Paper 3306-2019

Detecting Fraud and Other Anomalies Using Isolation Forests

Ryan Gillespie, SAS Institute Inc., Cary, NC

ABSTRACT

Methods of fraud and other financial crimes are becoming increasingly sophisticated. One method of detecting fraud is to use unsupervised machine learning techniques. In this breakout session, we'll explore how isolation forests can be used to detect anomalies without requiring previously labeled data. We'll also compare this method against other well-known anomaly detection methods to evaluate its performance on selected data sets.

INTRODUCTION

Fraud is a growing concern for companies all over the globe. While there are many ways to fight and identify fraud, one method that is gaining increased attention is the use of unsupervised learning methods to detect anomalies within customer or transactions data. By analyzing customers or transactions relative to each other, we're able to spot unusual observations. These observations can potentially be indicative of fraud and, by identifying them, we are able to examine what is occurring and if it is of a fraudulent nature.

This paper examines one method that can be used to identify anomalies indicative of potential fraud using a method referred to as isolation forests. We'll examine how the method works and test it out on a publicly available data set constructed for fraud detection.

SUPERVISED LEARNING AND UNSUPERVISED LEARNING

When trying to identify fraud with machine learning, two approaches are commonly used. The first approach is with methods associated with supervised machine learning. This method involves using historical data that contains examples of the type of fraud that the user is trying to find. The algorithm can then learn to detect the fraudulent event by training a model using the examples of fraudulent and non-fraudulent cases. Typical modelling methods used for this type of fraud detection are logistic regression, decision trees, random forests, gradient boosting, neural networks, and other types of classification models.

In addition to supervised methods, there are also unsupervised methods that are being used to detect fraudulent cases. These methods are referred to as unsupervised because there is no historical information about fraudulent cases that is used to train the model. Whereas a data set used to train a supervised model would have a variable indicating fraud/non-fraud that could be used to train the model, a data set used for unsupervised modeling likely doesn't contain a variable with this information. Instead, unsupervised methods are used to find anomalies by locating observations within the data set that are separated from other heavily populated areas of the data set.

The assumption behind this is that fraudulent behavior can often appear as anomalous within a data set. It should be noted that just because an observation is anomalous, it doesn't mean it is fraudulent or of interest to the user. Similarly, fraudulent behavior can be disguised to be hidden within more regular types of behavior. However, without labeled training data, unsupervised learning is a good method to use to begin to identify deviant accounts or transactions.

There are several reasons why a user might want to use unsupervised methods instead of supervised methods. As described above, a user might not have historical examples of the type of fraud they are trying to detect. Without labelled observations, options for supervised learning are limited.

Another reason for using unsupervised methods instead of supervised methods, or in addition to them, is to try to find new types of fraud that may not have been captured within the historical data. Fraud patterns can evolve or change and so it is important to constantly be searching for ways to identify these new patterns as early as possible. If purely relying on supervised models built with historical data, these new patterns can be missed. However, since the unsupervised methods are not limited by the patterns present in the historical data, they can potentially identify these new patterns as they may represent behavior that is unusual or anomalous.

One unsupervised method we will examine to identify anomalies are isolation forests.

WHAT ARE ISOLATION FORESTS?

An isolation forest is an implementation of the forest algorithm that is used to detect anomalies instead of a target variable. (SAS Institute Inc. 2018) The main idea behind why it can help find anomalies is that observations that are more anomalous will have shorter paths from the root node of the tree in the forest to the leaf node. These observations will be isolated by fewer splits of the data. By averaging the path lengths for each observation, we can find observations that are more distinct within the data set.

The shortest distance path to isolating an observation can also be understood by visual examination. In Figure 1, we see a two-dimensional data set with two anomalies that are highlighted, one in red and one in orange. By examining how many divisions of the X1 and X2 variable we would require to isolate each observation, we can get a sense of how anomalous the observation is considered.

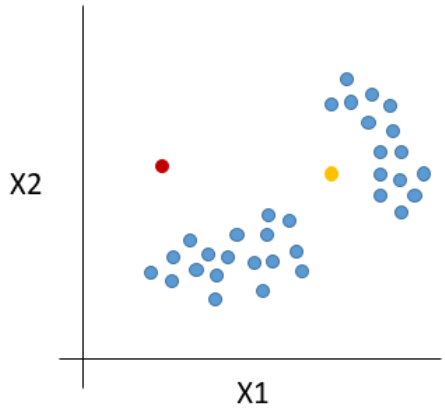

Figure 1: Two-dimensional space with two highlighted observations

Looking at the observation that is colored red, we can see within Figure 2 that it takes two divisions to isolate this record from the remaining records in the data set.

