Anomaly Detection in CAN Bus Driving Data Using Gaussian Mixture Models

Victoria Hong
Advisors: Prof. Dario Pompili and PhD student Vidyasagar Sadhu
Department of Electrical & Computer Engineering, Rutgers University, New Brunswick, N.J.

Introduction

As artificial intelligence is becoming significantly popular, several algorithms have been developed to detect anomalies in various data types. However, their application in autonomous-driving (AD) vehicles is still relatively unexplored due to the complexity of the technology. This research is vital to improve the safety and efficiency of AD vehicles as it is becoming a trend in the automobile industry. In this project, we test the effectiveness of the Gaussian Mixture Model (using the Expectation-Maximization algorithm) on a set of time-series sensor data obtained from the Honda Research Center in California, USA.

What is an Anomaly?

To simply put, an anomaly is a pattern that appears infrequently in a data set and does not match the expected behavior, similar to an outlier on a graph [1].

Related Works

Clustering, an anomaly detection technique, is a collection of objects that are grouped based on the basis of similarity and dissimilarity [2].

- K-Means is a popular clustering technique that searches for a predetermined number of clusters within an unlabeled data set and uses the euclidean distance to the centroid to assign data points to a cluster.
- We found K-Means is not ideal for every scenario because it does not use a probabilistic approach for grouping data points and taking account of all cluster shapes, unlike the Gaussian Mixture Model.

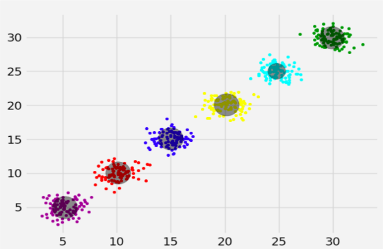


Fig. 1. K-Means Clustering uses a group of data to classify anomalies. Here, there are six clusters and their respective centers.

Proposed Solution

Gaussian Mixture Model (GMM), another method for anomaly detection, assumes there are a number of Gaussian distributions (or clusters) with each distribution being grouped with a probabilistic model and soft clustering approach [3]. The probability formula is given by:

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\Pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- GMMs are used to find clusters (or assume the number of clusters) within a given data set when the location or shape is unknown, making it the most effective anomaly detection method because it takes into consideration all shapes [1].
- The Expectation-Maximization algorithm, a step in the GMM process, computes the log-likelihood of each data point belonging to a cluster with two steps: Expectation (E-Step) and Maximization (M-Step).

E-Step

$$r_{ic} = \frac{\pi_c N(x_i|\mu_c,\Sigma_c)}{\Sigma_{k=1}^{K}\pi_k N(x_i|\mu_k,\Sigma_k)}$$

M-Step

$$\sum_c = \frac{1}{m_c}\sum_i r_{ic}(x_i-\mu_c)^T(x_i-\mu_c)$$

Log-Likelihood

$$\ln p(X|\pi,\mu,\Sigma) = \sum_{I=1}^{N}\ln\left(\sum_{k=1}^{K}\pi_k N(x_i|\mu_k,\Sigma_k)\right)$$

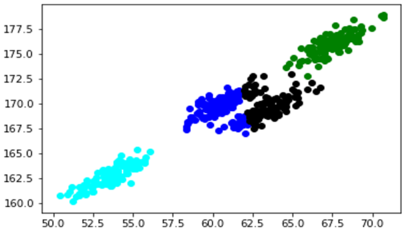


Fig. 2. The Gaussian Mixture Model illustrates that there are four Gaussian distributions, proving the variety in shape and size of a cluster.

Performance Evaluation

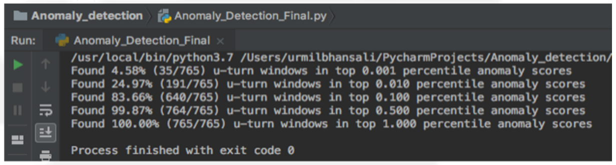
How the car data was handled:

- The data utilized consisted of 266 hours of predominantly daytime raw driving data from February 2017 to March 2018, with a frequency of 100Hz that was downsampled to 5Hz.
- The data consisted of 6 modalities: steer angle, steer speed, speed, yaw, pedal angle, and pedal pressure.
- The time-series data was segmented into windows using a sliding window approach with 5s window size and a stride length of 0.5s.
- 70% of the data was used for training, and the remaining 30% for testing.
- The U-turn was not used in the training process, but instead considered as anomalous for testing purposes.
- Using 11 maneuvers, the GMM classifier classified any window corresponding to a U-turn as an anomaly.

