Term Project

CMPT - 318 / Fall 2022

Group 7
Michael Kuby ID# 301562996
Harsh Behal ID# 301342771

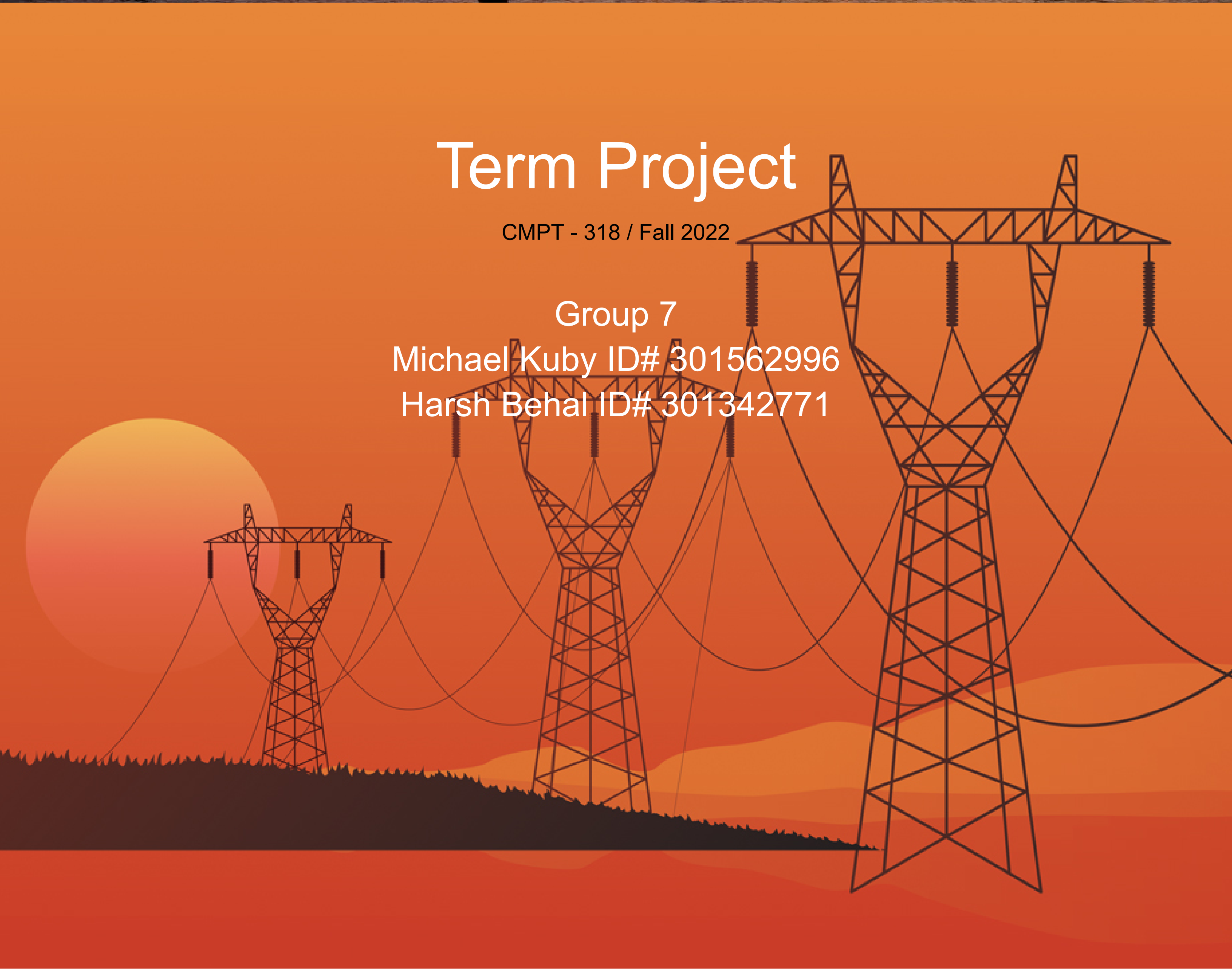


Table of Contents:

1) Introduction - Abstract

2) Response Variable Analysis via PCA

3) Observation Time Window

4) Data Partitioning

5) Hidden Markov Models: Training and Assessment

6) Anomaly Detection Across Three Selected Data Sets

7) References

Abstract

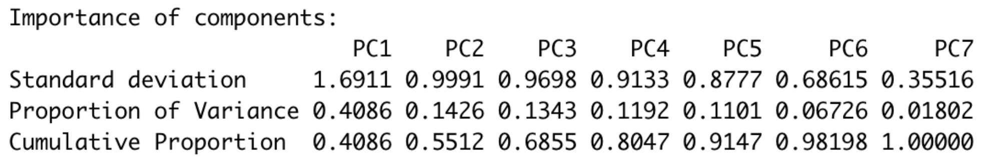
The following is an implementation of Hidden Markov Modeling (HMM) for anomaly detection and analysis for multivariate time series data. The trend toward automation of critical infrastructure and the increasing reliance on supervisory control systems has led to widening attack surfaces for advanced persistent threats to attempt to exploit. Here we explore techniques for detecting such threats by looking for anomalies in given multivariate time series data with the use of HMMs.

Specifically, the following looks at multivariate time series data from late 2006 - late 2009 that tracks 7 different features: Global active power, Global reactive power, Voltage, Global intensity, as well as Sub metering 1-3. We use Principal Component Analysis (PCA) to help reveal those features that best explain the variation in the data. Our analysis led us to use the principal components themselves for training the models. We explore some of the perceived benefits and note some of the limitations of the use of PCs for training HMMs in the section below.

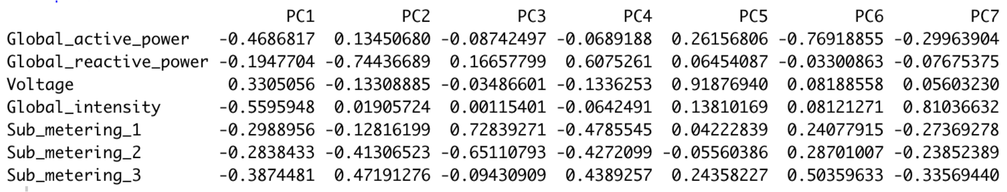
With our chosen time window between 03:00 - 06:00, and Tuesday as our day of choice, we found that a 40-state model optimized for both log Likelihood as well as BIC values when tested against train data. However, a considerably simpler 24 state model performed nearly as well in all metrics, and thus was our model of choice. We used this 24 state model to assess the degree of anomalousness amongst three sets of data. Our findings indicate that DataWithAnomalies3.txt was the most anomalous of the three, DataWithAnomalies1.txt was the second most anomalous, and DataWithAnomalies2.txt was the least anomalous.

