HMMs for Prediction of High-Cost Failures

Julio Martinez
juliomz

Christianne Dennison
cdenn

Zhengyi Lian
zylian

Abstract

PACCAR, a commercial vehicles company, has made available a large dataset of sensor snapshots that were taken using communication networks during the operation of several vehicles, as well as repairs that were performed for specific failures on these vehicles. Experimenting with Hidden Markov Models (HMMs), we analyze this dataset to determine the probability that a sequence of snapshots will lead to potential high-cost failures within a specific time range and with what cost.

1. Introduction

PACCAR, a commercial vehicles company, has made available a large dataset of field snaps (snapshots) that were taken using communication networks during the operation of several vehicles. In addition, repairs experienced by these vehicles were made by various dealerships. Data on these repairs is also available. The field snaps are taken in an asynchronous manner meaning the readings are made periodically whenever the vehicles are in operation. These include temporal data readings such as barometric pressure, acceleration, and other observations on the condition and environment of the vehicle. Repair data include but are not limited to mileage of the vehicle when submitted for repair, identified fault codes, and general repair cost labels. PACCAR has made this dataset available with the intention to discover a number items that could prove beneficial in finding precursors to expensive repairs.

For instance, they are interested in predicting when a fault that requires a costly repair is likely to occur, or what sensor readings are indicative of impending costly faults and repairs which could then potentially be mitigated by preemptive maintenance. We aim to apply HMMs[4] to determine the probability that a sequence of snapshots will lead to a specific failure within a specific time range and with what cost.[1] This can help to inform what type of preemptive maintenance is necessary and when it should be carried out, as well as guide PACCAR on their future sensor investments for field data collection.

2. Background

Predictive maintenance is a vast and expanding field, but primarily studied by those in industry. [8] has a very similar setup to ours, using vehicle sensor readings to predict a repair type. They used windows of 1, 15, 30, and 60 days before the ensuing repair, and found that the DTC (diagnostic trouble code) field was the best earliest predictor of the vehicle needing maintenance. They also used linear regression and random forests for classification. Above a certain threshold of likelihood, they would predict a failure. To simplify our model, we are choosing only the maximal prediction, and unlike their model, we are trying to predict time to failure along with type of failure, not only type of failure within a fixed window.

As pointed out in [6], there are three main approaches to predictive vehicle maintenance: Remaining Useful Life (RUL) prediction, Deviation Detection, and supervised classification. Deviation detection can easily be solved but cannot easily help determine the fault that will occur; it is used primarily for online detection of potential failures. RUL is typically done component-wise. A combination of supervised classification and RUL is best for our application given that we have repair data and we wish to predict the RUL for each system of the vehicles as well as classify by the failure codes.

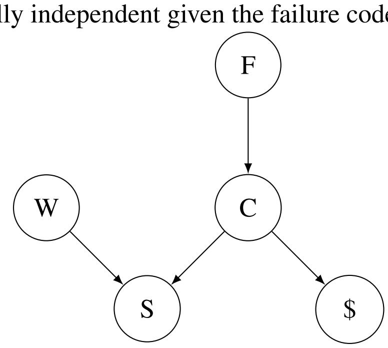
[7] uses KNN, C5.0, and Random Forest learning algorithms to predict a failure within some fixed set of window sizes as [8] did, which differs from our approach to predict the time to failure. Their approach was also to report cost savings of executing on the early maintenance within simulation instead of accuracy, which also differs from our approach to aim for the highest accuracy of prediction. [3] is yet another example of using random forests to predict the time to failure on lead-acid batteries.

Hidden Markov Models (HMMs) have been used in many other applications for prediction in time series, but also predictive maintenance. [5] proposes a new HMM-based structure called a Multi-Branch HMM (MB-HMM), specifically to account for the case where multiple modes of decay can occur. For our work, we are assuming that each vehicle is decaying at the same rate for simplicity. [2] found that hidden semi-Markov models (HSMMs) performed better at predictive maintenance in airplanes than normal HMMs, since violating the Markov property can overcome modeling limitations. However, given the support of the HMM library we chose, we restricted our experimentation to only HMMs. Finally, [9] suggests that careful feature selection is important for using HMMs. In their paper, features were selected based on what they call “feature saliencies”, which represent the degree to which a feature can distinguish between states. However, their primary motivation for feature selection is to reduce “test cost”, or the acquisition of data, for which we have no concern, since all the data is provided to us up front.

