EE 257: Machine Learning

Course Project

"Gas Sensor Array Drift"

Prof. Birsen Sirkeci- Spring 2019

Project Report Date: 05/15/2019

Submitted By

Shubhankar Kulkarni- 013734422

(With Ajitesh Hule - 013755963)

Introduction to the Sensor Drift Problem

Sensor drift refers to the deviation in sensors readings from the actual values over a period of time. As the sensors become old they tend to be affected by several internal and external factors which affect their performance. We try to understand how the sensor drift problem hampers the accuracy of classifying the gases correctly. We implement different types of classifiers and evaluate the performance of the classifiers using different classification metrics. To improve the classification accuracy we also analyse the performance of the ensemble methods.

Dataset Description and Data Cleaning

This dataset uses time series format and has readings taken from 16 gas sensors which are used to classify one of the six gases from ammonia, acetaldehyde, acetone, ethylene, ethanol and toluene. Each gas sensor has 8 features which leads to total of 128 features. To understand the effect of sensor drift the entire dataset has been divided into 10 batches consisting data from different months. Below is a table showing the different months belonging to each batch. There are no missing values in the dataset and the format of the sensor readings of 8 features for each gas sensor was of the form x: value where x is the column number. We removed the colon and the column number in excel and used the cleaned data for processing.

Batch ID	Month ID's
Batch 1	1 & 2
Batch 2	3,4,8,9 & 10
Batch 3	11,12 & 13
Batch 4	14 & 15
Batch 5	16
Batch 6	17,18,19 & 20
Batch 7	21
Batch 8	22 & 23
Batch 9	24 & 30
Batch 10	36

Table 1

Related Work

Paper named “Gas Sensor Drift Migration using Classifier Ensembles” by Alexander Vergara, Tuba Ayhan and Shankar Vembu has used this data set.

- This paper addresses the Gas sensor drift issue by using the ensemble of classifiers approach. SVM was chosen as base classifier on ensemble.
- The observations collected over the period of 36 months were divided into 10 batches each containing almost same number of observations.
- The idea behind using the ensemble method was to address the sensor drift issue by weighted combination of classifiers trained over a period of time.
- Following were the experimental results:

1. For every month multi class classifier was trained with data from previous month and tested on current month. This was able to solve the sensor drift problem up to a certain extent.
2. When the ensemble classifier is tested on more number of batches the sensor drift problem tends to decrease.

MODEL DEVELOPMENT

We used the following classification models to train the data:

1 – Random Forest
2 – Decision Tree
3 – Support Vector Machines

We have evaluated the performance of the above mentioned classifiers on the test data using metrics such as accuracy, precision and recall. To understand the performance of the classifiers and to gain a better insight into the sensor drift problem, we train the classifiers on the previous month's data and test it on the current month data. We have analysed four different scenarios wherein we took different sets of previous month's data to train the classifiers and then evaluate the test error for each scenario. The motive for using different variations of the training data was to find out which set of past data would lead to a more accurate prediction of current or future data.

Details of the four scenarios:
Scenario1: All the past available data was used for training.
Scenario2: Training data from Batch 1 to Batch 7. Most recent data not used for training.
Scenario3: Training data from Batch 4 to Batch 9. Skipping earliest training data when the sensors were new.
Scenario4: Training data from Batch 7 to Batch 9. Skipping some more past data in comparison to scenario3.

Test data used to evaluate the performance of the classifiers is the same for all the 4 scenarios i.e. Batch 10.

Scenario	Training Data Batch Number	Test Data Batch Number
1	1,2,3,4,5,6,7,8,9	10
2	1,2,3,4,5,6,7	10
3	4,5,6,7,8,9	10
4	7,8,9	10

Table 2

Model Performance and Evaluation

1 – Random Forest Classifier

The Random forest classifier is an ensemble method that creates multiple models and then combines them to produce better than results than any one of the single models individually. In case of random forests, several decision trees are combined to form a more accurate model.
We analyse the Random Forest Classifier's performance on all the 4 scenarios. We have used different values of number of trees while training the random forest classifier. As the number of trees increase, the test accuracy increases and it starts to drop for larger number of trees. Scenario 1 & 2 perform the best as more past training data is used to train the classifier while scenarios 3 & 4 perform poorly as only the most recent training data has been used when the sensors had started to drift.