Response Variable Analysis Via PCA

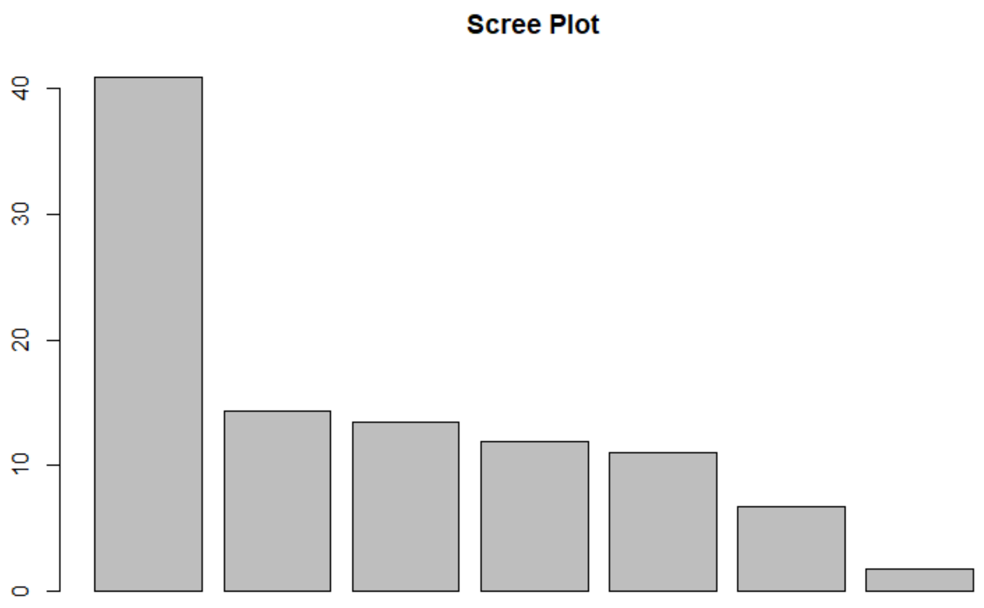
Principal Component Analysis, or PCA, is a feature analysis technique used to analyze and/or reduce the dimensionality of a set of features in a dataset into a smaller number of features, called principal components. These resultant components are new variables constructed from linear combinations of the original variables, with the added benefit that the principal components are completely uncorrelated [1]. With a large number of dimensions per observation, PCA helps to unveil which features of a data set best explain the data's variation. The results of our PCA analysis are as follows:



```
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7
Standard deviation     1.6911 0.9991 0.9698 0.9133 0.8777 0.68615 0.35516
Proportion of Variance 0.4086 0.1426 0.1343 0.1192 0.1101 0.06726 0.01802
Cumulative Proportion  0.4086 0.5512 0.6855 0.8047 0.9147 0.98198 1.00000
```



```
                             PC1         PC2         PC3        PC4         PC5         PC6         PC7
Global_active_power   -0.4686817  0.13450680 -0.08742497 -0.0689188  0.26156806 -0.76918855 -0.29963904
Global_reactive_power -0.1947704 -0.74436689  0.16657799  0.6075261  0.06454087 -0.03300863 -0.07675375
Voltage                0.3305056 -0.13308885 -0.03486601 -0.1336253  0.91876940  0.08188558  0.05603230
Global_intensity      -0.5595948  0.01905724  0.00115401 -0.0642491  0.13810169  0.08121271  0.81036632
Sub_metering_1        -0.2988956 -0.12816199  0.72839271 -0.4785545  0.04222839  0.24077915 -0.27369278
Sub_metering_2        -0.2838433 -0.41306523 -0.65110793 -0.4272099 -0.05560386  0.28701007 -0.23852389
Sub_metering_3        -0.3874481  0.47191276 -0.09430909  0.4389257  0.24358227  0.50359633 -0.33569440
```



The first thing to notice here is that the proportion of variance in the data explained by PC1 is a considerable .4086, or 40.86%. PC2-5 are each somewhat equally important in this regard, each explaining between 11.01% and 14.26% of the data's variation. Turning our attention to the cumulative proportion of variance, we see that by considering PC1 - PC3 we can explain 68.55% of the data's total variance.

The next step in our PCA analysis was to choose a subset of features for our HMM training based on their perceived impact on the data. From the second chart, we can see what is called the load of the features in relation to the primary components[2]. The load values can alternatively be considered coefficients for the features with respect to the primary components. Here we are looking for features of large magnitude that correspond to the most impactful primary components. For example, scanning the column for PC1, it is clear that Global_intensity, Global_active_power, and Sub_metering_3 have the greatest impact on PC1, with Voltage not far behind. Similarly, for PC2, Global_reactive_power has far and away the most impact on the value of PC2.

Seeking some sort of methodical way of choosing features required some creativity. By thinking of the coefficient values as a matrix, and the proportion of variance values corresponding to each PC as a column vector, we performed matrix multiplication on the given matrices. This computed the cumulative sum of products of a features coefficient multiplied by the proportion of variance as explained by each PC. Visually:

```
-0.4686817  0.13450680 -0.08742497 -0.0689188  0.26156806 -0.76918855 -0.29963904     0.40860     -0.220615454
-0.1947704 -0.74436689  0.16657799  0.6075261  0.06454087 -0.03300863 -0.07675375     0.14260     -0.087438689
 0.3305056 -0.13308885 -0.03486601 -0.1336253  0.91876940  0.08188558  0.05603230     0.13430      0.203129294
-0.5595948  0.01905724  0.00115401 -0.0642491  0.13810169  0.08121271  0.81036632     0.11920     -0.198166234
-0.2988956 -0.12816199  0.72839271 -0.4785545  0.04222839  0.24077915 -0.27369278     0.11010     -0.083712979
-0.2838433 -0.41306523 -0.65110793 -0.4272099 -0.05560386  0.28701007 -0.23852389     0.06726     -0.304364593
-0.3874481  0.47191276 -0.09430909  0.4389257  0.24358227  0.50359633 -0.33569440  X  0.01802  =   0.003278792
```

Where each value can be considered as the sum of the weighted distributed impact on the variance of the data, with the topmost value corresponding to Global_active_power, and the bottommost value corresponding to Sub_metering_3. What we see then is that

Sub_metering_2, Global_active_power, Voltage, and Global_intensity have the largest effect on the variance of the data (in that order). That is, should we reduce the number of features used to train our model, a reasonable choice would be to choose the top two or three features from this list.

