Machine Learning

Outlier Detection

Claudio Sartori

DISI
Department of Computer Science and Engineering – University of Bologna, Italy
claudio.sartori@unibo.it

Outlier Detection in Machine Learning: Overview

- Outlier detection involves identifying data points that deviate significantly from the majority of the dataset
- Such anomalies can indicate:
    - Noise or errors in the data
    - Rare but important events (e.g., fraud, equipment failure)
    - Variability in the underlying process being studied
- Applications include fraud detection, predictive maintenance, healthcare, and finance

Key Techniques for Outlier Detection

- Statistical Methods
  - Identify outliers based on assumptions of the data distribution
  - Examples:
    - Z-Score: Measures the distance from the mean in standard deviations
    - IQR (Interquartile Range): Identifies outliers using quartiles
- Distance-Based Methods
  - Measure the distance of each point to its neighbors
  - Examples:
    - k-Nearest Neighbors (k-NN)
    - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
- Model-Based Methods
  - Train machine learning models to distinguish normal from abnormal data
  - Examples:
    - Isolation Forest
    - Autoencoders for anomaly detection

Challenges in Outlier Detection

- High Dimensionality
  - Outliers become harder to detect in high-dimensional spaces
  - Distance measures lose significance due to the curse of dimensionality
- Imbalanced Data
  - Outliers are rare, making standard classification approaches less effective
- Noise in Data
  - Differentiating between true outliers and random noise can be challenging
- Dynamic Datasets
  - Continuous data streams may introduce new patterns over time

Evaluation Metrics for Outlier Detection

- Precision
  - Fraction of detected outliers that are true outliers
- Recall
  - Fraction of true outliers that are successfully detected
- F1-Score
  - Harmonic mean of precision and recall
- Area Under Curve (AUC)
  - Evaluates model performance across various thresholds
- Execution Time
  - Important for real-time applications such as fraud detection or predictive maintenance

Applications of Outlier Detection

- Fraud Detection
    - Identify unusual patterns in transactions that indicate fraud
- Predictive Maintenance
    - Detect anomalies in sensor data to predict equipment failures
- Healthcare
    - Identify rare disease cases or anomalies in medical imaging data
- Finance
    - Detect unusual trading activity or financial irregularities
- Network Security
    - Identify abnormal network traffic patterns signaling potential attacks

Significance of Outlier Detection

- Enhancing Data Quality
  - Removing or correcting outliers improves model accuracy
- Critical Insights
  - Detecting rare events can lead to significant operational improvements
- Preventing Losses
  - Early detection of anomalies reduces financial, operational, or reputational damage

Anomaly detection

- Find objects that are different from most other objects
- How to measure such dissimilarity/exceptionality/inconsistency?
- How to explore the data set to find outliers?
- Anomaly does not imply necessarily a small number
- An anomaly can also be caused by errors in data collection

Synonim

anomaly $\leftrightarrow$ outlier

Focus on some applications of anomaly detection - I

- Fraud detection

  Example Change in purchasing behaviour for credit card customers

- Network intrusion detection

  Example Monitor packets in communication networks to discover attacks

- Ecosystem disturbances

  Example Predict hurricanes, floods, etc. on the basis of metereological parameters

Focus on some applications of anomaly detection - II

- Medical diagnosis
  - Example: Unusual sympthoms can indicate potential health problems
- Public health
  - Example: Anomalous diseases can indicate problems in vaccination campaign

Causes of anomalies - I

Data from different classes

Most item of the previous slide are examples of anomalies that represent a different class of objects

Hawkins' definition of an Outlier

An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism

Causes of anomalies - II

Natural variation
When data can be modeled by a normal distribution, most objects are near a center, extreme values have low likelihood, nevertheless they can be interesting

Data measurement and Collection errors
Incorrect recordings due to human errors, device-related errors, noise; removal of such errors is usually named Data Cleaning

Approaches to Anomaly Detection - I

Model–based techniques

Build a model of the data (e.g. estimate the parameters of a probability distribution): outliers will fit poorly in the model