3. Targets

3.1. Assumptions

We know that the probability of the sequence depends on the window (W) that it occurs within, and the probability of seeing a particular failure code (C) is dependent on the occurrence of a failure (F) at all. To simplify the model, we are assuming that the cost ($) and the sequence (S) are conditionally independent given the failure code (C):



We assert that this is a reasonable assumption since a cost of a repair has several factors that are unknown, such as the cost of towing or the efficiency and rate of the mechanics working on the repair.

We are also assuming that every vehicle that has a failure is always repaired, since we have no way to know otherwise:

	f^0	f^1
$\exists C$	0	1
$\nexists C$	1	0

In this table, $\nexists C$ the case where there were no failures for the vehicle, and $\exists C$ is where at least one failure occurred with some code C. We will treat $\nexists C$ as simply another failure code (e.g. $C = 0$) for calculation.

3.2. Priors

For our calculations, we needed several priors. For the total probability of failure, we took the proportion of snapshots that did not lead up to a failure and divided by the total number of snapshots: $P(f^1) = 0.45$

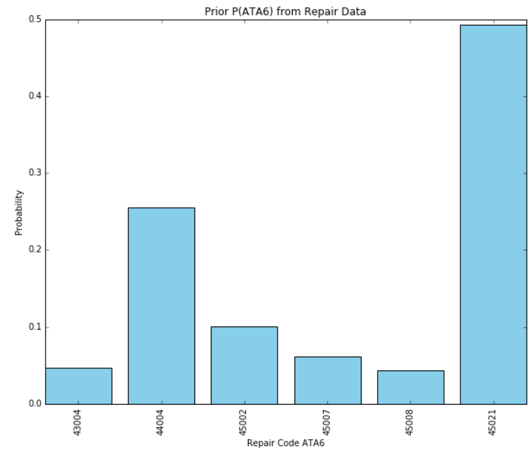


Figure 1. Prior probabilities of six ATA6 failure codes that are of interest from the repair dataset (normalized).

For the prior conditional probability on failure codes $P(C \mid f^1)$, we recorded the proportion of that code occurring among all of the failures. Thereafter, the priors for six failure codes of interest were selected and then normalized (see Figure 1). Section 4.2 contains details on how the six failure codes were selected and section 8 discusses about potential selection bias and how they can be addressed n future work. With this defined and figure 3.1, we computed the full prior distribution as $P(C \mid F)P(F)$, with corresponding values shown in Figure 2.

F	Failure Code C	P(F)	Normalized P(C | F)	Prior = P(C|F)P(F)
0	0	0.55	1	0.55
1	43004	0.45	0.05	0.02
	44004		0.26	0.11
	45002		0.10	0.05
	45007		0.06	0.03
	45008		0.04	0.02
	45021		0.49	0.22

Figure 2. Prior probabilities corresponding to the case of no failure (f^0, c^0) and six failure codes of interest.

The prior distribution on the window $P(W)$ where $W \in \{1, \ldots, k\}$) is difficult to determine and assumed to be uniform as an initial guess.

3.3. Timesteps to Failure

In order to determine the time for the particular failure, we need a model for each time window before the failure occurrence, up to some limit number of windows. Therefore each failure type model will be composed of k models, where k is the hyperparameter of the number of windows. Each HMM gives us $P(s \mid w, c)$. We want $P(w \mid s)$:

$$
\begin{aligned}
P(w \mid s) &= \frac{P(w,s)}{P(s)} \\
P(w \mid s) &= \sum_{f=0}^{1} \sum_{c \in C} \frac{P(s,w,f,c)}{P(s)} \\
&= \sum_{f=0}^{1} \sum_{c \in C} \frac{P(s \mid w,c)P(c \mid f)P(f)P(w)}{P(s)} \\
&\propto \sum_{f=0}^{1} \sum_{c \in C} P(s \mid w,c)P(c \mid f)P(f)
\end{aligned}
$$

