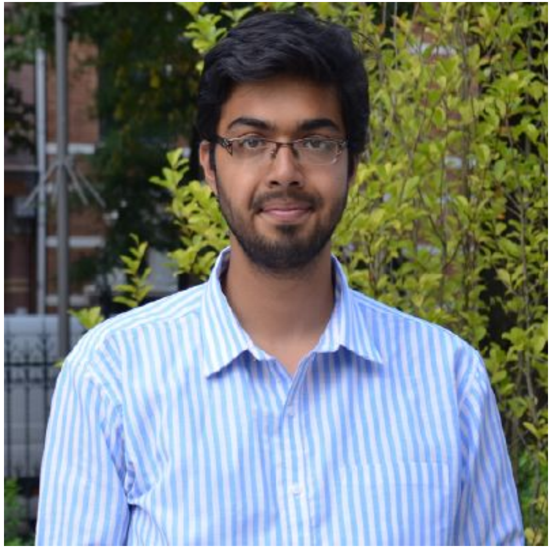
OOD/Anomaly Detection

28th November 2020

Mantissa DS Webinars

$WHOAMI

Data Scientist at Vahan.ai
Founder at Mantissa Data Science
Previously: Co-founder & CTO at Recreate.ai
Data Scientist, Indegene
Masters in ITIS, TU Braunschweig
Bachelors in ECE,
MS Ramaiah Institute of Technology



Agenda

1. What is OOD?
2. Why is it important - Applications involving OOD
3. Algorithms/Techniques to find OOD data points
4. Factors to consider when choosing an Anomaly Detector
5. Other applications of Anomaly Detectors
6. Question and Answers

What is OOD?

1. Out-of-distribution Detection
2. Anomaly Detection
3. Outlier Detection
4. Novelty Detection
5. One Class Classification

01 **Out-of-distribution/Anomaly Detection**

- OOD Detection deals with detecting whether a test sample is from in-distribution (i.e., training distribution by a classifier) or out-of-distribution sufficiently different from it. [1]

02 **Outlier Detection**

- An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism and outlier detection deals with detecting such observations.[9]

03 **Novelty Detection**

- Novelty detection is the identification of new or unknown in-distribution data that a machine learning system is not aware of during training. [3]

In short, it’s about finding things that don’t fit a pattern

Why is it Important?

Some Applications

1. Are there an unusual amount of login attempts from a particular IP address?
2. Are any customers buying more than the typical number of products at a given hour?
3. Which homes are consuming above-average amounts of water during a drought?
4. Which judges convict an unusual number of defendants?
5. Should a patient's blood tests be considered normal, or are there outliers that require further checks and examinations?
6. Are transactions being done through a credit card that may indicate fraudulent behaviour?

Techniques/Algorithms used

Types of Techniques

1. Probability Distribution (Normal, Poisson etc) Based
2. Machine Learning Based

Distribution based / rule based

- Standard deviation is a measure of how spread out the data is.
- From the empirical rule (or the 68-95-99.7 rule), we know that 68% of the values lie within the first standard deviation, 95% of the values within the 2nd std. deviation and 99.7% of the values within the 3rd std. Deviation.
- One simple way to do this is to set a cutoff, often done at two or three standard deviations.
- This is simple, but it has its shortcomings. For instance, it works only on a single dimensional vector.

Machine learning based

The following types of methods have been identified as the top recommended anomaly detectors [4]:

1. Proximity-based methods
2. Isolation-based methods
3. Kernel-based methods

Proximity based - LOF algorithm

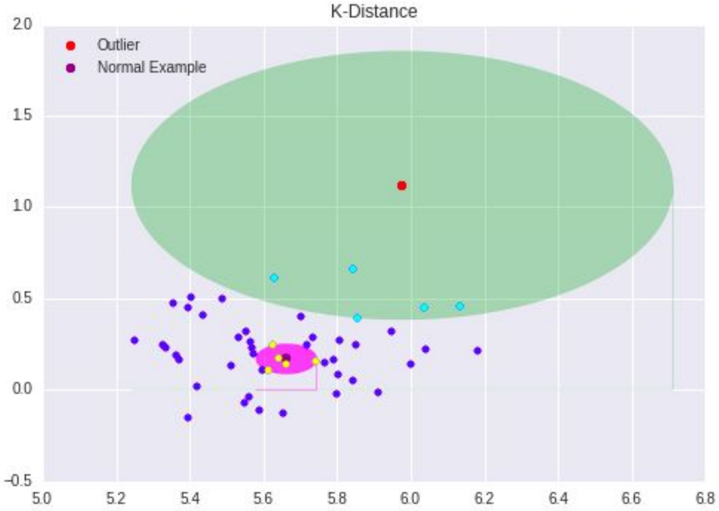
1. Local Outlier Factor (LOF) algorithm - unsupervised anomaly detection method.
2. Computes the local density deviation of a given data point with respect to its neighbours.
3. The Local Outlier Factor algorithm can be essentially broken down into four parts:

- K-Distance and K-Neighbors
- Reachability-Distance
- Local Reachability Density
- Local Outlier Factor calculation

Proximity based – LOF algorithm

1. **K-Distance and K-Neighbors**

- The first step is creating a framework to model how "isolated" each data point is.
- We choose the number k of neighboring points we want to consider and for an arbitrary data point p, we find the necessary radius r to have k points within distance r of p.
- How to choose k:
  - Fix k to a particular number
  - Let k represent a proportion of the data



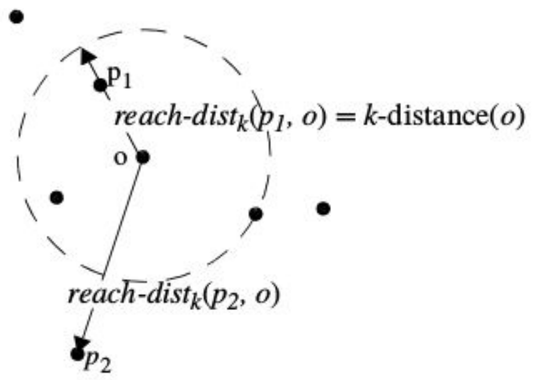
- The K-Distance provides a meaningful heuristic to reason about isolated data points. The more isolated a data point is, the farther we will have to search to find k neighboring points.

1. https://stealthbits.com/wp-content/uploads/2016/03/uba-graph-2.png

Proximity based – LOF algorithm

2. **Reachability-Distance**

- Reachability-Distance (o,p) = max{Distance(o,p), K-Distance(o)}
- If a point p is one of point o's k-nearest neighbors,
  then Distance(o,p) will be less than K-Distance(o),
  so Reachability-Distance(o,p) = K-Distance(o).
- If a point p is not one of point o's k-nearest neighbors,
  then Distance(o,p) is greater than K-Distance(o),
  so Reachability-Distance(o,p) = Distance(o,p).

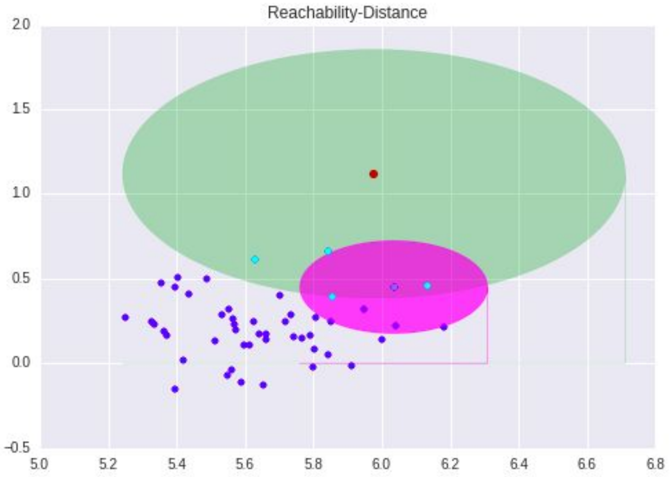


- In essence, we create a circle with radius K-Distance(o)
  around the point o. All points within the circle are "pushed" to the boundary of the circle.
- As a result, all of point o's k-nearest neighbors are considered equidistant from o under the Reachability-Distance.