- if the model is a set of clusters, then the outlier will not fit well to any cluster
- if the model is a regression then the outlier will be far from the predicted value
- classification-based techniques could fail, because it is difficult to build a model of the, relatively rare, anomalies

Approaches to Anomaly Detection - II

Proximity–based techniques
: Anomalous objects are those that are distant from most of the other objects In two or three dimension, scatter plots allow visual identification of anomalies

Density–based techniques
: Density follows straightforwardly from the proximity measure. Object in low-density regions can be considered outliers. Density can be measured extensively or estimated with some approximation

The use of class labels - I

Supervised There exists a training set with both anomalous and normal objects

- objects labelled as normal or anomalous
- problem of imbalanced classes

Unsupervised Labels are not available

- learn from the training set a way to assign to each object a score reflecting the degree of anomaly
- anomalies should be different one from the other

The use of class labels - II

Semi–supervised The training set contains only normal objects

- compute the anomaly score from the information available for normal objects
- in this case a relation among anomalies does not affect the result
- a.k.a. one class classification

Issues - I

Number of attributes used

- single attribute values can be anomalous, e.g. person's height of mt0.40
- common values can be anomalous when considered together, e.g. a person with (height=mt1.50,weight=kg120)

Global versus Local Perspective

An object may seem unusual w.r.t. all object, but usual w.r.t. its neighborhood

Degree of Anomaly

Instead of a binary decision, the degree allows to set a threshold that can be adjusted in a tuning step

Issues - II

Operation

Discover one-anomaly-at-a-time versus many-anomalies-at-once

- find the most anomalous object, remove it from the data set and loop
- find a set of anomalous objects
- the latter is prone to the problem of masking, e.g. several similar anomalies mask each other
- and also to the problems of swamping, e.g. the anomalies distort the data model and normal objects seem to be anomalous

Issues - III

Evaluation

Usual measures for the evaluation of classifiers (precision, recall, ...) are uneffective due to the unbalancing of normal and anomalous class

Efficiency

Classification and statistical methods are usually expensive to set up but lightweight run–time; proximity methods in principle should compare each object with all the others, and tend to have $O(n^2)$ complexity

Probabilistic definition of an Outlier

Definition

An object that has a low probability w.r.t. a probability distribution model of the data

- Probability distribution model created from the data by estimating the parameters for a user–specified distribution
- Statistical tests to identify discordant observations

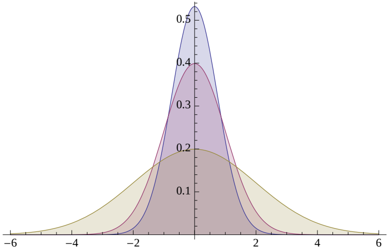
Issues in probabilistic definition

Identifying the specific distribution
: E.g. Gaussian, Poisson, Binomial, . . . , a wrong model invalidates the results

Number of attributes used
: There are techniques available for multi–variate data

Mixtures of distributions
: Model the data as a mixture of different distributions, usually of the same type but with different parameters (e.g. mixture of Gaussians and Expectation Maximization algorithm)

Detecting Outliers in a Univariate Normal Distribution



Normal PDF for $\sigma = 0.75, 1, 2$

Definition

An object with an attribute value x from a Normal distribution $Nb(0, 1)$ is an outlier if $|x| \geqslant c$, where $prob(|x|) \geqslant c = \alpha$

α is the probability of false positive, i.e. that a regular object is labeled as outlier

Strengths and Weaknesses

- strong theoretical foundations, well established techniques, such as parameter estimation
- plenty of methods for "outlierness" tests for univariate data
- fewer methods available for multivariate data
- bad performance with high-dimensional data

Proximity-based Outlier Detection

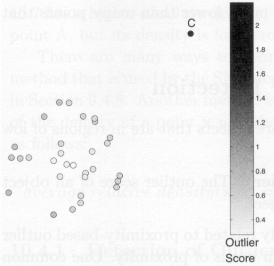
- An object is anomalous if it is distant from most points
- Relies on a proximity measure
- For each object, make (in principle) a sorted list of its neighbors, according to proximity

