Localized Model to Segmentally Estimate Miles Per Gallon (MPG) for Equipment Engines

Jiulin LUO[1], Haojing LUO[2,*], Aimin LI[1] and Haohan WANG[3]

[1]Department of Technical Support Engineering, Academy of Armored Forces Engineering

[2]College of Business and Economics, Australian National University

[3]School of Computer Science, Carnegie Mellon University

*Corresponding author

Keywords: Engine Parameters, Fuel Economy, Regression, Clustering, Machine Learning.

Abstract. In this paper, we built a localized regression model to estimate the miles per gallon (MPG) characteristic for equipment engines based on a serious physical features of this engine. First, we statistically viewed these parameters to build up a basic understanding of the data we collected. Then, with the belief that engines with similar characteristics will perform similarly, we proposed a novel localized model with a novel optimal function based EM algorithm and a novel self-adjusted optimal clustering algorithm to estimate MPG based on the other fully studied engines with similar physical features.

I. Introduction

Engine is always one of the most important modules of an equipment since it gives the power for this equipment to move. As society develops, industry and military build all kinds of engines for many different areas. For different applications, the standards of these engines vary dramatically, so as one of the most important feature of an engine: miles per gallon (MPG). [1]

MPG, as the indicator of the fuel economy [2] of engine industry, it reflects the relationship of between the distance traveled and the amount of fuel it burns. Therefore, it is a very important parameter for the industry of engines. When a manufacturer builds up an engine, often they are responsible for testing the MPG of this engine. However, this test could consume time and energy. More importantly, if the MPG does not reach a level they expect, this built-up engine could be a waste of materials and time. Therefore, we believe that it could be very convenient that we can infer the MPG even before the engines are built.

In this paper, we are proposing a system that can predict the MPG of one engine even when this engine is in the design process and only a few parameters are known. With the state of art technology of machine learning and statistical methods of understanding data, we found some relationship between the physical parameters and the MPG of this engine.

First, we perform some statistical methods to understand distributions of the data and then, with the understanding of these data, we found that if we infer a relationship from those physical parameters and MPG, it could easily result in overfitting, and it could be a very complex model. As a result, we decided to automatically segment the data into different areas and then build models for each area. Also, our localized method is intuitively built on the assumption that there are differences in behaviors for engines in different applications. For example, the engine of a small car must be different from the engine of an equipment, so the relationship between physical parameters and MPG could also be different. In order to solve this problem, we proposed a distance based novel EM clustering algorithm and a self-adjusted number of clustering selection strategy, as well as regression model based on infinite kernel.

This paper is showed as following. In the next section, we will talk about our data and its statistical features. Then we will talk about our methodology. After that, we will show our results compared with a traditional regression model. Then the conclusion of this paper is drawn in the last section, with some interesting aspects that we will consider in the future.

II. Introduction of Data

In this section, we will introduce the statistical analysis of our data [3,4], including the distribution of each feature and the relation of each feature with the MPG label. Our data is sampled from real world equipment engines; there are seven features and one label.

Dataset

There are totally 392 instances in our data set, each instance has seven features and one numerical label. Our task is to estimate the numerical label as accurate as possible based on the seven features. The features are 1) the number of cylinders, 2) electric displacement, 3) horsepower, 4) weight, 5) acceleration, 6) model year. Details are listed in Table. 1

Table 1. Basic statistics of features

Feature	Min	Max	Mean	Median	STD
Cylinder	3	8	5.47	4	1.70
displacement	68.0	455.0	194.41	151.0	104.51
horsepower	46.0	230.0	104.46	93.5	38.44
weight	1613.0	5140.0	2977.58	2803.5	848.31
acceleration	8.0	24.8	15.54	15.5	2.75
Model year	1970	1982	1976	1976	3years, 8months
MPG	9.0	46.6	23.44	22.75	7.79

Besides some statistical characteristics shown in the table, we also plot the histogram of each feature and the predicting label, as in Figure 1. We can get a clearer understanding of these features from Figure 1.