1. https://stealthbits.com/wp-content/uploads/2016/03/uba-graph-2.png

Proximity based - LOF algorithm

3. **Local Reachability Density**

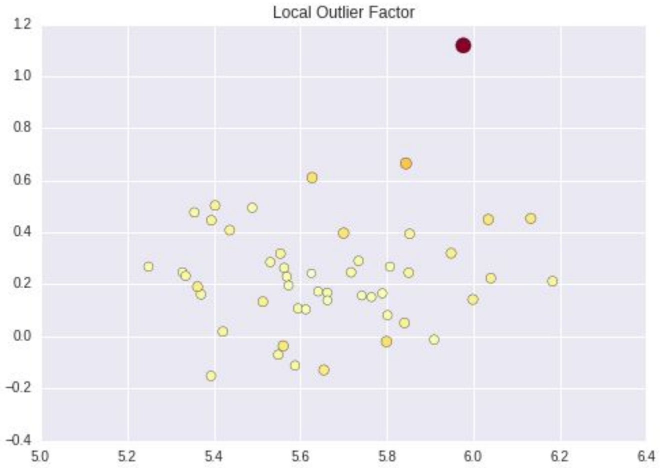


- Local Reachability Density for a point o is:
  LRD(o) = Average (Reachability-Distance of o's neighbors)
- We expect our outliers to be in less dense (sparser) regions compared to normal points.
- Therefore, our outliers should have lower local reachability densities, showing that the density around our outlier is lower than those of the other points in the dataset.
- Local Reachability Density provides us with an estimate of the "statistical density" for each point.

1. https://stealthbits.com/wp-content/uploads/2016/03/uba-graph-2.png

Proximity based - LOF algorithm

4. **Local Outlier Factor calculation**



- Local Outlier Factor value for point o:
  LOF(o) = Average (LRD of K-Neighbors of o)
- The LOF is a ratio that shows the relative density of neighbors.
- LOF(o) ~ 1 means Similar density as neighbors.
  LOF(o) < 1 means Higher density than neighbors (Inlier).
  LOF(o) > 1 means Lower density than neighbors (Outlier).

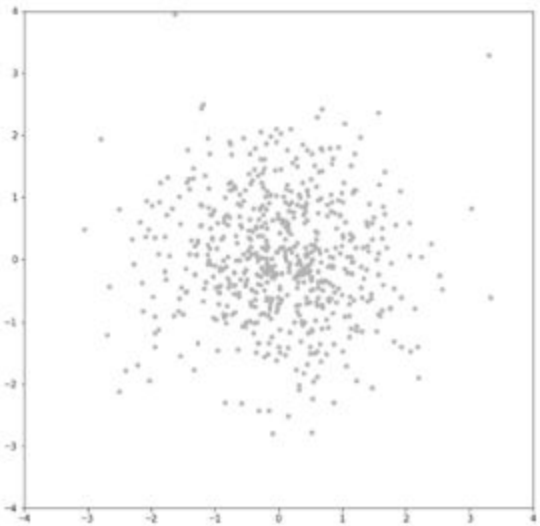
1. https://stealthbits.com/wp-content/uploads/2016/03/uba-graph-2.png

Isolation based

1. In the previous method, we saw how we created a profile for normal instances and then used that to flag a data point as an anomaly.
2. There are two major drawbacks of such an approach:
    a. It is optimized to profile normal instances and not to detect anomalies. As a result, the results of anomaly detection might not be as good as expected (too many False Positives).
    b. Many existing methods work only with low dimensional data and small data size because of the high computational complexity.
3. Isolation Based methods concentrate on the minority data points and their attribute values.
4. The idea being that anomalies are 'few and different' and are therefore more susceptible to isolation than normal points.
5. One popular Isolation based technique is Isolation Forests.

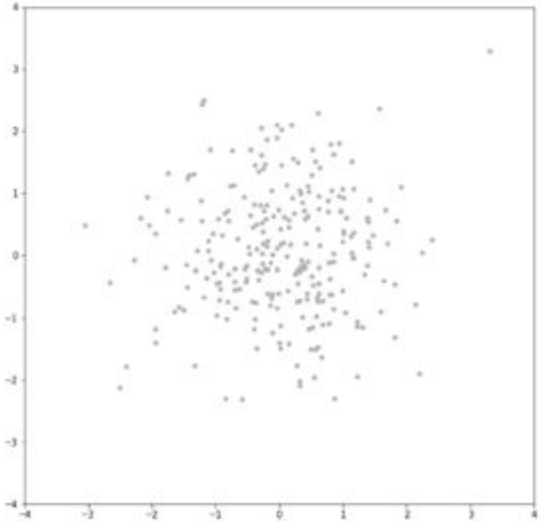
Isolation based – Isolation Forest

- 'Few and different' can be isolated quicker than many and normal.