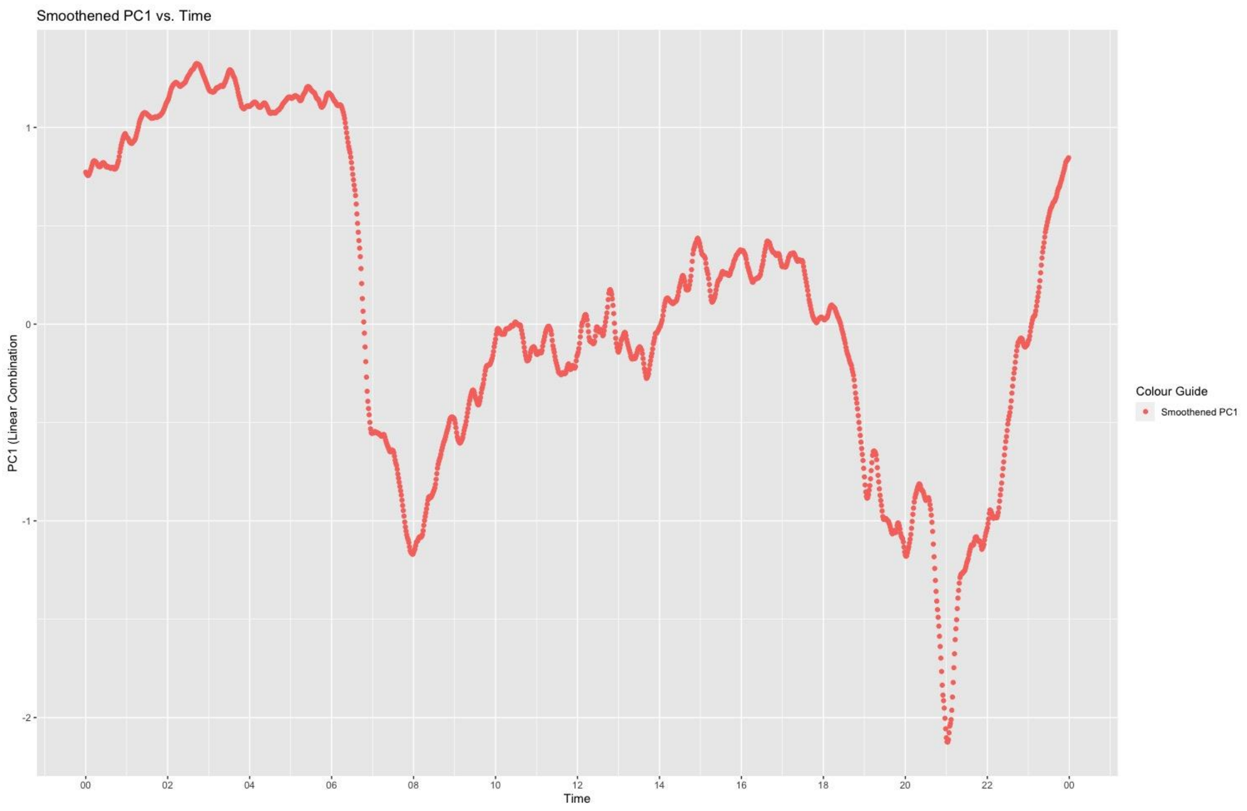
An alternative approach to feature reduction performed in the above way would be to use the PCAs themselves as the features for training the model. That is, based on the PCA analysis done thus far, we can be certain that by choosing PCs 1-3, fully 68.55% of the variance in the data would be explained. There is comfort in using this approach as it presents a solid metric to base our choosing. Using PCs also has the added value in that, by nature of the PCA approach, each of the PCxs' is guaranteed to be wholly uncorrelated, something we cannot be so sure of using the original features [1]. The third benefit to this approach is that, because the principal components themselves are linear combinations of the original features, by using them we are in some sense maintaining the impact of all of the original features, and the effect of each on the form of the data as a whole should be maintained to some degree.

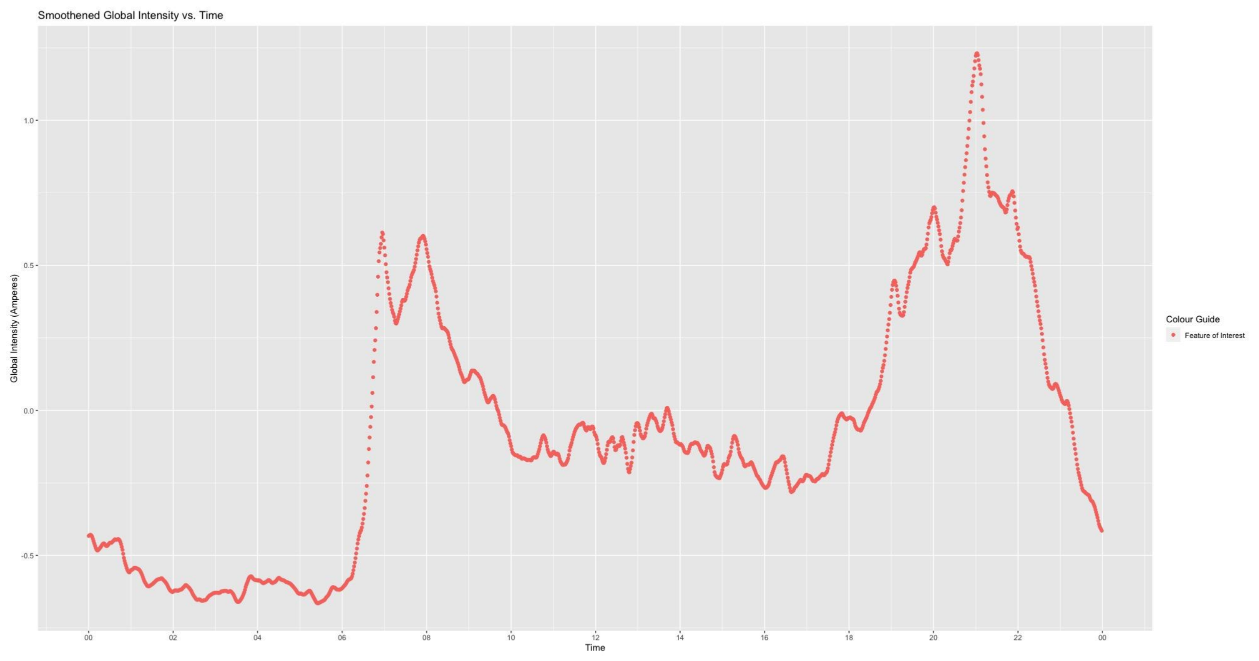
However, the authors also recognize the drawbacks to using the principal components themselves as the features for training the model: namely that any inferences to be drawn about the results of the model and the input of the data itself will themselves be, well, hard to draw [1]. This is a result of the fact that the principal components are linear combinations of the original features. This makes it hard to do any analysis other than determining the log-likelihood values of sequences of observations given an HMM. Fortunately, that is exactly what we are planning to answer.

Considering both approaches, we decided to move ahead with the more creative approach of using the PCs to train our model, as we suspect that the ability to train a model that is able to perceive anomalies may be enhanced, albeit at the expense of inference; a tradeoff we are willing to live with for the assigned task.