Random Forest	Test Set Accuracies			
Number of Trees	Scenario 1 Accuracies	Scenario 2 Accuracies	Scenario 3 Accuracies	Scenario 4 Accuracies
50	59.822	60.267	55.793	56.432
100	61.073	61.017	56.738	58.044
200	62.101	61.739	57.822	58.572
300	62.267	61.906	57.655	58.627
500	61.323	61.462	57.627	58.266

Table 3

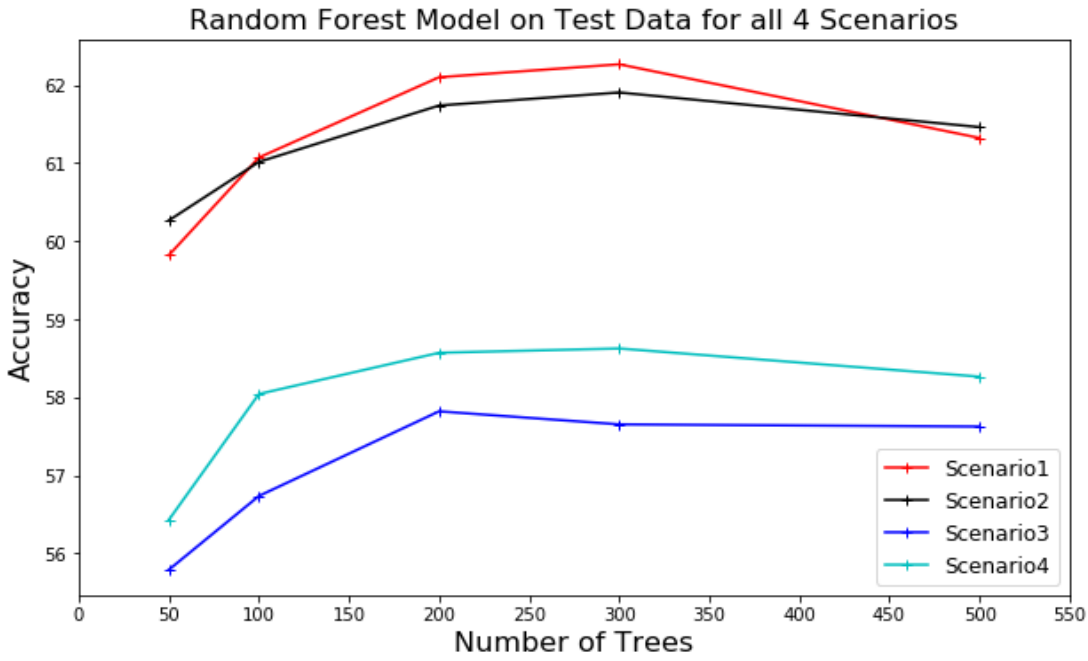

Figure 1

2 – Support Vector Machine

We have used different values of C which is the penalty parameter or the tuning parameter while training the Support Vector Classifier. The tuning parameter controls the bias – variance trade-off, when the tuning parameter C is small we tend to overfit the model on training data which would lead to higher accuracy on the training data but would perform poorly on the unseen test data. Conversely when the tuning parameter C is high, we allow the model to not fit the training data extremely accurately which in turn leads to a better performance on the test data (high bias low variance).

Support Vector Classifier	Test Set Accuracies			
Penalty Parameter C	Scenario 1 Accuracies	Scenario 2 Accuracies	Scenario 3 Accuracies	Scenario 4 Accuracies
1	66.88	65.24	57.627	65.518
10	72.937	71.214	60.795	65.991
50	71.853	70.047	59.767	66.852
100	72.548	70.158	59.739	67.574
500	71.353	69.38	61.573	58.683
1000	70.631	69.519	61.517	58.016

Table 4

As seen in the table above, increasing the value of C leads to a better performance for all the scenarios. However as we keep increasing the value of C, the test set accuracies start decreasing. As we can see in the figure below the test set accuracies for scenario 1 are on the higher side as expected as all the past training data has been used to train the classifier. Scenario 2 performs slightly worse than scenario 1 as only the most recent data from batches 8 and 9 was skipped where the sensors had become old. In comparison scenarios 3 & 4 perform poorly as only data from few of the recent batches where the sensors had already drifted was used to train the classifier.

