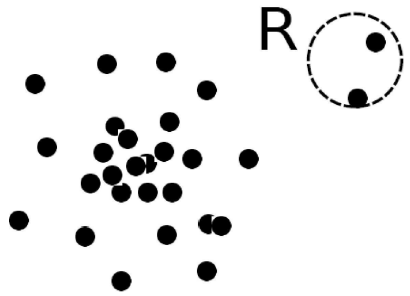
Outlier Mining

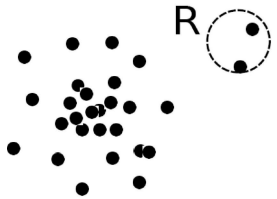
Noise refers to random variations or irregularities in data that do not carry meaningful information.

What Are Outliers?

- Outlier: A data object that **deviates significantly** from the normal objects as if it were **generated by a different mechanism**
  - Ex.: Unusual credit card purchase, sports: Mo Salah, …
- Outliers are different from the noise data
  - Noise is random error or variance in a measured variable
  - Noise should be removed before outlier detection
- Outliers are interesting: They violate the mechanism that generates the normal data
- Outlier detection vs. novelty detection: early stage, outlier; but later merged into the model
- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Medical analysis



Types of Outliers (I)



Global Outlier

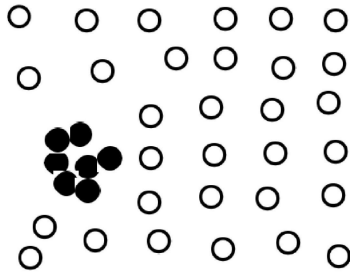
- Three kinds: *global*, *contextual* and *collective* outliers
- **Global outlier** (or point anomaly) significantly deviates from the rest of the data set
  - ex. Very high speed
  - How to find ? Assuming data is clustered, find an appropriate measurement of deviation.

- **Contextual outlier** (or *conditional outlier*) deviates significantly based on a selected context
  - Is 30º C an outlier ? Is 40km/h on the highway and outlier ?
  - Attributes of data objects should be divided into two groups
    - Contextual attributes: defines the context, e.g., time & location
    - Behavioral attributes: characteristics of the object, used in outlier evaluation, e.g., temperature
  - Issue: How to define or formulate meaningful context?

Types of Outliers (II)

Collective Outliers, a subset of data objects *collectively* deviate significantly from the whole data set, even if the individual data objects may not be outliers

e.g., intrusion detection: (checking the personal data of one client vs. of 100 clients), one car deviating from the shortest path vs. 100 cars doing so.



Collective Outlier

- **Detection** of collective outliers
  - Consider not only behavior of individual objects, but also that of groups of objects
  - Need to have the background knowledge on the relationship among data objects, such as a distance or similarity measure on objects.
- A data set may have multiple types of outlier

Challenges of Outlier Detection

- Modeling normal objects and outliers properly
  - Hard to enumerate all possible normal behaviors in an application
  - The border between normal and outlier objects is often a gray area
- Application-specific outlier detection
  - Choice of distance measure among objects and the model of relationship among objects are often application-dependent
  - E.g., clinic data: a small deviation could be an outlier; while in marketing analysis, larger fluctuations
- Handling noise in outlier detection
  - Noise may distort the normal objects and blur the distinction between normal objects and outliers. The detection can confuse noise and outliers.
- Understandability
  - Understand why these are outliers: Justification of the detection
  - Specify the degree of an outlier: the unlikelihood of the object being generated by a normal mechanism

Outlier Detection I: Supervised Methods

- Two ways to categorize outlier detection methods:
  - Based on whether user-labeled examples of outliers can be obtained:
    - Supervised, semi-supervised vs. unsupervised methods
  - Based on assumptions about normal data and outliers:
    - Statistical, proximity-based, and clustering-based methods
- Outlier Detection I: **Supervised Methods**
  - Modeling outlier detection as a classification problem
    - Samples examined by domain experts used for training & testing
  - One class classification:
    - Model normal objects & report those not matching the model as outliers, or
  - Challenges
    - Imbalanced classes, i.e., outliers are rare
    - Catch as many outliers as possible, i.e., recall is more important than accuracy

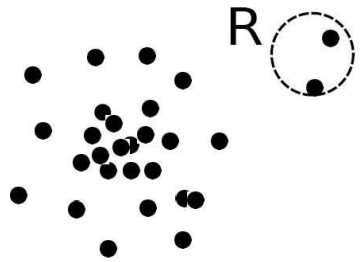
Outlier Detection II: Unsupervised Methods

- Assume the normal objects are somewhat *clustered* into multiple groups, each having some distinct features
- An outlier is expected to be far away from any groups of normal objects
- Weakness: Cannot detect collective outlier effectively
  - Normal objects may not share any strong patterns, but the collective outliers may share high similarity in a small area