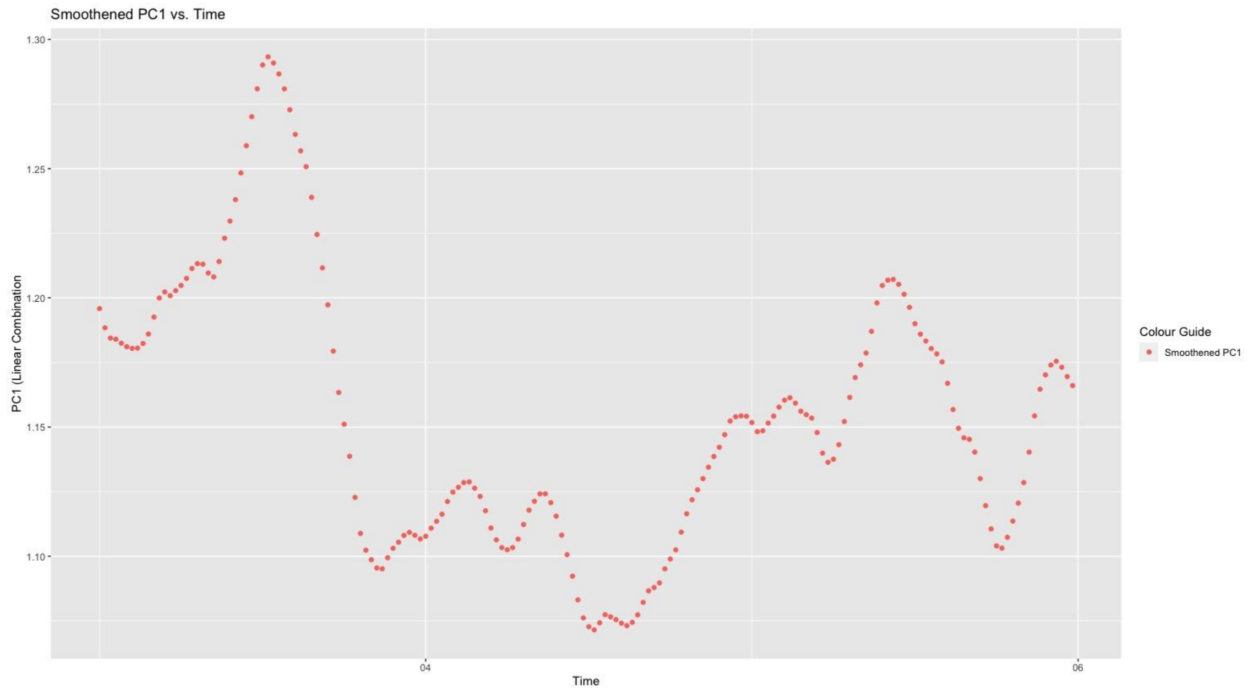
Observation Time Window

To determine a time window for the specific weekday that showed a clearly recognizable consumption pattern we computed the average smoothened week for PC1 for each week in 2007 by smoothening the PC1 data for each week using the moving average, with a time window of 10 consecutive observations. We then computed the average smoothened week by finding the average value for all times for one whole week. Choosing Tuesday as our day of choice, we plotted the average Tuesday to try to identify an interesting and definable time window. We repeated this process for Global intensity so that we could visually compare how PC1 and Global Intensity might coincide. Note that Tuesday was chosen as it represented the heart of the work week.





In the graphs above we notice the inverse relationship between PC1 and Global Intensity. Hence, we proceed under the assumption that PC1 has an inverse relationship with global energy consumption: that is, peaks in PC1 correspond to periods of relative calm, while the valleys correspond to periods of heavy use.



Based on the above graphs we decided to choose the three-hour time window just before morning consumption: 03:00 - 06:00. We suspected that, because of the relative calm, identifying anomalies during this window may be somewhat easier to identify. Above is the graph depicting our chosen time window. Note the positive value of PC1, indicating relatively low consumption, and the small amount of variation in the values.

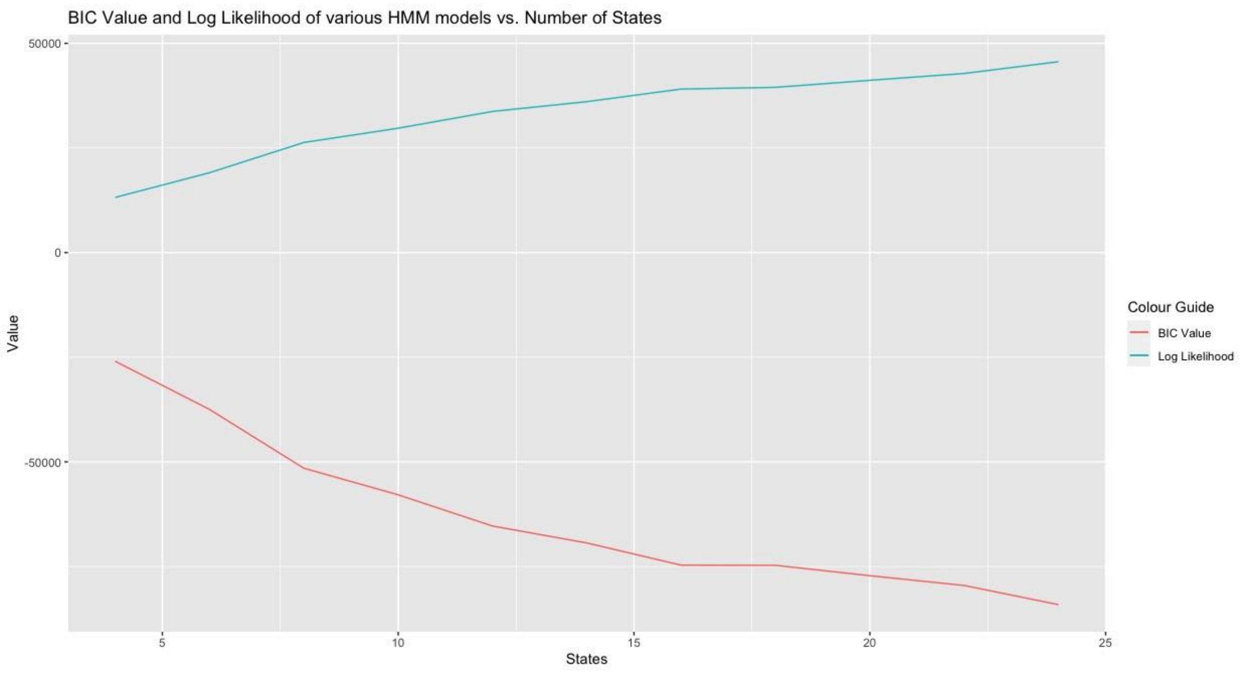
Data Partitioning

We partitioned our data into two sets: train and test. The divide was 70:30 train data to test data. Of primary concern was *how* to partition the data: a common method is to randomly sample to obtain a split; however, this approach would not work for our purposes as the underlying principle of the HMM is the fact that the next state of the model is predicated on the current state, but not on any of the past states (otherwise known as the Markov Property). Thus, the sequence is extremely important, as it is what enables the model to learn about the underlying pattern and structure of the data.

With this in mind, we chose to partition the data into contiguous blocks. The total data set consisted of 154 total weeks of data. The first block, the train data set: 107 contiguous weeks starting in late 2006 and going all the way up to the end of 2008, accounting for 70% of the data. The second block, the test data set: 47 weeks starting at the beginning of 2009 and ending just before the start of December that year, which accounted for the other 30% of the data.