Table 1
Comparison of the Percentage of U-Turns Detected (Qualitative Results) by Analyzing Top Anomaly Scores

Percentile Top Scores	Multi-Class LSTM Autoencoder	Gaussian Mixture Model	Multi-task Learning Model
0.001%	0.39% (3/765)	**4.58% (35/765)**	7.97% (61/765)
0.010%	1.96% (15/765)	**24.97% (191/765)**	29.02% (222/765)
0.100%	13.33% (102/765)	**83.66% (640/765)**	48.63% (646/765)
0.500%	73.64% (562/765)	**99.87% (764/765)**	84.44% (646/765)
1.000%	100.00% (765/765)	**100.00% (765/765)**	100.00% (765/765)

Table 1. compares the results that we found with the results from [5]. The same data set was used but different anomaly detection methods were applied. GMM was found to be better than the Multi-Class LSTM Autoencoder for anomaly detection but it is less accurate than Multi-Learning Model for the first two percentiles.



```
/usr/local/bin/python3.7 /Users/urmilbhansali/PycharmProjects/Anomaly_detection/
Found 4.58% (35/765) u-turn windows in top 0.001 percentile anomaly scores
Found 24.97% (191/765) u-turn windows in top 0.010 percentile anomaly scores
Found 83.66% (640/765) u-turn windows in top 0.100 percentile anomaly scores
Found 99.87% (764/765) u-turn windows in top 0.500 percentile anomaly scores
Found 100.00% (765/765) u-turn windows in top 1.000 percentile anomaly scores

Process finished with exit code 0
```

Fig. 3. The Gaussian Mixture Model Results

Conclusion and Future Work

We found that the Gaussian Mixture Model (GMM) was better than previous methods (Multi-Class LSTM Autoencoder and Multi-Learning Model) that were used for anomaly detection. Despite the Multi-Learning Model being the most accurate in the first two percentiles, GMM proves to be the best at detecting the most anomalies in a given set overall.

In the future, we plan to:

- Incorporate the Generative Adversarial Network (GAN) concept (with backpropagation) into our algorithm for better precision in detecting real from non-real data.
- Continue our journey in discovering better anomaly detection methods to increase the safety and efficiency in AD vehicles for users.

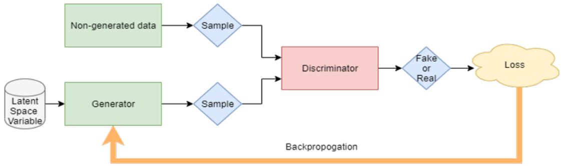


Fig. 4. The Generative Adversarial Network (GAN) structure displays the process of creating and determining real from non-real data. The backpropagation feature makes the process more precise at detection by generating "non-real" data that resembles the given data set.

References

[1] M. G. Gumbao, "Best clustering algorithms for anomaly detection," 2019. [Online]. Available: https://towardsdatascience.com/best-clustering-algorithms-for-anomaly-detection-d5b7412537c8

[2] G. Seif, "The 5 clustering algorithms data scientists need to know," 2019. [Online]. Available: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

[3] Y. P. Raykov, A. Boukouvalas, F. Baig, and M. A. Little, "What to do when k-means clustering fails: A simple yet principled alternative algorithm," 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0162259

[4] N. Janakiev, "Understanding the covariance matrix," 2018. [Online]. Available: https://datascienceplus.com/understanding-the-covariance-matrix

[5] V. Sadhu, T. Misu, and D. Pompili, "Deep multi-task learning for anomalous driving detection using CAN bus scalar sensor data," CoRR, vol. abs/1907.00749, 2019. [Online]. Available: http://arxiv.org/abs/1907.00749

Acknowledgements

I would like to acknowledge and thank my partners Yisel Breton and Shreya Suresh for always being there whenever I needed help throughout this project and most importantly, Dr. Dario Pompili and PhD student Vidyasagar Sadhu for guiding and teaching me skills that will be useful for my career in research.

RUTGERS