MATLAB EXPO 2018
Ampliando MATLAB Analytics con Kafka y Servicios en la Nube

Lucas García
Agenda

1. Access and Explore Data
   - Files
   - Databases
   - Sensors

2. Preprocess Data
   - Working with Messy Data
   - Data Reduction/Transformation
   - Feature Extraction

3. Develop Predictive Models
   - Model Creation e.g. Machine Learning
   - Parameter Optimization
   - Model Validation

4. Integrate with Production Systems
   - Desktop Apps
   - Enterprise Scale Systems
   - Embedded Devices and Hardware

5. Visualize Results
   - 3rd party dashboards
   - Web apps

MATLAB EXPO 2018
The Need for Large-Scale Streaming

Predictive Maintenance
*Increase Operational Efficiency*
*Reduce Unplanned Downtime*

More applications require near real-time analytics

Jet engine: ~800TB per day
Turbine: ~2 TB per day

Medical Devices
*Patient Safety*
*Better Treatment Outcomes*

Connected Cars
*Safety, Maintenance*
*Advanced Driving Features*

Car: ~25 GB per hour
Example Problem – How’s my driving?

- A group of MathWorks employees installed an OBD dongle in their car that monitors the on-board systems

- Data is streamed to the cloud where it is aggregated and stored

- We would like to use this data to score the driving habits of participants
Example: Fleet Analytics with MATLAB
Fleet Analytics Architecture
The first step is to clean up the incoming data.
The Data: Timestamped messages with JSON encoding

```
{  "vehicles_id": {"$oid":"55a3fd0069702d5b41000000"},
    "time": {"$date":"2015-07-13T18:01:35.000Z"},
    "kc" : 1975.0, "kff1225" : 100.65293, "kff125a" : 110.36619, ...
}
```

Key

```
{  "vehicles_id": {"$oid":"55a3fe3569702d5c5c000020"}
    "time": {"$date":"2015-07-13T18:01:53.000Z"},
    "kc" : 2000.0, "kff1225" : 109.65293, "kff125a" : 115.36619,
      ...
}
```

Timestamp

```
{  "vehicles_id": {"$oid":"55a4193569702d115b000001"}
    "time": {"$date":"2015-07-12T19:04:04.000Z"},
    "kc" : 2200.0, "kff1225" : 112.65293, "kff125a" : 112.36619,
      ...
}
```

Values
1. Access and Explore Data

Access a Sample of Data

Raw Data

<table>
<thead>
<tr>
<th>timestamp</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-Jan-2015 22:12:23</td>
<td>&quot;_id&quot;:</td>
<td></td>
</tr>
<tr>
<td>15-Jan-2015 22:12:24</td>
<td>&quot;_id&quot;:</td>
<td></td>
</tr>
<tr>
<td>15-Jan-2015 22:12:25</td>
<td>&quot;_id&quot;:</td>
<td></td>
</tr>
<tr>
<td>15-Jan-2015 22:12:26</td>
<td>&quot;_id&quot;:</td>
<td></td>
</tr>
</tbody>
</table>

✓ Decode JSON data
✓ Create Timetable

Timetable

<table>
<thead>
<tr>
<th>t = 4647×40 timetable</th>
</tr>
</thead>
<tbody>
<tr>
<td>trip_id</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
</tbody>
</table>
Develop a Preprocessing Function