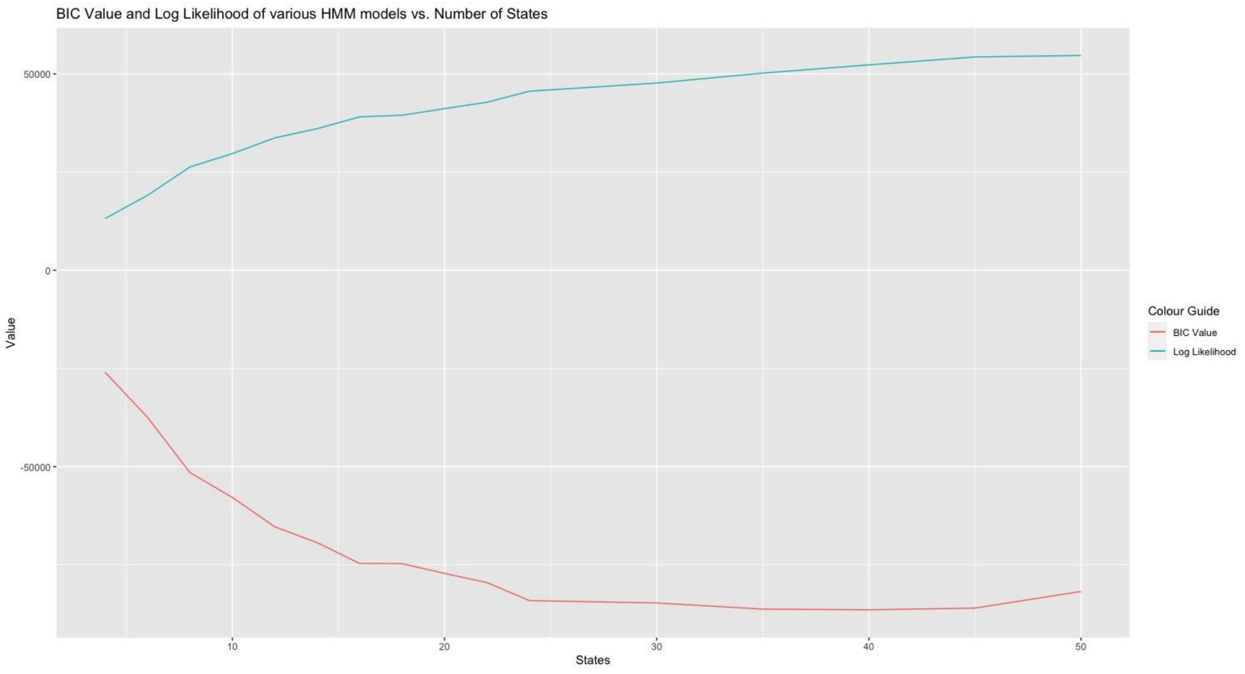
Hidden Markov Models: Training and Assessment

We trained 11 models with states 4, 6, 8, …, 24. It is important to note that during the process of PCA analysis we scaled our data through standardization. Standardizing our data consisted of scaling the features such that they have characteristics of a standard normally distributed graph with a mean of 0 and standard deviation of 1. This allowed us to make some assumptions about our features when training the model, namely that the distributions of the features were characteristically gaussian. Fitting each of the models we found the following log likelihoods and BIC values:



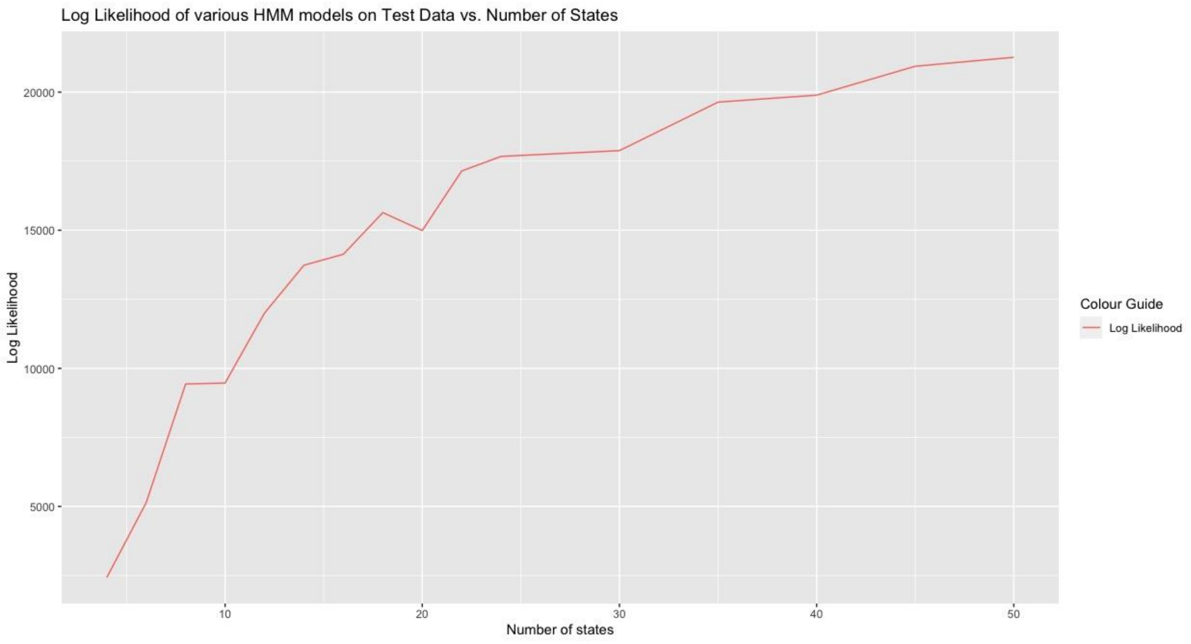
What we are looking for here is a maximized log-likelihood accompanied by a minimized BIC value. Based on the training data and the above graph it thus looked as though the ideal number of states was 24, and possibly training models with greater than 24 states would have found an optimal model. Based on the above assessment it is clear then that the 24-state model is our chosen model for testing.

Out of interest, however, we decided to continue to train models with higher states until we saw classical overfitting as seen in a positive regression of the BIC values. That is, we trained models with 30 - 50 states in 5-state increments in an attempt to find the “optimal” state model. Of 16 total models we trained then, we found that the best model was the 40-state model and that at 45 and 50 states the BIC values did finally begin to positively regress. The graph below bores this out:



It is clear that there is very limited benefit to using the 40-state model over 24, and due to the increase in complexity, the 24-state model would be preferred in a real-world situation.

We then assessed the models against the test data. We found that an increase in states consistently led to an increase in log likelihood values. That is, the 50-state model performed best. Our results are shown below.



Again we noticed that the greatest benefit is accrued throughout the first 24 state increases, and there is a distinct tapering afterward. This reinforces our belief in the 24-state model as the ideal balance of performance and simplicity.

