Outlier Detection and Analysis

Lindsey Curtis & Scott Onestak

*Using real and synthetic Yahoo! production traffic to determine the robustness of outlier detection algorithms*

**Introduction**

**Background and Motivation**

When attempting to learn from data, it is important to identify observations that deviate substantially from other data. Anomalies can result from many different causes, and understanding origins can significantly affect the way data is interpreted. Outliers may present themselves as a result of incorrect or bad data, in which case it is often beneficial to identify and deal with outlier cases individually or remove them from the data set.

In today’s digital world, it is not only important to be able to identify outliers, but be able to do so in real time. As real time outlier detection becomes more accurate, many important, everyday systems become more reliable. If banks can detect outliers of credit card purchases more efficiently in real time, less credit card identifications are able to be stolen for use. If websites and applications are able to determine outliers more quickly and efficiently, more news stories are able to be written and spread to the people faster. Additionally, if intelligence agencies are able to determine outlier behavior in real time, more tragic events, such as terrorism, have the possibility of being thwarted.

In our project, we plan on displaying a number of various methods that can be used when handling outlier detection, as well as comparing their efficiency and applicability to different situations.

**What Makes an Anomaly?**

The Merriam-Webster Dictionary defines an anomaly as “something that is unusual or unexpected.”[[1]](#footnote-1) In terms of data mining, anomalies are defined as a “data object that deviates significantly from the rest of the objects, as if it were generated by a different mechanism.”[[2]](#footnote-2)

The types of outliers can be classified into three different subcategories, the first of which being the global outlier. This is a data object that “deviates significantly from the rest of the data set.”[[3]](#footnote-3) Often with global outliers, the challenge is determining the appropriate measure from which an object is determined an outlier or not.3

The next type of outlier is the contextual outlier. These are outliers that may not necessarily deviate from the entire data set as an outlier, but with respect to its location in the data set, makes the data point an outlier.[[4]](#footnote-4) For example the observed high on December 11, 2015, was 62 degrees Fahrenheit. In the entire set of data points for the year, 62 degrees would not be considered an outlier. However, it is extremely rare to have a day this warm in December, so the data point may be an outlier relative to the other data points around it. The challenging part about detecting contextual outliers is that they are subjective to the selected context.4 Therefore, depending on the context in which the data points are analyzed could change whether a data point is classified as an outlier or not.

The last classification of outliers are collective outliers, where “objects as a whole deviate significantly from the entire data set.”[[5]](#footnote-5) Collective outliers add a new challenge to anomaly detection because now groups of outliers that may not individually be outliers must be detected based on the group behavior, which often requires background knowledge of the relationship among data points.[[6]](#footnote-6)

**Data Acquisition**

To test our methods, we acquired Yahoo!’s Labeled Anomaly Detection data set that includes real and synthetic Yahoo! production traffic. Released early in 2015, the data set brands itself as being the first of its kind to be a data stream data set with labeled anomalies.

This data set includes four subsets of the data, the first being the A1Benchmark. This subset of data includes 67 files of real data where each timestamp contains a value specific to the amount of traffic on that site for the interval of one hour. Each of these data points is then labeled whether it is or is not an anomaly. Each data point in the A1Benchmark data was labeled an anomaly by a human being, so the authors of the data set warn that this data may be best suited to measure recall since the labeling of the anomalies may be subjective to the labelers of the data.

The next of the four subsets is the A2Benchmark data, which consists of the same formatting as the A1Benchmark data, but now the data is synthetic. Each data point in the time-series of the 100 different test files now has a random seasonality, trend, and noise, with outliers inserted randomly into data set.

The A3Benchmark data set only contains outliers in its 100 files, excluding change-points that would be classified as anomalies. The data attributes have varying noise and trend within the data, and like previous data set, include anomalies at random positions in the data.

Finally, data attributes have varying noise and trend in the A4Benchmark data set. However, change-point anomalies have been included randomly as outliers in addition to the other types of outliers in the 100 testing data sets.

**Performance Evaluation**

Because the Yahoo! data set is the first of its kind to label the anomalies in the time series data, the data is now analyzable from the accuracy of the algorithms used. Therefore, not only will runtime be available to analyze, but now the recall, precision, and F-score of each data set will be analyzable in order to determine the most efficient and effective algorithms to detect the outliers.