**Timetable**

```matlab
% Sort rows
t = sortrows(t);
% Remove missing values
% MinNumMissing denotes the minimum number of missing values
% width(t) refers to the width of the timetable
% Here, we are removing rows with less than 2 missing values
% This is achieved by setting the missing values to NaN
% and then removing the rows where the sum of NaN values is greater than 2
% This is done using the rmmissing function
% The second argument 'MinNumMissing' specifies the minimum number of values
% that must be non-missing for a row to be kept
% The third argument 'width(t)-2' specifies the number of columns
% to be considered for the missing value check
% This is particularly useful when working with time series data
% where data points are missing due to sensor failures or other issues
% The code snippet below demonstrates this

% Perform windowed calculations
% Here, we calculate the moving median of the speed column
% The 'movmedian' function is used
% The third argument specifies the window size
% In this case, a window size of 3 is used
% This calculates the median of the last 3 values
% The result is then assigned back to the speed column
% This is done for each row of the timetable

% Calculate differences
% Here, we calculate the difference between consecutive
% values in the speed column
% The 'diff' function is used
% The result is assigned to the D1 column
% This is done for each row of the timetable

% Get time bounds
% Here, we get the minimum and maximum time values
% This is done using the 'bounds' function
% The result is assigned to the tmin and tmax variables
% These variables can be used
% for further time-related calculations or visualizations

% Convert time
% Here, we convert the time values from the timetable
% back to the original format
% The 'retime' function is used
% The first argument specifies the format
% The second argument specifies the type of time
% The third argument specifies the conversion
% The result is assigned to the tnew variable
% This is done for each row of the timetable
% The 'histcounts' variables can be used
% for further time-related calculations or visualizations
```

- **Clean up**
- **Enrich**
- **Restructure**
Ad Hoc Access to Data from MATLAB

Access the data in S3

Bring up the AthenaClient

```matlab
athenaClient = aws.athena.Client();
athenaClient.Database = 'trainingdata';
athenaClient.initialize();
```

Create a query and submit

```matlab
athenaClient.submitQuery('SELECT * FROM "trainingdata"."sampledata" limit 100', 's3://fleettrainingdata')
```

Fetch data as a table for easy analysis

```matlab
ds = datastore('s3://fleettrainingdata/*.csv');
ds.NumHeaderLines = 2;
data = table(ds);
```

Your usual MATLAB workflow goes here
Develop a Predictive Model

Production System

Kafka Connector

Storage Layer

Analytics Development

Algorithm Developers

MATLAB Distributed Computing Server

MATLAB

SDK

Business Decisions

Power BI

Qlik

Spotfire

Tableau

End Users

Business Systems

Edge Devices

API Gateway

AWS Lambda

kafka

Develop Predictive Models

MATLAB EXPO 2018
Everything you need to develop a predictive model is found in MATLAB

3 Develop Predictive Models

Label Events

Represent Signals

Scale Up

Evaluate tall expression using Spark Cluster:
- Pass 1 of 2: Completed in 11 sec
- Pass 2 of 2: Completed in 2.3333 min
Evaluation completed in 2.6167 min

Validate Model

Train Model

MATLAB EXPO 2018
Develop a Predictive Model in MATLAB
Integrate Analytics with Production Systems

Production System

Analytics Development

Kafka Connector

MATLAB Production Server

MATLAB Compiler SDK

MATLAB

Algorithm Developers

AWS Lambda

API Gateway

kafka

Storage Layer

Business Decisions

End Users

Business Systems

Power BI

Qlik

Spotfire
A quick Intro to Stream Processing

- **Batch Processing** applies computation to a finite sized historical data set that was acquired in the past

- **Stream Processing** applies computation to an unbounded data set that is produced continuously
Why stream processing?

Integrate with Production Systems

- Why stream processing?
- MATLAB Distributed Computing Server, MATLAB Compiler
- Stream Processing with MATLAB Production Server
- Edge Processing with MATLAB Coder

Value of data to decision making:
- Preventive / Predictive
- Actionable
- Reactive

Time critical decisions

Big Data processing on historical data

MATLAB Distributed Computing Server, MATLAB Compiler

Today’s example focuses here

Kafka, Event Hub, Kinesis

Integrate with Production Systems

- Near Real time decisions
- Big Data processing on historical data
- Today’s example focuses here

Time:
- Real-Time: Seconds, Minutes, Hours, Days, Months
- Historical

MATLAB EXPO 2018
Streaming data is treated as an unbounded Timetable.