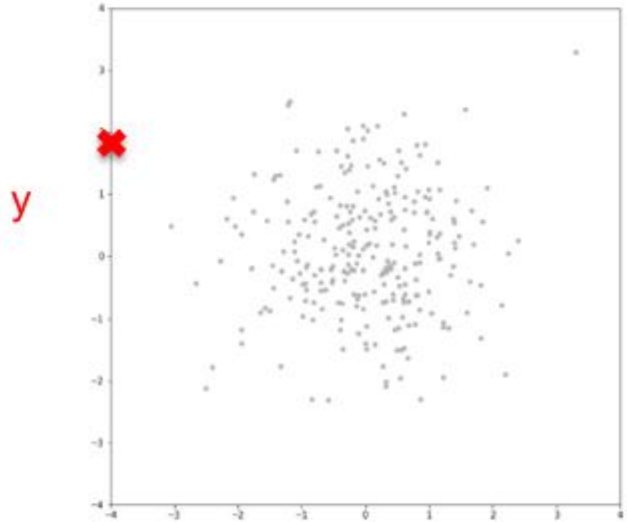
Isolation based - Isolation Forest

- ‘Few and different’ can be isolated quicker than many and normal.
- For each tree:
    - Get a sample of the data



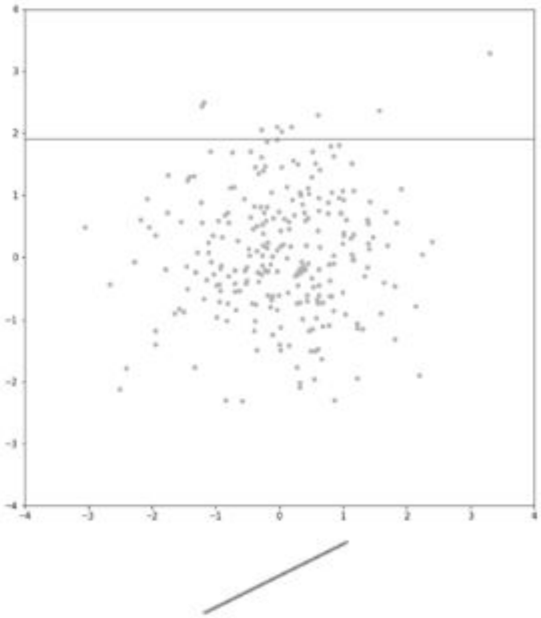
Isolation based – Isolation Forest

- ‘Few and different’ can be isolated quicker than many and normal.
- For each tree:
    - Get a sample of the data
    - Randomly select a dimension
    - Randomly pick a value in that dimension



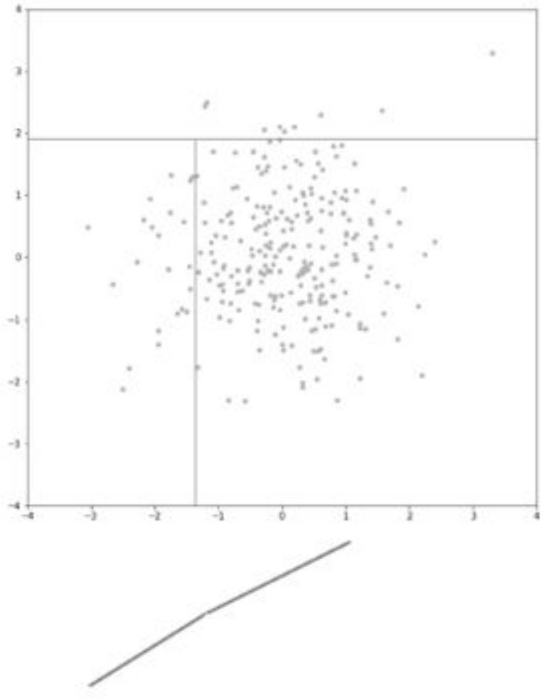
Isolation based – Isolation Forest

- ‘Few and different’ can be isolated quicker than many and normal.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data



