MATLAB EXPO 2018
KOREA
MATLAB EXPO 2018

MATLAB을 이용한 머신 러닝
(기본)

Senior Application Engineer
엄 준 상 과장
Machine Learning is Everywhere

Solution is too complex for hand written rules or equations
- Speech Recognition
- Object Recognition
- Engine Health Monitoring

Solution needs to adapt with changing data
- Weather Forecasting
- Energy Load Forecasting
- Stock Market Prediction

Solution needs to scale
- IoT Analytics
- Taxi Availability
- Airline Flight Delays

learn complex non-linear relationships
update as more data becomes available
learn efficiently from very large data sets
Bazille’s Studio
Bazille 1870

Shuffleton’s Barbershop
Rockwell 1950
Artistic Style Classification

Image Feature Extraction

Visual Features

Machine Learning Classification

- **Style** Classifier (SVM)
  - **Style**: Regionalism

- **Genre** Classifier (SVM)
  - **Genre**: Interior

- **Artist** Classifier (SVM)
  - **Artist**: Rockwell
Machine Learning

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

**Standard Approach**

**Hand Written Program**

If RSI > 70
then “SELL”
If MACD > SIG and RSI <= 70
then “HOLD”
...

**Formula or Equation**

\[ Y_{trade} = \beta_1 X_{RSI} + \beta_2 X_{MACD} + \beta_3 X_{TSMom} + \ldots \]

**Machine Learning Approach**

**Inputs → Outputs**

**Prediction** = \( F(\text{factors, trade decision}) \)

MATLAB EXPO 2018
Different Types of Learning

**Type of Learning**

- **Supervised Learning**
  - Develop predictive model based on both input and output data

- **Unsupervised Learning**
  - Discover an internal representation from input data only

**Categories of Algorithms**

- **Classification**
  - Output is a choice between classes (True, False) (Red, Blue, Green)

- **Regression**
  - Output is a real number (temperature, stock prices)

- **Clustering**
  - No output - find natural groups and patterns from input data only
**Objective:**
Train a classifier to classify human activity from sensor data

**Data:**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-axial Accelerometer</td>
<td>![activity icons]</td>
</tr>
<tr>
<td>3-axial Gyroscope</td>
<td>![activity icons]</td>
</tr>
</tbody>
</table>

**Approach:**
- Import data
- Interactively train and compare classifiers
- Test results on new sensor data
Example: Regression

Objective:
Easy and accurate computation of day-ahead system load forecast

Type of Learning
- Supervised Learning
  - Develop predictive model based on both input and output data
- Unsupervised Learning
  - Discover an internal representation from input data only

Categories of Algorithms
- Classification
- Regression
- Clustering

Objective:
Easy and accurate computation of day-ahead system load forecast
Example: Clustering

Machine Learning
- Supervised Learning
  - Develop predictive model based on both input and output data
- Unsupervised Learning
  - Discover an internal representation from input data only

Type of Learning
- Classification
- Regression
- Clustering

Categories of Algorithms

Objective:
Given data for engine speed and vehicle speed, identify clusters
Different Types of Learning

- **Machine Learning**
  - **Supervised Learning**
    - Develop predictive model based on both input and output data
  - **Unsupervised Learning**
    - Discover an internal representation from input data only

**Categories of Algorithms**

- **Classification**
  - Support Vector Machines
  - Discriminant Analysis
  - Naive Bayes
  - Nearest Neighbor

- **Regression**
  - Linear Regression
  - GLM
  - SVR, GPR
  - Ensemble Methods
  - Regression Trees
  - Neural Networks

- **Clustering**
  - kMeans, kmedoids
  - Fuzzy C-Means
  - Hierarchical
  - Gaussian Mixture
  - Neural Networks
  - Hidden Markov Model
Case Study

Machine learning techniques for algorithmic trading

- **Goal**
  - Developing a trading strategy

- **Data**
  - Multiple factors
  - Based on fundamentals or price data
  - Tested on historical data
The Challenge

- **Persona:** FX Trader

- **Question:** Can we predict the future price/return of a currency pair
  - E.g. 60 minutes into the future

- **Using:** Historical intra-day data
  - Recent returns
  - Technical Indicators

- **Creating:** A predictive model
  - Regression / machine-learning
  - Backtest over a suitable period of time
Data

- Currency Pair: EURUSD
- Data: Ten years of one-minute bar prices
  - bid/ask/mid
- Stored: In timetable objects