**Methodology and Experimentation**

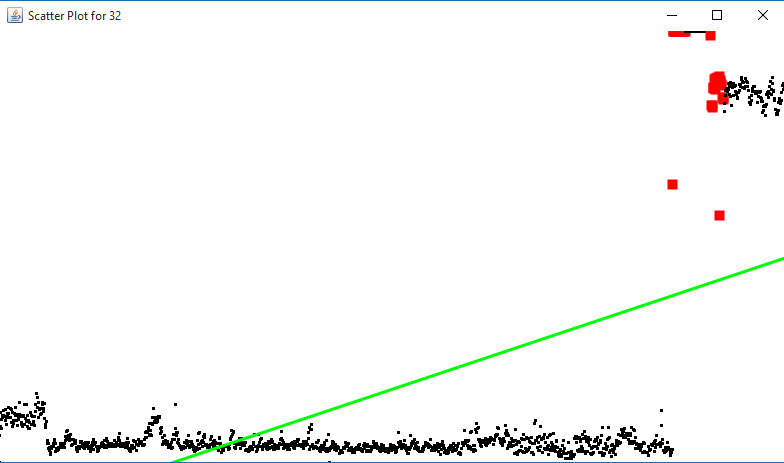
Our methods include the use of algorithms from many areas of outlier detection methodology. These included partition-based clustering, regression analysis, moving window methodology, and autoregressive modeling.

**Linear Regression**

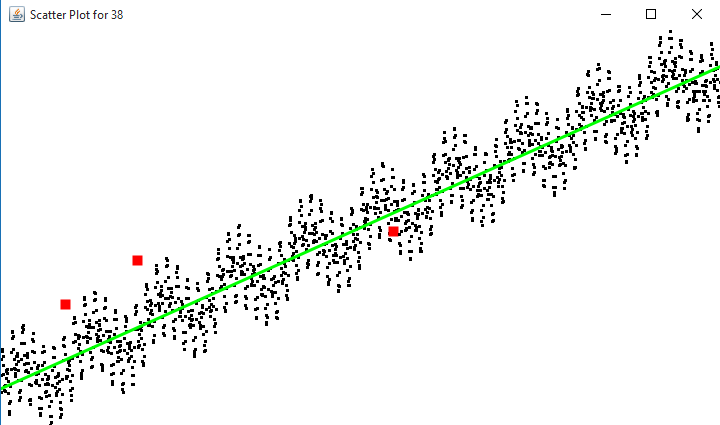
The first of these, used as a base case from which to compare the algorithms to, was a simple linear regression where the values were regressed onto the timestamps to form the linear regression model. From the model, the residuals of the data points, or the distances between the model prediction and the actuality of the data point, were calculated. Based on the residual distances, outliers can be detected if they reside more than three standard deviations from the mean of the average residual, which is zero because the primary condition on which the Ordinary Least Squares model is constructed is that the sum of the residuals be equal to zero.

The advantages of the linear regression model are few. One advantage would be that the linear regression model is fairly easy to construct, so the algorithm should run fairly efficiently. Additionally, the algorithm should do an accurate job of detecting global outliers.

However, there are many disadvantages to the linear regression model. The first of which is that the model assumes the data is linearly related. Often in the data sets, the data did not display any linearity at all, but a cyclic or even no relationship between the timestamps and the values, as can be seen in the plot for file 32 of the A1Benchmark data where true positives are purple, true negatives are black, false positives are orange, and false negatives are red .



Additionally, the linear regression model assumes that the variability in the residuals is consistent throughout the data. Therefore, even if the data did appear to exhibit a linear relationship between the timestamps and values, if the residual values were not evenly spread throughout the data, or displayed heteroscedasticity, then the model may still not be able to correctly predict the outliers. This can be seen in the plot for data set 38 of the A3Benchmark data. Though the model exhibits a linear relationship between timestamps and the values, there is a cyclical variation around the regression line. Therefore, the values vary cyclically with timestamps as well, and the model is unable to predict some of the outliers.