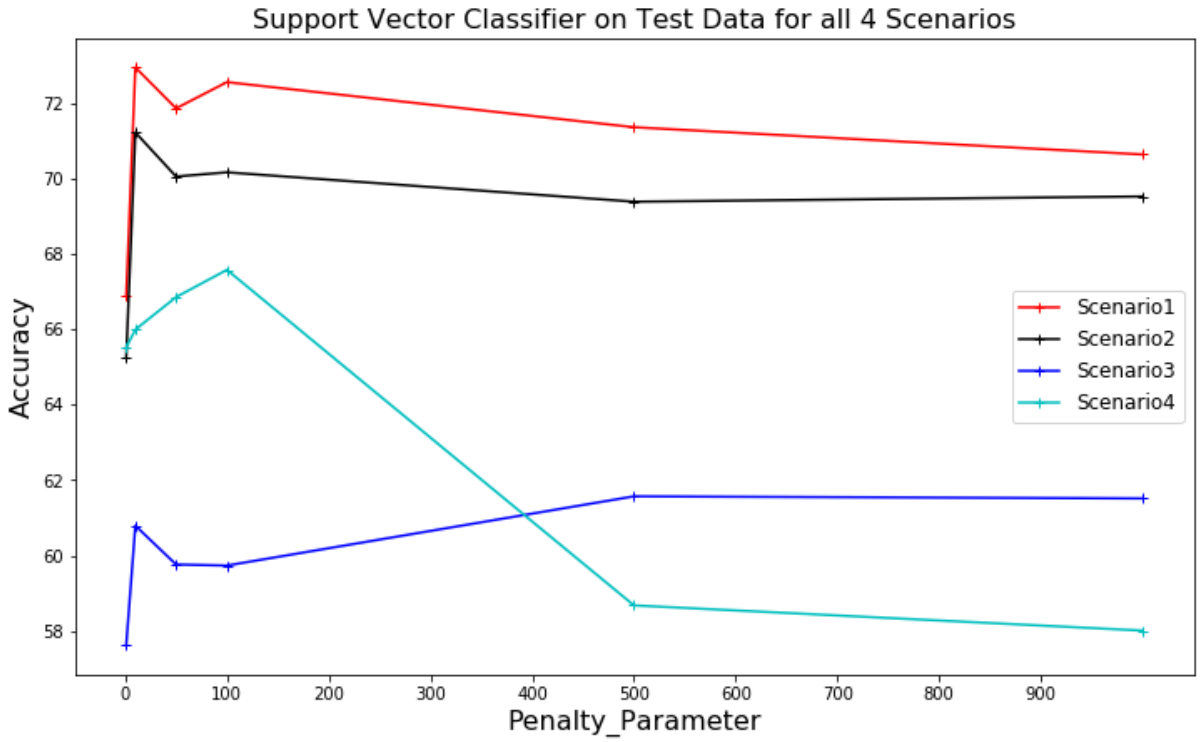

Figure 2

3- Decision Tree Classifier

Decision trees have higher interpretability and perform well on both linear and nonlinear problems. However they are prone to overfitting and perform poorly on smaller datasets. The test set accuracies for all the 4 scenarios are substantially less in comparison to random forests which combine decision trees to form a classifier.

Decision Tree Classifier	Test Set Accuracy
Scenario 1	36.232
Scenario 2	38.65
Scenario 3	41.819
Scenario 4	54.182

Table 5

Solving the Sensor Drift Problem using Ensemble Methods

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. Ensemble of classifiers tend to perform much more accurately than the individual classifiers that make them up. Compared to using a single classifier model for prediction, classifier ensemble methods improve the performance as shown in the table below, provided that the base models are sufficiently accurate and diverse in their predictions.

