Extended Isolation Forest

by Sahand Hariri, Matias Carrasco Kind, Robert J. Brunner

Presenter: Jaehyeong Ahn

Department of Applied Statistics, Konkuk University

jayahn0104@gmail.com

Contents

Motivation

What makes this problem?

Solution

Notation

Methodology

1 Training Stage

2 Evaluation Stage

Empirical Results

Summary

Motivation

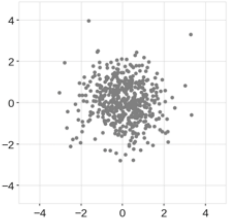
- Anomaly scores produced by Isolation Forest reveals that they are inconsistent
- The best way to see the problem is to examine very simple datasets where we have an intuition of how the scores should be distributed and what constitutes an anomaly
- We will see three motivational situations with synthetic datasets which show the problem of standard Isolation Forest

Motivation

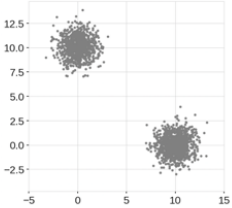
Normally distributed data

- Let there is a two dimensional dataset sampled from a 2-D normal distribution with zero mean vector and covariance given by the identity matrix

- We expect to see an anomaly score map with an almost circular and symmetric pattern with increasing values as we move radially outward(i.e., similar score values for fixed distances from the origin)



(a) Single Blob

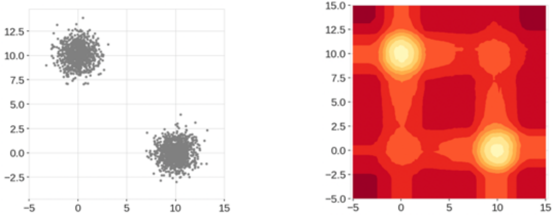


(b) Anomaly Score Map

Motivation

Two normally distributed clusters

- Let there are two separate clusters of normally distributed data concentrated around (0,10) and (10,0)



(a) Multiple Blobs

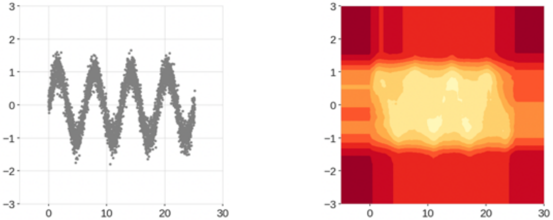
(b) Anomaly Score Map

- We can observe "ghost" clusters close to (0,0) and (10,10) which raise a significant problem
- Not only does this increase the chances of false positive, it also wrongly indicates a non-existent structure in the data

Motivation

Sinusoidal data points with Gaussian noise

- The data has an inherent structure, the sinusoidal shape with a Gaussian noise added on top



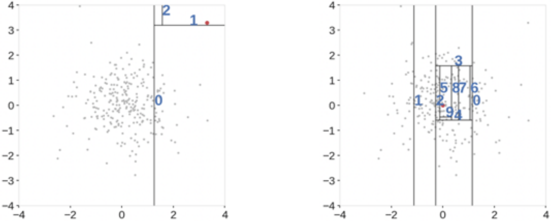
(a) Sinusoidal data points with Gaussian noise

(b) Anomaly Score Map

- We can see that the algorithm performs very poorly
- Isolation Forest failed to capture the structure of data

What makes this problem?

- The problem arises because of the way the branching of the tree in Isolation Tree
- Below figure shows the branching process during the training phase for an anomaly and a nominal points using the standard Isolation Tree for the normally distributed data



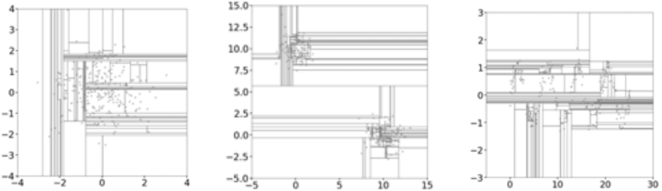
(a) Anomaly point

(b) Nominal point

- In this case the random cuts are all either vertical or horizontal because of the nature of the Isolation Tree

What makes this problem?