**Input Table**

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Vehicle</th>
<th>RPM</th>
<th>Torque</th>
<th>Fuel Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>18:01:10</td>
<td>55a3fd</td>
<td>1975</td>
<td>100</td>
<td>110</td>
</tr>
<tr>
<td>18:10:30</td>
<td>55a3fe</td>
<td>2000</td>
<td>109</td>
<td>115</td>
</tr>
<tr>
<td>18:05:20</td>
<td>55a3fd</td>
<td>1980</td>
<td>105</td>
<td>105</td>
</tr>
<tr>
<td>18:10:45</td>
<td>55a3fd</td>
<td>2100</td>
<td>110</td>
<td>100</td>
</tr>
<tr>
<td>18:30:10</td>
<td>55a419</td>
<td>2000</td>
<td>100</td>
<td>110</td>
</tr>
<tr>
<td>18:35:20</td>
<td>55a419</td>
<td>1960</td>
<td>103</td>
<td>105</td>
</tr>
<tr>
<td>18:20:40</td>
<td>55a3fe</td>
<td>1970</td>
<td>112</td>
<td>104</td>
</tr>
<tr>
<td>18:39:30</td>
<td>55a419</td>
<td>2100</td>
<td>105</td>
<td>110</td>
</tr>
<tr>
<td>18:30:00</td>
<td>55a3fe</td>
<td>1980</td>
<td>110</td>
<td>113</td>
</tr>
<tr>
<td>18:30:50</td>
<td>55a3fe</td>
<td>2000</td>
<td>100</td>
<td>110</td>
</tr>
</tbody>
</table>

**Output Table**

<table>
<thead>
<tr>
<th>Time window</th>
<th>Vehicle</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>18:00:00</td>
<td>55a3fd</td>
<td>5</td>
</tr>
<tr>
<td>18:10:00</td>
<td>55a3fe</td>
<td>...</td>
</tr>
<tr>
<td>18:20:00</td>
<td>55a419</td>
<td>...</td>
</tr>
<tr>
<td>18:10:00</td>
<td>55a3fd</td>
<td>7</td>
</tr>
<tr>
<td>18:20:00</td>
<td>55a3fe</td>
<td>3</td>
</tr>
<tr>
<td>18:30:00</td>
<td>55a419</td>
<td>...</td>
</tr>
<tr>
<td>18:30:00</td>
<td>55a3fd</td>
<td>...</td>
</tr>
<tr>
<td>18:40:00</td>
<td>55a3fe</td>
<td>5</td>
</tr>
<tr>
<td>18:40:00</td>
<td>55a419</td>
<td>8</td>
</tr>
</tbody>
</table>
Introducing MATLAB Production Server

Data
- Databases: DynamoDB, SQL Server, MongoDB, Cosmos DB
- Cloud Storage: Azure Blob, AWS S3
- Streaming: Kafka, MQTT, Azure IoT Hub

Analytics
MATLAB Production Server
- Request Broker

Business System
- Dashboards: Qlik, Tableau, Microsoft Power BI, Spotfire
- Web: Microsoft IIS, Apache Tomcat, WebSphere
- Custom Apps: Google Cloud Platform, Azure, Amazon Web Services, Rackspace, OpenStack, VMware

Platform

MATLAB EXPO 2018
MATLAB Production Server is an application server that publishes MATLAB code as APIs.
Connecting MATLAB Production Server to Kafka

- Kafka client for MATLAB Production Server feeds topics to functions deployed on the server
- Configurable batch of messages passed as a MATLAB Timetable
- Each consumer process feeds one topic to a specified function
- Drive everything from a simple config file
  - No programming outside of MATLAB!