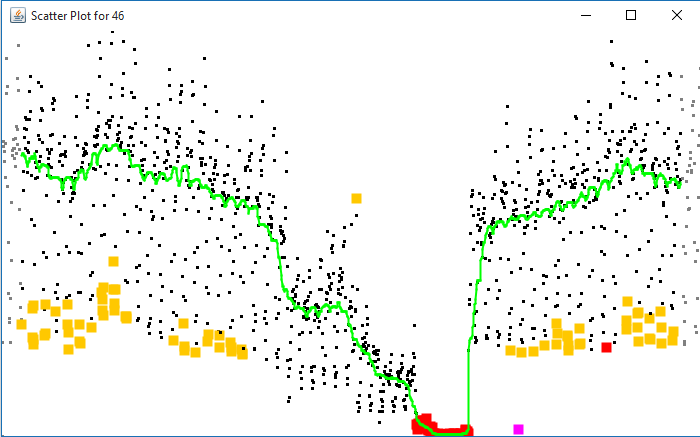


**OutlierMAD (Median Average Deviation)**

The next algorithm implemented was OutlierMAD, which introduces the concept of a moving window. In order to implement this algorithm, the algorithm was transposed from the R-package Pracma to Java so that the algorithm could be implement for the project.[[7]](#footnote-7) The algorithm uses the concept of the moving window by finding the median value of the objects within a specified range of the object being analyzed. If the data object deviates further from the median of the set of data objects than the test statistic, then the object is classified as an outlier.

The OutlierMAD algorithm has its advantages. The first of which being that the measurements will be robust to outliers. The median, as opposed to a mean, will not be influenced by the existence of an outlier in the range used for calculating the median. Additionally, the introduction of a moving average allows the estimate for each data point to continuously move up and down dependent on the data. Therefore, unlike linear regression, the assumption of linearity or any shape of the model is no longer necessary to more effectively analyze the data.

However, the OutlierMAD did have its drawbacks as well. The first of these is that the model cannot detect collective outliers because all the outliers would be in the same range, so the element being analyzed would not deviate from the values next to it because they are also outliers. Additionally, this model had a tendency to overfit the data and did not work well on the real data. This can all be seen below in the plot for file 46 of the A1Benchmark data.



Furthermore, the fact that the algorithm detect outliers using data objects on both sides of the point being analyzed makes this algorithm fairly ineffective in the real world. In the real world, credit card companies cannot wait for the next few purchase in order to detect outliers that indicate fraud. This concept also extends to all other areas of outlier detection in time-series data. The information to detect an outlier is necessary when the event happens because the information is time sensitive.

**Median Average Approach**

Building off the concept of the OutlierMAD method, we set out to develop a median-based moving window approach that determined whether an element in the data set was an outlier without having to analyze data points on both sides of the element. The result was this median based approach that only used data from previous data points.

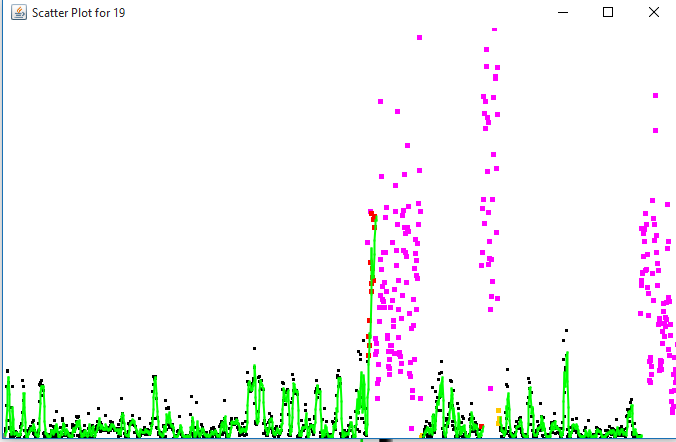
While the algorithm did tend to prevent overfitting of the data better as well as detect more outliers, two fundamental problems still persisted. The first was that the algorithm could still not detect collective outliers. Secondly, not all previous values in the range for analyzing should be weighted the same. Data points closer to the one being analyzed should be weighted more than points further away. In using medians to analyze the data, it becomes relatively impossible to effectively use a weighting system for the objects because it would move the estimation fairly little. This is due to the fact that we have to select one of the points in the range as the expected value. It would be better to have a prediction for the value that could change with each weight differential.

**Mean Average Approach**

In order to attempt to solve for the weighting problem of the median average approach, we attempted to implement a means-based approach so that the predicted values would be more responsive to the weights. In doing so, the means approach presents new challenges that must be overcome.

