Anomaly Detection with Isolation Trees using the Numenta Anomaly Benchmark Data

Nathen Byford

Contents

1 Introduction

Many modern data pipelines rely on data streamed at high rates to make near instantaneous decisions for business needs. One of the largest issues among all areas of data collection is the identification and understanding of anomalies in real-time data. As organizations utilize the insight of real-time data for informed decision-making, the potential impact of anomalies or deviations from expected patterns or behaviors, cannot be understated.

There are many possible causes for anomalies in data. Whether caused by errors, fraud, or unforeseen events, anomalies possess the capacity to significantly influence the integrity and reliability of insights derived from data. Identifying anomalies is the first step to determining the cause of the anomaly. Knowing the cause of the anomaly can help determine if the abnormalities should be included in the analysis of the data and the decision-making process.

2 Data

The data for this project comes from the Numenta Anomaly Benchmark data set, specifically the twitter tag count portion of the data set. This portion of the data has 158,631 observations of twitter tag counts every 5 minutes from Feb. 26, 2015 to Apr. 23, 2015. These observations are split between 10 companies, Apple, Amazon, Salesforce, CVS, Facebook/Meta, Google, IBM, Coca-cola, Phizer, and UPS. For each observation there is a logical assigned, true if it is an anomaly and false otherwise.

2.1 Exploritory Data Analysis

In figure 1 the time series of each company with the identified anomalies overlaid as Xs is shown. Looking at the time series' in figure 1 it is clear that there is an imbalance problem in the data, as there are far more non-anomalous points compared to the number of anomalous points. This leads to the possibility of the models performing extremely well by determining that all points are normal. In anomaly detection there is often a high false positive rate due to the nature of anomalies being uncommon. Another observation of the time series' in figure 1 is not all peaks are anomalies. This points to something else being present in the decision making process of a true anomaly, this is an inherant problem with univariate time series anomaly detection. There are often more variables necessary to be consider for identifying true anomalies versus outliers.

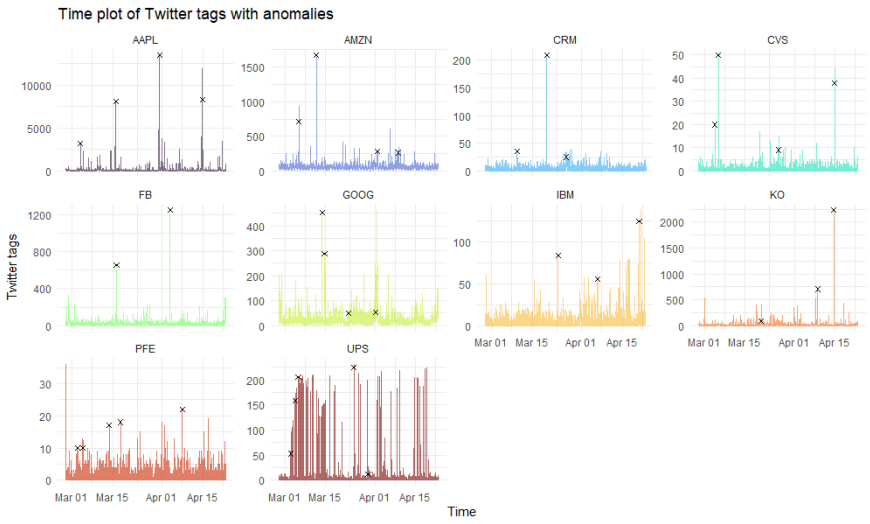

Figure 1: Plot of each company time series

3 Methods

In this report the performance of four anomaly detection algorithms/methods will be compared. These methods are: linear regression, seasonal decomposition of time series by Loess, neural networks, and isolation forests. These models can be separated into two groups, statistical models and machine learning models. Some of these models are post hoc and intended to be performed after the data is collected, others can utilize cross validation techniques to be trainied on different data than the decision making data. All models will be trained on the first half of the time series if necessary and all test measures are from the second half of the time series to maintain the time dependence structure.

3.1 Statistical models

The following models are considered statistical models, one is more of a baseline simple model and the other is a standard method for anomaly detection.

3.1.1 Linear regression model

Anomalies are similar to outliers in statistics, so why not use outlier detection methods to identify anomalies? When using a simple linear regression model we test for points with high leverage that have a large impact on the coefficients of the model. Here we can identify high leverage points from a simple linear regression on the time series. Once points have been identified as high leverage points using Cook's distance, covarince ratio, or DF beta measures, they are labled as anomalies. This method is a simple yet surprisingly effective way of identifying anomalies in time series data. This method is post hoc and is applied to the full data

after it has been collected to identify anomalies, here it will only be applied to the testing data.

In figure 2 an example of this model is shown. The blue line is the simple linear regression, the green points are identified outliers/anomalies using this method, and the red x is a true anomaly from the data set. We can see that there are many more anomalies identified than there truly are, but the methof does correctly identify the true outlier in this case. In a typical linear regression we are interested in the coefficients of the model, here we are not interested in the coefficients. The focus of this model is a byrpoduct of the diagnostics for the regression line fit.