We can calculate $P(s \mid w) = \sum_{c \in C} P(s \mid w, c)$ by summing the rows of our HMM matrix, $HMM_{w,c}(s)$. The most likely window can be determined by taking the maximal probability over that summation:

$$
w^* = \arg \max_{w \in 1 \ldots k} \sum_{f=0}^{1} \sum_{c \in C} P(s \mid w,c)P(c \mid f)P(f)
$$

3.4. Most Likely Failure

In order to determine the probability that a sequence will lead to a specific failure, we need to determine the probability distribution of failure codes given the sequence, $P(C \mid s)$. Each HMM model, fitted to a specific failure code c and time window w prior to failure, gives us $P(s \mid w, c)$. We want $P(c \mid s)$:

$$
\begin{aligned}
P(c \mid s) &= \frac{P(s,c)}{P(s)} \\
&= \sum_{f=0}^{1} \sum_{w=0}^{k} \frac{P(s \mid c,w)P(w)P(c \mid f)P(f)}{P(s)} \\
&\propto \sum_{f=0}^{1} \sum_{w=0}^{k} P(s \mid c,w)P(c \mid f)P(f)
\end{aligned}
$$

Assuming a uniform prior $P(w)$, we can calculate $P(s \mid c) = \sum_{w=0}^{k} P(s \mid c, w)$ by summing the columns of our HMM matrix, $HMM_{w,c}(s)$. The most likely failure code can be determined by taking the maximal probability over that summation:

$$
c^* = \arg \max_{c \in C} \sum_{f=0}^{1} P(s \mid c)P(c \mid f)P(f)
$$

3.5. Cost of Failure

Finally, we must determine the potential repair cost given a sequence. The repair data allows us to compute the prior conditional probability distribution of repair cost given failure code $P(\$ \mid C)$ as the proportion of failures under failure code C which led to a repair cost \$. For consistency, we defined repair cost $\$ = 0$ when there is no failure (i.e. failure code $c = 0$) and $P(\$ = 0 \mid c^0) = 1$. The prior conditional cost probability distribution for the full set of interested cost and failure code pairs are shown in Figure 3.

P(\$ | C)		Failure Code C						
		0	43004	44004	45002	45007	45008	45021
Repair Cost \$	0	1	0	0	0	0	0	0
	very low	0	0.46	0.08	0.46	0.47	0.42	0.89
	low	0	0.38	0.32	0.14	0.06	0.42	0.02
	medium	0	0	0.58	0.07	0.18	0	0.09
	high	0	0.15	0.01	0.07	0.18	0.17	0
	very high	0	0	0	0.25	0.12	0	0

Figure 3. Prior conditional probability distribution of repair cost \$ given failure code C.

With the calculated $P(\$ \mid C)$ from the data, we can calculate $P(\$ \mid s)$:

$$
\begin{aligned}
P(\$ \mid s) &= \frac{P(\$,s)}{P(s)} \\
&= \frac{\sum_{f=0}^{1} \sum_{c \in C} \sum_{w=0}^{k} P(\$,s,c,w,f)}{P(s)} \\
&\propto \sum_{f=0}^{1} \sum_{c \in C} \sum_{w=0}^{k} P(\$ \mid c)P(s \mid w,c)P(c \mid f)P(f)
\end{aligned}
$$

$$
P(s \mid c) = \sum_{w=0}^{k} P(s \mid w,c) \text{ (see 3.4)}
$$

$$
P(\$ \mid s) \propto \sum_{f=0}^{1} \sum_{c \in C} P(\$ \mid c)P(s \mid c)P(c \mid f)P(f)
$$

The predicted cost given a sequence is then the following:

$$
\$^* = \arg \max_{\$} \sum_{f=0}^{1} \sum_{c \in C} P(\$ \mid c)P(s \mid c)P(c \mid f)P(f)
$$

3.6. Intended Experiments

We will split up the data into train, validation and test sets, fit HMMs using different hyperparameters, such as the maximum number of time-windows k and window size $|p|$. The prediction performance of the fitted HMMs can be evaluated using the validation set to identify an optimum set of hyperparameters. The prediction performance of the corresponding optimum HMM is then evaluated using the test set.