- Many clustering methods can be adapted for unsupervised methods
  - Find clusters, then outliers: not belonging to any cluster
  - Problem 1: Hard to distinguish noise from outliers
  - Problem 2: Costly since first clustering: but far less outliers than normal objects

Outlier Detection (2): Proximity-Based Methods

- An object is an outlier if the nearest neighbors of the object are far away
- Model the proximity of an object using its 3 nearest neighbors
  - Objects in region R are substantially different from other objects in the data set.
  - Thus the objects in R are outliers



- The effectiveness of proximity-based methods highly relies on the proximity measure.
- Often have a difficulty in finding a group of outliers which stay close to each other
- Two major types of proximity-based outlier detection
  - Distance-based vs. density-based
  - Distance-based outlier detection: An object o is an outlier if its k-nearest neighborhood happens at a relatively large distance
  - Density-based outlier detection: An object o is an outlier if its density is relatively much lower than that of its neighbors

Distance-Based Outlier Detection

Formally, let r $(r \geq 0)$ be a *distance threshold* and π $(0 < \pi \leq 1)$ be a fraction threshold. An object, $\boldsymbol{o}$, is a $DB(r, \pi)$-**outlier** if

$$\frac{\|\{\boldsymbol{o}' | dist(\boldsymbol{o}, \boldsymbol{o}') \leq r\}\|}{\|D\|} \leq \pi, \tag{12.10}$$

where $dist(\cdot, \cdot)$ is a distance measure.

Equivalently, we can determine whether an object, $\boldsymbol{o}$, is a $DB(r, \pi)$-outlier by checking the distance between $\boldsymbol{o}$ and its k-nearest neighbor, $\boldsymbol{o}_k$, where $k = \lceil \pi \|D\| \rceil$. Object $\boldsymbol{o}$ is an outlier if $dist(\boldsymbol{o}, \boldsymbol{o}_k) > r$, because in such a case, there are fewer than k objects except for $\boldsymbol{o}$ that are in the r-neighborhood of $\boldsymbol{o}$.

Distance-Based Outlier Detection

Algorithm: Distance-based outlier detection.

Input:

- a set of objects $D = \{o_1, \ldots, o_n\}$, threshold r ($r > 0$) and π ($0 < \pi \leq 1$);

Output: $DB(r, \pi)$ outliers in D.

Method:

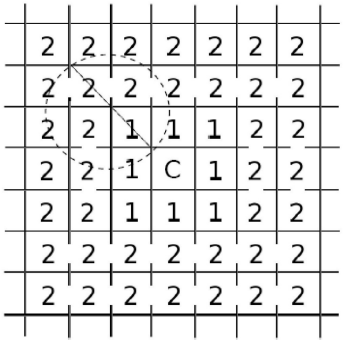
```
for i = 1 to n do
    count ← 0
    for j = 1 to n do
        if i ≠ j and dist(o_i, o_j) ≤ r then
            count ← count + 1
            if count ≥ π · n then
                exit {o_i cannot be a DB(r, π) outlier}
            endif
        endif
    endfor
    print o_i {o_i is a DB(r, π) outlier according to (Eq. 12.10)}
endfor;
```

Distance-Based Outlier: A Grid-Based Method

- Why efficiency is still a concern? When the complete set of objects cannot be held into main memory, cost I/O swapping
- The major cost: (1) each object tests against the whole data set, why not only its close neighbor? (2) check objects one by one, why not group by group?
- Grid-based method (CELL): Data space is partitioned into a multi-D grid. Each cell is a hyper cube with diagonal length r/2

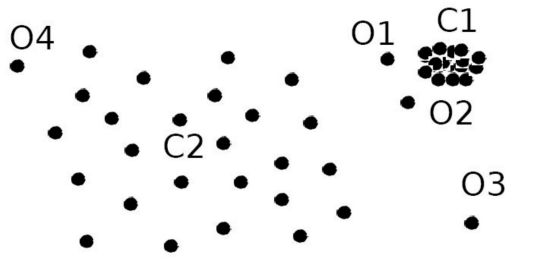
Level-1 cell pruning rule: Based on the level-1 cell property, if $a + b_1 > \lceil \pi n \rceil$, then every object o in C is not a $DB(r,\pi)$-outlier because all those objects in C and the level-1 cells are in the r-neighborhood of o, and there are at least $\lceil \pi n \rceil$ such neighbors.

Level-2 cell pruning rule: Based on the level-2 cell property, if $a + b_1 + b_2 < \lceil \pi n \rceil + 1$, then all objects in C are $DB(r,\pi)$-outliers because each of their r-neighborhoods has less than $\lceil \pi n \rceil$ other objects.