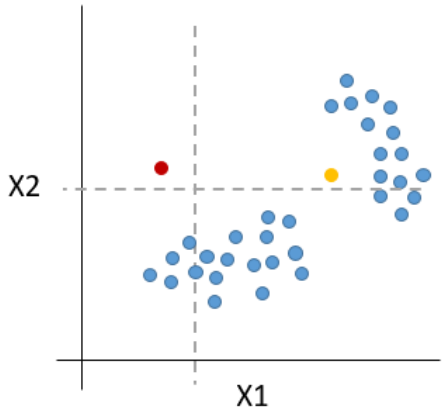

Figure 2: Two-dimensional space with the red observation isolated

However, when we try to isolate the orange observation, we can see within Figure 3 that it takes three divisions.

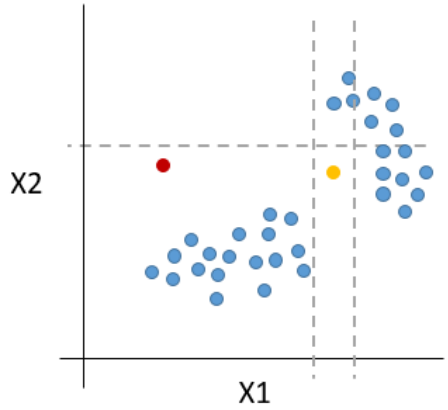

Figure 3: Two-dimensional space with the orange observation isolated

Based on the two divisions, we would say that the red colored observation would be more anomalous than the orange. This is the central idea behind isolation forests, however, during the implementation of the algorithm, it is applied as an average of path lengths that make up the ensemble of decision trees used to construct the forest.

The isolation forest algorithm in SAS® Visual Analytics will return an anomaly score for each observation based on the path lengths for that observation within the constructed trees. The formula used to calculate the anomaly score is (SAS Institute Inc. 2018):

$$Anomaly\ score = 2^{-\left(\frac{Average\ path\ length\ for\ observation}{Average\ length\ of\ all\ paths}\right)}$$

With this formula, the algorithm will return an anomaly score between zero and one. With this anomaly score we can then analyze these rare observations for their likelihood of fraud. One way to measure the validity of the isolation forest in finding fraud events is to test it against a data set that does contain labelled fraudulent observations.

IMPLEMENTING AN ISOLATION FOREST ON THE PAYSIM DATA SET

To test the validity of our isolation forest model, we'll evaluate it against a data set that contains synthetic transactions for mobile payments, known as PaySim. (E. A. Lopez-Rojas , A. Elmir, and S. Axelsson 2016) This sampled data set contains transaction data for a variety of transactions, with 11 variables and 6,362,620 observations.

For the purposes of our comparison, we'll use the 'isFraud' variable as a check for our unsupervised model's ability to detect the fraudulent observations. However, as inputs to the model, we will only use the amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest and newbalanceDest variables. We will also only examine the 'CASH_OUT' and 'TRANSFER' transaction types as these are the categories containing the fraudulent observations. Limiting the data set to these categories reduces the total number of observations to 2,770,409. For the purposes of quick evaluations, we will sample this at 20% to examine different hyperparameter scenarios.

To implement the isolation forest, we will use the forestTrain action from the decisionTree action set in SAS Visual Analytics.

We'll test the action at several different hyperparameter settings. The first being the default settings, and then several options testing different combinations of the number of trees used, the number of bins used for continuous variables, the number of input variables chosen for each node split, and the maximum depth of each tree. An example of the action for one of the options using the SAS SWAT package in Python is shown here:

```
s.decisionTree.forestTrain(isolation=True,
        table={'name':'training_sampled', 'where':'_PartInd_ eq 1.0'},
        inputs=['amount', 'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', 'newbalanceDest'],
        nTree=50,
        nBins=32,
        m=3,
        leafSize=1,
        maxLevel=8,
        seed=99,
        casout = {"name" : "iso_model", "replace" : True},
)
```

The results for several of the selected options are shown in Figure 4. The table in Figure 4 illustrates the results for the default option, as well as two other options that were better than the default result in at least one of two evaluation metrics.