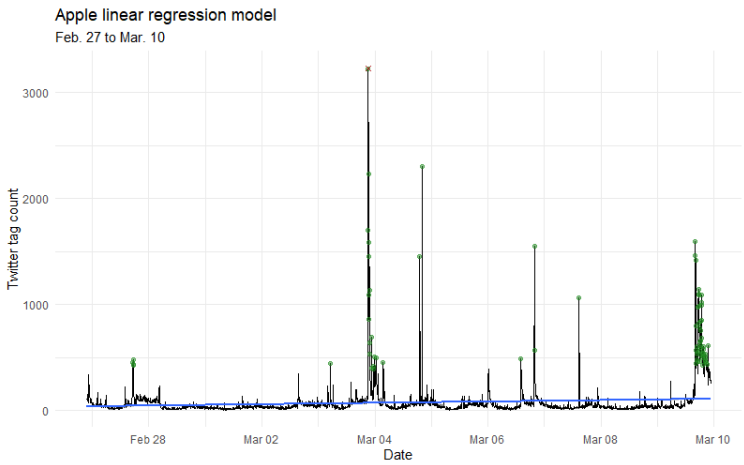

Figure 2: Example linear model

3.1.2 Seasonal decomposition of time series by Loess (STL)

A common method of anomaly detection with time series data is to utilize the seasonal decomposition of time series by Loess or STL. The STL is as its name suggests a decomposition of the time series by its trend component, seasonal component, and some residual or remainder component. This can be seen in figure 3. For anomaly detection we can place a bound on the remainder portion of the STL, the bottom time series in figure 3, and determine that any remainder outside the these bounds is an anomaly or deviation from the norm. This would be an indication that the values observed are further than expected from the seasonal plus trend components of the time series.

Similar to the linear regression model, the STL model is post hoc and uses the full data, as the classification comes as a byproduct of the model fit to the complete data. The coefficients of the seasonal and trend portions are not if interested. The package that contains the function in R for STL anomaly detection is `timetk`.

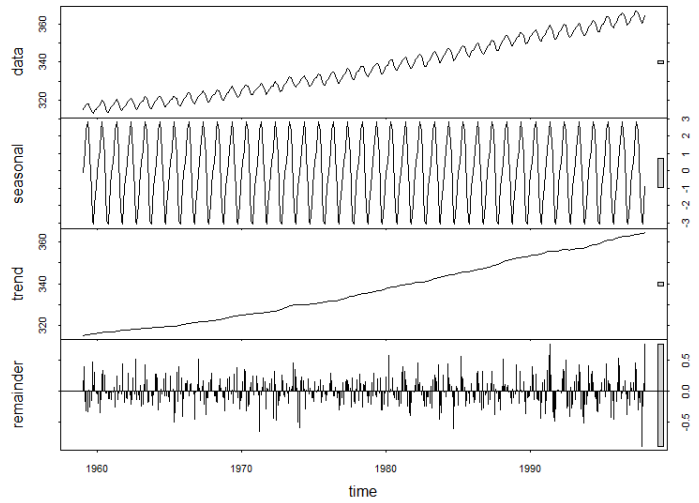

Figure 3: Example plot of STL

3.2 Machine Learning methods

The other class of methods used in this project are machine learning methods. The following are the machine learning models used for the anomaly detection of the twitter data in this project.

3.2.1 Artificial neural network

Neural networks try to replicate how the human brain thinks when classifying data. Neural networks can identify connections that other methods have difficulty finding, but the downside is they are more of a black box method where we don't really know what is driving the decision. Using the package `ANN2` in R for artificial neural networks for anomaly detection we can make a classification network based on stochastic gradient descent using a log loss function. This method is designed for anomaly detection, using cross validation the best parameters for the model are identified to be 1 hidden network as more or less result in a model guessing every value is non-anomalous. It is important to note that this mothod is not specifically designed for time series and the time stamp is needed to be converted to a numeric value for this method to work. More work is being done for time series AI, but it is not yet ready for use.

3.2.2 Isolation forest

The last method being compared is isolation forests, this method is similar to random forest and regression trees with a different goal and measure. Like tree models the data is split it into partitions as seen in figure 4, the data will continue to be partitioned until all points are the only member of their own partition. The

data are split randomly by variable and value, this is why an ensemble of these isolation trees is utilized. As each point gets its own partition we count the number of splits necessary to isolate the point, this is the isolation depth of the point for the tree. In the case of the point X_j in figure 4 it has an isolation depth of 4, it takes 4 splits to isolate the point. The basic assumption of isolatin forests is that it will be easier to isolate anomalous points as they are not the norm. Using the isolation depth of each point for many trees in the ensemble, we take the average depth of each point and that is the "anomaly score". Placing a threshold on the anomaly score we are able to classify the anomaly status of each point.