Distance to k–Nearest Neighbor

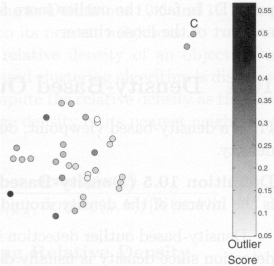
The outlier score of an object is the distance to its k–nearest neighbor

- Highly sensitive to the value of k

Outlier score - I

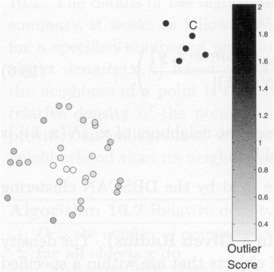


Outlier score based on 5–th nearest neighbor

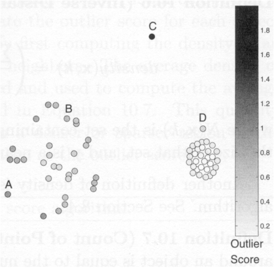


Outlier score based on 1-st nearest neighbor. Nearby outliers have low outlier scores

Outlier score - II



Outlier score based on
5–th nearest neighbor
A small cluster becomes
a set of outliers



Outlier score based on
5–th nearest neighbor
Clusters of different density

Proximity–based solutions

- Problem: finding the top m outliers in a data set
- Based on the notion of distance of the k-th Nearest Neighbors
- Brute force solution has complexity $\mathcal{O}(N^2)$
- There exists much more efficient algorithms

The Bay's algorithm [Bay and Schwabacher(2003)]

A simple nested loop with simple pruning

- For each example in $\mathcal{X}$ keep track of the k nearest neighbors found so far
- Determine the cutoff value of the score as the distance of the k-th nearest neighbor of the top m-th outlier found so far
- When an example achieves a score lower than the cutoff it is removed, because it can no longer be an outlier
- Later iterations find increasing scores, and the efficiency of pruning is increasing
- If data are in random order the average complexity is near linear
- Worst case complexity is $\mathcal{O}(N^2)$

Density-based Outlier Detection

Intuition

Outliers are found in low-density areas

Definition - Density–based outlier

The outlier score of an object is the inverse of the density around the object

Definition - Inverse distance

$$density(\mathrm{x}, k) = \left(\frac{\sum_{y \in Nb(\mathrm{x},k)} distance(\mathrm{x}, \mathrm{y})}{k} \right)^{-1}$$

where $Nb(\mathrm{x}, k)$ is the set containing the k–nearest neighbors of x

In practice, this is the inverse of the average distance to the objects in its neighborhood

Alternative definition of density

Definition - Count of Points within a Given Radius

The density around an object is defined as the number of objects that are within a specified distance d from the object

Issues in the choice of d

If d too small the density can be underestimated, with an effect similar to the choice of k too large in distance-based methods

Average Relative Density

- Problems in identification of outliers when data contains regions of different densities
- Find a definition relative to the neighborhood of the object
- Example: in slide 30, right, point D has higher absolute density than point A, but its density is lower relative to its nearest neighbors Jump to figure

$$\mathit{average\ relative\ density}(\mathrm{x}, k) = \frac{\mathit{density}(\mathrm{x}, k)}{\sum_{y \in N(\mathrm{x},k)} \mathit{density}(\mathrm{y}, k)/k}$$

It is the density normalized to the average density of the objects in the neighborhood

Detection of Outliers using Relative Density

Algorithm 1: Detection of Outliers using Relative Density

Input: k: number of nearest neighbors

Output: Outlier scores $outlier\_score(x, k)$ for all objects x

foreach object x **do**

  Determine $Nb(x, k)$, the k–nearest neighbors of x.

  Compute $density(x, k)$ using the objects in $Nb(x, k)$.

foreach object x **do**

  Set $outlier\_score(x, k) \leftarrow average\_relative\_density(x, k)$.