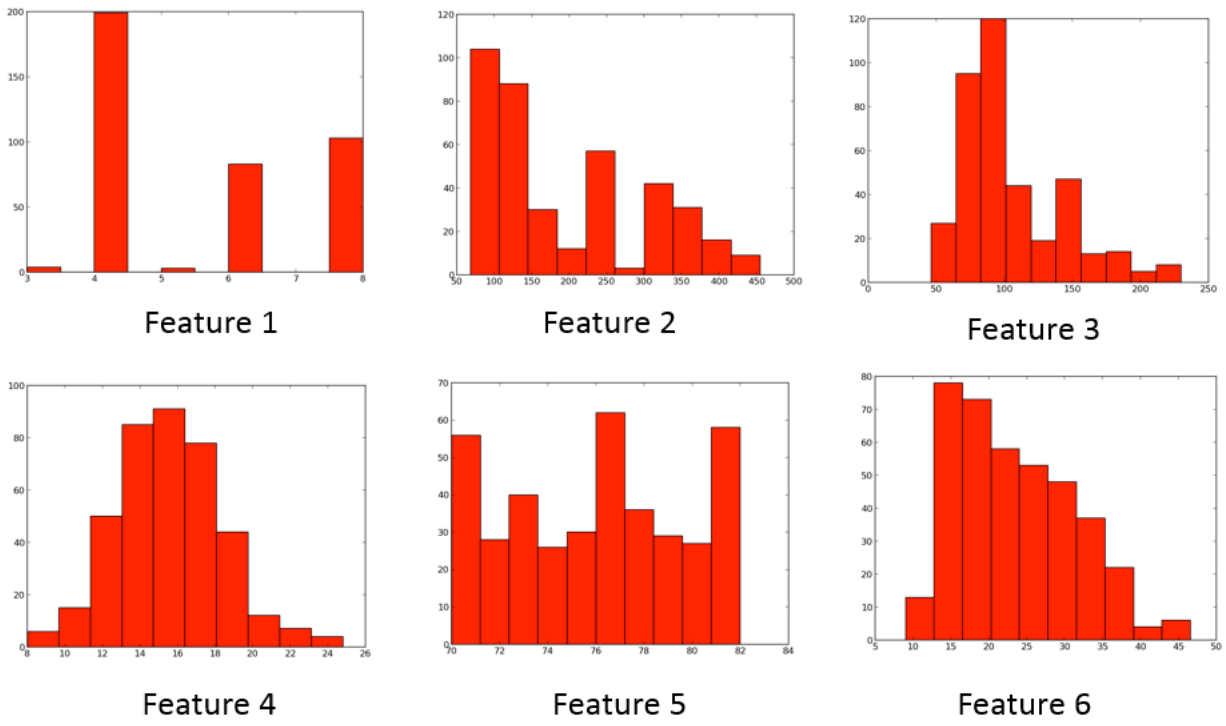

Figure 1. Histograms of features

We also scatter the points to see how these features are correlated with features. In Figure 2, we can see the distribution of each features and its relation with the label.

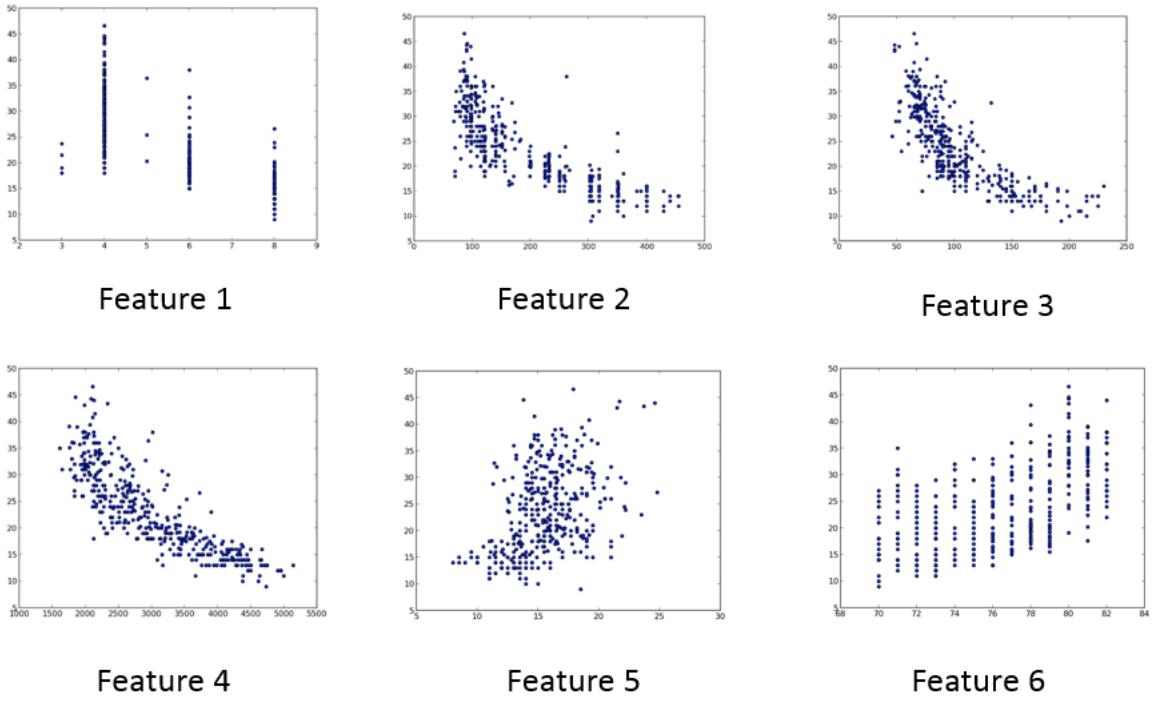

Figure 2. Scattered plot of features and predicting label

As shown in Figure 2, some features are distributed like a distribution of two-dimensional Gaussian, like acceleration while the most of the other features show a negative correlation with the labels. The model year shows a positive correlation with our feature, which is consistent with our common understanding that the later an engine is built, the better its performance.
This distribution of data give us some intuition that the predicted could be done by a simple solution of linear combination of all these features. However, in the later section we will show the advantage of our methods over simple regression model.

III. Methodology

In this section, we will talk our self-adjusted segmented regression model. We will first talk about how to segment our data automatically to build localized models and then, we will talk about our regression methods.

Self-adjusted segmenting method

Our solution is based on an intuitive thinking that the efficiency of an engine is only similar with the efficiency of other engines with similar physical characters. We believe that a general model across all the engines and predict MPG with much information from other engines may not behave well since there is a trade off between the information of every engine and the noises every other engine could generate [5]. The main contribution of our self-adjusted segmenting model is that we considered the balance of information and noise and to maximize the efficiency of our segmenting, so that we can build better regression model with our method in the second part of our method.
When we only have the data set, the most straightforward method to cut the data set into different segments is clustering. K-means clustering [6] or hierarchical clustering [7] play important roles in putting data into different boxes. However, the flaws of these clustering methods are also very apparent. K-means clustering requires K as prior knowledge of clustering which we do not know. Enumerating all possible Ks for higher accuracy may easily result in overfit. Hierarchical clustering does not require prior K because it cluster every instance at the very end, however, it requires too much computation power and a similar problem will be raised when we need to consider the

methodology to judge if two subsets can be merged to a super set. Therefore, in our method, we didn't consider these two clustering. Another clustering algorithm is EM clustering [8,9]. EM clustering believe there is a hidden model to govern the distribution (usually Gaussian distribution) of the data sets.

However, the disadvantages of this algorithm are also obvious. The first one is that the features do not show a Gaussian distribution as we showed in Figure 1. Moreover, we cannot draw a solid conclusion of the distribution of our data based on the observation of histograms. Secondly, the problem of K-means still exist, we do not have the prior knowledge of the number of clusters.

We first address the first problem. Our solution cannot directly solve this problem because as we showed that determining the distribution cannot be done. However, we bypass this problem by assuming a distribution without maximizing it. More specifically, our target function is not a likelihood function of the data and model given parameters, but a distance function of inter-cluster distance over intra-cluster distance. With the belief that a most promising clustering result will give a result that maximizes the inter-cluster distance, the modified EM algorithm we proposed iteratively searching for the parameters maximize this distance. In order to avoid the extreme situation, we also add the constraint of intra-cluster distance. Our algorithm is showed in detail as following:

E-step:

First, what we want to measure is the expectation of our data given parameters, here, we set total Kn parameters, where n is the number of data. Each parameter θ_{i,x_j}^t gives a prior probability that data point x_j belongs to cluster i, in the t round. The equation is showed as following:

$$P\left(y_j = i \middle| x_j, \theta^{t-1}\right) = P(y_i = i | x_j, \theta_{1,x_j}^{t-1}, \dots, \theta_{K,x_j}^{t-1}) \tag{1}$$

Then, as we want to measure the inter-distance, we believe that this data point with label i will maximize the function with inter-cluster distance over intra-distance, as following:

$$P\left(y_j = i \middle| x_j, \theta^{t-1}\right) = \frac{\sum_{m=1}^{N}\sum_{n=1}^{N}||x_{m,i}\theta_{i,x_m}^{t-1} - x_{n,i}\theta_{i,x_n}^{t-1}||^2 / \sum_{m=1}^{N} I(x_m \in i)}{\sum_{i=1}^{K}\sum_{m=1}^{N}\sum_{n=1}^{N}||x_{m,i}\theta_{i,x_m}^{t-1} - x_{n,i}\theta_{i,x_n}^{t-1}||^2} \tag{2}$$

in which, $x_{m,i}$ stands for the data point m that is in the cluster i. $I()$ is an identity function.

M-step:

Now, with the expectation, we can move to maximization step with the following equation,

$$\theta_{i,x_j}^t = \frac{P(y_l = i | x_j, \theta^{t-1})}{\sum_{l=1}^{n} P(y_l = i | x_j, \theta^{t-1})} \tag{3}$$

Different from traditional EM, we have to update the parameter Kn times.

Now, we focus on the second problem, how to select the optimal K based on training data. There are several existing methods regarding the change of inter-cluster distances, also known as residue error of clustering. A common solution is to search for the "elbow" point of the curve of the function of inter-cluster distance and K. However, this method does not work very well for our clustering method since that we intentional maximize this distance.

In our method, we consider three different metrics and aggregate the influence of them. In order to lower the bias of our clustering result, we run our clustering algorithm for each K 100 times and calculate the mode of inter-variance and intra-variance. Thus, we exclusively consider the result that appear the most times in the 100 run. We will evaluate the score of a K with the following equation:

$$s_k = \frac{(Ia_k - Ia_{k-1})^2}{(Ie_k - Ie_{k-1})^2} k \tag{4}$$

where Ia_k is the intra-variance when K=k and Ie_k is the intre-variance when K=k.
We consider the ratio of inter-cluster distance and intra-cluster distance with ratio K so that we can compare directly with different choices of K with this score. However, if for certain K, each of the clustering result only appears once, which means there are totally 100 different clustering results, so no Mode can be voted, this clustering is not considered as a valid one and will be assigned a negative score.

Regression Model
After the clustering, we can start to build regression models [10] for each clustered data. In order to deal with the fact of lacking features, kernel tricks are implemented. We use infinite kernel [11] for the regression model.

Prediction
When a new instance comes, our algorithm first assigns it to a certain category and then predict the label with regression model.
KNN classification is used to assign an instance to a category with Euclidean distance function. The equation for predict a new instance x is showed here:

$$y = \alpha \sum_{i=1}^{n} K_{inf}(x, x_{c,i}) + \beta \quad (5)$$

where $x_{k,i}$ stands for the i[th] data point and it is clustered in the c[th] cluster. c is selected with the KNN clustering.
One thing to clarify is that the K here is totally unrelated with the K we got from clustering.

Eager Learning V.S. Lazy Learning
As a further explanation of our work, it is difficult to categorize our learning algorithm as either eager learning or lazy learning [12]. Eager learning is a group of machine learning algorithms that generalizes training sets during the training process while lazy learning is a group of machine learning algorithms that generalizes training sets only after the a prediction is queried. In our algorithm, we generalize the parameters when we train the regression model but all the training instances are still held since the first step of prediction is based on lazy learning.

IV. Evaluation and Experiment

In this section, we will first talk about the evaluation methods we applied and then we showed our experiment results.

Evaluation
We evaluated with Root Mean Squared Logarithmic Error ("RMSLE") to measure the accuracy of our regression model, the equation is showed as following:

$$\varepsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2} \quad (6)$$

Where:

- ϵ is the RMSLE value (score)
- n is the total number of reviews in the data set
- p_i is the predicted number of useful votes for data instance i
- a_i is the actual number of useful votes for review i

We use the logarithm of the number of values so that error scales properly with the magnitude of values. An absolute error of one value is much more significant to a MPG of 20 than it is to a MPG of 100.

Experiment

We tested our data with our method, in order to show the comparison, we showed our experiment comparing with traditional regression model. In addition, we showed our result with a number of different clustering methods, including traditional K-means clustering, EM clustering and our proposed self-adjusted EM clustering. The results are showed as following:

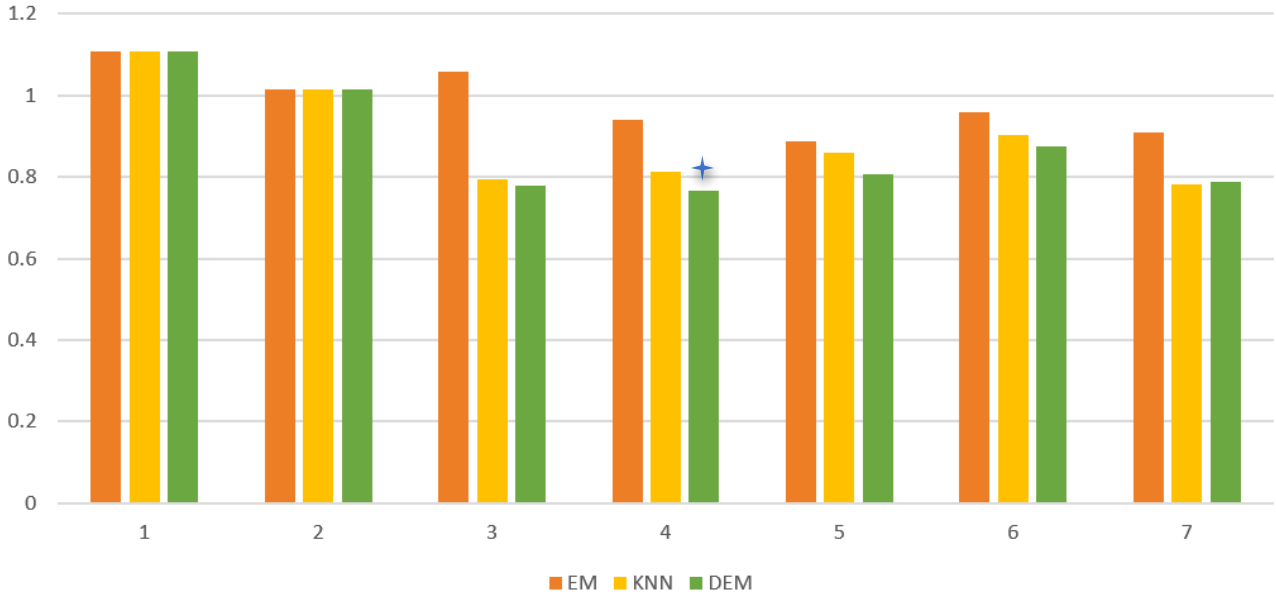

Figure 3 Experiment result of regression model based on different clustering algorithms, y is RMSLE and x is K

Figure 3 shows the result, as we can see that, by comparing with K=1, which is the leftmost line, we can clearly see that the segmented regression model works very well. Then, by comparing the three columns for each K, we can see that our algorithm generally works the best. The star on K=4 indicates that this is the number of clusters that our algorithm automatically searched out. We can see a quadratic shape of curve for these result while the lowest point is K=4. However, for K=7 and even bigger K, we can see the error drops down again. However, there is only one clustering result for each 100 run of clustering, we believe the result cannot be generalized very well.

V. Conclusion & Future Work

In this paper, we are aimed to predict the MPG of an engine based on a two-part algorithm. We proposed this algorithm as a solution to predict how an engine will perform before we actually build it. We believe that our methods will help to reduce the spending on design and build engines.

Our algorithm first trains a model with some information about engines with two part of an algorithm. The first part is to split the entire data sets into some separate data sets because we believe that data with too much distance always from a certain point may introduce too much noise for our model. The second part is to build regression model on our data set. Then we tested our algorithm with some real world engine data and we get a promising result for our algorithm.

In future, we will focus on improving the second part of our algorithm; we will focus on build regression model with lower MSE with fewer data instances, so that we can segment our data set into even smaller chunks and improve the general accuracy.

Acknowledgement

This work is supported by Ministry of Science and Technology of China under National 973 Basic Research Program Grant No. 2011CD302600, Grant No. 2011CB302805, Grant No. 2011CB302601, and Grant No. 2012CB315800. This work is also supported by China NSFC A3 Program (No.61161140320)

References

[1] Schweitzer, Paul H., and Carl Volz. Electronic Optimizer Control for IC engine: most MPG for any MPH. No. SAE# 750370. 1975.

[2] Greene, David L. "Vehicle Use and Fuel Economy: How Big is the" Rebound" Effect?." The Energy Journal 13, no. 1 (1992): 117-144.

[3] Quinlan, J. Ross. "Combining Instance-Based and Model-Based Learning." In ICML, pp. 236-243. 1993.

[4] Kolyukhin, Dmitriy, and Anita Torabi. "Statistical analysis of the relationships between faults attributes." Journal of Geophysical Research: Solid Earth (1978–2012) 117, no. B5 (2012).

[5] Tröltzsch, Fredi. Optimal control of partial differential equations: theory, methods, and applications. Vol. 112. American Mathematical Soc., 2010.

[6] Gnanadesikan, Ram. Methods for statistical data analysis of multivariate observations. Vol. 321. John Wiley & Sons, 2011.

[7] Jain, Anil K. "Data clustering: 50 years beyond K-means." Pattern Recognition Letters 31, no. 8 (2010): 651-666.

[8] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the Royal statistical Society 39, no. 1 (1977): 1-38.

[9] Ordonez, Carlos, and Edward Omiecinski. "FREM: fast and robust EM clustering for large data sets." In Proceedings of the eleventh international conference on Information and knowledge management, pp. 590-599. ACM, 2002.

[10] Sen, Ashish, and Muni S. Srivastava. Regression analysis: theory, methods, and applications. Springer, 1990.

[11] Gehler, Peter, and Sebastian Nowozin. "Infinite kernel learning." (2008).

[12] Hendrickx, Iris, and Antal Van Den Bosch. "Hybrid algorithms with instance-based classification." In Machine Learning: ECML 2005, pp. 158-169. Springer Berlin Heidelberg, 2005.