm changes
Agenda

- Introduction

- Predictive Modeling
  - Supervised Machine Learning
  - Time Series Modeling

- Big Data Analysis
  - Load, Analyze, Discard workflows
  - Scale computations with parallel computing
  - Distributed processing of large data sets

Moving to Production with MATLAB
Financial Modeling Workflow

Small/Big Data
- Access
  - Files
  - Databases
  - Datafeeds

Predictive Modeling
- Explore and Prototype
  - Data Analysis & Visualization
  - Financial Modeling
    - S=31; K=30
    - C=blsprice
    - P=C-S+K*ex
- Application Development

Deploy
- Share
  - Reporting
  - Applications
  - Production

Scale
Deployed Applications

- Example: Portfolio optimization and simulation
- Example: Day-ahead system load forecasting
MATLAB Production Server

- Enterprise framework for running packaged MATLAB programs

- Scalable & reliable
  - Service large numbers of concurrent requests

- Use with web, database & application servers
  - Easily integrates with IT systems (Java, .NET, C++, Python)
Integrating with IT systems

MATLAB Compiler SDK™

Web Server

MATLAB Production Server

- Portfolio Optimization
- Pricing
- Risk Analytics

Database Server

Web Applications

Desktop Applications

Excel®
Benefits of the MATLAB Production Server

- Reduce cost of building and deploying in-house analytics
  - Quants/Analysts/Financial Modelers do not have to rewrite code in another language
  - Update deployed models easily without restarting the server
  - Single environment for model development and testing

- IT can efficiently integrate models.analytics in to production systems
  - Centrally manage packaged MATLAB programs
  - Handoff from Quant to IT only requires function signatures
  - Easily support analytics built with multiple releases of MATLAB
  - Simultaneous multiple instances of MATLAB Production Server
## Summary

<table>
<thead>
<tr>
<th>Challenges</th>
<th>MATLAB Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (loss of productivity)</td>
<td><strong>Rapid analysis and application development</strong></td>
</tr>
<tr>
<td></td>
<td>Easily access big data sets, interactive exploratory analysis and visualization, apps to get started, debugger</td>
</tr>
<tr>
<td>No “one-size-fits-all”</td>
<td><strong>Multiple algorithms and programming constructs</strong></td>
</tr>
<tr>
<td></td>
<td>Regression, machine learning, time series modeling, parfor, MapReduce, datastore</td>
</tr>
<tr>
<td>Big data and scaling</td>
<td><strong>Work on the desktop and scale to clusters</strong></td>
</tr>
<tr>
<td></td>
<td>Hadoop support, no algorithm changes required</td>
</tr>
<tr>
<td>Time to deploy &amp; integrate</td>
<td><strong>Ease of deployment and leveraging enterprise</strong></td>
</tr>
<tr>
<td></td>
<td>Push-button deployment into production</td>
</tr>
</tbody>
</table>
Financial Modeling Workflow

Access
- Files
- Databases
- Datafeeds

Research and Quantify
- Data Analysis and Visualization
- Financial Modeling
- Application Development

Share
- Reporting
- Applications
- Production

Financial Instruments
Econometrics
Financial
Statistics & Machine Learning
Optimization

Spreadsheet Link EX
Database
Datafeed
Trading

Report Generator
Production Server
MATLAB Compiler SDK
MATLAB Compiler

MATLAB
Parallel Computing
Neural Networks
MATLAB Distributed Computing Server
Curve Fitting
Learn More: Predictive Modeling with MATLAB

To learn more, visit:
www.mathworks.com/machine-learning

Basket Selection using Stepwise Regression

Classification in the presence of missing data

Regression with Boosted Decision Trees

Hierarchical Clustering
Learn More: Big Data

- MATLAB Documentation
  - Strategies for Efficient Use of Memory
  - Resolving "Out of Memory" Errors

- Big Data with MATLAB

- MATLAB MapReduce and Hadoop
  - www.mathworks.com/discovery/matlab-mapreduce-hadoop.html
# Training Services

- **Classroom Training**
  - Customized curriculum
  - Usually 2-5 day consecutive format

- **Live Online**
  - Flexible scheduling
  - Full or Half Day Sessions

- **Self-Paced**
  - Learn whenever you want and at your own pace
  - Online discussion boards and live trainer chats

---

**GARP CPE APPROVED PROVIDER:** Earn one CPE credit per hour of content.

[mathworks.com/training]
Training Roadmap

MATLAB for Financial Applications

Data Analysis and Modeling
- Statistical Methods
- Machine Learning
- Time-Series Modeling (Econometrics)
- Risk Management
- Optimization Techniques

Application Development
- Programming Techniques
- Interactive User Interfaces
- Parallel Computing

Content for On-site Customization
- Asset Allocation
- Interfacing with Databases
- Interfacing with Excel
Consulting Services

Accelerating return on investment

A global team of experts supporting every stage of tool and process integration

- Process Assessment
- Advisory Services
- Jumpstart
- Migration Planning
- Component Deployment
- Full Application Deployment
- Process and Technology Automation
- Process and Technology Standardization
- Continuous Improvement

Research | Advanced Engineering | Product Engineering Teams | Supplier Involvement