APPLICATION OF MACHINE LEARNING ALGORITHMS TO ON-BOARD DIAGNOSTICS (OBD II) THRESHOLD DETERMINATION

TONY GULLITTI
ROBERT BOSCH, LLC

MATHWORKS AUTOMOTIVE CONFERENCE 2017
Machine Learning for OBD

Background: On-Board Diagnostics & Boundary

- On-Board Diagnostics is the requirement that vehicles must light the Malfunction Indicator Light (MIL) if an emissions related component is reporting a signal outside its expected operating range
- A threshold is determined based on empirical data
  - $4\sigma/3\sigma$: Preferred Guideline
  - More separation is better

![Diagram showing passing and failing data with thresholds at $4\sigma$ and $3\sigma$]
Machine Learning for OBD

Background: On-Board Diagnostics & Boundary

- Separation is needed to minimize:
  - False failure
  - False pass

- Diagnostic should run consistently on
  - The certification test cycle: FTP75
  - In the field: In Use Monitoring Performance Ratio (IUMPR)
Machine Learning for OBD
Example: Pressure Sensor Diagnostic

- **OBD Goal:**
  - Monitor pressure and determine whether connection is connected or disconnected
  - Delta pressure sensor (dP) measures pressure relative to atmosphere
    - Disconnected/Failing – positive pressure
    - Connected/Passing – negative pressure
  - All data units are normalized
  - Select the range (enable criteria) where separation is acceptable
    - Use values of Input #1 and Input #2 where there is no overlap
  - Determine the separation boundary
    - dP above the boundary – Light the MIL !!!
OBD Boundary Determination: Classic Method

- Make a grid of Input #1 & #2
- Each set of grid points has a column of passing & failing data
- Within each of the columns calculate for dP:
  - Mean (passing, failing) \( \mu_p, \mu_f \)
  - Standard Deviation (passing, failing) \( \sigma_p, \sigma_f \)
- Compute acceptable boundaries:
  - Failing: \( \mu_f - 3\sigma_f \)
  - Passing: \( \mu_p + 4\sigma_p \)
- Separation is achieved where
  - \( \mu_p + 4\sigma_p < \mu_f - 3\sigma_f \)
Machine Learning for OBD

OBD Boundary Determination: Classic Method

- Adjust the enable criteria to:
  - Enable the monitor at any coordinates where separation is achieved
  - Where separation is marginal, enable the monitor by selectively adding coordinates until monitor runs frequently enough to meet in use monitor criteria

- Calculate the boundary based on the larger of:
  - $\mu_p + 4\sigma_p$, $\mu_f - 3\sigma_f$

Use statistical analysis to guide us before we put the enable criteria and boundary in the controller!
Machine Learning for OBD

Rationale for Machine Learning

▪ Data is continuous and plentiful
  ▪ Fit a continuous function

▪ What is Machine Learning?
  ▪ The ability of computers to learn without explicitly being programmed
  ▪ Inputs predict the outputs via a model which has been fit
  ▪ Fitting is known as training or optimization
    ▪ Multiple Linear Regression
    ▪ Logistic Regression
    ▪ Multivariate Gaussian Distribution
    ▪ Principle Component Analysis
Freely Available: Coursera Machine Learning Course

www.coursera.org

Machine Learning Course, Andrew Ng
Machine Learning for OBD
Supervised Machine Learning

New Dataset

Training Dataset
Population

Regression
Output has continuous possible range

Classification
Output has a discrete possible range

Given the input feature set, predict the most likely output label value.
Regression

\[ y_p(x_1, x_2) = f\left( \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \ldots \right) + \theta_4 x_1^2 + \theta_5 x_2^2 \]

\[ y_f(x_1, x_2) = f\left( \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \ldots \right) + \theta_4 x_1^2 + \theta_5 x_2^2 \]

- Where:
  - \( x_1 \) = Input #1
  - \( x_2 \) = Input #2
  - \( y_p(x_1, x_2) \) = Delta Pres Passing data
  - \( y_f(x_1, x_2) \) = Delta Pres Failing data

- \([b, bint, r, rint, stats] = \text{regress}(y, X, alpha);\)
  - \( \text{alpha} \) is the requested confidence interval to be returned in \( \text{bint} \)
Machine Learning for OBD
Multiple Linear Regression

- Determine the parameters, $y(x_1, x_2) : \theta$ such that:
  - $y(x_1, x_2)$ predicts the mean value of $dP$
  - Confidence interval of fit determines separation
    - $-3\sigma = 99.557\%$ Confidence interval failing
    - $4\sigma = 99.987\%$ Confidence interval passing

- Adjust the threshold as before

- Give the algorithm some help
  - Filter out passing & failing data that overlap
Machine Learning for OBD
Logistic Regression: A Classification Approach

- Define an equation to be used as the boundary
  \[ h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1 x_2 x_3 + \cdots + \theta_5 x_1^2 + \theta_6 x_2^2 + \theta_7 x_3^2 \]
  - \( x_1 = \text{Input #1} \)
  - \( x_2 = \text{Input #2} \)
  - \( x_3 = \text{dP} \) ← Now an input!