The first of which is that a mean estimation point containing an outlier will greatly influence the expected value of the data points the outlier is being used to predict. Often times, these points will incorrectly be classified as outliers themselves due to the inclusion of this variable. In order to handle this issue, a recalculation method is implemented when the outlier is detected. This method recalculates the expected value of those data points, excluding the outlier in the estimation.

Using this technique of removing outliers from the estimation also provided the ability to detect collections of outliers as well. If a calculation for the expected value is done in which there are no data points in the range to detect new outliers, then it is assumed the algorithm has found a collection of outliers and classifies that point as an outlier as well. Though the algorithm still needs more work in order to become more robust to collections of outliers, it can be seen that the algorithm can work well in detecting outliers and collections of outliers it was not able to detect before. Below is file 19 from the A1Benchmark data set, which is the data set that contains the real data.



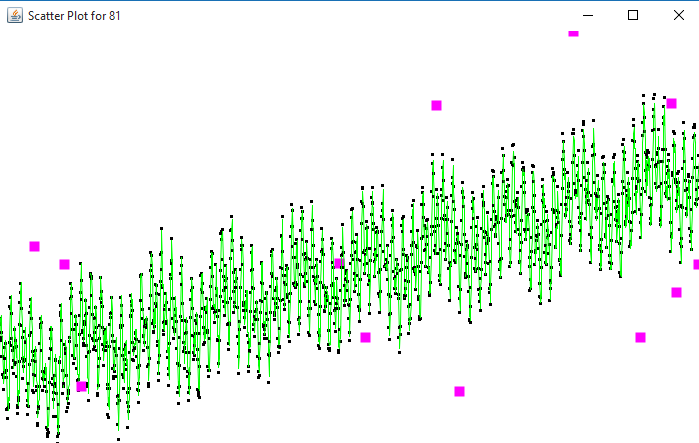
However, the algorithm has some disadvantages of its own. To begin, it now classifies every data object after the last outlier in the collection of outliers as an outlier until it reaches the number data object one greater than the range used to calculate the mean estimates. Additionally, the results are subjective to the size of the range used. Typically, the range used is small, so there is more variability in the predictions, which may cause the algorithm to overlook some outliers. On the other hand, if the range becomes too large, the algorithm may smooth out, have a larger variability in residuals of the expected versus reality and miss collections of outliers, such as the ones shown above. Therefore, there is a double-edged sword when determining how many data objects you wish to use in order to predict the next point.

**Autoregression Approach**

The next approach was to attempt an autoregressive model. Many models of times series data, including Twitter’s own algorithm to detect outliers, at least in part use an autoregressive model to determine outliers. The assumption made by using this model is that the previous value or values are good predictors of what the next value in the time series will be.[[8]](#footnote-8) Therefore, the previous value or values can be regressed onto the next value in order to form a model that predicts the next value.

This works better than the mean average approach in predicting the next value. This is because the mean average approach’s estimate must lie somewhere between the largest and smallest value of the estimators. However, using the autoregressive model can predict a value outside the range of values. This is because the model would look at the trend of the data. If data points are trending upward, then the model would be able to predict a value greater than that of the largest value in the range, and vice versa for a downward trend.

Therefore, an AR(1) model, in which only the previous data point is used to predict the next value, was constructed for the outlier detection. The method proved robust in detecting contextual outliers all previous methods had missed, especially in the A3Benchmark data sets.



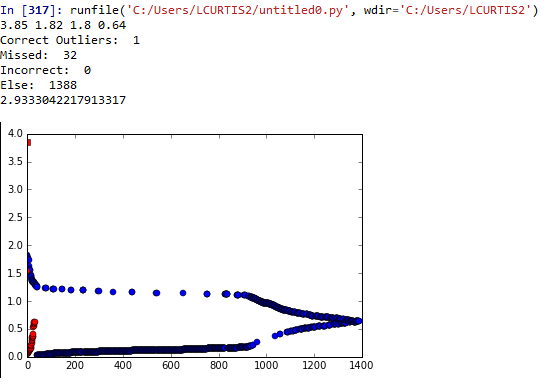
However, the AR(1) model also had its disadvantages as well. Since the model only uses the previous value in order to predict the next, the AR(1) model would not be robust to a collection of outliers. Furthermore, the algorithm did not appear to work as well at detecting outliers when heteroscedasticity was present in the data, such as many data sets in the A4Benchmark.