4. Pre-processing

4.1. Data Cleaning

Inconsistent formating of dates, duplicate rows, and missing data were identified and dealt with in both the sensor snapshot and repair data. The snapshot data were largely complete with most columns having less than 1% missing data, except for one column ('Ignition Cycle Counter': 69% missing) which was removed. The remaining missing data in each row were imputed with their respective mean values corresponding to that row's vehicle ID. These transformations maximize the number of complete snapshots for training our model. The data were then shifted and whitened to have mean 0 and variance 1.

The repair data contains key information like vehicle ID, repair date, repair cost and failure codes ATA3, ATA6 and ATA9 which narrow down the failure location. The vehicle ID and repair date are two necessary keys for identifying specific sensor snapshots that led up to each repair. Unfortunately, 24% of rows in the repair data were missing the vehicle ID and another 13% of rows had no corresponding snapshot data before the repair date. These rows had to be discarded because the necessary snapshot data needed for predicting the corresponding repairs in those rows either cannot be identified or are unavailable. In addition, an anomalous high-cost engine repair that resulted from a truck undecking accident was found and removed because the snapshots before the repair will be irrelevant to predicting that accident. These issues reduced the number of valid repairs in our data from 1128 to 713.

Once the data were cleaned, the data needed to be put into a format that was useful for processing. We created a map of failure types, which had arrays of snapshots that corresponded to the hyperparameter k, the maximum number of windows of snapshots before the failure. The window $w = 0$ corresponded to the window immediately before the failure. The window size hyperparameter, $|p|$, determined the number of sequential snapshots contained in each window. Each sequence of snapshots in our training data can be associated with multiple (c, w) pairs and will be assigned to train the corresponding $HMM_{c,w}$ models. When we do prediction of the most likely c and w given a sequence s of snapshots from the validation or test set, we will feed the given sequence into all the fitted HMM models (see sections 3.3 and 3.4 for details).

4.2. Selection of Failure Codes to Predict

Given limited 'high' and 'very high' cost repairs (21 out of 713) to train our prediction model, we reframed the problem into predicting suitable failure codes that constitute a decent percentage of our repair data and are associated with higher cost repairs. We found that 95 out of the 98 repairs in the 'medium', 'high' or 'very high' cost categories only occur in three locations specified by their ATA3 codes (43:exhaust system, 44:fuel system, 45:power plant). In addition, 77 out of those 95 repairs can be traced further to just six ATA6 codes (43004, 44004, 45002, 45007, 45008, 45021) which dominate the repairs in the 'medium', 'high'

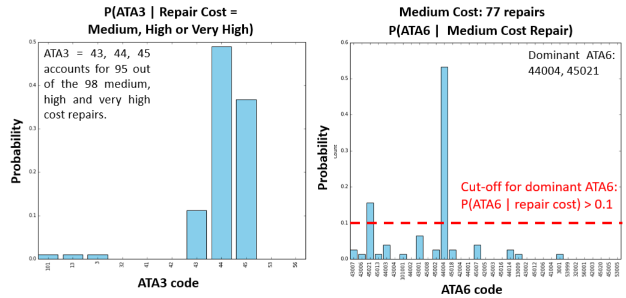


Figure 4. Left: Identifying ATA3 codes strongly associated with medium, high and very high cost repairs. Right: Identifying ATA6 codes that are more strongly associated with medium cost repairs.

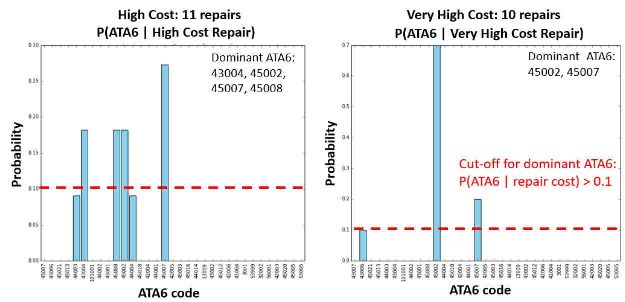


Figure 5. Identifying ATA6 codes that are more strongly associated with high and very high cost repairs.