Ensemble Voting Classifier	Test Set Accuracy	Precision	Recall
Scenario 1	74.493	0.758	0.745
Scenario 2	75.827	0.774	0.758
Scenario 3	68.714	0.716	0.687
Scenario 4	71.714	0.75	0.717

Table 6

Classifiers used for Voting in Ensemble method:
1 – Logistic Regression
2 – Support Vector Classifier
3 – Linear Discriminant Analysis
4 – Random Forest Classifier
5 – Gradient Boosting
6 – K nearest Neighbours Classifier

Hard voting method has been used which uses predicted class labels for majority rule voting.

Fine Tuning the Models and the Feature Extraction

We have applied a dimension reduction technique called 'Principal Component Analysis' which is a popular approach for deriving a lower set of features from our original set of features. In our case we have used n_components as 48 while applying PCA which means that we have derived a lower set of 48 features from our original 128 features.
We trained the Random Forest model using the reduced set of 48 features for different values of number of trees to train the model for all our 4 scenarios. After training the model we then evaluate if our test set accuracies increase in comparison to the Random Forest model which we had fitted with all the 128 features

Random Forest with PCA	Test Set Accuracies			
Number of Trees	Scenario 1 Accuracies	Scenario 2 Accuracies	Scenario 3 Accuracies	Scenario 4 Accuracies
50	65.852	66.574	62.962	65.129
100	64.407	65.324	62.49	65.046
200	63.212	66.324	62.795	64.907
300	62.267	66.88	63.073	64.935
500	61.434	65.49	64.379	65.213

Table 7

The test set accuracies do increase in comparison to when the model was fitted with all the 128 features for all the four scenarios as seen in the table above.

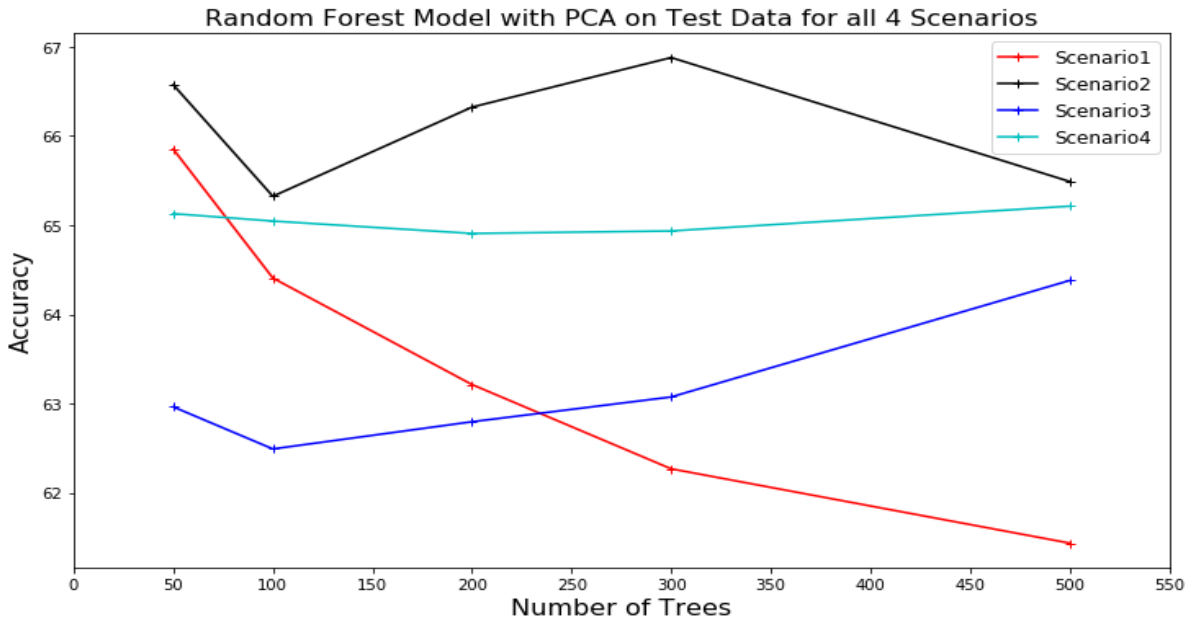

Figure 4

In addition to PCA we have also applied GridSearchCV algorithm which does an exhaustive search over specified parameter values for an estimator which in our case can be any classifier. While applying PCA we give a dictionary with hyper parameter names as keys and list of parameter settings to try as values .The GridSearchCV then gives a set of hyper parameter values from the parameter dictionary which give the best performance on the training set for that model.

