Multivariate Anomaly Detection: Evaluating Isolation Forest

A Technical Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Alan Phlips

May 9, 2023

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Alan Phlips

Technical advisor: Briana Morrison, Department of Computer Science

Multivariate Anomaly Detection: Evaluating Isolation Forest

CS4991 Capstone Report, 2023

Alan Phlips
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
alp5ws@virginia.edu

ABSTRACT

In 2020, data provided to banks from Credit Bureaus used to evaluate potential customers began to contain contradicting values in certain fields. Isolation Forest, an unsupervised machine learning algorithm used for outlier detection, is a potential solution to this multivariate anomaly. Using the scikit-learn library in Python, the technique was evaluated starting with small, synthetic data with known outliers. A model was then trained on and then provided a sample of real bureau data to identify what it believed to be anomalous. The model successfully found records that stood out relative to the rest of the sample provided. However, it was not able to identify the multivariate anomaly that this project set out to resolve. While Isolation Forest was not successful for this task, there are other anomaly detection algorithms that could be tested to potentially address this and similar issues in the future.

1. INTRODUCTION

With hundreds of millions of credit cards and billions spent using them in the United States, it is important for banks and card issuers to have accurate data from credit bureaus (ABA, 2021). This data is used in many analyses, from evaluating potential new customers to creating products that are suited to specific groups of customers.

The primary sources of such data are credit bureaus such as Equifax, Experian, and Transunion (USAGov, 2022). These credit bureaus provide anonymized monthly updates to the card issuer that contains information on all of the open and active accounts for that issuer. The data contains fields that are useful in themselves, but many companies will combine them to create new, derived attributes. These proprietary attributes are used to evaluate how worthy of credit a current or new customer is which is helpful for approvals and credit line increases.

Beginning in data from May of 2020, some of the bureau-provided fields gave contradicting values. Some fields contain values that depend on the values of others. For example, if the value of a first field is zero, then the dependent field should also have a value of zero, but this was not always the case. Such anomalies need an algorithm or method that can detect or prevent them in the data.

Anomaly detection is an effective application of machine learning that has been implemented in industry and at scale. Isolation Forest, a machine learning anomaly detection algorithm, can quickly identify the most anomalous records in a dataset as potential outliers. In the context of credit bureau data, this hopefully means it can identify instances of the multivariate anomalies previously described.

2. RELATED WORKS

Isolation Forest is a model-based method of anomaly detection designed by Liu, et al (2008) and presented at the 2008 Eighth IEEE Conference on Data Mining. This

original paper defines several aspects of the algorithm in words with visual examples to guide understanding along with written "psuedocode" to demonstrate how it could be implemented. The paper does not include or reference examples of actual code or implementations of the algorithm.

The paper also describes the performance characteristics, advantages, and disadvantages of the algorithm, with test data and comparisons to other algorithms to support its claims. Since this is the original publication of Isolation Forest, it does not reference and real-world implementations of the method for outlier detection.

However, LinkedIn Engineering (Verbus, 2019) implements and utilizes the algorithm for anomaly detection for spam and abuse prevention. Their article also describes the algorithm in words in addition to providing a coded implementation. They describe the advantage of Isolation Forest for their needs, emphasizing the unlabeled and adversarial nature of abuse on LinkedIn, ending their article with other use cases for the algorithm.

3. PROCESS DESIGN

The process to evaluate isolation forest can be broken down into several steps. First, data provided to the algorithm has to be cleaned and otherwise prepared for use. From here, an isolation forest model has to be declared, trained, and scored at which point manual analysis of results can take place.

3.1 Data Preparation

The Isolation Forest algorithm can only work correctly when all features of the dataset in question are strictly quantitative. This is the case since in creating the "trees"' that make up the "forest,"' a feature is chosen at random and then a split value for that feature is chosen. This is done to partition the entries of the dataset and look for isolated points. Such a partition cannot be performed for a feature of the data that is not represented by a number. Consequently, any such columns of the dataset were removed. For the intended dataset of this project, the only such column was for the calendar date of a credit cycle, information that is not relevant to whether or not a data point is anomalous.

Isolation Forest is also an unsupervised machine learning technique. This means that the algorithm does not use "labels" for the data, instead working without them. Practically speaking, it means all features of the dataset used to train the algorithm could contribute to whether or not the point is believed to be an outlier, even the entry number. For the credit bureau data, this manifested itself in having to remove the scrambled entry number for each account in the data set. While this had the potential to make cross-referencing each point very difficult, pandas.DataFrame, the data structure used to store the data and give to the algorithm to operate on, keeps track of rows independently, addressing this difficulty. All that needed to be done was to remove the row number and account number for each data record before training and running the algorithm.

3.2 Running Isolation Forest

Assuming proper data preparation, use of the scikit-learn implementation of Isolation Forest in python is simple. First an instance of the Isolation Forest needs to be declared, then trained or fitted with some dataset, and finally some dataset (can be the same as the training data or different) is scored or predicted to find what the algorithm believes to be the most anomalous data points within the scored dataset.