- Let's consider one randomly selected fully grown tree with all its branch rules and criteria



(a) Single blob (b) Multiple Blobs (c) Nominal point

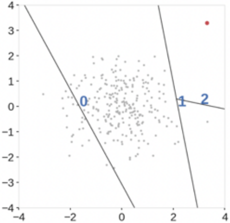
- Note that in each step, we pick a random feature (dimension), X_j, and a random value, v, for this feature

- As we move down the branches of the tree, the range of possible values for v decreases and so the line tend to cluster where most of the data points are concentrated

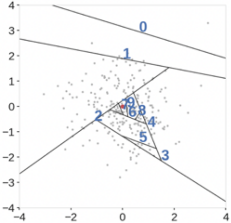
- Because of the constraint that the branch cuts are only vertical and horizontal, regions that don't necessarily contain many data points end up with many branch cuts

Solution

- (Sahand Hairiri et.el., 2018) proposes the "Extended Isolation Forest" which uses hyperplanes with random slopes (non-axis-parallel) at each split to overcome the drawbacks



(a) Anomaly



(b) Nominal

- For Extended Isolation Forest, the selection of the branch cuts requires only two pieces of information:
  1. A random slope for the branch cut
  2. A random intercept for the split which is chosen from the range of available values of the training data
- This is equivalent to standard Isolation Forest which also requires two pieces of information (a random feature, a random value for the feature)

Solution

Random Slope for the branch cut

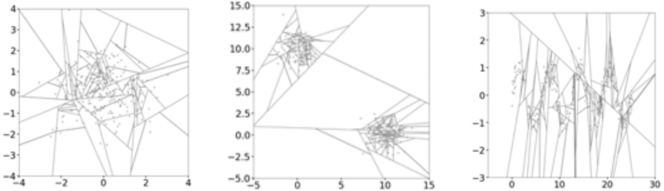
- For a D dimensional dataset
- Selecting a random slope for the branch cut is the same as choosing a normal vector $\vec{d}$ uniformly over the unit D-Sphere
- This is equivalent to draw a random number for each coordinate of $\vec{d}$ from $\mathcal{N}(0,1)$ which is the standard normal distribution
- For the intercept, $\vec{p}$, we simply draw from a uniform distribution over the range of values present at each branching point
- The branching criteria for the data splitting for a given point $\vec{x}$ is as follows:

$$(\vec{x} - \vec{p}) \cdot \vec{d} \leq 0 \tag{1}$$

- If the condition is satisfied, the data point $\vec{x}$ is passed to the left branch, otherwise it moves down to the right branch

Solution

- Fully grown tree with all its branch rules and criteria using **random slope**



(a) Single blob (b) Multiple Blobs (c) Sinusoidal

- Note that despite the randomness of the process, higher density points are better represented schematically

- We can observe that there are no regions that artificially receive more attention that the rest

- The results of this are score maps that are free of artifacts previously observed

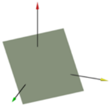
Solution

High Dimensional Data

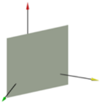
- The algorithm generalizes readily to higher dimensions
- In this case, the branch cuts are no longer straight line, but $D - 1$ dimensional hyperplanes
- The same criteria for branching process specified by inequality (1) applies to the high dimensions

Extension Levels

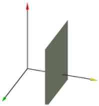
- We can consider D levels of extension by setting coordinates of the normal vector to zero
- For example, In the case of three dimensional data



(a) Ex 2



(b) Ex 1



(c) Ex 0

- So for any given D dimensional dataset, the lowest level of extension of the Extended Isolation Forest is coincident with the standard Isolation Forest (i.e., it's a generalization of isolation forest)

Notation

- $X = \{x_1, \cdots, x_n\}, \quad x_i \in \mathbb{R}^d$ for all $i = 1, \cdots, N$
    - (only consider continuous valued attributes)
- T is a node of an isolation tree
    - T is either an external-node (terminal-node) with no child, or an internal-node with one test and exactly two daughter nodes (T_l, T_r)
    - A test consists of a normal vector $\vec{d}$ and a intercept $\vec{p}$ such that the test $(\vec{x} - \vec{p}) \cdot \vec{d} \leq 0$ divides data points into T_l and T_r
- $h(x)$: Path Length of a point x which is measured by the number of edges x traverses an iTree from the root node until the traversal is terminated at an external node
- $c(n)$: Average path length of unsuccessful search in BST(Binary Search Tree)[B. R. Preiss,1999]. This is used for normalization of $h(x)$, $(H(i) \approx ln(i) + 0.5772156649)$
$$c(n) = 2H(n-1) - (2(n-1)/n) \tag{2}$$
- $s(x, n)$: Anomaly score
$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{3}$$

- $s(x,n)$: Anomaly score

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

  - when $E(h(x)) \rightarrow c(n), \quad s \rightarrow 0.5$
  - when $E(h(x)) \rightarrow 0, \quad s \rightarrow 1$
  - when $E(h(x)) \rightarrow n-1, \quad s \rightarrow 0$