ATA6	P(Repair Cost \$ | ATA6)			Training Data
	\$ = Medium	\$ = High	\$ = Very High	(% of repairs)
43004	0.00	0.15	0.00	2%
44004	0.58	0.01	0.00	10%
45002	0.07	0.07	0.25	4%
45007	0.18	0.18	0.12	2%
45008	0.00	0.17	0.00	2%
45021	0.09	0.00	0.00	19%
				39%

Figure 6. Conditional probabilities of repair costs given specific ATA6 failure codes. Some ATA6 failure codes are indicative of 'medium', 'high' or 'very high' cost repairs based on their relatively higher conditional probabilities (highlighted in yellow).

or 'very high' cost categories (see Figures 4 and 5). We used $P(c \mid \$) > 0.1$ as a cut-off to identify dominant ATA6 codes in each repair cost category.

As the ATA3 is a broad failure code encompassing many 'low' and 'very low' cost repairs, we decided to predict the ATA6 failure code instead, since it is a more meaningful predictor of potential repair cost. Among the 713 valid repairs in our data, 278 of them ($\sim$39%) had the six ATA6 codes that are more strongly associated with 'medium', 'high' or 'very high' cost repairs, which are our target cost categories. Figure 6 shows $P(\$ \mid c)$ for 'medium', 'high' and 'very high' cost repairs given the six ATA6 codes of interest.

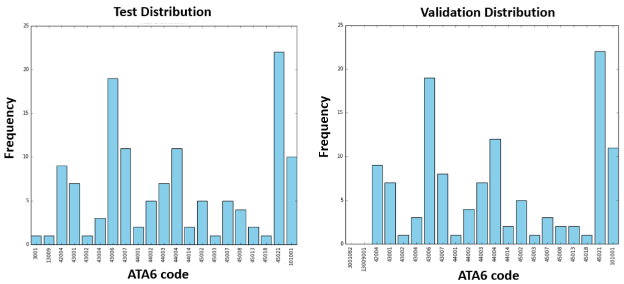


Figure 7. Comparison of test and validation distributions over ATA6 fault codes to ensure no test-validation mismatch.

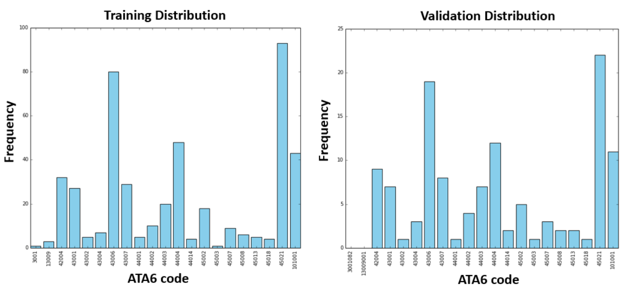


Figure 8. Comparison of training and validation distributions over ATA6 fault codes to ensure no train-validation mismatch.

4.3. Train, Validation, and Test Set

The repairs corresponding to each ATA6 code were split further based on their specific ATA9 codes, and then assigned into training, validation and test sets roughly in the proportion of 70%, 15%, 15% via systematic sampling. This minimizes distribution mismatch among the test, validation and training sets across the ATA9, ATA6 and ATA3 codes (note: first 3 digits of ATA9 and ATA6 = ATA3, first 6 digits of ATA9 = ATA6). Figures 7 and 8 show the resulting test, validation and training set distributions across their common ATA6 failure codes. Next, vehicles which never appeared in the repair data are assumed to have 'healthy' snapshots and also split into test, validation and training sets in a similar proportion.

5. HMM

5.1. Training Setup

We capture failure code and time dependencies by fitting a different HMM for each ATA6 failure code c and time window w prior to a repair with failure code c.

The model assumes that each vehicle has an underlying hidden state at time t which we will denote as i_t. The state of any given vehicle evolves over time as specified by the state transition matrix $T(i_t, i_{t+1})$ which is learned during the training of the HMM along with the initial state occupation distribution, the mean and covariance parameters for each state. These parameters, which we denote all together as Θ, allow us to infer how a sequence of snapshots (i.e. observations) s_t is generated according to a state-dependent observation likelihood function which specifies the log likelihood probability of observing sequence s_t given the underlying state i_t at time t. Training the HMM entails estimating these parameters via an expectation-maximization (EM) algorithm from the set of all observation sequences for the failure code c, and time window w.