Our 24-state model predicted has the following properties:

```
Convergence info: Log likelihood converged to within tol. (relative change)
'log Lik.' 45606.29 (df=719)
AIC:  -89774.59
BIC:  -84119.09
```

```
Initial state probabilities model
  pr1   pr2   pr3   pr4   pr5   pr6   pr7   pr8   pr9  pr10  pr11  pr12  pr13  pr14  pr15  pr16  pr17  pr18  pr19  pr20  pr21  pr22  pr23  pr24
0.009 0.011 0.076 0.009 0.033 0.029 0.064 0.019 0.037 0.093 0.069 0.020 0.049 0.036 0.037 0.065 0.052 0.046 0.015 0.051 0.077 0.056 0.000 0.047

Transition matrix
        toS1  toS2  toS3  toS4  toS5  toS6  toS7  toS8  toS9 toS10 toS11 toS12 toS13 toS14 toS15 toS16 toS17 toS18 toS19 toS20 toS21 toS22 toS23 toS24
fromS1  0.828 0.001 0.000 0.003 0.000 0.000 0.000 0.000 0.002 0.086 0.008 0.000 0.004 0.011 0.000 0.002 0.000 0.003 0.001 0.000 0.002 0.019 0.025 0.005
fromS2  0.000 0.741 0.000 0.000 0.000 0.000 0.000 0.025 0.032 0.051 0.000 0.000 0.000 0.109 0.042 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
fromS3  0.000 0.000 0.885 0.000 0.016 0.000 0.000 0.006 0.000 0.000 0.007 0.002 0.006 0.047 0.001 0.001 0.000 0.000 0.007 0.000 0.005 0.002 0.009 0.007
fromS4  0.020 0.005 0.004 0.029 0.005 0.000 0.000 0.000 0.000 0.060 0.000 0.000 0.000 0.006 0.000 0.000 0.000 0.005 0.000 0.000 0.000 0.000 0.866 0.000
fromS5  0.002 0.000 0.014 0.002 0.850 0.000 0.000 0.000 0.000 0.000 0.035 0.000 0.029 0.000 0.001 0.000 0.000 0.001 0.035 0.000 0.000 0.008 0.001 0.021
fromS6  0.006 0.000 0.000 0.006 0.000 0.901 0.030 0.000 0.000 0.006 0.000 0.000 0.000 0.000 0.000 0.042 0.000 0.000 0.000 0.000 0.004 0.000 0.000 0.006
fromS7  0.000 0.000 0.000 0.000 0.000 0.005 0.854 0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.003 0.027 0.002 0.000 0.039 0.042 0.000 0.000 0.026
fromS8  0.002 0.006 0.006 0.003 0.000 0.000 0.000 0.897 0.000 0.002 0.000 0.000 0.000 0.005 0.000 0.005 0.000 0.070 0.001 0.000 0.000 0.000 0.003 0.000
fromS9  0.000 0.015 0.000 0.000 0.000 0.000 0.000 0.000 0.821 0.125 0.000 0.000 0.000 0.000 0.024 0.015 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
fromS10 0.046 0.008 0.000 0.008 0.003 0.000 0.000 0.002 0.032 0.858 0.000 0.004 0.007 0.001 0.001 0.000 0.000 0.000 0.001 0.000 0.000 0.006 0.022 0.001
fromS11 0.002 0.000 0.005 0.000 0.020 0.000 0.000 0.000 0.000 0.001 0.807 0.017 0.011 0.000 0.002 0.000 0.000 0.000 0.043 0.000 0.029 0.018 0.004 0.042
fromS12 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.005 0.000 0.006 0.025 0.817 0.010 0.006 0.014 0.013 0.010 0.000 0.012 0.019 0.024 0.019 0.020 0.000
fromS13 0.001 0.000 0.000 0.000 0.033 0.000 0.000 0.002 0.000 0.014 0.011 0.012 0.832 0.007 0.016 0.000 0.000 0.000 0.018 0.000 0.000 0.023 0.021 0.007
fromS14 0.012 0.014 0.055 0.000 0.000 0.000 0.000 0.003 0.000 0.009 0.000 0.003 0.017 0.824 0.005 0.001 0.000 0.002 0.013 0.000 0.000 0.035 0.006 0.000
fromS15 0.000 0.009 0.000 0.000 0.002 0.000 0.000 0.000 0.016 0.000 0.002 0.025 0.026 0.008 0.821 0.009 0.000 0.000 0.012 0.000 0.000 0.069 0.000 0.000
fromS16 0.000 0.000 0.000 0.000 0.000 0.030 0.003 0.007 0.015 0.000 0.000 0.026 0.000 0.000 0.008 0.842 0.041 0.000 0.006 0.018 0.003 0.000 0.000 0.000
fromS17 0.000 0.000 0.000 0.000 0.000 0.001 0.028 0.000 0.000 0.000 0.000 0.013 0.000 0.000 0.003 0.042 0.860 0.000 0.000 0.017 0.031 0.002 0.003 0.000
fromS18 0.000 0.000 0.003 0.000 0.000 0.000 0.001 0.029 0.000 0.000 0.002 0.000 0.000 0.001 0.000 0.000 0.000 0.955 0.002 0.002 0.001 0.000 0.001 0.003
fromS19 0.010 0.000 0.006 0.003 0.038 0.000 0.000 0.000 0.000 0.007 0.084 0.006 0.016 0.001 0.010 0.000 0.000 0.000 0.761 0.005 0.002 0.039 0.004 0.007
fromS20 0.000 0.000 0.003 0.000 0.000 0.012 0.046 0.003 0.000 0.000 0.000 0.011 0.000 0.001 0.001 0.013 0.030 0.001 0.003 0.806 0.066 0.000 0.000 0.004
fromS21 0.002 0.000 0.001 0.000 0.000 0.003 0.020 0.000 0.000 0.000 0.044 0.023 0.000 0.000 0.000 0.000 0.010 0.001 0.005 0.036 0.808 0.001 0.002 0.044
fromS22 0.013 0.002 0.000 0.000 0.011 0.000 0.000 0.000 0.001 0.017 0.023 0.009 0.014 0.010 0.045 0.000 0.001 0.000 0.030 0.000 0.002 0.819 0.000 0.001
fromS23 0.018 0.000 0.010 0.350 0.014 0.005 0.000 0.002 0.000 0.041 0.020 0.040 0.047 0.000 0.002 0.000 0.000 0.004 0.004 0.012 0.006 0.000 0.403 0.023
fromS24 0.000 0.000 0.013 0.003 0.073 0.000 0.040 0.000 0.000 0.000 0.160 0.007 0.006 0.000 0.000 0.000 0.007 0.012 0.025 0.000 0.165 0.000 0.016 0.472

Response parameters
Resp 1 : gaussian
Resp 2 : gaussian
Resp 3 : gaussian
     Re1.(Intercept) Re1.sd Re2.(Intercept) Re2.sd Re3.(Intercept) Re3.sd
St1            1.210  0.151          -0.470  0.107          -0.009  0.064
St2           -0.151  0.333          -1.344  0.381           0.283  0.102
St3            0.689  0.248           0.308  0.296           0.009  0.067
St4            1.177  0.154          -0.275  0.104          -0.133  0.024
St5            1.391  0.098           0.501  0.042          -0.031  0.018
St6            1.613  0.164          -0.447  0.134          -0.080  0.061
St7            2.029  0.127           0.218  0.045          -0.093  0.018
St8           -0.659  0.892           0.105  0.455          -0.078  0.109
St9            0.696  0.180          -1.840  0.233           0.304  0.070
St10           0.930  0.147          -1.127  0.196           0.142  0.069
St11           1.569  0.087           0.422  0.036          -0.048  0.018
St12           1.423  0.107          -0.295  0.073           0.088  0.019
St13           1.146  0.128          -0.182  0.073           0.110  0.026
St14           0.339  0.262          -0.471  0.307           0.168  0.082
St15           1.002  0.134          -1.002  0.160           0.279  0.040
St16           1.357  0.152          -1.115  0.263           0.161  0.083
St17           1.649  0.192          -0.543  0.116           0.087  0.032
St18          -0.490  0.625           1.217  0.355          -0.278  0.095
St19           1.342  0.158          -0.006  0.087           0.050  0.029
St20           1.721  0.173          -0.159  0.117           0.013  0.031
St21           1.760  0.094           0.329  0.035          -0.068  0.019
St22           1.228  0.127          -0.421  0.099           0.155  0.026
St23           1.221  0.146          -0.212  0.114          -0.021  0.033
St24           1.374  0.225           0.434  0.109          -0.130  0.040
```

The 24-state model computed the log-likelihood of the test data with a value of 17673.86. To compare train and test scores we see divide by the number of samples to see that:

- Train data: 45606.29 / 19260 = 2.370
- Test data: 17673.86/8460 = 2.089

While we see some falloff between the train and test data, this is to be expected. The scores are rather comparable, so we have quite a bit of confidence in our model.