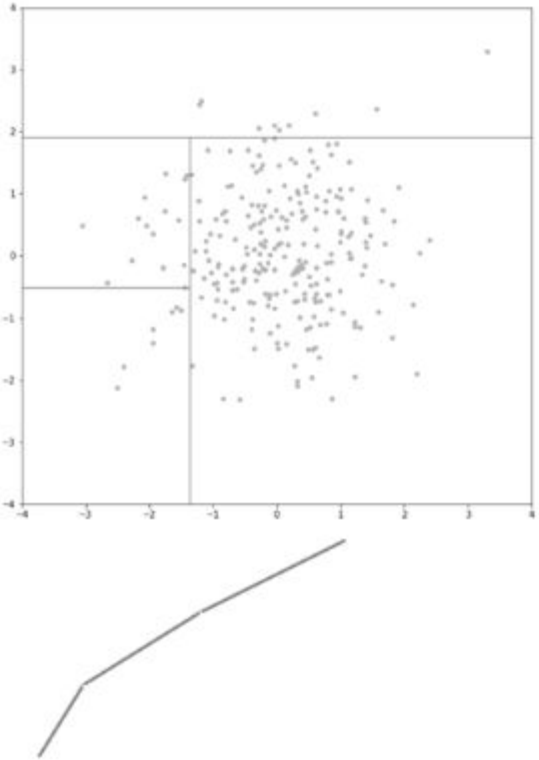
Isolation based – Isolation Forest

- ‘Few and different’ can be isolated quicker than many and normal.
- For each tree:
    - Get a sample of the data
    - Randomly select a dimension
    - Randomly pick a value in that dimension
    - Draw a straight line through the data at that value and split data
    - Repeat until tree is complete



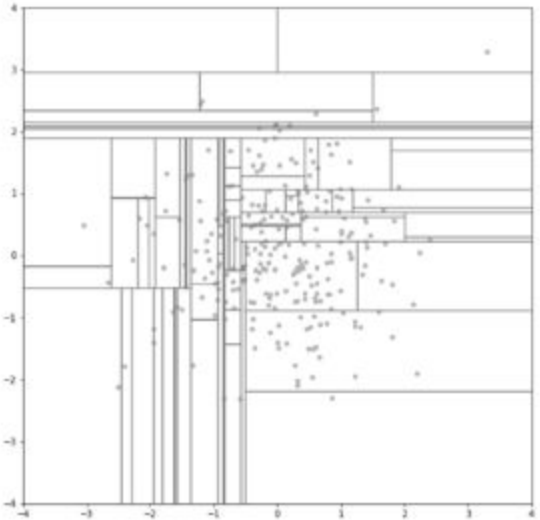
Isolation based – Isolation Forest

- ‘Few and different’ can be isolated quicker than many and normal.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete



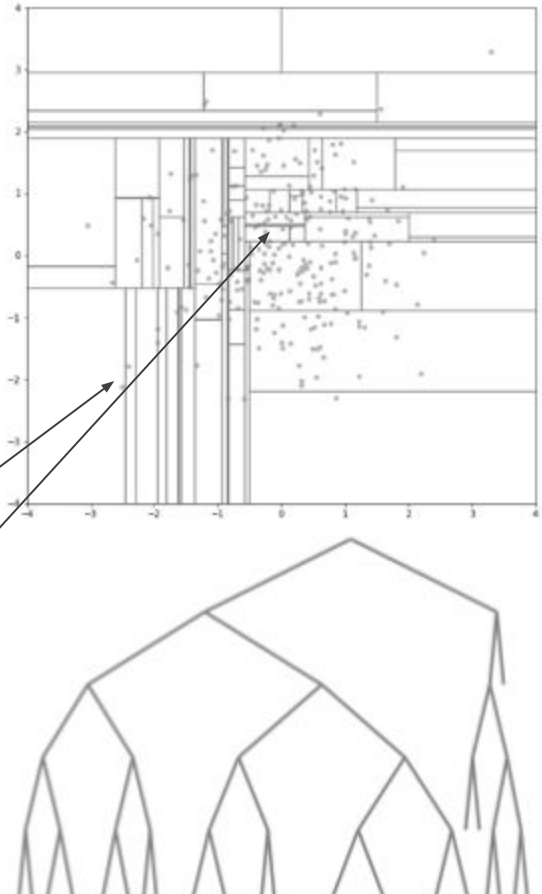
Isolation based - Isolation Forest

- ‘Few and different’ can be isolated quicker than many and normal.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees -> forest