<table>
<thead>
<tr>
<th>Time</th>
<th>Mid</th>
<th>Bid</th>
<th>Ask</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-Jan-2007 00:00:00</td>
<td>1.31916</td>
<td>1.31908</td>
<td>1.31924</td>
</tr>
<tr>
<td>01-Jan-2007 00:01:00</td>
<td>1.31929</td>
<td>1.31921</td>
<td>1.31937</td>
</tr>
<tr>
<td>01-Jan-2007 00:02:00</td>
<td>1.31954</td>
<td>1.31946</td>
<td>1.31962</td>
</tr>
<tr>
<td>01-Jan-2007 00:03:00</td>
<td>1.31963</td>
<td>1.31958</td>
<td>1.31960</td>
</tr>
<tr>
<td>01-Jan-2007 00:04:00</td>
<td>1.31952</td>
<td>1.31945</td>
<td>1.31959</td>
</tr>
<tr>
<td>01-Jan-2007 00:05:00</td>
<td>1.3195</td>
<td>1.31942</td>
<td>1.31958</td>
</tr>
<tr>
<td>01-Jan-2007 00:06:00</td>
<td>1.31945</td>
<td>1.3194</td>
<td>1.3195</td>
</tr>
<tr>
<td>01-Jan-2007 00:07:00</td>
<td>1.31965</td>
<td>1.31962</td>
<td>1.31968</td>
</tr>
<tr>
<td>01-Jan-2007 00:08:00</td>
<td>1.31958</td>
<td>1.31953</td>
<td>1.31963</td>
</tr>
<tr>
<td>01-Jan-2007 00:09:00</td>
<td>1.319525</td>
<td>1.31945</td>
<td>1.3196</td>
</tr>
</tbody>
</table>
Factor Creation - Predictors

- A mixture of factors from the Financial Toolbox and hand written
  - Toolbox
    - rsindex (5, 10, 15, 20, 25, 30 & 60 minute)
    - macd
  - Derived
    - N-Minute return (5, 10, 15, 20, 25, 30 & 60 minute)
The Trading Model

- Train a model based around a number of factors
  - Technical Indicators & Short Term Returns
  - Attempt to predict positive or negative future returns using current information
  - Trade on this prediction

- Model Selection
  - Linear regression and stepwise
  - Classification Tree

- Backtesting
  - Test over ten years, taking into account bid/offer spread as trading cost
  - Varying our length of in-sample and out-sample
Step1 : Data Regression

- Continuous Data to Discrete Data

- Linear Model and Stepwise Regression
  - fitlm, stepwiselm

- Two month In-Sample and one month Out-Sample
  - timerange
Machine Learning App

- Point and click interface – no coding required
- Quickly evaluate, compare and select regression models
- Export and share MATLAB code or trained models
Regression Learner App

Same workflow as Classification Learner:

- Linear Regression
- Trees
- SVMs
- Gaussian Process Regression
- Ensembles
Demo – Long, Short Classification

- Repeat this process, switching to machine learning and **supervised learning**

- **Supervised learning**
  - *The machine learning task of inferring a function from labelled training data*

- **Our labelling**
  - Look at the median bid/offer spread (in pips)
  - Classify our problem as
    - +1 - where the future return is +ive & greater than the spread (i.e. go long)
    - -1 - where the future return is -ive & greater than the spread (go short)
    - 0 – all other cases
### Building the classification model

- **Form the predictor/response table**
  - Yesterday’s factor row, today’s return

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>...</th>
<th>FN</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1(1)</td>
<td>F2(1)</td>
<td>F3(1)</td>
<td>F4(1)</td>
<td>F5(1)</td>
<td>...</td>
<td>FN(1)</td>
<td>R(2)</td>
</tr>
<tr>
<td>F1(2)</td>
<td>F2(2)</td>
<td>F3(2)</td>
<td>F4(2)</td>
<td>F5(2)</td>
<td>...</td>
<td>FN(2)</td>
<td>R(3)</td>
</tr>
<tr>
<td>F1(3)</td>
<td>F2(3)</td>
<td>F3(3)</td>
<td>F4(3)</td>
<td>F5(3)</td>
<td>...</td>
<td>FN(3)</td>
<td>R(4)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>F1(M-1)</td>
<td>F2(M-1)</td>
<td>F3(M-1)</td>
<td>F4(M-1)</td>
<td>F5(M-1)</td>
<td>...</td>
<td>FN(M-1)</td>
<td>R(M)</td>
</tr>
</tbody>
</table>
Classification Learner App

- App to apply advanced classification methods to your data
  - Discriminant analysis
  - Dimension reduction via PCA
  - Parallel coordinates plot
  - Categorical predictors
  - Train classifiers in parallel

- Also: Table and categorical support via command line
Demo – Historical Backtesting

- Use MATLAB scripting as a backtesting environment

- Loop through our dataset using `datetime` and `dateshift`
  - Run `fitlm` and `fittree` at each iteration