Anomaly Detection Across Three Selected Data Sets

The next step in our journey is to use our trained 24-state HMM to help us identify the degree to which three sets of data are found to be anomalous. Our approach here was to check the log-likelihood of the given sequence of events during the same day and time period as our chosen model was trained on. Recall that our day of choice was Tuesday and the time period was three hours from 03:00 - 06:00. With one observation per minute across those three hours, each continuous sequence consisted of 180 contiguous observations. Note that these numbers were the same as for our training procedure. We will also say here, as this holds across the three data sets, that each of the DataWithAnomalies*.txt files tested consisted of 51 weeks of data, and thus consisted of 51 * 180 = 9180 total observations.

DataWithAnomalies1.txt

Our model showed that the log Likelihood of the occurrence of the sequences presented by the data in DataWithAnomalies1.txt was -3221.862. Normalizing this value we have:

- -3221.862 / 9180 = -0.3510

DataWithAnomalies2.txt

Our model showed that the log Likelihood of the occurrence of the sequences presented by the data in DataWithAnomalies2.txt was -2577.61 Normalizing this value we have:

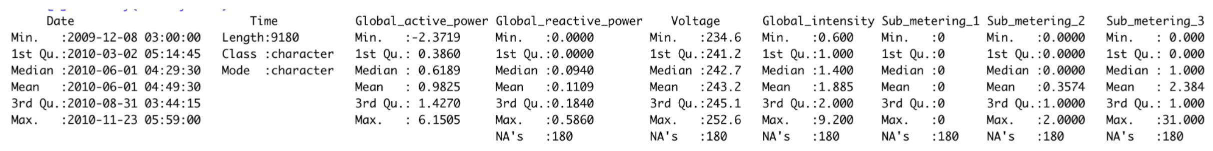
- -2577.61 / 9180 = -0.2809

DataWithAnomalies3.txt

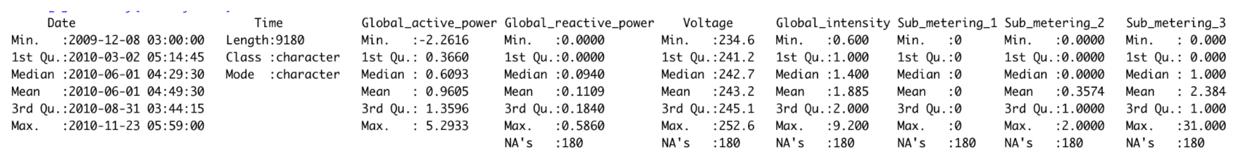
To our surprise, our model showed that the log Likelihood of the occurrence of the sequences presented by the data in DataWithAnomalies3.txt was NaN, the value in r for “not a number”. We triple-checked our work to make sure that we had not made any mistakes and indeed the code is written as functions, so each of the data sets were analyzed identically. We are not sure what would cause this result, nor particularly happy with it, but we can only make an inference based on possible reasons. Our suspicion is that the log-likelihood computed by our model was essentially zero. In other words, the data presented in DataWithAnomalies3.txt may be so anomalous that the value computed for its probability was so small that it caused the value to underflow. Further inspection of the data seems to reinforce this conclusion.

Data set inspection

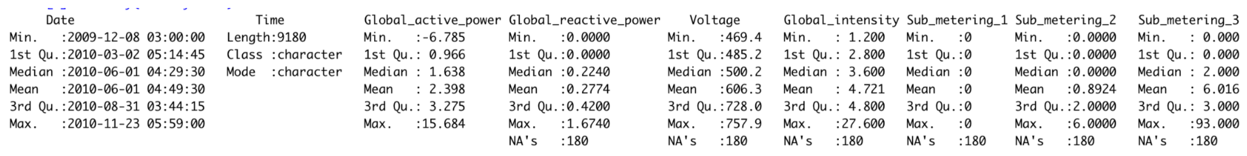
To aid in the confirmation of our suspicion about data set 3, we computed summary statistics of the three given data sets. The following are for DataWithAnomalies1.txt, DataWithAnomalies2.txt, and DataWithAnomalies3.txt, respectively:



```
      Date                        Time              Global_active_power Global_reactive_power    Voltage       Global_intensity Sub_metering_1 Sub_metering_2  Sub_metering_3
 Min.   :2009-12-08 03:00:00   Length:9180        Min.   :-2.3719     Min.   :0.0000        Min.   :234.6   Min.   :0.600    Min.   :0      Min.   :0.0000  Min.   : 0.000
 1st Qu.:2010-03-02 05:14:45   Class :character   1st Qu.: 0.3860     1st Qu.:0.0000        1st Qu.:241.2   1st Qu.:1.000    1st Qu.:0      1st Qu.:0.0000  1st Qu.: 0.000
 Median :2010-06-01 04:29:30   Mode  :character   Median : 0.6189     Median :0.0940        Median :242.7   Median :1.400    Median :0      Median :0.0000  Median : 1.000
 Mean   :2010-06-01 04:49:30                      Mean   : 0.9825     Mean   :0.1109        Mean   :243.2   Mean   :1.885    Mean   :0      Mean   :0.3574  Mean   : 2.384
 3rd Qu.:2010-08-31 03:44:15                      3rd Qu.: 1.4270     3rd Qu.:0.1840        3rd Qu.:245.1   3rd Qu.:2.000    3rd Qu.:0      3rd Qu.:1.0000  3rd Qu.: 1.000
 Max.   :2010-11-23 05:59:00                      Max.   : 6.1505     Max.   :0.5860        Max.   :252.6   Max.   :9.200    Max.   :0      Max.   :2.0000  Max.   :31.000
                                                                      NA's   :180           NA's   :180     NA's   :180      NA's   :180    NA's   :180     NA's   :180
```



```
      Date                        Time              Global_active_power Global_reactive_power    Voltage       Global_intensity Sub_metering_1 Sub_metering_2  Sub_metering_3
 Min.   :2009-12-08 03:00:00   Length:9180        Min.   :-2.2616     Min.   :0.0000        Min.   :234.6   Min.   :0.600    Min.   :0      Min.   :0.0000  Min.   : 0.000
 1st Qu.:2010-03-02 05:14:45   Class :character   1st Qu.: 0.3660     1st Qu.:0.0000        1st Qu.:241.2   1st Qu.:1.000    1st Qu.:0      1st Qu.:0.0000  1st Qu.: 0.000
 Median :2010-06-01 04:29:30   Mode  :character   Median : 0.6093     Median :0.0940        Median :242.7   Median :1.400    Median :0      Median :0.0000  Median : 1.000
 Mean   :2010-06-01 04:49:30                      Mean   : 0.9605     Mean   :0.1109        Mean   :243.2   Mean   :1.885    Mean   :0      Mean   :0.3574  Mean   : 2.384
 3rd Qu.:2010-08-31 03:44:15                      3rd Qu.: 1.3596     3rd Qu.:0.1840        3rd Qu.:245.1   3rd Qu.:2.000    3rd Qu.:0      3rd Qu.:1.0000  3rd Qu.: 1.000
 Max.   :2010-11-23 05:59:00                      Max.   : 5.2933     Max.   :0.5860        Max.   :252.6   Max.   :9.200    Max.   :0      Max.   :2.0000  Max.   :31.000
                                                                      NA's   :180           NA's   :180     NA's   :180      NA's   :180    NA's   :180     NA's   :180
```