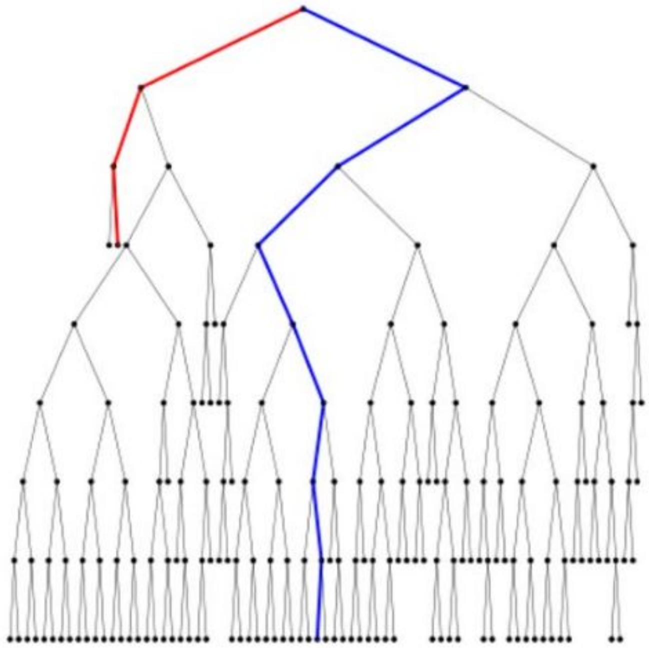
Isolation based - Isolation Forest

- 'Few and different' can be isolated quicker than many and normal.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees -> forest
- Anomalies will be isolated in a few steps - path length is smaller
- Normal points take more steps - path length is greater



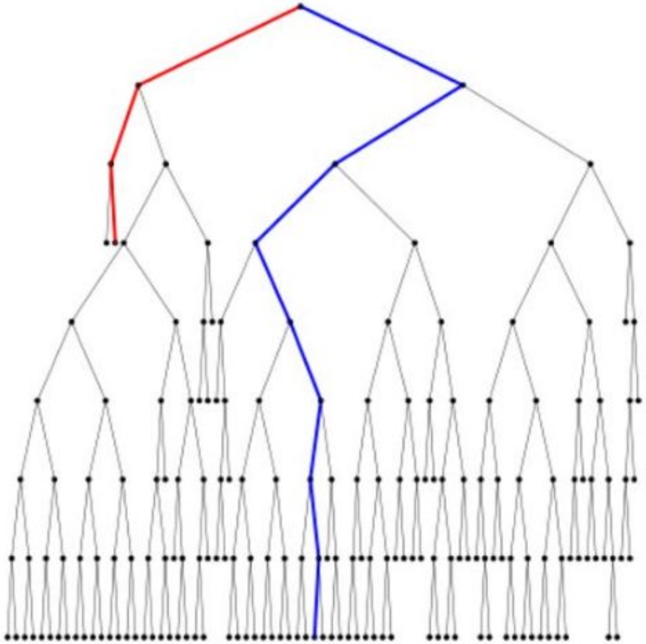
Isolation based - Isolation Forest

- The one in red shows the path of an anomaly data point from the root node to the terminating node.
- The one in blue shows the path of a normal data point from the root node to the terminating node.
- The average path length of the anomaly data point is lesser than the average path length of the normal data point.



Isolation based – Isolation Forest

- The anomaly score s of an instance x is given by:
  s(x, n) = 2^(- E(h(x))/c(n))
  where,
  - n -> the number of instances in the dataset
  - c(n) -> average path length of unsuccessful search in BST
  - E(h(x)) -> average of h(x) from a collection of isolation trees
  - h(x) -> path length of point x measured by number of edge traversals from root to terminal node
- If s ≈ 1, then they are definitely anomalies.
- If s ≈ 0.5, then no distinct anomaly.
- If s is much smaller than 0.5, then they can be safely regarded as normal instances.