3.2.1 Declaring an Instance

Isolation Forest is a model-based technique of anomaly detection. Declaring an instance is effectively creating a blank model to then tailor to the relevant use of the algorithm later. Declaring an instance of

Isolation Forest from the scikit-learn library includes first importing the ensemble learning library of scikit-learn. From here, an instance can be declared in accordance with the scikit-learn documentation (2011), specifying any of the parameters desired for the model to use in training or scoring. For this project, the default parameter values were used.

3.2.2 Fitting the Model

Once an instance of Isolation Forest has been declared, it is ready to be trained using a baseline dataset. This can be different from the dataset used for scoring/predicting, but for this project, each sample of credit data was used to train and score an instance of Isolation Forest. Fitting the model creates the trees that make up the forest and prepares the model/instance for scoring the data that will eventually be given to the model to look for anomalies.

In practice, this step consists of calling the fit() method on the previously declared instance of Isolation Forest, passing in the prepared dataset as a parameter to the fit function. Sample weights for each data sample can be specified for this step, but were not done for the credit bureau data as all were equally weighted in terms of importance. With datasets consisting of around 9-10 million entries, the fit() method took around 15 minutes to complete creating the forest of trees from the dataset each time.

Isolation Forest constructs its trees by first randomly choosing a feature of the dataset, and then randomly choosing a value between the minimum and maximum values for that feature. This partitions the data into two sets, one with a value for that feature below the randomly chosen threshold, and one above. This is in theory repeated on the two subsets of data until all entries are isolated at various endpoints along the tree. In practice, there is a maximum depth parameter that can be specified. Since anomalies make up only a small portion of the dataset and will tend to be isolated after fewer partitions of the data set, or "lower tree depth," past a certain depth all points can safely be assumed to not be outliers and thus no further recursions are needed. This process is repeated for all of the trees in the forest until the number of estimators, or trees, is reached.

For credit bureau data, since the data was provided for each credit cycle (i.e., every month) a model was trained on data from a given credit cycle.

3.2.3 Scoring Data

Once the model has been fitted, any data with the same feature set can be scored using same Isolation Forest. For this project, the same data used in training was used for scoring. Like fitting, scoring a dataset of around 9 million entries took about 20 minutes to score each time.

Using the same instance of Isolation Forest declared and fitted before, the predict() and score_samples() method can be used to detect anomalies within the datasets passed as parameters to each method. The score_samples() method works by running each individual record through each of the trees in the forest creating using the fit() method. In generally, entries that are outliers will be isolated with fewer partitions of the dataset than inliers. This on average means the depth that this point will be isolated at in the forest is lower than inliers. This average depth that the point is found is used to assign an anomaly score to each record. In this implementation, lower anomaly score means more abnormal. Details on the calculation of this can be found in the scikit-learn documentation or the original paper. The predict() method is the same as the score_samples() method, just taking one additional step to predict if a given entry is an outlier or not using the anomaly score.

Fitting and predicting could be combined into one step using the fit_predict() method, assuming the same dataset was desired for

both steps. Results of this single method matched the individual methods running in succession. This method was used interchangeably with the two individual ones with the credit bureau data.

4. RESULTS

Upon completing the process design, the model was successfully able to identify anomalous entries. However, these entries were not anomalous in a way that was a solution to the multivariate data anomaly for which this algorithm was intended.

Often, the points with the highest anomaly score from the algorithm had the highest balance or credit limit for the month as opposed to demonstrating the discrepancy in values across several features that was intended to be detected. As a result, the algorithm was not pursued further for this issue and others like it in credit bureau data.

5. CONCLUSION

Despite failing to solve the multivariate anomaly issue, isolation forest is a useful outlier detection technique with potential for many future implementations. The algorithm's unsupervised nature made it not well suited to handle the specific anomalies present in the dataset, but still able to identify records that did stand out from the rest. Its speed and effectiveness in detecting outliers make the algorithm appropriate for large scale industry adoption as a means of catching potential errors before they become too problematic or costly.

6. FUTURE WORK

While the original multivariate anomaly issue was addressed, other like it may occur. Isolation forest may not be a viable solution, but other outlier detection methods such as Local Outlier Factor, Kth-Nearest Neighbors, or Support Vector Machines may be effective at preventing such data issues. Additionally, isolation forest itself has many desirable properties for outlier detection, making it a possible solution to anomalies in different contexts or datasets.

REFERENCES

American Bankers Association (ABA). (2021, May 10). *2020 Q4 Credit Card Market Monitor.*https://www.aba.com/-/media/documents/reports-and-surveys/2020-q4-credit-card-market-monitor.pdf?rev=a3370871ac564a6f8767daf6698f618a&hash=51E065537CABDD452198890D55A4E2BB

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *Eighth IEEE International Conference on Data Mining*.

sklearn.ensemble.IsolationForest. (2011). scikit-learn. https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

USAGov. (2022, July 12). *Credit Reports and Scores*. https://www.usa.gov/credit-reports

Verbus, J. (2019, August 13). *Detecting and preventing abuse on LinkedIn using isolation forests*. LinkedIn Engineering. https://engineering.linkedin.com/blog/2019/isolation-forest