Density-Based Outlier Detection

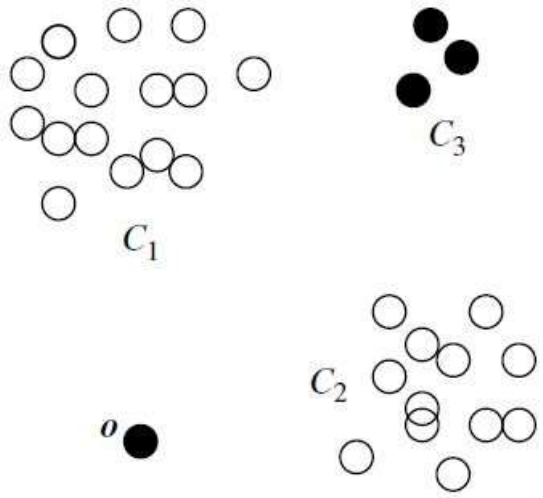
- Contextual outliers: Outliers comparing to their local neighborhoods, instead of the global data distribution
- Distance-based outlier detection won't work.



- Intuition (density-based outlier detection): The density around an outlier object is significantly different from the density around its neighbors
- Method: Use the relative density of an object against its neighbors as the indicator of the degree of the object being outliers
- k-distance of an object o, $dist_k(o)$: distance between o and its k-th NN
- k-distance neighborhood of o, $N_k(o) >= \{o' |\ o' \text{ in } D, dist(o, o') \leq dist_k(o)\}$
  - $N_k(o)$ could be bigger than k since multiple objects may have identical distance to o

Clustering-Based Methods

- An object is an outlier if
  - (1) it does not belong to any cluster,
  - (2) there is a large distance between the object and its closest cluster ,
  - or (3) it belongs to a small or sparse cluster

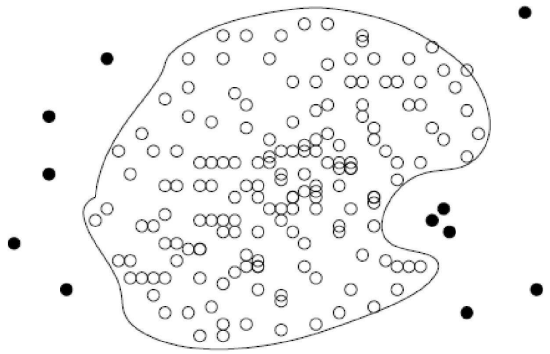


One-Class Model- classify instances into a single class or category. This means that the model is trained to recognize and assign instances to one specific class, and all other instances are considered as anomalies or outliers.
One-class models are often used in anomaly detection or novelty detection tasks, where the focus is on identifying instances that deviate from the norm rather than classifying them into multiple categories.

Two-Class Model- In a two-class model, the objective is to classify instances into one of two classes or categories. The most common scenario is binary classification, where the two classes are often labeled as positive and negative, or class 0 and class 1. Binary classification is used in a wide range of applications, such as spam detection (spam or not spam), disease diagnosis (diseased or not diseased), and many others.

Classification-Based Method I: One-Class Model

- Two-class classification
  Idea: Train a classification model that can distinguish "normal" data from outliers
  - Requires many abnormal samples.
  - Abnormal might not well cluster.



- One-class model: A classifier is built to describe only the normal class.
  - Learn the decision boundary of the normal class
  - Any samples that do not belong to the normal class (not within the decision boundary) are declared as outliers
  - Adv: can detect new outliers that may not appear close to any outlier objects in the training set
  - Extension: Normal objects may belong to multiple classes

Credits and Readings

- These slides, except when explicitly stated, use material from:
  - Charu C.Aggarwal. Data Mining The Textbook, Springer.
  - Data Mining: Concepts and Techniques, 3rd ed. Slides by Jiawei Han.

One-Class Classifier:

Purpose: A one-class classifier, also known as a one-class SVM (Support Vector Machine) or anomaly detection model, is designed to identify instances that deviate from the norm or are considered outliers.
Training Data: Typically trained on a single class, representing the normal or majority class. It learns the characteristics of this class and is then able to detect instances that do not conform to these characteristics during testing.
Use Case: Anomaly detection is a common use case for one-class classifiers. For example, fraud detection in financial transactions, network intrusion detection, or identifying defective products in manufacturing.

Two-Class Classifier:
Purpose: A two-class classifier is designed to classify instances into one of two classes - a binary classification problem. It aims to learn a decision boundary that separates the two classes based on the features provided.
Training Data: Requires labeled data with instances belonging to either the positive class or the negative class. The model learns to distinguish between these two classes.
Use Case: Common applications include spam detection (spam or not spam), sentiment analysis (positive or negative sentiment), medical diagnosis (healthy or diseased), and more.
Differences:
Training Data: One-class classifiers are trained on a single class, while two-class classifiers require data with instances from both classes.
Objective: One-class classifiers focus on identifying anomalies or deviations from the norm, while two-class classifiers aim to classify instances into one of two categories.