Strengths and weaknesses

- works well when data has regions of different density
- natural complexity $O(N^2)$ in the number of objects
- can be reduced to $O(N \log N)$ for low–dimensional data with special data structures
- parameter selection quite difficult

Isolation Forest: Overview

- Isolation Forest (iForest) is an unsupervised anomaly detection algorithm
- It isolates anomalies by leveraging the fact that they are "few and different."
- Instead of profiling normal data, iForest directly isolates anomalies

Core Concept

- Isolation Principle
  - Anomalies are easier to isolate compared to normal points
  - Isolation is performed using random splits on data dimensions
- Tree Construction
  - Randomly select a feature and split value for each node
  - Continue splitting until:
    - A single data point remains in the partition
    - A maximum depth is reached
- Path Length
  - The depth of a point in the tree measures how quickly it gets isolated
  - Anomalies have shorter path lengths compared to normal points

Mathematical Details

- Path Length for a Point
  - Let $h(x)$ denote the path length of a point x in a tree
  - For n data points, the average path length $c(n)$ is approximated as:
    - $c(n) = 2H(n-1) - \frac{2(n-1)}{n}$,
    - where $H(i)$ is the i-th harmonic number $H(i) = \sum_{k=1}^{i} \frac{1}{k}$
- Anomaly Score
  - The anomaly score $s(x)$ is calculated as:
    - $s(x) = 2^{-\frac{E(h(x))}{c(n)}}$,
    - where $E(h(x))$ is the average path length of x across all trees
  - Interpretation:
    - $s(x) \to 1$: x is an anomaly
    - $s(x) \to 0.5$: x is a normal point

Computational Complexity

- Tree Construction
  - For n data points and t trees, the complexity is:
    - $O(t \cdot n \cdot \log(n))$,
    - as each tree requires $O(n \log(n))$ time
- Scalability
  - The algorithm scales well with large datasets due to its linear complexity in n
- Memory Efficiency
  - Each tree is built using a random subset of data (sub-sampling)
  - Reduces memory usage and improves efficiency

Advantages of Isolation Forest

- Unsupervised
  - No labels are required for training
- Efficient
  - Linear time complexity with respect to the number of samples
- Handles High Dimensions
  - Works well with datasets having many features
- Interpretability
  - Path lengths provide an intuitive understanding of anomalies

Limitations of Isolation Forest

- Random Splits
  - May lead to inconsistent results for small datasets
- Assumes Independent Features
  - Ignores correlations between features during splitting
- Suboptimal for Complex Data
  - May require fine-tuning for datasets with intricate patterns

Applications of Isolation Forest

- Fraud Detection
  - Identify fraudulent transactions based on behavioral patterns
- Predictive Maintenance
  - Detect sensor anomalies indicating potential equipment failures
- Network Security
  - Spot unusual network activity that could indicate cyberattacks
- Healthcare
  - Identify rare but critical events in patient monitoring data

Outlier Detection Methods in sklearn

Method	Class	Type	Key idea / notes
Local Outlier Factor	`LocalOutlierFactor`	Unsupervised	Compares local density of a point to its neighbors to detect anomalies.
Isolation Forest	`IsolationForest`	Unsupervised / novelty	Isolates points using random splits; anomalies are easier to isolate, and it scales to large datasets.
One-Class SVM	`OneClassSVM`	Novelty detection	Learns a boundary enclosing most of the data in a high-dimensional feature space.
Elliptic Envelope	`EllipticEnvelope`	Outlier detection	Fits a robust Gaussian (elliptical) model to identify low-likelihood points.
Robust Covariance	`MinCovDet`	Robust statistics	Estimates covariance robustly; used to compute Mahalanobis distance for outlier scoring.
DBSCAN	`DBSCAN`	Clustering	Density-based clustering; points labeled `-1` are treated as outliers (noise).

Bibliography I

- Stephen D. Bay and Mark Schwabacher.
  Mining distance-based outliers in near linear time with randomization and a simple pruning rule.
  In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003, pages 29–38, 2003.
  doi: 10.1145/956750.956758.
  URL http://doi.acm.org/10.1145/956750.956758.
- Edwin M. Knorr and Raymond T. Ng.
  Algorithms for mining distance-based outliers in large datasets.
  In Proceedings of VLDB Conference, 1998.