Clustering and classification-based outlier detection approaches

Introduction

Clustering-Based Approaches

- Unsupervised methods so no labeled data required
- Detect outliers by examining relationships between data objects and clusters.
- Outliers belong to small or remote clusters or none at all.
- Highly related to the concept of clustering

Classification-Based Approaches

- Supervised methods so requires labeled training data
- Treat outlier detection as a classification problem.
- Trains a model to differentiate normal data from outliers using labeled training data.

Clustering-Based Approaches

Three Clustering-based Approaches

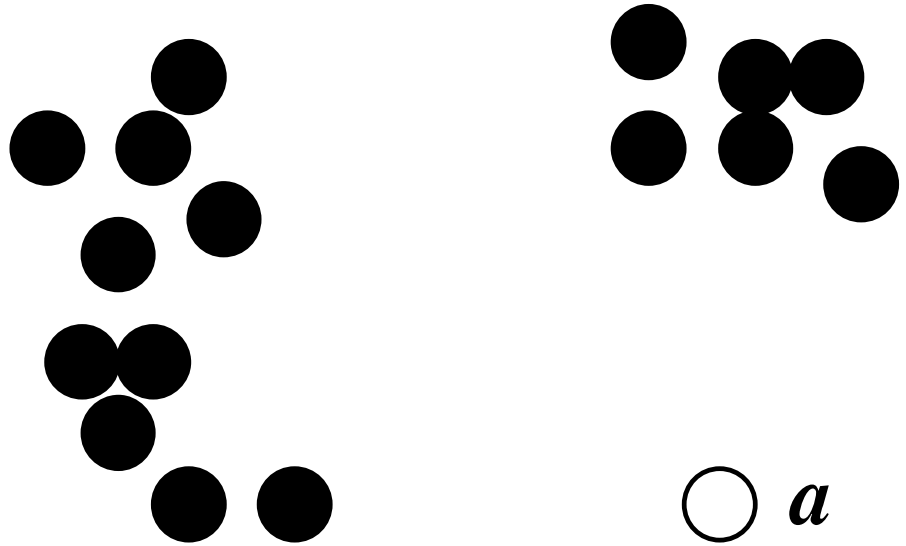
- **Approach 1**: Does the object belong to any cluster? If not, then it is identified as an outlier.

- **Approach 2**: Is there a large distance between the object and the cluster to which it is closest? If yes, it is an outlier.

- **Approach 3**: Is the object part of a small or sparse cluster? If yes, then all the objects in that cluster are outliers.

Approach 1: Detecting outliers as objects that do not belong to any cluster

- Example: Animals moving in flocks (e.g., goats, deer)
- Using outlier detection, we can identify outliers as animals that are not part of a flock. Such animals may be either lost or wounded.
- Using a density-based clustering method, such as **DBSCAN**, we note that the black points belong to clusters.
- The white point, a, does not belong to any cluster, and thus is declared an outlier.

