Smart Outliers Detection

Elad Bilman and Yedidya Bacher

Department of Computer Science, Bar-Ilan University

Project is part of course: Tabular Data Science 89547, Dr. Amit Somech

Abstract

In Data Science, the errors or "noise" in a data-set are often referred as outliers. Those outliers may carry important information, as such outliers are a major candidate for abnormal data that may lead into misspecification in prediction. In this project we try to address this problem by detecting the outliers and removing them from the data-set, by creating a new Outlier Detection algorithm which combines known outlier detection algorithms. Therefore, we have a baseline approach in which we will compare how our algorithm affects the prediction model accuracy against know algorithms.

1 Introduction - Problem Description

In data-sets more often than not we have errors/noises in the data, as such this points can influence greatly our prediction and therefore mislead our model into false prediction or improve it. For this reason in this project we try to improve the Data Cleaning element in the Data Science pipeline, as this element suffers from no clear and broad solution to the effects of outliers. In particular one problem is that although there are efficient and proven outliers detection algorithms, their effects on the model are greatly dependent on the data-set itself, which makes it harder to identify the outliers and their effect. Thus, we try to combine those algorithms each one with it‘s own benefit, so that the detection algorithm is more accurate about it‘s conclusions and cater to more databases.

2 Solution Overview

Our objective is to highlight possible anomalies in any given data-set, and possibly removing them from the data-set. We achieve this by trying multiple solutions that take a couple of detection algorithms and infer from the data they provide on the data-set a better list of anomalies. We achieve this by running an anomalies detection algorithm and then refine the given anomaly list.

Figure 1: Work flow of the experiment. The yellow boxes are the processes we will change and will experiment on in our different solutions

2.1 Anomaly Detection Algorithms Usage

As a more baseline approach we take multiple detection algorithms and use them on the data-sets to compare our solutions to known and proven detection algorithms in order to measure our results. Each detection algorithm we chose has a attribute that makes him different from the rest and bases itself on a different approach. We ran each algorithm on is own on the data-sets and got the results to compare to.

- Local Outlier Factor(LOF)[1] - LOF is a quantification to outlierness of points, which adjusts the variations in different local densities. For a given data point X, firstly finds its distance, reachability distance and reachability density to its nearest neighbors. And count out important indexes into three sets, namely the set of kdistances, the set of k-distance neighborhood of X and reachdistance. The value of this quantitative factor is just the average of local accessible density of X's k-neighbors to X's local accessible density. Thus solves us the problem finding anomalies in data-sets that have a couple of dense areas.

- Elliptic Envelope[2] - The elliptic envelope finds the center of the data samples and then draws an ellipsoid around that center. The radii of the ellipsoid are measured using the Mahalanobis distance as the euclidean distance. After the ellipsoid is drawn, the elements that are drawn

outside it are called outliers. The Gaussian distribution is defined by the single-valued mean and variances, then the multivariate gaussian distribution is defined by matrices for mean and covariances. The multivariate Gaussian distribution is used to draw what is normal and outside it is called the outlier. This detection algorithm covers the statistical based detection approach.

- One Class SVM[3] - When modeling, One-Class captures the density of the majority class and classifies examples on the extremes of the density function as outliers. Although SVM is a classification algorithm and One-Class SVM is also a classification algorithm, it can be used to discover outliers in input data for both regression and classification data-sets. Hence, One-Class is based on the approach of learning where the most nonzero data is.

- Isolation Forest (IF/iForest)[4] - The principle of the detection method is similar with another ensemble technique, random forests, but the way to score data is different. As a set of isolation trees, each data would be recursively partitioned by cutting chosen partition with axisparallel randomly, so that each space has fewer instances. Until the points are isolated into singleton nodes, which contain only one instance. iForest discovers anomalies purely by concept of isolation without employing any proximity-based indexes, which is quite different from any other methods. Due to its point to cut dimensions, there're also lots of untapped attributes, which cause the reduction of method's reliability. It considers little about correlations between attributes, but is still a method who cares only about the global outliers. For this reason, this algorithm covers a more broad approach that doesn't care about the relations between attributes and bases itself on a more global relation based approach.