REGRESSION

Dataset Description

This data set contains 13,910 measurements from 16 chemical sensors exposed to 6 gases at different concentration levels. This dataset is an extension of the Gas Sensor Array Drift dataset which we have used earlier to classify the gases, provides now the information about concentration level at which the sensors were exposed for each measurement. The resulting dataset comprises recordings from six distinct pure gaseous substances, namely Ammonia, Acetaldehyde, Acetone, Ethylene, Ethanol, and Toluene, dosed at a wide variety of concentration levels in the intervals (50,1000), (5,500), (12,1000), (10,300), (10,600), and (10,100) ppmv, respectively.

Model Development and Performance Evaluation

We have implemented the linear regression model to predict the concentration level of the gas to which the sensors were exposed to for each measurement using the class number and the measurements from the 16 gas sensors each having 8 features which results in a 128 feature vector.
Following models were implemented:

1 – Linear Regression
2 – Ridge Regression with Cross Validation
3 – Lasso Regression with Cross Validation

We have used the Mean Squared Error as a performance metric to evaluate the performance of models used for regression. Similar to the combinations used for evaluating the classification models we have used 4 sets of training and test data combinations to train our models and then evaluate the accuracy on the test set.

Scenario	Training Data Batch Number	Test Data Batch Number
1	1,2,3,4,5,6,7,8,9	10
2	1,2,3,4,5,6,7	10
3	4,5,6,7,8,9	10
4	7,8,9	10

Table 8

The Mean Square Error for all the regression models for all the scenarios mentioned above are as shown in the table below

Model	MSE for Scenario 1	MSE for Scenario 2	MSE for Scenario 3	MSE for Scenario 4
Linear Regression	48416.842	50888.144	54324.85	35448.889
Ridge Regression with Cross Validation	23272.385	22357.002	42612.169	32636.154
Lasso Regression with Cross Validation	27248.613	25964.589	34351.919	25624.056

Table 9

Similar to classification models, performance of all the models for scenarios 1 & 2 are better in comparison to the rest.

Conclusions: Ensemble Methods to Solve Sensor Drift Problem

The Ensemble method voting classifier gives the best performance amongst all the models. Test accuracies for the ensemble method classifier are greater in comparison to the random forest, decision tree and support vector classifiers. Support Vector classifiers perform better than random forest classifiers for all the scenarios.

In terms of the different training and test set combinations used for evaluating the performance of different classifiers, all the classifiers give a better performance when gas sensor readings from all the previous months are used to train the classifier. Scenarios 1 and 2 use majority of previous month's gas sensor readings to classify gases on the test data resulting in better performance of all the models for these 2 scenarios. Conversely when less amount of previous month's data is used or more importantly when the data from months where the gas sensors were new is skipped, the performance of all the classifiers degrades considerably. We can conclude that when all the previous months data is used to classify the gases, test accuracies for classifying the gas are on the higher side. This stems from the fact that when we take all the previous months' available data, it includes readings taken by the gas sensors when the gas sensors were new and had not started to drift.
Even if we skip some of the most recent month's data and ensure that data when the sensors were new is included we get test accuracies slightly lower than when we take all the previous data for training the model. On the other hand when sensor readings from months where the sensors were new and had not begun to drift is not included in training the classifier , the test set accuracies on classifying the gases is considerably low for all the classifiers.

Thus the use of ensemble methods such as ensemble voting classifier and the efficient use of training data can lead to better classification of gases.

References

1 - “Sensor Drift Compensation in Time Series Prediction through Regularized Ensemble of Classifiers”, International Journal of Advanced Research in Computer and Communication Engineering, Vol 4, Issue 2, February 2015.

2 - https://archive.ics.uci.edu/ml/datasets/gas+sensor+array+drift+dataset