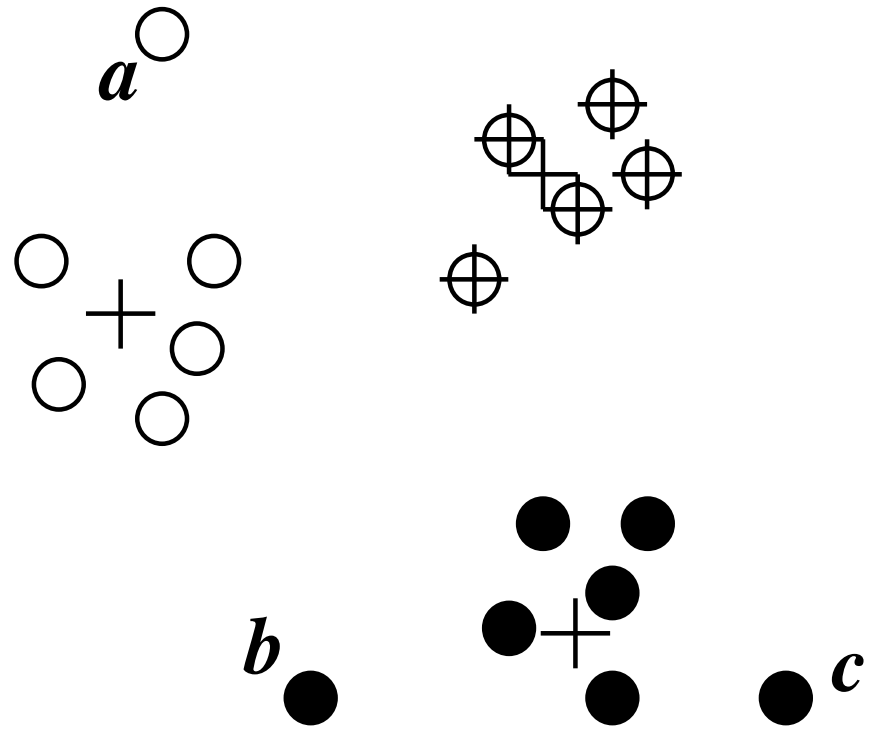


Approach 2: Using distance to the closest cluster

- The **k-means clustering** method groups data into clusters, each represented by a center marked with a plus (+) sign.
- An outlier-ness score can be calculated for each data point to quantify how much of an outlier it is.
- A higher score suggests that the point is an outlier.
- The outlier score for object o is calculated this way:

$$\frac{dist(o, c_o)}{l_{c_o}}$$

- Where c_o is the closest center to object o and l_{c_o} is the average distance between c_o and the objects assigned to c_o
- In the example, points labeled a, b, c are outliers because they are significantly distant from any cluster center.

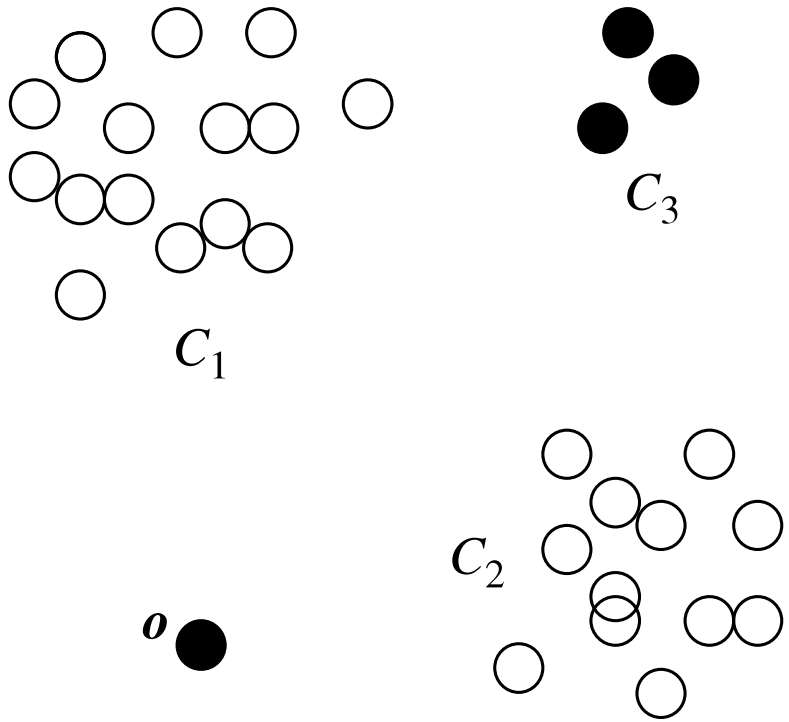


Intrusion detection using Approach 2

- **Step 1: Pattern Identification**
  - The TCP connection data is divided into segments (e.g., by dates), and frequent itemsets (common patterns) in these segments are identified as "**base connections**," representing normal data patterns.
- **Step 2: Clustering**
  - Connections that contain these base connections are grouped together, assuming they are normal and not under attack.
- **Step 3: Outlier Detection**
  - The original data points are then compared to these clusters. Any connection that doesn't fit well with the clusters (an outlier) is flagged as a possible attack.

Approach 3: Detecting outliers in small clusters

- Previous methods detect outliers by comparing individual data points to larger data clusters one by one.
- In large datasets, some outliers may not be isolated but can form their own small clusters.
- In cases like intrusion detection, hackers using similar methods might create such a small cluster, which can be misleading for traditional outlier detection.
- To address this issue, a new method identifies entire small or sparsely populated clusters as outliers.
- The FindCBLOF algorithm is an example of this third approach that targets small clusters to find outliers.