$\Rightarrow s$ is monotonic to $h(x)$

- $0 < s \leq 1$ for $0 < h(x) \leq n-1$
  - (a) if instances return s very close to 1, then they are definitely anomalies,
  - (b) if instances have s much smaller than 0.5, then they are quite safe to be regarded as normal instances,
  - (c) if all the instances return $s \approx$ 0.5, then the entire sample does not really have any distinct anomaly

Methodology

Algorithm 0: $iForest(X, t, \psi)$

```
Inputs  : $X$ - input data, $t$ - number of trees, $\psi$ - sub-sampling size
Outputs: a set of $t$ $iTrees$
begin
    set height limit $l = ceiling(\log_2 \psi)$;
    for $i = 1$ to $t$ do
        $X^{'} \leftarrow sample(X, \psi)$;
        $Forest \leftarrow Forest \cup iTree(X^{'}, 0, l)$;
    end
    return $Forest$
end
```

Methodology

Algorithm 1: $iTree(X, e, l)$

```
Inputs  : $X$ - input data, $e$ - current tree height, $l$ - height limit
Outputs: an iTree
begin
    if $e \geq l$ or $|X| \leq 1$ then
        return $exNode\{Size \leftarrow |X|\}$;
    else
        Randomly select a normal vector $\vec{d} \in \mathbb{R}^{|X|}$ by drawing each
          coordinate of $\vec{d}$ from a standard Gaussian distribution
        Randomly select an intercept point $\vec{p} \in \mathbb{R}^{|X|}$ in the range of $X$
        Let coordinate of $\vec{d}$ to zero according to extension level
        $X_l \leftarrow filter(X, (X - \vec{p}) \cdot \vec{d} \leq 0)$
        $X_r \leftarrow filter(X, (X - \vec{p}) \cdot \vec{d} > 0)$
          return $inNode\{Left \leftarrow iTree(X_l, e + 1, l),$
                              $Right \leftarrow iTree(X_r, e + 1, l),$
                              $Normal \leftarrow \vec{d},$
                              $Intercept \leftarrow \vec{p}\}$
    end
end
```

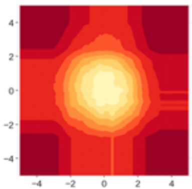
Methodology

Algorithm 2: $PathLength(\vec{x}, T, e)$

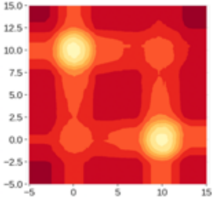
```
Inputs  : $\vec{x}$ - an instance, $T$ - an iTree, $e$ - current path length;
to be initialized to zero when first called
Outputs: path length of $\vec{x}$
begin
    if $T$ is an external node then
        return $e + c(T.size)\{c(.)$is defined in Equation (2)$\}$
    end
    $\vec{d} \leftarrow T.Normal$;
    $\vec{p} \leftarrow T.Intercept$;
    if $(\vec{x} - \vec{p}) \cdot \vec{d} \leq 0$ then
        return $PathLength(\vec{x}, T.left, e + 1)$;
    else if $(\vec{x} - \vec{p}) \cdot \vec{d} > 0$ then
        return $PathLength(\vec{x}, T.right, e + 1)$
    end
end
```

Empirical Results

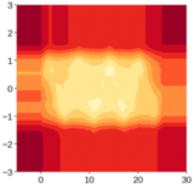
Score Maps



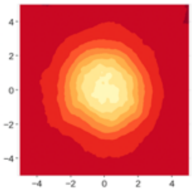
(a) Standard IF



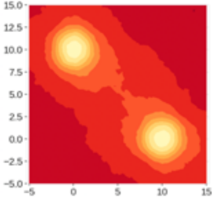
(b) Standard IF



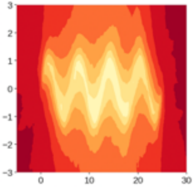
(c) Standard IF



(a) Extended IF



(b) Extended IF

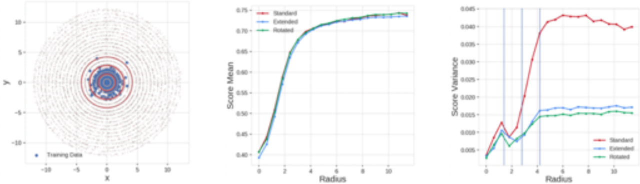


(c) Extended IF

Empirical Results

Variance of anomaly scores

- Comparing the mean and variance of the anomaly scores of points distributed along roughly constant score lines for various cases between Standard IF and Extended IF
- Let there is a normally distributed data, anomaly scores along each circle should remain more or less a constant