MATLAB EXPO 2018
Develop and Deploy a Stream Processing Function

Integrate with Production Systems

Edge Devices

API Gateway

AWS Lambda

kafka

Production System

Kafka Connector

MATLAB Production Server

MATLAB Analytics

Storage Layer

Analytics Development

MATLAB Compiler SDK

Algorithm Developers

Business Decisions

End Users

Power BI

Qlik

Spotfire

Tableau

Business Systems

MATLAB EXPO 2018
Develop a Stream Processing Function in MATLAB

Integrate with Production Systems

Process each window of data as it arrives

Current score

Previous state

Current window of data to be processed

MATLAB EXPO 2018
Develop a Stream Processing Function in MATLAB

Develop a Streaming Function

```matlab
function current_data = preprocessData(current_data)
    % Preprocess and perform calculations
    % Remove records with all missing data
    current_data = rmmissing(current_data,'MinNumMissing',width(current_data)-1);
    % Smooth and calculate approximate gradients
    current_data.Speed = movmedian(current_data.kff1001,5);
    current_data.D1 = [0;diff(current_data.kff1001)];
    current_data.D2 = [0;0;diff(current_data.kff1001,2)];

    % Count events for each ten second window
    countsByTime = countEvents(current_data);

    % Predict driving events
    current_data = predictEvents(current_data);

    % Write discrete data to mongodb
    updateResultsStore(car_id,countsByTime,resultsStore);

    % Update new state
    new_state = updateState(countsByTime,old_state);
end
```

Apply your pre-processing algorithm
Develop a Stream Processing Function in MATLAB

Develop a Streaming Function

```matlab
function new_state = calculateScores(car_id, current_data, old_state, resultsStore)
Preprocess and perform calculations
    current_data = preprocessData(current_data);
Predict driving events
    current_data = predictEvents(current_data);
Count events for each ten second window
    countsByTime = countEvents(current_data);
Write discrete data to mongdb
    updateResultsStore(car_id, countsByTime, resultsStore);
Update new state
    new_state = updateState(countsByTime, old_state);
end
```

Use the model you created with Classification Learner App

```matlab
function current_data = predictEvents(current_data)
    % Predict events for current data based on machine learning model
    predictorNames = {'kff1005','kff1006','kff125a','k10','kff1249','Speed','D1','D2','
                      'kff1001','kff1220','kff1221','kff1222','kff1223','...
                      'k47','kff124d'};
    predictors = current_data(:, predictorNames);
    mdl = load('machineLearningModel.mat');
    current_data.Event = predict(mdl.model, predictors);
end
```
Develop a Stream Processing Function in MATLAB

Develop a Streaming Function

function new_state = calculateScores(car_id, current_data, old_state, resultsStore)

Preprocess and perform calculations

current_data = preprocessData(current_data);

Predict driving events

current_data = predictEvents(current_data);

Count events for each ten second window

countsByTime = countEvents(current_data);

Write discrete data to mongodb

updateResultsStore(car_id, countsByTime, resultsStore);

Update new state

new_state = updateState(countsByTime, old_state);
end

Update Mongo database
- Count of events by type and location
- Results of driver scoring

MATLAB EXPO 2018
Debug a Stream Processing Function in MATLAB
Debug a Stream Processing Function in MATLAB

MATLAB EXPO 2018
Tie in your Dashboard Application

Integrate with Production Systems

Edge Devices

Analytics Development

Production System

Kafka Connector

MATLAB Production Server

MATLAB Analytics

Storage Layer

Business Decisions

Business Systems

End Users

API Gateway

AWS Lambda

kafka

Algorithm Developers

MATLAB Compiler SDK

MATLAB

4

AWS Lambda

Power BI

Qlik

Tableau

Qlik

Tableau

MATLAB EXPO 2018
Complete Your Application

Visualize Results

Fleet Summary

Fleet Statistics

Accelaration/Deceleration Events, 2014 - 2017

- Aggressive acceleration: 24296 Events
- Speeding: 1636 Events
- Aggressive deceleration: 15303 Events
- Moderate deceleration: 1909 Events
- Safe acceleration: 815 Events
- Slow deceleration: 1128 Events
Scalable Analytics with Enterprise BI Tools

TIBCO Spotfire

Tableau

Tableau

MATLAB EXPO 2018
Key Takeaways

➢ MATLAB connects directly to your data so you can quickly design and validate algorithms

➢ The MATLAB language and apps enable fast design iterations

➢ MATLAB Production Server enables easy integration of your MATLAB algorithms with enterprise production systems

➢ You to spend your time understanding the data and designing algorithms
Resources to learn and get started

- Data Analytics with MATLAB
- MATLAB Production Server
- MATLAB Compiler SDK
- Statistics and Machine Learning Toolbox
- Database Toolbox
- Mapping Toolbox
- MATLAB with TIBCO Spotfire
- MATLAB with Tableau
- MATLAB with MongoDB