Kernel based – ONE Class SVM

- The concepts are similar to the Support Vector Machine for binary classification.
- Recall that a regular SVM for classification finds a max-margin hyperplane that separates the positive examples from the negative ones.
- But instead, here we have only one class that we want to model.
- There are two different techniques of building a One Class SVM:
  - Finding a hypersphere in the feature space around the data while minimising the volume of this hypersphere.
  - Finding a hyperplane that separates all data points from origin in the feature space and maximises the distance from this hyperplane to the origin. [7]

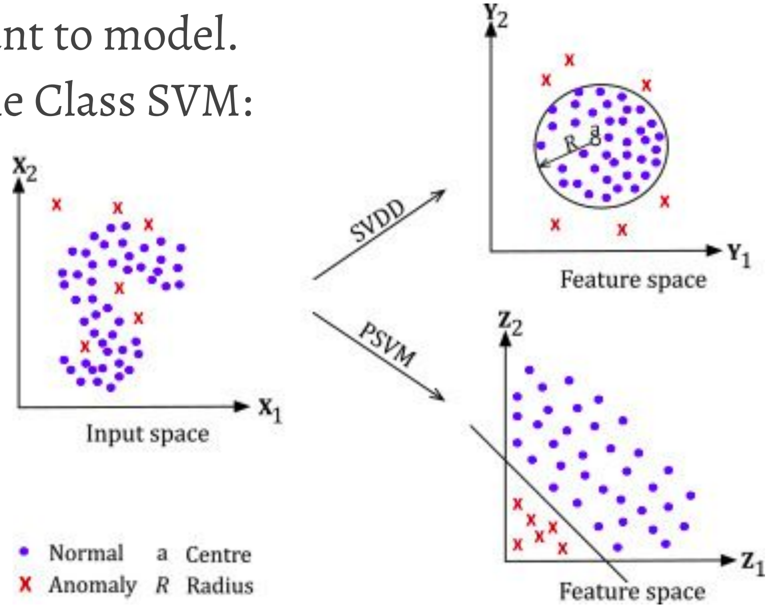


Image source: https://ars.els-cdn.com/content/image/1-s2.0-S0031320316300267-gr2.jpg

Factors to consider when choosing an anomaly Detector

Factors to consider when choosing an anomaly detector

1. Fewer parameters
2. Fast runtime
3. Low space complexity
4. Known behaviour under different data properties
5. Can deal with many different types of anomalies
6. Understanding the nature of anomaly and use appropriate technique/algorithm

Other applications of Anomaly Detectors

Other applications of anomaly detectors

1. Classification under streaming emerging new class - Instances of emerging new classes are 'outlying anomalies' w.r.t. the instances of known classes. The assumption is that anomalies of the known classes are more "normal" than the "outlying" anomalies.
2. Measuring similarity between two points - Compute the distance/density between two points and use it to find similarity.
3. Ranking, for example, in Information Retrieval - Uses anomaly score of the query in isolation trees to rank their relevance to the query

Thanks for listening!

Question and Answers

References

1. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples, Kimin Lee, Honglak Lee, Kibok Lee, Jinwoo Shin
2. https://scikit-learn.org/stable/modules/outlier_detection.html
3. Out-of-Distribution Detection in Deep Learning: A Survey, Saikiran Bulusu , Bhavya Kailkhura, Bo Li, Pramod K. Varshney, Dawn Song
4. Which Anomaly Detector should I use?, Kai Ming Ting & Sunil Aryal, Federation University Australia, Takashi Washio, Osaka University Tutorial on 20 November 2018 at IEEE International Conference on Data Mining, Singapore
5. https://stealthbits.com/blog/local-outlier-factor-part-1/
6. Liu, Fei Tony, Ting, Kai Ming and Zhou, Zhi-Hua. “Isolation forest.” Data Mining, 2008. ICDM’08. Eighth IEEE International Conference.

References

7.http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/

8. https://www.sciencedirect.com/science/article/abs/pii/S0360835205000100

9. LOF: Identifying Density-Based Local Outliers, Markus M. Breunig, Hans-Peter Kriegel , Raymond T. Ng , Jörg Sander