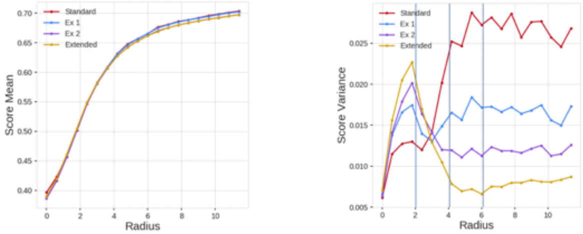


(a) Data (b) Score Mean (c) Score Variance

- After about 3σ, the variance among the scores computed by the Extended IF is much more stable and smaller than Standard IF
- Which means that Extended IF is a much more robust anomaly detector

Empirical Results

- Let's consider the higher dimensional case and the extension levels
- Choose 4-D blobs of normally distributed data in space around the origin with unit variance in all directions
- Since it is the case of 4 dimension, we can execute 3 extension levels



(a) Score Mean

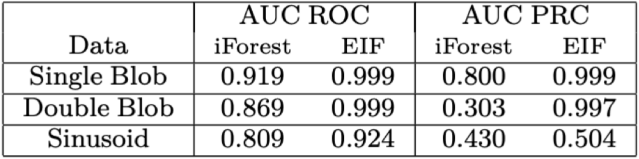
(b) Score Variance

- After about 3σ, similar to 2-D case, the Extended IF produces much lower values in the score variance than Standard IF
- We also can observe that the variance decreases as the extension level increases (i.e., the Extended IF produces the most reliable and robust anomaly scores)

Empirical Results

AUC Comparison

- In this section we report AUC values for ROC and PRC

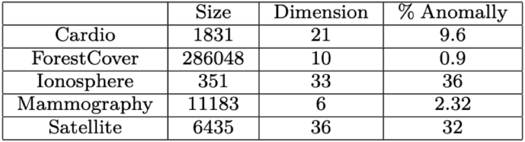


	AUC ROC		AUC PRC	
Data	iForest	EIF	iForest	EIF
Single Blob	0.919	0.999	0.800	0.999
Double Blob	0.869	0.999	0.303	0.997
Sinusoid	0.809	0.924	0.430	0.504

- In all cases the improvement is obvious, especially in the case of the double blob ans the sinusoid

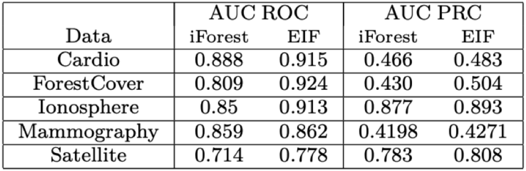
Empirical Results

Real data experiments



	Size	Dimension	% Anomally
Cardio	1831	21	9.6
ForestCover	286048	10	0.9
Ionosphere	351	33	36
Mammography	11183	6	2.32
Satellite	6435	36	32

Figure: Table of data properties



	AUC ROC		AUC PRC	
Data	iForest	EIF	iForest	EIF
Cardio	0.888	0.915	0.466	0.483
ForestCover	0.809	0.924	0.430	0.504
Ionosphere	0.85	0.913	0.877	0.893
Mammography	0.859	0.862	0.4198	0.4271
Satellite	0.714	0.778	0.783	0.808

Figure: Table of AUC values for both ROC and PRC

Summary

- In this work, it has shown an extension to the anomaly detection algorithm known as Isolation Forest
- The motivation for the study arose from the observation that score maps in two dimensional datasets demonstrated unexpected artifacts
- It proposed that this problem can be solved by obtaining **random slopes** (non-axis-parallel) at each split in branching procedure in the algorithm
- Further more, we have seen that this kind of solution earned improvement in detecting anomalies not only in synthetic data, but also in real datasets
- Additionally, it insists that this method does not sacrifice computational efficiency compared to the Standard IF

Future works

- dimensionality reduction (either feature selection or feature extraction) for anomaly detection in high dimensional data
- Dealing with categorical variables

References

1 Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest." In 2008 Eighth IEEE International Conference on Data Mining, pp. 413-422. IEEE, 2008.

2 Hariri, Sahand, Matias Carrasco Kind, and Robert J. Brunner. "Extended isolation forest." arXiv preprint arXiv:1811.02141 (2018).