Given our setup, let $S_{c,w}^{v_j}$ be the sequence of snapshots for vehicle v_j in time-window w preceding a repair with ATA6 fault code c. If there are m such vehicles, it follows that

$$S_{c,w} = \{S_{c,w}^{v_1}, S_{c,w}^{v_2}, ..., S_{c,w}^{v_m}\}$$

is the set of sequences corresponding to vehicles associated with failure code c and time window w. Using the training portion of this set, we trained a single HMM as $HMM_{c,w}$ and repeated this for all c and w on their respective training sets. After training is completed, each $HMM_{c,w}$ can be seen as a function that takes some new unseen sequence s' and predicts the likelihood of seeing that sequence.

$$HMM_{c,w}(s') = \ell(s' \mid \Theta_{c,w})$$

where $\ell(s' \mid \Theta_{c,w})$ is the log-likelihood of seeing that sequence given failure code c and time window w, and $\Theta_{c,w}$ are the corresponding learned HMM parameters. The log-likelihoods are then converted into relative probabilities by computing a softmax of $\ell(s' \mid \Theta_{c,w})$ over all c and w. This yeilds the resulting conditional probabiltiies of a sequence of snapshots given failure code and time window.

$$P(S = s' \mid C = c, W = w)$$

This is then used to predict the most likely failure code c and number of time-windows to failure w as described in 3.4 and 3.3.

5.2. Hyperparameters

In order to train the models given our data, a number of assumptions were made.

- The number of underlying hidden states set to 2.
- All features are treated as continuous resulting in each HMM having Gaussian Emissions.
- Each state uses a diagonal covariance matrix.
- Priors for initial state occupation set using Dirichlet prior distribution with parameters as 1.
- Number of iterations was set to 100.

The number of latent variables (i.e. hidden states) was chosen to be 2. This was found by setting the number of hidden states as a hyperparameter and performing a grid search. Using two hidden states consistently gave significantly better validation error than all other values for all failure codes and time windows. Hence there is strong evidence to suggest that there are 2 underlying hidden states.

All features were treated as continuous. We observe that there are 62 features for each snapshot, of which 9 are categorical and the rest are continuous by nature. Other than 4 binary features, the remaining 5 categorical features were treated as continuous variables instead of using one-hot encoding. One-hot encoding was avoided due to the high dimensionality issue from variables with large and sparse number of categories such as DTCID which has 343 categories.

During training, setting the covariance for each state to be diagonal was found to give better results than using a full covariance matrix, which also improved running time. Furthermore, the Dirichlet priors were a suitable choice for initialization of state occupation distribution priors.

The maximum number of iterations for each HMM training session was set to 100 after running several training sessions and discovering that no further improvement was being made for larger iterations as the EM Algorithm converged to maxima (possibly local maxima) by or before the 100th iteration for all HMM models.

In order to find the best performing model, a grid search was performed over different window sizes (i.e. number of snapshots per window) and number of windows (i.e. span of time considered). The grid search was performed from 10 to 80 for number of windows and 10 to 80 for window size.

6. Accuracy and Error Calculation

All accuracies and errors are averaged over all snapshot sequences in the validation or test set.

6.1. Code Prediction

The failure code c and time-window w that maximizes $P(s \mid c, w)$ from the $|C||W|$ HMMs will be the predicted code and window. However, each vehicle j can experience more than one repair corresponding to either the same or different failure codes c. About $\frac{1}{5}$ of vehicles have more than 1 repair, so even though the intended (i.e. nearest) failure code c_{sj} that occurs after a snapshot sequence s for vehicle j may not be predicted, that prediction $\hat{c}_{sj}$ is still valid if it falls within the set of failure codes C_{sj} experienced by vehicle j after that sequence s. Given this circumstance, we needed a customized accuracy and error function. Our code accuracy calculation takes into account the other valid failure codes for a prediction by awarding the model α if the prediction is still a valid repair, and the inverse for an error:

$$acc(c_{sj}, \hat{c}_{sj}) = \begin{cases} 1, & \text{if } c_{sj} = \hat{c}_{sj} \\ \alpha, & \text{if } c_{sj} \in C_{sj} \\ 0, & \text{otherwise} \end{cases}$$

$$err(c_{sj}, \hat{c}_{sj}) = \begin{cases} 0, & \text{if } c_{sj} = \hat{c}_{sj} \\ 1-\alpha, & \text{if } c_{sj} \in C_{sj} \\ 1, & \text{otherwise} \end{cases}$$