The analysis decided to focus on two result metrics to determine the efficacy of the model. The first was to evaluate the area under the receiver operating characteristic curve (AUC). This metric helps evaluate the effectiveness of classifiers with a score of 1.0 for a model that perfectly classifies the observations and a score of 0.5 indicating the model is no better than randomly guessing. However, AUC can also potentially be misleading when evaluating rare event classifiers due to the imbalance of false negatives that can skew the false positive rate associated with the curve.

In fraud or other rare event detection, a company may also be limited by the amount of events we can investigate or assess. As such, the isolation forest is also evaluated by looking at the number of events captured within the 1,000 observations with the highest anomaly score. The thinking being that if the company only had the time to look at a set number of observations, they would prefer to maximize the amount of true positives within that resource constrained span.

The results for both metrics for the default and selected options are:

	Default	Option 1	Option 2
Number of Trees	50	50	50
Number of Bins	20	16	32
Input Variables for Node Split	3	3	3
Max Depth of Tree	6	8	8
AUC	**0.830**	**0.845**	**0.784**
Fraud Cases in Top 1000 Scores	**62**	**77**	**158**

Figure 4: Isolation Forest Results for Default and Selected Metrics

We can see within Figure 4 that the default options return an AUC of 0.830 and contain 62 fraud cases within the top 1,000 scores. This indicates that the model does have predictive ability in finding a portion of the 1,643 cases of fraud within the 554,082 events.

We can see that Option 1 and Option 2 also perform better than the default option in at least one of the two chosen result metrics. Option 1 shows a slight increase for the AUC of the model and a 24% increase in the number of fraud cases found within the highest scoring observations. Interestingly, Option 2 shows a lower AUC score but a substantial increase in the number of fraud cases detected within the 1,000 highest scoring cases. Evaluating these results in the different ways can help companies determine what might be the correct model for them based on the priorities, costs and resource constraints of their process.

We've seen how an isolation forest can be used to find fraud cases within the PaySim data set. Now let's turn to another commonly used unsupervised method for finding anomalies to evaluate how well it can locate examples of fraud within the same data.

IMPLEMENTING CLUSTERING TO FIND ANOMALIES ON THE PAYSIM DATA SET

Another popular method for identifying anomalies in unlabeled data sets is to use clustering. By clustering data into common groups, we can examine the groups with low numbers of records as potentially anomalous and see how many fraud events are contained within these groups.

For the clustering analysis, the k-means algorithm was used from SAS Visual Statistics. Twenty clusters were chosen as the input for the algorithm after reviewing results from clustering with five and ten clusters. The number of clusters was selected based on trying to isolate approximately 1,000 observations as anomalous from the data set in order to compare to the results from the isolation forest model.

CLUSTERING RESULTS

Looking at the most anomalous 1,000 observations from the clustering, it can be found that 149 observations are flagged as fraud within these clusters. This is substantially better than the default results seen with the isolation forest but 6% less than what was seen with the top selected isolation forest results for the data set.

One aspect where the isolation forest provides additional clarity over traditional clustering is within the anomaly score. The isolation forest provides a way to rank the entire data set, whereas the clustering method relies more heavily on subjective parsing to determine a ranking for how to distinguish between observations within different clusters.

CONCLUSIONS

This paper has demonstrated a use case for implementing isolation forests to detect anomalies and other rare events such as fraud. The results indicated that tuning the isolation forest could result in large increases in traditional classification metrics, such as AUC, as well as unconventional metrics that might be of relevance to resource constrained businesses. A comparison with a clustering approach has also shown how both options can be worthwhile to explore anomaly detection and how the results from isolation forests could potentially be easier to interpret across an entire data set.

REFERENCES

SAS Institute Inc. 2018 *SAS® Visual Analytics 8.3: Procedures*. Cary, NC: SAS Institute Inc.

E. A. Lopez-Rojas , A. Elmir, and S. Axelsson. "PaySim: A financial mobile money simulator for fraud detection". In: *The 28th European Modeling and Simulation Symposium-EMSS*, Larnaca, Cyprus. 2016

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Ryan Gillespie
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Ryan.Gillespie@sas.com
http://www.sas.com