**K-Means Approach**

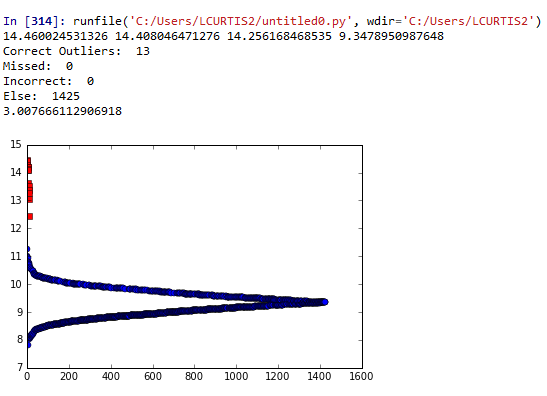
Another implementation we created followed from the concept of k-means clustering and attempted to improve upon regressive techniques. The algorithm implements an initial k-means partitioning, a concept that proves useful when working with Euclidean distance evaluation to cluster data. Ideally, clusters could be used in the identification of outliers by enabling the algorithm to finalize an outlier cluster as a result. For our experimental design, the clustering analysis is used to provide an overall average through the number of clusters taken at random points during the k-means process. The improved number taken from the iterative k-means clustering is provided to check each data point against, where they are sorted by distance. To choose outliers, a distance threshold based on this range was chosen, and data points were labeled accordingly.

For the initial implementation over 20 datasets, a threshold level of 50 percent was chosen arbitrarily, labeling of any data more than 50 percent away from the improved average as an outlier. The result was an array of detection abilities, focused on the 20 datasets to show how different sets affected the algorithm. A few of the datasets returned perfect detection, with no false positives or false negatives. About half of the sets returned average results, but adjustment of the threshold to a higher or lower percentage significantly improved the results of the data.

For a few of the sets, the detection method was not applicable to return helpful results due to the nature of the data. Provided is an example of a dataset distribution that is not conducive to the algorithm:



The true negative exists as a red square at the edge of the graph, while the points represented as red circles are false negatives. Compared to the next dataset:



In this set, the outliers can be easily identified due to their location outside of the normal distribution of the rest of the data.

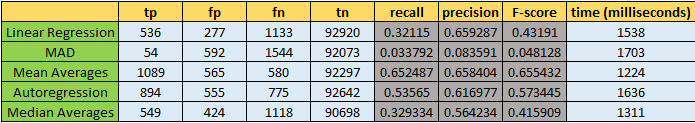
Overall, the benefits of such a clustering technique could be applied more effectively to two-dimensional data sets, while it attempted to provide a more applicable average that is less affected by outliers. However, in the datasets which provided outliers that are significant distance from the rest of the data, the technique proved to be very effective.

**Results**

From the algorithms described, the true positives, false positive, true negatives, and false negatives were collected as well as runtime. From this, precision and recall could be computed for each of the subclasses of the data sets to compare classification performance using an F-score.

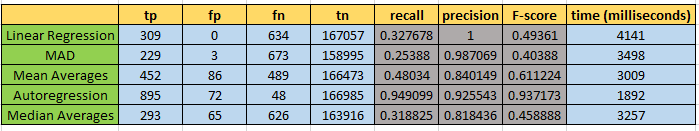
For testing purposes, the k-means data was only tested on a subset of the A1Benchmark data set, which was the real data. This was due to the longer runtime as well as the implementation of adjustment to the threshold for more accurate results.

For the datasets applied to all the testing files provided by the data set, the mean average approach worked best at detecting outliers in the A1Benchmark data subset, as shown below.



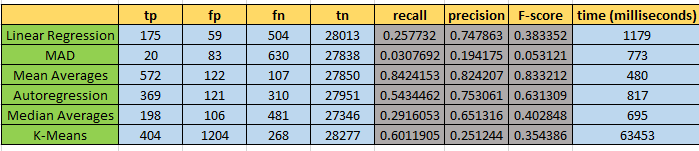
When looking at the types of data sets provided in this subset, it is easy to see why this is the case. The mean averages algorithm is more robust detecting collections of outliers, so it was able to detect more of the large groupings of outliers prevalent throughout the A1Benchmark data subset.