2.2 Data and Baseline Result Setup

Firstly in order to test our solutions we take a data-set, then we learn the data-set to the best prediction results. This is done so that we can compare the accuracy of the model before and after the data-set change, thus giving us a result to compare to our solution. As discussed in class many data-sets suffer from data that is incomprehensible such as NaN, infinity and negative infinity, therefore we cleaned our data by deleting all of those points. It is important to notice that we picked data with major eligible data.
We use 4 data-sets in our experiment:

- Boston Housing Prices
- Insurance Pricing
- Yacht Hydrodynamics
- Diabetes Prediction

For each data-set we build a prediction model to the best accuracy we could achieve.

2.3 Simple Solution

We try the very simple solution of running all the detection algorithms individually and classify as anomalies all the points given from all the algorithms. This is a very paranoid approach and isn't recommended unless the data-set is big or it is known that the anomalies impact on the model is undoubtedly negative.
This solution gives another point of reference but more importantly a base method to improve. As this method suffers greatly from loss of data we can improve the amount of data we take out (classify as anomalies), moreover we can improve the accuracy by giving the correct algorithm more weight in the final decision.

2.4 Second Solution

In this section we ran various combinations of the output of anomaly detection algorithm. In each experiment we took the two best algorithms that gave us the best results in the previous step. In experiments 3 and 4 it is not possible to use the results of LOF because it does not have the function decision function
The first experiment (delete_all):
In this experiment we take the markings of algorithms 1 and -1 (1 no outlier -1 outlier) Any line identified as an exception in one of the algorithms will be marked as an exception. After marking all possible anomalies we deleted the appropriate lines and ran the linear regression algorithm.
The second experiment (Add_and_delete):
In this experiment we take the markings of algorithms 1 and -1 (1 no outlier -1 outlier) We've put all the exception marker lines into a new array. We selected the lowest value in the array and marked each place value as an exception. All other locations are not exceptional After marking all possible anomalies we deleted the appropriate lines and ran the linear regression algorithm.
The third experiment(uniform_average):
In this experiment we will use the output of a decision function that returns the Average anomaly score of X of the base classifiers.(You can expand on the description of the algorithms on the sklearn website about the function work), The lower the value the more outlier it is. And we will perform a uniform average on it. A certain percentage of the data that received the lowest values after the average was then deleted and ran the linear regression algorithm.
The fourth experiment(ratio_average):
Very similar to previous experiment but ratio according to the results of the first algorithm.

2.5 Third Solution

In this section we have built a function that goes through all the possible combinations of the algorithms and runs them on the first two experiments we performed(in the second solution). In each run we tested whether the new value we received improves our accuracy result. Finally we return the accuracy results and two lists of -1 and 1 of anomaly detection, a list for each precision metric because sometimes a particular list of anomalies is more appropriate for a particular precision

metric.

3 Experimental Evaluation

In this section we will examine our results and infer the conclusion from each solution.

3.1 Solution 1

In the first part we ran the Linear Regression algorithm before and after running the anomaly algorithms and deleting the anomalies so that we can compare the algorithms to the normal results to see if they are effective. It is important to note that we have run the algorithms several times with different parameter values to reach a situation where at least some of the algorithms will be able to find us effective anomalies. Because we run with a learning algorithm that also uses randomness so not every run we get the same results and sometimes different algorithms will get different results but we will be able to notice consistent results. After optimizing the parameters of the anomaly detection algorithm we got results in which there is an improvement when we run the anomaly detection algorithm. It is not possible to see a particular algorithm that surpasses its capabilities in any data-set, this is because each algorithm has its own advantages that are suitable for different data sets.

Part 1 - Data diabetes

	mean_squared_error	mean_absolute_error
No deletions	0.489	0.521
LOF	0.445	0.518
EE	0.475	0.524
OneClassSVM	0.458	0.511
IForest	0.455	0.53

Part 1 - Data housing

	mean_squared_error	mean_absolute_error
No deletions	24.145	3.656
LOF	23.383	3.59
EE	24.087	3.663
OneClassSVM	24.209	3.664
IForest	22.912	3.494

Part 1 - Data yacht_hydrodynamics

	mean_squared_error	mean_absolute_error
No deletions	72.389	7.166
LOF	70.693	7.054
EE	72.598	7.146
OneClassSVM	70.006	6.77
IForest	72.182	7.054

Part 1 - Data insuranc

	mean_squared_error	mean_absolute_error
No deletions	35609191.266	4079.335
LOF	35367398.807	4156.553
EE	35634212.928	4071.309
OneClassSVM	35464712.461	4128.648
IForest	35745020.284	4030.51

