MATLAB EXPO 2016
Develop Predictive Maintenance Algorithms using MATLAB

Dr. Sarah Drewes, MathWorks Consulting Services
Different Types of Learning

Machine Learning

- Supervised Learning
  - Classification
  - Regression
  - Output is a choice between classes
    - (True, False) (Red, Blue, Green)
  - Regression
    - Output is a real number (temperature, stock prices)

- Unsupervised Learning
  - Discover a good internal representation
  - Learn a low dimensional representation
Classification in Predictive Maintenance

- Parameters/Predictors: Sensor data, control settings
- Classes/States: Failure states, time horizon until failure/ material fatigue

Goal: Predict failure from sensor data

Prerequisites:
- Machine-readable data format
- Sufficient historical data containing meaningful information
Classification model generation @MONDI Gronau

Which sensor measurements indicate machine failure?
Classification model generation

Basic Workflow

1. Preprocess Data
2. Choose Algorithm
3. Fit Model
4. Evaluate Model
5. Choose Model
6. Make Predictions
Classification model generation - Prepare data

- Preprocess sensor data: clean invalid data, disregard constant values, identify data types
- Aggregate per time stamp

Sensor Data (10-100 /plant)

Quality State

update ~ 1 min.
update ~ 60-90 min.
Classification model generation

Basic Workflow

1. Preprocess Data
2. Choose Algorithm
3. Fit Model
4. Evaluate Model
5. Choose Model
6. Make Predictions
Classification model generation
Choose algorithms

Possible Classification Methods

Statistics and Machine Learning

**Nearest Neighbor Classification**

**Support Vector Machines (SVMs)**

**Classification Trees**

**Naive Bayes Classification**

**Discriminant Analysis**

**Neural Network**
Classification model generation
Choose an algorithm

- Distinguish 'categorical' (= discrete) and other (= continuous) predictors

- A priori analysis of data, e.g., test for normal distribution

- Reduce dimension of predictor variables, e.g., principal component analysis (PCA)

- Use ensemble learning to reduce sensitivity of learning algorithms, e.g. TreeBagger for classification trees
## Classification model generation

### Choose an algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Function</th>
<th>Categorical Predictors?</th>
<th>Data</th>
<th>Functions to Examine Data</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest Neighbor</td>
<td>fitcknn</td>
<td>Y (but not both)</td>
<td>Normalize (distance-based calculation).</td>
<td>pdist pdist2</td>
<td>Better results in lower dimensions. High memory usage.</td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td>fitcdiscr</td>
<td>N Y</td>
<td>Multivariate normal distribution by class.</td>
<td>cov vartestn anoval kruskalwallis</td>
<td>Determines mean and covariance for each class. Can specify linear or quadratic discriminant type.</td>
</tr>
<tr>
<td>Trees</td>
<td>fitctree fitrtree</td>
<td>Y</td>
<td>Any arrangement. Binary comparisons and structure of tree can be examined/adjusted.</td>
<td>view</td>
<td>Computationally efficient. Highly sensitive to training data.</td>
</tr>
<tr>
<td>SVM</td>
<td>fitcsvm fitcecoc</td>
<td>N Y</td>
<td>Linearly separable hyperplane (can specify nonlinear kernel).</td>
<td>ksdensity</td>
<td>Can specify nonlinear kernel distributions. Can adjust optimization parameters.</td>
</tr>
<tr>
<td>Neural Network</td>
<td>patternnet</td>
<td>N Y</td>
<td>Transpose (columns are observations). All data must be numeric.</td>
<td>dummyvar plotconfusion plotroc</td>
<td>Use dummyvar for categorical classes. Models are available as Simulink® blocks.</td>
</tr>
</tbody>
</table>
Classification model generation

Basic Workflow

1. Preprocess Data
2. Choose Algorithm
3. Fit Model
4. Evaluate Model
5. Choose Model
6. Make Predictions
Classification model generation
Fit model

Fit model based on historic data

Training Data, e.g. 70% of historic data

$\text{PredictionModel} = \text{fitcxxx}(\text{PARAMETER}, \text{STATE})$
Classification model generation