However, when you look at the rest of the subsets, the autoregressive model outperformed the other methods by a considerable margin. Below is the results from the A3Benchmark data set.



This phenomenon occurred because there were very few collections of outliers in the other data subsets. Therefore, the autoregressive model’s ability to detect contextual outliers better than the other algorithms allowed it to perform better on the other data subsets.

For the analysis of k-means with the rest of the algorithms, we restricted the data files to a sample of the first 20 A1Benchmark data sets due to the longer runtime and attempt to make the algorithm more robust to detecting outliers. The results below show k-means was the second worst algorithm on the dataset, which may be attributed to the lack of dimensionality in the dataset.



**Conclusions**

Of the algorithms implemented, the autoregressive model appeared most robust across all the datasets because it was able to better detect contextual outliers. Yet, the mean averages was able to better detect outliers in the data containing real values because it was more robust to the collective outliers present in the data.

Overall, the OutlierMAD appeared to be the worst performing algorithm, especially on the data set containing real values. Linear regression, median averages, and k-means appeared to perform relatively equal in their abilities to detect outliers in the data set, but k-means had the longest runtime by far. The other algorithms were able to run on the entirety of the testing files in the subset in the time it took k-mean to run on one test file.

However, it is important to note that k-means would probably become more robust as an algorithm in detecting outliers as the number of dimensions grew larger. Since the data set is one dimensional, there is very little room between clusters. As the dimensionality would grow, the data would spread out, and it would be easier for the algorithm to determine which data objects are similar and which are outliers.

Since the testing only implement an AR(1) model, future testing on this data set would need to include the addition of even more previous data points in predicting the current one. With more data points, more information may be able to provide more accurate results and detect more outliers than the AR(1) model does.

Based on the findings of this research, future research may be needed in a hybrid method of the autoregressive model and mean averages. This is because the autoregression seems to be robust in detecting contextual outliers while the mean average is better at detecting collective outliers. Combining these two methods into one would be able to detect more outliers, but could possible be offset by detecting more data points which are not outliers.

In short, analysis of various outlier detection methods has highlighted various types of outliers and how different methods may detect them. Certain algorithms perform functions more effectively than others do, and an understanding of the benefits and costs of each can provide the opportunity to employ them in the most effective environment.

**References**

"14.1 - Autoregressive Models." Regression Methods. 2015. Accessed December 13, 2015. https://onlinecourses.science.psu.edu/stat501/node/358.

Han, Jiawei, Micheline Kamber, and Jian Pei. "Outlier Detection." In *Data Mining Concepts and Techniques*, 543-584. 3rd ed. Morgan Kaufmann Publishers, 2012.

"SCM Repository." R-Forge. Accessed December 13, 2015. https://r-forge.r-project.org/scm/viewvc.php/pkg/pracma/R/outlierMAD.R?view=markup&root=optimist&pathrev=5.

*Merriam-Webster Dictionary*, s.v. “anomaly,” accessed December, 15, 2015, http://www.merriam-webster.com/dictionary/anomaly.

1. 2006. Anomaly | Definition of anomaly by Merriam-Webster. http://www.merriam-webster.com/dictionary/anomaly.. [↑](#footnote-ref-1)
2. Jiawei Han, Micheline Kamber, and Jian Pei. “Outlier Detection.” In *Data Mining Concepts and Techniques*, 544. 3rd ed. Morgan Kaufmann Publishers, 2012. [↑](#footnote-ref-2)
3. Ibid., 545. [↑](#footnote-ref-3)
4. Hans, “Data Mining Concepts,” 546. [↑](#footnote-ref-4)
5. Ibid., 547. [↑](#footnote-ref-5)
6. Ibid., 548. [↑](#footnote-ref-6)
7. "SCM Repository." R-Forge. Accessed December 13, 2015. https://r-forge.r-project.org/scm/viewvc.php/pkg/pracma/R/outlierMAD.R?view=markup&root=optimist&pathrev=5. [↑](#footnote-ref-7)
8. "14.1 - Autoregressive Models." Regression Methods. 2015. Accessed December 13, 2015. https://onlinecourses.science.psu.edu/stat501/node/358. [↑](#footnote-ref-8)