MATLAB EXPO 2018

엔터프라이즈, 빅 데이터 및 애널리틱 솔루션 활용을 위한 MATLAB 적용기술 소개

성 호 현 부장
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

Medical Devices
- Patient Safety
- Better Treatment Outcomes

Connected Cars
- Safety, Maintenance
- Advanced Driving Features

Jet engine: ~800TB per day
Turbine: ~ 2 TB per day
Car: ~25 GB per hour

More applications require near real-time analytics
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

Edge Devices

Production System

Analytics Development

Business Decisions

- API Gateway
- AWS Lambda

- Kafka Connector
- Kafka

- MATLAB Production Server
- MATLAB Analytics
- Storage Layer

- MATLAB Compiler SDK

- Algorithm Developers

- End Users

- Power BI
- Qlik
- Spotfire
- Tableau

MATLAB EXPO 2018
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, ...
}
{  "vehicles_id": {"$oid":"55a3fe3569702d5c5c000020"},  "time": {"$date":"2015-07-13T18:01:53.000Z"},  "kc": 2000.0,  "kff1225": 109.65293,  "kff125a": 115.36619, ...
}
{  "vehicles_id": {"$oid":"55a4193569702d115b000001"},  "time": {"$date":"2015-07-12T19:04:04.000Z"},  "kc": 2200.0,  "kff1225": 112.65293,  "kff125a": 112.36619, ...
}
```
Access a Sample of Data

Raw Data

<table>
<thead>
<tr>
<th>timestamp</th>
<th>1 value</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-Jan-2015</td>
<td>...</td>
</tr>
<tr>
<td>22:12:23</td>
<td>...</td>
</tr>
<tr>
<td>15-Jan-2015</td>
<td>...</td>
</tr>
<tr>
<td>22:12:24</td>
<td>...</td>
</tr>
<tr>
<td>15-Jan-2015</td>
<td>...</td>
</tr>
<tr>
<td>22:12:25</td>
<td>...</td>
</tr>
<tr>
<td>15-Jan-2015</td>
<td>...</td>
</tr>
<tr>
<td>22:12:26</td>
<td>...</td>
</tr>
</tbody>
</table>

Timetable

<table>
<thead>
<tr>
<th>t</th>
<th>timetable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sun Jul 12 16:18:41 UTC 2015</td>
</tr>
<tr>
<td>2</td>
<td>Sun Jul 12 16:18:42 UTC 2015</td>
</tr>
<tr>
<td>3</td>
<td>Sun Jul 12 16:18:43 UTC 2015</td>
</tr>
<tr>
<td>4</td>
<td>Sun Jul 12 16:18:44 UTC 2015</td>
</tr>
<tr>
<td>5</td>
<td>Sun Jul 12 16:18:45 UTC 2015</td>
</tr>
<tr>
<td>6</td>
<td>Sun Jul 12 16:19:13 UTC 2015</td>
</tr>
<tr>
<td>7</td>
<td>Sun Jul 12 16:19:14 UTC 2015</td>
</tr>
<tr>
<td>8</td>
<td>Sun Jul 12 16:19:15 UTC 2015</td>
</tr>
<tr>
<td>9</td>
<td>Sun Jul 12 16:19:16 UTC 2015</td>
</tr>
</tbody>
</table>

✓ Decode JSON data
✓ Create Timetable
Develop a Preprocessing Function

Preprocess Data

Timetable

Preprocess data

\[ t = \text{sortrows}(t); \]
\[ t = \text{rmmissing}(t, 'MinNumMissing', \text{width}(t)-2); \]

Perform windowed calculations

\[ t.\text{Speed} = \text{movmedian}(t.\text{SpeedGPS}, 3); \]
\[ t.D1 = [0; \text{diff}(t.\text{SpeedGPS})]; \]

\[ [tmin, tmax] = \text{bounds}(t.\text{time}); \]
\[ \text{tnew} = tmin: \text{seconds}(10): tmax; \]
\[ \text{countsByTime} = \text{retime}(t(:,:, 'Event'), \text{tnew}, @\text{histcounts}); \]
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://fleetrainingdata')
```

Fetch data as a table for easy analysis

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

Your usual MATLAB workflow goes here
Develop a Predictive Model

Production System

Analytics Development

Business Decisions

Edge Devices

MATLAB Distributed Computing Server

Kafka Connector

API Gateway

AWS Lambda

kafka

Storage Layer

MATLAB

SDK

Algorithm Developers

End Users

Power BI

Tableau

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

1. Label Events
2. Represent Signals
3. Train Model
4. Scale Up
5. Validate Model

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

**Production System**
- Kafka Connector
- MATLAB Production Server
- Storage Layer

**Analytics Development**
- MATLAB Compiler SDK
- Algorithm Developers

**Business Decisions**
- Power BI
- Tableau

**End Users**
- Business Systems

**Edge Devices**
- API Gateway
- AWS Lambda

**Integrate with Production Systems**

---

MATLAB EXPO 2018

16
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?

Edge Processing with MATLAB Coder

Stream Processing with MATLAB Production Server

- Reactive
- Actionable
- Preventive/Predictive

Today's example focuses here

- Real-Time Seconds
- Minutes
- Hours
- Days
- Months

Time

Value of data to decision making

MATLAB Distributed Computing Server, MATLAB Compiler

Integrate with Production Systems

Kinesis Event Hub

Big Data processing on historical data

MATLAB EXPO 2018
Streaming data is treated as an unbounded Timetable
Introducing MATLAB Production Server

Data

- Databases
  - DynamoDB
  - SQL Server
  - Cassandra
  - Cosmos DB

- Cloud Storage
  - Azure Blob

- Streaming
  - AWS Kinesis
  - 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
  - OSIsoft PI System
  - Kafka
  - MQTT

Platform

Google Cloud Platform

Microsoft

Amazon Web Services

Rackspace

OpenStack

VMware

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!
Develop and Deploy a Stream Processing Function
Develop a Stream Processing Function in MATLAB

```matlab
function new_state = calculateScore(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
```

- Process each window of data as it arrives
- Current score
- Previous state
- Current window of data to be processed
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)];
end
```

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

Use the model you created with Classification Learner App

```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 mongodb
    updateResultsStore(car_id, countsByTime, resultsStore);

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

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,centsByTime,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
Debug a Stream Processing Function in MATLAB

Production System

- Kafka Connector
- kafka
- Storage Layer

Analytics Development

- MATLAB Compiler SDK
- Algorithm Developers

Integrate with Production Systems

Edge Devices

- Edge Devices
- Production System

Business Decisions

- Power BI
- Qlik
- Spotfire
- Tableau

End Users

MATLAB EXPO 2018
Debug a Stream Processing Function in MATLAB
Tie in your Dashboard Application
Complete Your Application

Visualize Results
Scalable Analytics with Enterprise BI Tools

TIBCO Spotfire

Tableau
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