3.2 Solution 2

In the results it is difficult to see a clear trend but it can be seen that there may be room for more experiments. In housing data we did not see a better result than the iforest algorithm when working separately. In diabetes data almost all the combinations gave better results than the algorithms individually which gives us hope that combining algorithms is the right direction. In insurance data there were algorithms that brought good results but not in all the indices something that gives us an idea to divide the experiment with separate results for the two metrics. In yacht data the results seem close to the table of part 1 but are not good in both indices at the same time

Part 2 - Data diabetes

	mean_squared_error	mean_absolute_error
delete_all	0.478	0.536
Add_and_delete	0.42	0.512
uniformy_average	0.544	0.523
ratio_average	0.528	0.524

Part 2 - Data housing

	mean_squared_error	mean_absolute_error
delete_all	23.408	3.449
Add_and_delete	23.918	3.653
uniformy_average	24.334	3.674
ratio_average	24.924	3.724

Figure 2: All the combinations gave better results than the algorithms individually.

Part 2 - Data insuranc

	mean_squared_error	mean_absolute_error
delete_all	35482255.626	4127.166
Add_and_delete	35689829.936	4055.091
uniformy_average	35482634.931	4108.674
ratio_average	35410331.344	4089.33

Part 2 - Data yacht_hydrodynamics

	mean_squared_error	mean_absolute_error
delete_all	71.259	6.946
Add_and_delete	72.088	7.112
uniformy_average	70.227	6.751
ratio_average	70.215	6.735

3.3 Solution 3

The results were compared to the first part where we ran each algorithm separately. It can be seen that the new algorithm brings better results on all data-sets than the rest of the algorithms. As we mentioned earlier all the results can vary from run to run so the results here reflect the run we ran on the algorithm, but the good effect is consistent.

4 Related Work

- Research on Stacking-Based Integrated Algorithm of Anomaly Detection in Production Process
  Authors: Huichen Shu; Xiaoyong Zhao; Huan Luo; Chen Li

Part 3 - Data diabetes

	mean_squared_error	mean_absolute_error
No deletions	0.489	0.521
LOF	0.445	0.518
EE	0.475	0.524
OneClassSVM	0.458	0.511
IForest	0.455	0.53
combination	0.432	0.51

Part 3 - Data housing

	mean_squared_error	mean_absolute_error
No deletions	24.145	3.656
LOF	23.383	3.59
EE	24.087	3.663
OneClassSVM	24.209	3.664
IForest	22.912	3.494
combination	22.457	3.485

Part 3 - Data yacht_hydrodynamics

	mean_squared_error	mean_absolute_error
No deletions	72.389	7.166
LOF	70.693	7.054
EE	72.598	7.146
OneClassSVM	70.006	6.77
IForest	72.182	7.054
combination	69.983	6.706

Figure 3: We can see that this data-set did meet expectations and the elimination was helpful.

Part 3 - Data insuranc

	mean_squared_error	mean_absolute_error
No deletions	35609191.266	4079.335
LOF	35367398.807	4156.553
EE	35634212.928	4071.309
OneClassSVM	35464712.461	4128.648
IForest	35745020.284	4030.51
combination	35367398.807	4071.309

Figure 4: We can see that this data-set didn't meet expectations and the elimination wasn't helpful.

5 Conclusion

In conclusion, we found that a combination of anomaly detection algorithms can lead to a better performing model. As such if anomalies do interfere with the model predictions, combining multiple anomaly detection algorithms will be helpful in distinguishing the anomalies and eliminate them. Furthermore we notice that the combinations of detection algorithms parameters impact the results in a non-trivial way, and they may vary from data-set to data-set. We suggest as a following experiment to check the connection between the detection algorithms used and the data-set.

6 Bibliography

1 M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers", Acm Sigmod International Conference on Management of Data. ACM, 2000, vol. 29, no. 2, pp. 93-104.

2 Somasree Majumde, "Detecting outliers using the Elliptic Envelope","https://bishnupadamajumder32.medium.com

3 Bernhard Schölkopf,John C. Platt,John C. Shawe-Taylor,Alex J. Smola,Robert C. Williamson,"Estimating the Support of a High-Dimensional Distribution",Neural Computation,Volume 13,Issue 7,July

2001,pp 1443–1471.

4 F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation-based anomaly detection," Acm Transactions on Knowledge Discovery from Data, 2012, vol. 6, no. 1, pp. 1-39.