6.2. Window Prediction

The window accuracy and error calculation also requires some flexibility, since a prediction of being within 1 window of the true window should not be as big of an error as a prediction 2 or more windows off of the true window. We use an exponential decay function for the window accuracy calculation, and MSE for the error calculation:

$$acc(w_{sj}, \hat{w}_{sj}) = \frac{1}{1 + (w_{sj} - \hat{w}_{sj})^2}$$

$$err(w_{sj}, \hat{w}_{sj}) = (w_{sj} - \hat{w}_{sj})^2$$

6.3. Cost Prediction

Since we are calculating predicted cost given a sequence, the error is simply the mean squared error of the predicted repair cost of a particular sequence s from vehicle j is $\hat{\$}_{sj}$ and the actual cost of that sequence $\$_{sj}$ as follows:

$$err(\$_{sj}, \hat{\$}_{sj}) = (\$_{sj} - \hat{\$}_{sj})^2$$

7. Test Results

As shown in figures 9, 10, and 11 it is clear that for each number of windows used the accuracy drastically falls at a particular range of window sizes. Therefore, is it clear that the optimal choice for window size should be small. In this case, the smallest value is 10. For the number of windows, 10 on average outperforms other window sizes for all number of windows. This can also be seen by observing the mean squared error rate which is inversely proportional to the accuracies. Another way to see this is to compute an aggregation over the accuracies, i.e. the average accuracy over the ATA6 code, time window, and repair cost for all window sizes and number of windows pairs. The maximum average accuracy was consistent which the observed values of window size = 10 and number of windows = 10. The validation results are shown in table 7.

Validation: Num. Windows = 10, Window Size = 10		
Prediction	Error	Accuracy (%)
ATA6 Code	0.45	55.11
Time Window	21.86	34.53
Repair Cost	2.34	52.31
Aggregate	8.22	47.32

Finally, predictions for the test step are shown below in table 7 with the ATA6 code accuracy as the greatest deviation from the validation results.

Test: Num. Windows = 10, Window Size = 10		
Prediction	Error	Accuracy (%)
ATA6 Code	0.63	37.29
Time Window	21.84	33.54
Repair Cost	2.45	41.25
Aggregate	8.3	37.36

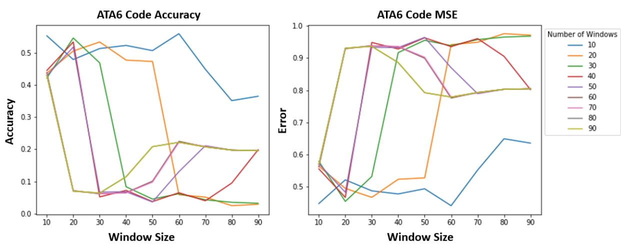


Figure 9. Sensitivity of ATA6 failure code prediction accuracy and error to hyperparameters (window size and number of windows).

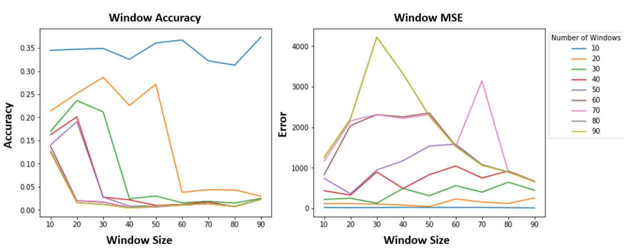


Figure 10. Sensitivity of window prediction accuracy and error to hyperparameters (window size and number of windows).

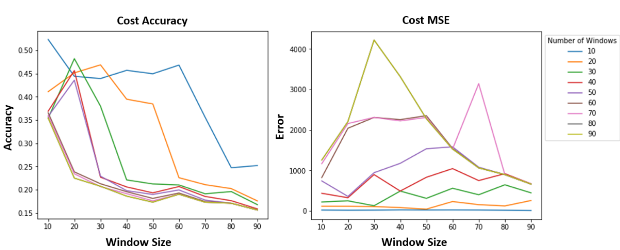


Figure 11. Sensitivity of window prediction accuracy and error to hyperparameters (window size and number of windows).