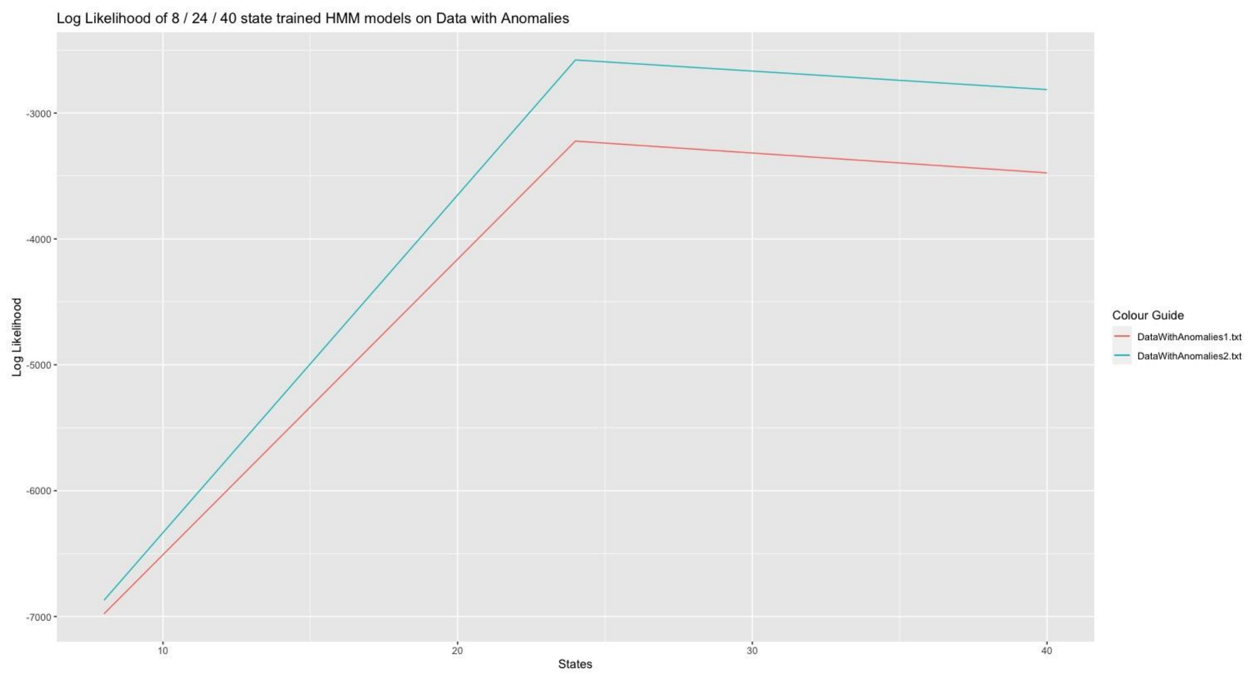
```
      Date                        Time              Global_active_power Global_reactive_power    Voltage       Global_intensity  Sub_metering_1 Sub_metering_2  Sub_metering_3
 Min.   :2009-12-08 03:00:00   Length:9180        Min.   :-6.785      Min.   :0.0000        Min.   :469.4   Min.   : 1.200    Min.   :0      Min.   :0.0000  Min.   : 0.000
 1st Qu.:2010-03-02 05:14:45   Class :character   1st Qu.: 0.966      1st Qu.:0.0000        1st Qu.:485.2   1st Qu.: 2.800    1st Qu.:0      1st Qu.:0.0000  1st Qu.: 0.000
 Median :2010-06-01 04:29:30   Mode  :character   Median : 1.638      Median :0.2240        Median :500.2   Median : 3.600    Median :0      Median :0.0000  Median : 2.000
 Mean   :2010-06-01 04:49:30                      Mean   : 2.398      Mean   :0.2774        Mean   :606.3   Mean   : 4.721    Mean   :0      Mean   :0.8924  Mean   : 6.016
 3rd Qu.:2010-08-31 03:44:15                      3rd Qu.: 3.275      3rd Qu.:0.4200        3rd Qu.:728.0   3rd Qu.: 4.800    3rd Qu.:0      3rd Qu.:2.0000  3rd Qu.: 3.000
 Max.   :2010-11-23 05:59:00                      Max.   :15.684      Max.   :1.6740        Max.   :757.9   Max.   :27.600    Max.   :0      Max.   :6.0000  Max.   :93.000
                                                                      NA's   :180           NA's   :180     NA's   :180       NA's   :180    NA's   :180     NA's   :180
```

Here we note the considerable difference in min, max, and mean values between DataWithAnomalies3.txt and the other two data sets. With max values of 300% across most features and mean values of similar difference, it is clear that DataWithAnomalies3.txt is the outlier of the trio. We do no further manual analysis here but we do revisit the fact that we discovered this anomalousness originally with our HMM, but as the NaN result for its log Likelihood was rather confusing to interpret, we felt this was a reasonable step of inquiry. For DataWithAnomalies1.txt and DataWithAnomalies2.txt we rely on our findings from the hidden Markov modeling process alone, as the summary statistics reveal nothing obvious. However, we do note that we proceed under the assumption that DataWithAnomalies3.txt is the most anomalous data set of the three.

Anomaly Detection Conclusions

Ranking the data sets in order from most to least anomalous, we find that DataWithAnomalies3.txt is the most anomalous. One conclusion is to say that, given a log-likelihood of NaN, our model would never predict this sequence of observations. It is somewhat unclear if this analysis is accurate, but the summary statistics bore this out to a large degree. Next, we found that DataWithAnomalies1.txt was the second most anomalous data set, with a normalized log-likelihood score of -0.3510. Finally, in third place, we found DataWithAnomalies2.txt to be the least anomalous of the three data sets, with a normalized log-likelihood score of -0.2809.

Out of interest, we also looked to see how three different models performed against the anomalous data sets. We used our trained 8-state, 24-state, and 40-state models. We predicted that the 8-state model would perform worst, the 40-state model would perform best, but that the 24-state model would perform similarly to the 40-state model.



Above we see the results of our experiment. We note that our predictions were fairly accurate: the log-likelihood values for the 8-state models were considerably lower than the 24 and 40-state models. More notable however is that the 8-state model saw very little difference between DataWithAnomalies1.txt and DatawithAnomalies2.txt. It is worth mentioning however that the 8-state model ranked the anomalousness of the datasets identically to the 24 and 40-state models (finding data set 2 to be the least anomalous).

We also see that the 24-state model performed very comparatively to the 40-state model. From the graph, we can visually see that the difference in log likelihoods between the two data sets for the 24-state model is much the same as the difference found by the 40-state model. The main difference here is that the 40-state model found the two data sets to generally be more anomalous, compared to the test and train data, than did the 24-state model.

The results of this experiment seem to reinforce our belief that the 24-state model still is the preferred model. Having 16 fewer states it is considerably simpler than the 40-state model and performs in a similar manner.

References :

[1] PCA: Using Principal Component Analysis (PCA) for machine learning
https://towardsdatascience.com/using-principal-component-analysis-pca-for-machine-learning-b6e803f5bf1e
[2] PCA: Loadings of weights
http://strata.uga.edu/8370/lecturenotes/principalComponents.html#:~:text=The%20elements%20of%20an%20eigenvector,to%20a%20particular%20principal%20component.