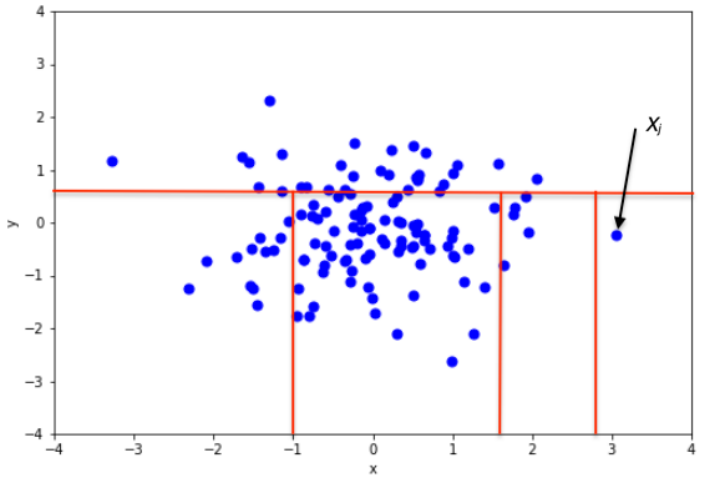

Figure 4: Example of data partition for isolation forest

There are other possible metrics to determine the anomaly score of the partitions, there is the average of the isolation depth as mentioned and another option is a ratio of the variables on each side of the split. This second measure is called the density anomaly score. The density isolation forest tests to preform better for categorical variables according to the liturature. Both mothods will be implimented and compared in the analysis of the twitter tags data. Both methods will be done using the `isotree` package in R.

4 Results

For each model and time series the area under the ROC curve (AUC) was calculated to compare each model. AUC makes it possible to see how the model compares to guessing the same class for every point, such a model would have an AUC of 0.5. Higher AUC is indicative of a better model. In table 1 we can see the resulting AUC for each model and time series. In this table the best test value for each time series is in bold. The first thing to note is that no model was able to do better than guessing for the UPS time series and most models preform worse than guessing. The model with the highest AUC on the most time series was the STL

anomaly detection method, it performed best on the Amazon, Salesforce, and Phizer data and has high AIC for Apple, CVS, Facebook/Meta, and Coca-cola as well. We can also see that another problem time sereis is Google. All but one method has an AUC of .5 or less, the only model to perform well on Gogle was the density isolation forest. Looking at the other AUC values of the density isolation forests it performes very well with all time series having an AUC of .96 or more excluding UPS.

Table 1: AUC results of each model

Model	AAPL	AMZN	CRM	CVS	FB	GOOG	IBM	KO	PFE	UPS
lm	0.9962	0.9888	0.9814	**0.9899**	0.9945	0.4851	**0.9899**	0.9968	0.9790	0.4939
stl	0.9554	**0.9920**	**0.9892**	0.8667	0.9868	0.4839	0.9869	0.9843	**0.9869**	0.4817
SGD	**0.9994**	0.5000	0.5000	0.5000	**0.9999**	0.5000	0.5000	0.5000	0.5000	**0.5000**
IF	0.9967	0.5000	0.5000	0.7497	0.5000	0.4999	0.5000	**0.9991**	0.5000	0.4981
Density IF	0.9665	0.9859	0.9713	0.9611	0.9905	**0.9736**	0.9746	0.9912	0.9758	0.4888

Another important metric for this case is the true posetive rate, since there are so many false positives we really want to see that the model gets at least the couple of anomalies correctly. In table 2 the true posetive rates of each model for each time series is listed. Again, here we see that none of the models were able to identify the anomalies in the UPS dataset. This is most likely due the anomalies that are identified in the data set. For the UPS and Google time series, looking at figure 1, the anomalies are different as they are not any of the peaks in the series. This is where we begin to see that the detection of anomalies cannot only be based on the time series of values, but must be used in conjunction with other data to identify anomalies.

Table 2: True positive rates

Model	AAPL	AMZN	CRM	CVS	FB	GOOG	IBM	KO	PFE	UPS
lm	1	1	1	1	1	0	1	1	1	0
stl	1	1	1	1	1	0	1	1	1	0
SGD	1	0	0	0.5	1	0	0	0.5	0	0
Isolation Forest	1	0	0	0.5	0	0	0	1	0	0
Density Isolation Forest	1	1	1	1	1	1	1	1	1	0

5 Conclusion

Based on the results discussed above, it is difficult to determine what model is truly the best. Since the best overall model based on AUC is the STL method, but the best in terms of true positive rate is the density isolation forest. The overall best model depends on the criteria that maters most, in anomaly detection it is far more important to have a model that identifies the true anomalies more often, so the best model is the density isolation forest model. Using density isolation forests, it is possible to build a reliable model to identify anomalous points if the anomalies are identified to be abnormal points or outlies as one would expect.