FindCBLOF Step 1: Cluster Identification and Size Categorization

- **Clusters are sorted** by size (number of points in each cluster) in descending order.
- A **threshold parameter (α)** is used to differentiate between **large clusters** and **small clusters**:
  - Clusters containing **≥ α% of the data points** are considered **large**.
  - The rest are labeled as **small clusters**.

FindCBLOF Step 2: Assigning CBLOF Scores

- **For points in large clusters**:
  - Their **CBLOF score** is the product of:
    - The size of the cluster (number of points).
    - The **similarity** between the point and the cluster.
- **For points in small clusters**:
  - Their **CBLOF score** is the product of:
    - The size of the small cluster.
    - The **similarity** between the point and the **closest large cluster**.

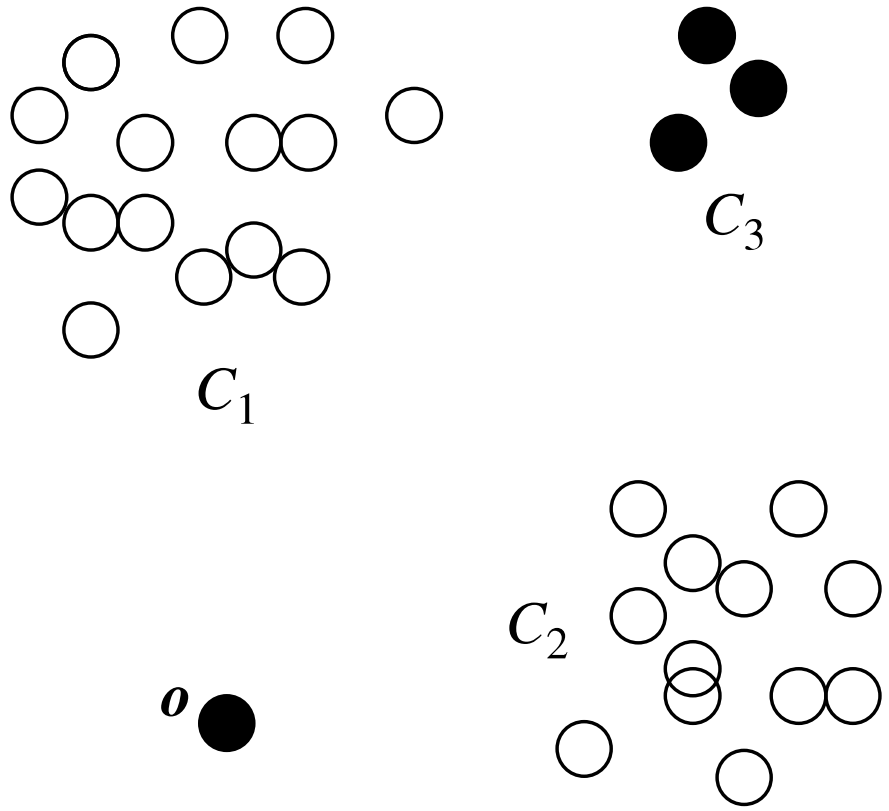
Note: The similarity is the inverse of the distance you see on the the plots.

FindCBLOF Step 3: Identifying Outliers

- **CBLOF scores** represent how well a point matches its cluster.
- Points with the **lowest CBLOF scores** are flagged as **outliers** because:
- They are in small clusters or are far from their closest large cluster.

FindCBLOF Example: Detecting outliers in small clusters

- The data is divided into three clusters: two large ones (C1 and C2) and one small cluster (C3). There is also an object 'o' that doesn't belong to any cluster.
- The FindCBLOF algorithm identifies 'o' and the points in the small cluster C3 as outliers because they are significantly different from the points in the larger clusters C1 and C2.
- For object 'o', the closest large cluster is C1, but it shares a low similarity with C1, thus it is considered an outlier.
- For points in the small cluster C3, the closest large cluster is C2. Despite there being three points in C3, their low similarity with C2 and the small size of C3 means these points also receive high CBLOF scores, marking them as outliers.



Textbook Example 11.22

Strengths and Weaknesses of clustering-based outlier detection approaches

Strengths

- **Unsupervised**: No labeled data required.
- Applicable to **various data types**.
- Clusters serve as **data summaries**, reducing the need to analyze all objects individually.
- **Fast outlier detection** once clusters are created, as the number of clusters is usually small.

Weaknesses

- Effectiveness depends on the clustering method used.
- Not always optimized for outlier detection.
- Costly for large data sets, which can create performance bottlenecks.