- Adding transaction costs where we trade
  - Once again, based on the bid/ask spread
BackTest Result
Feature Selection

Why?
- Reduce data size (compute/storage gains) and model complexity (prevent overfitting)

When?
- High dimensional datasets with poor feature to observation ratio

Capabilities
- Accuracy comparable to state-of-art techniques
- Regularization to control sparsity and redundancy
- Handles high dimensional data and scales to large datasets
Fine-tuning Model Parameters

Why?
- Manual parameter selection is tedious and may result in suboptimal performance

When?
- When training a model with one or more parameters that influence the fit

Capabilities
- Efficient compared to standard optimization techniques or grid search
- Tightly integrated with fit function API with pre-defined optimization problem (e.g. bounds)

Hyperparameter Tuning with Bayesian Optimization

Previously tuning these parameters was a manual process.
Machine Learning with Big Data

Why?
- Learning on larger datasets often leads to better generalization but they don’t fit in memory

When?
- Data does not fit in memory
- Data lives remotely on clusters

Capabilities
- Functions for deriving summary statistics and generating visualizations
- Machine learning algorithms for classification, regression and clustering

“Tall” data types and functions for out-of-memory data

Tall Data Types
- Text
- Spreadsheet (Excel)
- Database (SQL)
- Image
- table
- cell
- numeric
- cellstr & string
- Date & Time
- categorical

Exploration & Pre-processing
- Numeric functions
- Basic stats reductions
- Date/Time capabilities
- Categorical
- String processing
- Table wrangling
- Missing Data handling
- Summary visualizations:
  - Histogram/histogram2
  - Kernel density plot
  - Bin-scatter

Machine Learning
- Linear Model
- Logistic Regression
- Discriminant analysis
- K-means
- PCA
- Random data sampling
- Summary statistics
Convolutional Neural Networks (CNN)

- CNN take a fixed size input and generate fixed-size outputs.
- Convolution puts the input images through a set of convolutional filters, each of which activates certain features from the input data.
Time Series Analysis – LSTM Layers

To train a deep neural network to classify sequence data, you can use an LSTM network. An LSTM network enables you to input sequence data into a network, and make predictions based on the individual time steps of the sequence data.
## Deep Learning

<table>
<thead>
<tr>
<th>Classification</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ConvNets</strong></td>
<td><strong>ConvNets</strong></td>
</tr>
</tbody>
</table>
| % Define network architecture
layers = [ imageInputLayer([28 28 1])
  convolution2dLayer(5,20)
  fullyConnectedLayer(10)
  softmaxLayer()
  classificationLayer() ];
% Train the network.
options = trainingOptions( 'adam' );
net = trainNetwork( X, Y, layers, options ); |
| **LSTM Networks** | **LSTM Networks** |
| % Define network architecture
layers = [ sequenceInputLayer(25)
  lstmLayer(100)
  fullyConnectedLayer(10)
  softmaxLayer()
  classificationLayer() ];
% Train the network.
options = trainingOptions( 'adam' );
net = trainNetwork( X, Y, layers, options ); | % Define network architecture
layers = [ sequenceInputLayer(25)
  lstmLayer(100)
  fullyConnectedLayer(10)
  softmaxLayer()
  classificationLayer() ];
% Train the network.
options = trainingOptions( 'adam' );
net = trainNetwork( X, Y, layers, options ); |
CNN and LSTM Result
Data Analytics Workflow

Access and Explore Data
- Files
- Databases
- Sensors

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

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

Integrate Analytics with Systems
- Desktop Apps
- Enterprise Scale Systems
- Embedded Devices and Hardware
Solution for Data Analytics

Access and Explore Data

- MATLAB Analytics work with business and engineering data
- Repositories: Databases (SQL), NoSQL, Hadoop
- File I/O: Text, Spreadsheet, XML
- Web Sources: RESTful, JSON, HTML, Mapping, Financial data feeds

Preprocess Data

- MATLAB lets engineers do Data Science themselves
- Communication Protocols: CAN (Controller Area Network), DDS (Data Distribution Service), OPC (OLE for Process Control), XCP (Explicit Control Protocol)
- Real-Time Sources: Sensors, GPS, Instrumentation, Cameras, Communication systems, Machines (embedded systems)

Develop Predictive Models

- Machine Learning, Statistics, Neural Networks, Signal Processing

Integrate Analytics with Systems

- MATLAB Analytics run anywhere
- Excel Add-in, Hadoop, Standalone Application, C/C++, Java, .NET, C, C++, HDL, PLC
- Cloud, Web, Cluster, GPU, Desktop, Mobile
- Microcontrollers, DSP chips, FPGAs, Industrial automation systems

MATLAB EXPO 2018
Additional Resources

Documentation:
