

Customer Usage Space Classification and Representative Duty Cycle Development Using K-Means Clustering

Gaurav Gosain, Billy Holland, and Thomas McKinley
 Cummins Inc

ABSTRACT

Understanding customer usage space and its impact on engine, after treatment, and vehicle duty cycles poses challenges in terms of data noise, data variability and complex interrelations. Moreover, humans are only able to concurrently visualize at most 2 to 3 dimensions, limiting the number of engine parameters that can be considered. Previous studies in this field have been limited to understanding trends in data based on single duty cycle, comparatively short application period and time domain segmented clustering analysis. These techniques have been used to determine representative cycles for specific applications. In this paper, K-Means Clustering is used to classify customer usage space based on tens of dimensions, for multiple duty cycles, and over years of operation. The clusters are evaluated based on system, sub-system, and component-based metrics on a day based unsegmented engine parameter values. Some particular applications of this methodology are discussed in the paper including - Generalized Clustering and Post Processing Visualization Techniques, Critical Customer Identification and Customer Representative Nominal Short-Cycle Generation - a moving window approach. Case studies with real customer data for various On-Highway commercial diesel applications are discussed for different sub systems to demonstrate the applications and power of the tool.

CITATION: Gosain, G., Holland, B., and McKinley, T., "Customer Usage Space Classification and Representative Duty Cycle Development Using K-Means Clustering," *SAE Int. J. Commer. Veh.* 10(2):2017, doi:10.4271/2017-01-0204.

INTRODUCTION

Diesel engines are used in a wide range of applications, and within the broad categories of on- and off-highway there exist a large number of uses, each having its own unique attributes. Identifying and understanding these key attributes has always been a tough task due to non-linear and complex interrelations between subsystems of an engine; thus making it tough to draw out patterns from the data and basing similarities or differences between different branches or duty cycles. Also faster product development schedules demand smart testing and planning; ensuring coverage of variation in customer usage space with minimum amount of resources. The other hindrance comes from data quality and noise. Engine data is vulnerable to data noise coming from bad or faulty sensors, variations in simulated vs actual parameters, environmental factors, irregularities in drive patterns on some days and variations in data due to differences resulting from driver operation. Machine Learning algorithms can be applied to identify complex patterns inside the data. One such simple and fast algorithm is K-Means Clustering which focuses on classifying numerical data in higher dimensions into similar groups or clusters.

Clustering techniques have been used previously also to develop customer specific sample cycles. Most of the techniques have been limited to very few drive parameters such as vehicle speed, acceleration etc. These techniques also share a common approach of Kinematic sequencing of a drive cycle - which basically comprises of

time domain analysis of engine parameters when the vehicle moves between two stops or idle cases. In [1] the authors discuss an approach to use clustering to develop a sample cycle for HEVs. Vehicle speed of different applications was used to classify duty cycles into Urban, Road and Highway. Kinematic sequences of different duty cycles were analyzed for 15 different parameters. The statistics for the 15 parameters were used to cluster the kinematic sequences to develop a duty cycle. In [2] the same technique as [1] was used to derive a refuse hauler sample duty cycle. In [3], clustering of kinematic sequences was done to understand real world driving conditions of a transit bus in 2 different locations. In [4], gross vehicle weight statistics were added to kinematic parameters to analyze and classify kinematic sequences to develop a sample cycle. In [5] clustering of velocity profiles of kinematic sequences is used followed by a Markov Statistical Chain Model to develop test cycles for PHEVs.

A shortcoming in using vehicle speed (or other kinematic parameters) as primary criterion for classifying usage space is that it is highly dependent on geographical conditions, driver input and traffic conditions. Same duty cycle may classify as 2 different clusters just because of different traffic conditions and locations. The kinematic sequencing of data is also highly varying in real world conditions as every idling event cannot be identified as equal - it could be a stop signal, a red signal, a bus stop or a house stop for a refuse hauler - in between which the engine system level parameters might be different for the same duty cycle or same for different duty cycles. Hence, by

kinematic sequencing approach we are limited to only characterizing a single duty cycle and not multiple different duty cycles - unable to understand their engine system level parameter variation.

The approach discussed in this paper focuses more on system level parameters and identification of similarities and differences between multiple duty cycles based on the distribution of these parameters. K - Means clustering is utilized to cluster different operational duty cycles into groups based on a non-kinematic sequence basis. The use of this technique is demonstrated through 3 different case studies - (a) Fuel System Pressure Fatigue risk analysis, (b) General Customer Usage space clustering and (c) Identifying Representative Extreme Customer Cycle. The paper also discusses a new method of generation of real world customer representative nominal cycle. The following sections discuss the conceptual details and methodology architecture.

CLUSTERING IMPLEMENTATION

Clustering is an approach in data sciences to classify numerical data into clusters or groups such that - Inter-Cluster variation is maximized and Intra-Cluster variation is minimized. The center of a cluster is termed as cluster centroid. Various clustering techniques have been developed [6] of which K-Means is used to cluster data in this study. Reference [7] discusses the mathematical details of K - Means clustering algorithm. The process flow chart of the analysis is depicted in the pictorial representation of Figure 1.

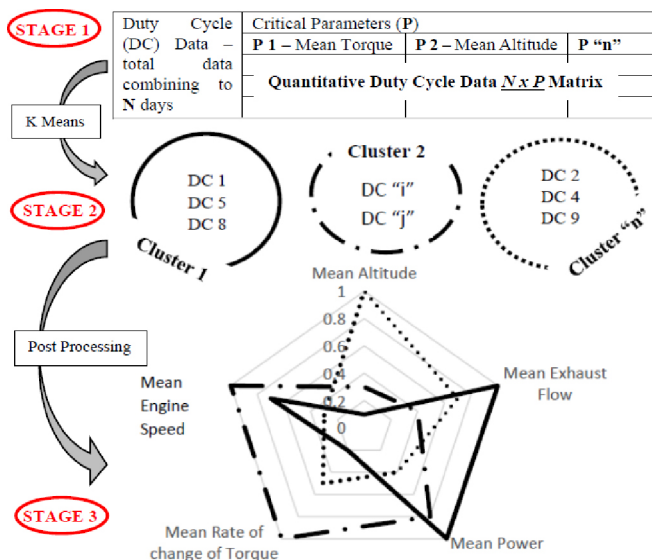


Figure 1. Customer Usage Space Classification Process Flow

The analysis can be completed in 3 stages all of which are independent of each other.

STAGE 1: Preprocessing Data and Generation of $N \times P$ Matrix

The 1st stage of the process comprises of synthesizing the customer data into a form that K-Means clustering can utilize. Usually, engine data whether it is recorded by the Engine Control Module or external

additional sensors is stored in form of a data-file. This data-file has all the recorded channels stored as vectors - the data-file for this study was for a whole day in a '.mat' file ('.mat' file is a data form for MATLAB). For this study, the channels are recorded at 1Hz - but the current process will work with different frequency of data as long as it is same for a channel for all the duty cycles.

The input to the clustering algorithm is a matrix of the form of $N \times P$ where 'N' is total number of data-files with all the duty cycles combined (refer Figure 1 Stage 1). This signifies that the rows of the $N \times P$ matrix represent datasets of all the duty cycles combined - the order of data-files is not important as the clustering algorithm considers each data-file individually, independent of the other; however for ease of processing all the data-files of a duty cycle should be placed at consecutive rows.

The columns of the $N \times P$ matrix comprise of the 'P' parameters that needed to be calculated for the each of the data-files. A data-file usually contains several data points and the parameters are calculated considering all the data points e.g. mean engine speed parameter would be estimated by calculating mean of all the data points within a data-file. Deciding the critical parameters for an analysis is the most important process and defines the quality of input to the clustering algorithm. From an overall/general point of view (considering the engine on whole), systems-interaction and integration teams can be asked for a feedback on general parameters. If the analysis concerns a particular sub-system/component then the parameters that are critical to that sub-system/component should be considered. The different case studies in the sections which follow provide insights on the choice of parameters. Linearly dependent parameters e.g. Average Vehicle Speed and Average Miles traveled should be avoided together- doing this implicitly weights these parameters more than others and hence, skews the analysis. The advantage with K - Means clustering is it can work with a combination of non - homogeneous units. For example, if 2 parameters are mean engine speed and mean net engine torque, net engine speed units could be Revolutions per Minute and that for net engine torque could be in Newton-meter. However, since K-Means clustering works on the principle of Euclidean distances between vectors in higher dimensions, if each dimension is not scaled equally then units of the physical quantities will result in unequal weighting of each parameter, which will skew the results of the analysis. To avoid this, we have chosen to normalize all the columns of $N \times P$ matrix. For this study, each parameter column was linearly normalized based off maximum and minimum values to a scale of 0-1. To filter out noise such as erroneous sensor readings or data logger 'drop-out' from the data - a normal distribution curve was fit through each parameter distribution and an upper and lower bound probability was tuned after examination to ignore data that was errored or unrealistic; while retaining extreme levels of realistic loading. The $N \times P$ matrix is populated by data from each data-file and for each parameter. Through this process the attributes of a duty cycle within every data-file is reduced a vector of P values that relate to how the duty cycle system level parameters varied during the duration of that data-file. This avoids the step of segmenting the data-file into kinematic sequences altogether.

STAGE 2: Clustering the $N \times P$ Duty Cycle Matrix

The $N \times P$ matrix serves as 1 of the inputs to the K-means clustering algorithm. The other major input is K which is the number of clusters. Deciding number of clusters has been a topic of detailed discussion in literature. In [8] the paper summarizes many well established and adopted criteria to identify the optimum number of clusters. For this study, 3 cluster evaluation criteria are used: Calinski Harabasz (should consider maximum value of this measure); Davies Bouldin (minimum value); and Silhouette (maximum value). The aim with finding optimum number of clusters is to find minimum clusters that convey adequate variations or attributes of the data. However, a larger number of clusters leads to the identification of more unique duty cycles. For the sake of product robustness, it may be advantageous to select a larger than optimal number of clusters based on the criteria above. In turn, the maximum permissible number of clusters will be limited by the available finite time and resources for validating the product.

The output of the clustering algorithm is an index vector with values in between 1 and K for every row of the $N \times P$ matrix. This allocates every data-file to a specific cluster. The classification of the duty cycle to a specific cluster then depends on the number of data-files each cluster share. As the case studies demonstrate, usually every duty cycle has clear allocation to a cluster but in some cases, the duty cycle will have data-files split over multiple clusters. This happens when the duty cycle operated differently on some days or shifts than its regular operation. For this study if more than 70% (this percent is the choice of the analyst and can be changed) of the data-files of a particular duty cycle belonged to a specific cluster, then that duty cycle was considered to be a part of the cluster. In addition the algorithm also specifies a $K \times P$ matrix which houses the location of the cluster centroids. The K rows here represent the K cluster centroids and their parameter values.

STAGE 3: Post Processing the Data and Visualization Techniques

Visualizing data beyond 3 dimensions is tough because of limitation of human visual perception. In addition, plotting every single of the N vectors introduces clutter and a lot of overlap. The case studies discuss some of the visualization techniques used with the tool, which primarily are - (a) Box Plot of normalized parameter variation, it helps in relatively understanding which parameters have more variation than others over the customer population; (b) Cluster Centroid Radar plot - instead of plotting individual data-file vectors, the cluster centroid vectors can be plotted on a normalized radar plot, it helps in readily understanding how the cluster centers are separated in higher dimensions; (c) Cluster Probability Distribution curves - these curves are made for each parameter by fitting normal distributions over the parameter values of individual clusters, this helps in understanding how much each cluster has an overlap over the other for a particular parameter.

Customer Representative Short Cycle Generation

After the duty cycles have been allocated to clusters, short duration customer representative cycles can be developed for each cluster. Representative cycles can be synthesized by combining different

pre-defined sequences of data derived from customer data at different time based kinematic sequences - this approach has been used extensively in the literature [2]. Instead, we use the moving window approach described below.

The objective is to find a short time window which is a good representation of 'average' operation for the cluster of interest. The 'average' operation is characterized by the centroid, which is a P dimensional vector. A three-dimensional illustration of this approach is shown in Figure 2. A simple measure of each data-file's difference from the cluster centroid is the difference $\vec{d}$ between its P dimensional vector $\vec{T}$ and the centroid $\vec{C}$. The ten data-files of the cluster having the smallest difference vectors by length are then found. This can be thought of as prioritizing the search to a representative time window within the ten most representative data-files. By doing so, we reduce the amount of data considered and therefore computational effort. However, the time duration of each data-files is still several hours to an entire 24 hour period. Therefore, we take an additional step to compress the time duration even more.

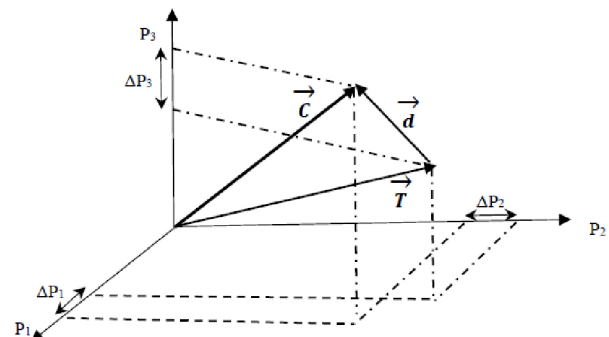


Figure 2. 3 Dimensional Illustration of Short Cycle Data File Search

Specifically, each of the selected data-files is segmented into consecutive time windows of a chosen length Δt (e.g. 20 to 30 minutes). This is illustrated for the engine speed histories shown in Figure 3 with 2 windows w_i and w_{i+1} . Each time window has its own P dimensional vector, and again the length of the difference between that vector and the cluster centroid is taken as a measure of how well it represents the cluster. The time window taken as the best representation of the cluster is the one with the shortest difference vector and also all components of its P dimensional vector within one standard deviation of the components of the cluster centroid. The standard deviation is computed for each of the P parameters independently. In this way, we ensure the time window is a good representation of all P parameters in aggregate and all P parameters individually. The power of this method is its ability to compress duty cycle time histories by several orders of magnitude using only real-world time histories. Furthermore, although the process above finds time histories representative of the centroid, it is easy to find extreme (e.g. "90th percentile") time windows by looking at large differences from the centroid or by changing the target vector away from the centroid. Case study C demonstrates this usage.