Fit model
Classification model generation

Basic Workflow

1. Preprocess Data
2. Choose Algorithm
3. Fit Model
4. Evaluate Model
5. Choose Model
6. Make Predictions
Classification model generation
Evaluate model

predictedState = PredictionModel(\text{Parameter})

Validation Data, e.g. 30% of historic data

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>PARAMETER</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'2015-07-14 00:49:12.0'</td>
<td>160 160 160 160 1000 7 1000 9 33 32</td>
<td>1</td>
</tr>
<tr>
<td>'2015-07-14 00:50:12.0'</td>
<td>160 160 160 160 1000 8 1000 10 33 32</td>
<td>1</td>
</tr>
<tr>
<td>'2015-07-14 00:51:13.0'</td>
<td>160 160 160 160 1000 8 1000 10 33 32</td>
<td>1</td>
</tr>
<tr>
<td>'2015-07-14 00:52:12.0'</td>
<td>160 160 160 160 1000 8 1000 10 33 32</td>
<td>1</td>
</tr>
<tr>
<td>'2015-07-14 00:53:12.0'</td>
<td>160 160 160 160 1000 8 1000 11 33 32</td>
<td>2</td>
</tr>
<tr>
<td>'2015-07-14 00:54:12.0'</td>
<td>160 160 160 160 1000 8 1000 12 33 32</td>
<td>2</td>
</tr>
<tr>
<td>'2015-07-14 00:55:12.0'</td>
<td>160 160 160 160 1000 8 1000 10 33 32</td>
<td>2</td>
</tr>
</tbody>
</table>

Misclassification rate 1 of 7: 14.28 %
Accuracy: 85.72 %
Classification model generation
Evaluate model - using Classification Learner App
Classification model generation

Basic Workflow

- Preprocess Data
- Choose Algorithm
- Fit Model
- Evaluate Model
- Choose Model
- Make Predictions

For each classification method
Choose Model with best misclassification rate

<table>
<thead>
<tr>
<th>Model</th>
<th>Nearest Neighbour</th>
<th>TreeBagger</th>
<th>NeuralNetwork</th>
<th>NaiveBayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>M151</td>
<td>24%</td>
<td>2%</td>
<td>8%</td>
<td>10%</td>
</tr>
<tr>
<td>M152</td>
<td>44%</td>
<td>5%</td>
<td>23%</td>
<td>13%</td>
</tr>
<tr>
<td>M153</td>
<td>23%</td>
<td>2%</td>
<td>13%</td>
<td>13%</td>
</tr>
<tr>
<td>M156</td>
<td>12%</td>
<td>2%</td>
<td>3%</td>
<td>9%</td>
</tr>
<tr>
<td>M157</td>
<td>11%</td>
<td>1%</td>
<td>10%</td>
<td>8%</td>
</tr>
<tr>
<td>M158</td>
<td>29%</td>
<td>2%</td>
<td>14%</td>
<td>17%</td>
</tr>
<tr>
<td>M159</td>
<td>21%</td>
<td>0%</td>
<td>3%</td>
<td>2%</td>
</tr>
<tr>
<td>M181</td>
<td>1%</td>
<td>0%</td>
<td>1%</td>
<td>2%</td>
</tr>
</tbody>
</table>
Classification model generation
Choose model
Classification model generation

Basic Workflow

1. Preprocess Data
2. Choose Algorithm
3. Fit Model
4. Evaluate Model
5. Choose Model
6. Make Predictions
Predictive monitoring at MONDI Gronau - Use the predictive model

Predict current machine states during operation.

State is: ok

Sensor Data (10-100 /plant)

Quality State

Sensor data (now)

Predicted State (now)

Update Prediction Model (historic data)
Process monitoring at MONDI Gronau – Domain knowledge and tools

Tools:
- MATLAB
- Database Toolbox
- Statistics and Machine Learning Toolbox
- Neural Network Toolbox
- MATLAB Compiler