8. Future Work

The selection of dominant ATA6 failure codes, together with the normalization of their priors can introduce selection bias in the predictions from our fitted HMM models. The degree of selection bias is influenced by how likely the other ATA6 failure codes occur, as well as how likely those failures will lead to higher cost repairs. One way to lower selection bias is to fit an HMM model for the omitted ATA6 failure codes, which can be an extension of our preliminary work. However, it should be recognized that there will still be selection bias associated with unobserved faults and repairs. Nevertheless, we believe the accumulation of repair data and reselection of dominant ATA6 failure codes using our framework has potential to reduce the degree of selection bias and improve the estimation accuracy of the relevant priors over time, which will improve the prediction accuracy of the fitted HMM models.

As described in [8], we could use a two-stage classification scheme. The first stage would be composed of models that predict a higher level code, and the second stage would be composed of models to predict the next level code. In our work, we focused only on ATA6 codes, but could potentially use a two-stage with ATA3 then ATA6, or even a three stage with ATA3, ATA6, then ATA9. This would likely give more precise failure predictions that would be more helpful for predictive maintenance.

Given the promise of HSMMs in [2], we could use HSMMs for this dataset and likely achieve higher accuracy results.

Categorical variables should be either treated differently using one-hot encoding, in which case more attention to the dimensionality of the problems would be needed, or by separating them from the continuous data and feeding separately into a multinomial HMM resulting in a mixture model of discrete transition probabilities.

Training with various initializations for HMM parameters could yield slight improvements in overall loss and accuracy. The reason for this is due to the EM Algorithm using a gradient based update. Because of this it is possible to get stuck in local optima. However, this would have to be done for all time windows and all repair codes increasing the complexity of the algorithm by a multiple of the current complexity, the multiple being the number of different initializations desired.

Given the results from the hyperparameter search, there may be a better model for windows smaller than 10 and number of windows smaller than 10. In future work, we could explore the hyperparameter space more.

If we were to pursue creating a production-ready model, we would need to do more work in feature selection, since that can result in cost savings in collection for PACCAR.

9. Joint Project

In addition to AA228, all three of us are taking CS229, and we are combining the projects from both courses into one. We have created models specific to the concepts learned in CS229 for the project in that class.

References

[1] Paccar. https://github.com/christydennison/paccar. Accessed: 2017-12-09. 1

[2] M. Dong, D. He, P. Banerjee, and J. Keller. Equipment health diagnosis and prognosis using hidden semi-markov models. *The International Journal of Advanced Manufacturing Technology*, 30(7):738–749, Oct 2006. 1, 7

[3] M. K. Erik Frisk and E. Larsson. Data-driven lead-acid battery prognostics using random survival forests. Annual Conference of the Prognostics and Health Management Society 2014. 1

[4] M. J. Kochenderfer, C. Amato, G. Chowdhary, J. P. How, H. J. D. Reynolds, J. R. Thornton, P. A. Torres-Carrasquillo, N. K. Üre, and J. Vian. *Decision Making Under Uncertainty: Theory and Application*. The MIT Press, 1st edition, 2015. 1

[5] T. T. Le, F. Chatelain, and C. Bérenguer. Hidden Markov Models for diagnostics and prognostics of systems under multiple deterioration modes. In *European Safety and Reliability Conference (ESREL 2014)*, pages 1197–1204, Wroclaw, Poland, Sept. 2014. Taylor & Francis - CRC Press/Balkema. 1

[6] R. Prytz. Machine learning methods for vehicle predictive maintenance using off-board and on-board data. *Halmstad University Dissertations*, 9, 09 2014. 1

[7] R. Prytz, S. Nowaczyk, T. Rognvaldsson, and S. Byttner. Analysis of truck compressor failures based on logged vehicle data. Las Vegas, Nevada, USA. 9th International Conference on Data Mining. 1

[8] S. Ramanujam. The data science behind predictive maintenance for connected cars. 2016. 1, 7

[9] R. C. Stephen Adams, Peter A. Beling. Feature selection for hidden markov models and hidden semi-markov models. *IEEE*, 2016. 2