- Predict the \textbf{probability} that signal is passing
- The probability is achieved by using the \textbf{sigmoid} function
  \[ \text{sigmoid}(h(x)) = \frac{1}{1 + e^{-h(x)}} \]

- Find the coordinates of Input #1, Input #2 and dP where:
  - \( \text{sigmoid}(h(x)) = 0.5 \)
  - The resulting surface is the boundary
Logistic Regression Training

Create $y$ such that:

- Optimize the parameters of $h(x)$: $\theta$
  - $y = \text{sigmoid}(h(x))$
  - Minimize the difference between
    - 0 and $\text{sigmoid}(h(x))$ for failing data
    - 1 and $\text{sigmoid}(h(x))$ for passing data

- MATLAB: `fminunc` – find the minimum of unconstrained multivariable function

- Objective function (Matlab syntax)
  $$J = \frac{1}{m} \left( \sum y \cdot \log(\text{sigmoid}(X\theta)) + (1-y) \cdot \log(1-\text{sigmoid}(X\theta)) \right) + \frac{\lambda}{2m} \sum \theta^2;$$

- **Regularized** Logistic Regression
To obtain the boundary

- Create a grid of points in the three dimensions – **fullfact**
  - In 2D can use contour plot
- Input the grid values to the trained equation
  - \( \text{sigmoid}(h(x)) \)
- Find the coordinates where
  - \( \text{sigmoid}(h(x)) = 0.5 \)
  - Range 0.45 to 0.55

Explore adjusting the probability value to achieve the desired separation

- Closer to 1 → passing bias
- Closer to 0 → failing bias
Logistic Regression Using Matlab Command Line

- Logistic Regression using Matlab routines
  - Script/Command line interface

```matlab
mdl = fitglm(X,y,'Distribution','binomial','Link','logit');
Xnew = grid of points
P = predict(mdl, Xnew);
ythres = P > 0.45 & P < 0.55;
Xnew(ythres) → boundary
```
Logistic Regression Using Matlab Classification Learner App

- Classification Learner App
  - \( T = \text{table}(X,y); \)
  - Logistic Regression & more!
  - Extract to the model to the workspace

\[
\text{mdl} = \text{trainedModel}.\text{GeneralizedLinearModel};
\]

\[
X_{\text{new}} = \text{grid of points}
\]

\[
P = \text{predict(mdl, } X_{\text{new}})\]

\[
ythres = P > 0.45 \text{ & } P < 0.55;
\]

\[
X_{\text{new}}(ythres) \rightarrow \text{boundary}
\]
Machine Learning for OBD

Further Approaches

- Multivariate Gaussian Distribution
- Principal Component Analysis
Machine Learning for OBD
Multivariate Gaussian Distribution

- Model data using Multivariate Gaussian Distribution
- Shown: $2\sigma$ of passing/failing
- Better applied to anomaly detection
  - Only passing data
  - Find outliers
Principal Component Analysis

- Principal Component Analysis Method
  - Reduce the dimensions from 3D to 2D
  - Boundary could be a curve instead of a surface
- Can be used for higher dimensional data
  - Handling of a monitor with 4+ inputs
    - Reduce to 3D or 2D
Machine Learning for OBD

Summary

- On-Board Diagnostics boundaries can be calculated using statistical techniques
  - Widely used - classic method using a grid of points
- Classic OBD boundary setting reduces the data to a grid of points as a basis for the boundary
- Machine learning introduces the concept of using continuous functions as a basis for the boundary
  - Regression: Linear Regression use confidence interval for $4\sigma/3\sigma$ separation
  - Classification: Logistic Regression provides a direct method for determining the boundary
- Striving for: Better quality boundaries, obtained more quickly, using less data
- OBD boundary determination is emerging as an excellent application for machine learning!
Machine Learning for OBD

PS

Youtube “big bang theory check engine light”

https://www.youtube.com/watch?v=KMhp2ShPVQw