CS 591.03
Introduction to Data Mining
Instructor: Abdullah Mueen

LECTURE 9: OUTLIER DETECTION

Chapter 12. Outlier Analysis

Outlier and Outlier Analysis

Outlier Detection Methods

Statistical Approaches

Proximity-Base Approaches

Clustering-Base Approaches

Classification Approaches

Mining Contextual and Collective Outliers

Outlier Detection in High Dimensional Data

Summary

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: Michael Jordon, Wayne Gretzky, ...

Outliers are different from the noise data

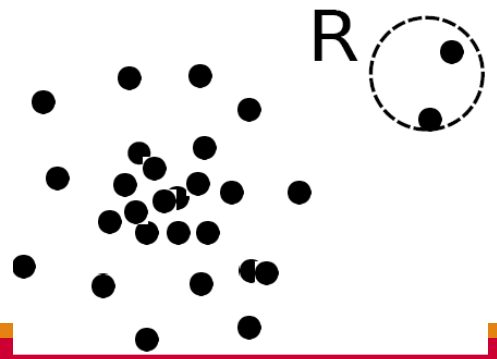
- Noise is random error or variance in a measured variable
- Noise should be removed before outlier detection

Outliers are interesting: It violates the mechanism that generates the normal data

Outlier detection vs. *novelty detection*: early stage, outlier; but later merged into the model

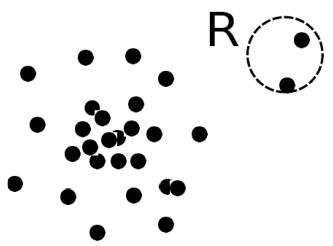
Applications:

- Credit card fraud detection
- Telecom fraud detection
- Customer segmentation
- Medical analysis



Types of Outliers (I)

Global Outlier

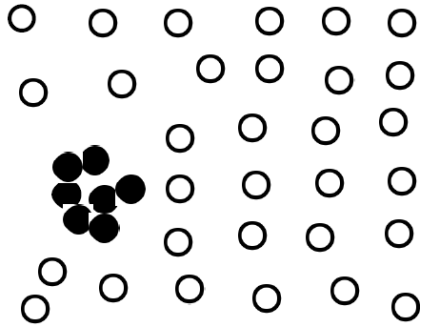


Three kinds: *global, contextual* and *collective* outliers

Global outlier (or point anomaly)
- Object is O_g if it significantly deviates from the rest of the data set
- Ex. Intrusion detection in computer networks
- Issue: Find an appropriate measurement of deviation

Contextual outlier (or *conditional outlier*)
- Object is O_c if it deviates significantly based on a selected context
- Ex. 80^o F in Urbana: outlier? (depending on summer or winter?)
- 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
- Can be viewed as a generalization of *local outliers*—whose density significantly deviates from its local area
- Issue: How to define or formulate meaningful context?

Types of Outliers (II)



Collective Outlier

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
- Applications: E.g., *intrusion detection*:
  - When a number of computers keep sending denial-of-service packages to each other
- 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
- One object may belong to more than one type 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. It may help hide outliers and reduce the effectiveness of outlier detection
- 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

Chapter 12. Outlier Analysis

Outlier and Outlier Analysis

Outlier Detection Methods

Statistical Approaches

Proximity-Base Approaches

Clustering-Base Approaches

Classification Approaches

Mining Contextual and Collective Outliers

Outlier Detection in High Dimensional Data

Summary

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
- Methods for Learning a classifier for outlier detection effectively:
  - Model normal objects & report those not matching the model as outliers, or
  - Model outliers and treat those not matching the model as normal
- Challenges
  - Imbalanced classes, i.e., outliers are rare: Boost the outlier class and make up some artificial outliers
  - Catch as many outliers as possible, i.e., recall is more important than accuracy (i.e., not mislabeling normal objects as outliers)

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

Ex. In some intrusion or virus detection, normal activities are diverse

- Unsupervised methods may have a high false positive rate but still miss many real outliers.
- Supervised methods can be more effective, e.g., identify attacking some key resources

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
  - Newer methods: tackle outliers directly

Outlier Detection III: Semi-Supervised Methods

Situation: In many applications, the number of labeled data is often small: Labels could be on outliers only, normal objects only, or both

Semi-supervised outlier detection: Regarded as applications of semi-supervised learning

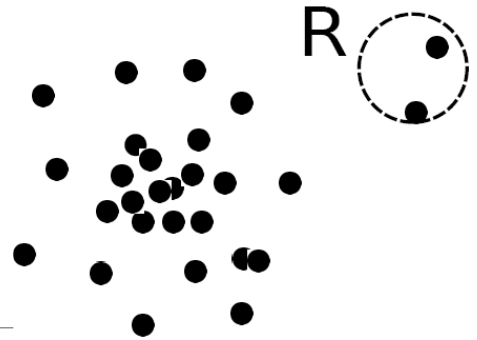
If some labeled normal objects are available

- Use the labeled examples and the proximate unlabeled objects to train a model for normal objects
- Those not fitting the model of normal objects are detected as outliers

If only some labeled outliers are available, a small number of labeled outliers many not cover the possible outliers well

- To improve the quality of outlier detection, one can get help from models for normal objects learned from unsupervised methods

Outlier Detection (1): Statistical Methods

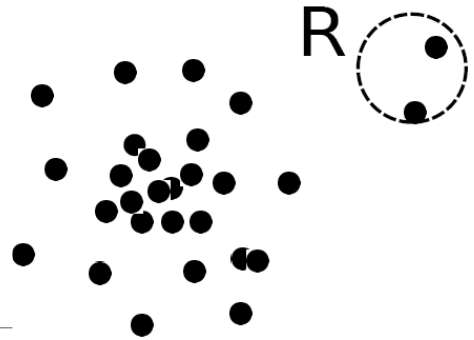


Statistical methods (also known as model-based methods) assume that the normal data follow some statistical model (a stochastic model)

- The data not following the model are outliers.

- Example (right figure): First use Gaussian distribution to model the normal data
  - For each object y in region R, estimate $g_D(y)$, the probability of y fits the Gaussian distribution
  - If $g_D(y)$ is very low, y is unlikely generated by the Gaussian model, thus an outlier
- Effectiveness of statistical methods: highly depends on whether the assumption of statistical model holds in the real data
- There are rich alternatives to use various statistical models
  - E.g., parametric vs. non-parametric

Outlier Detection (2): Proximity-Based Methods



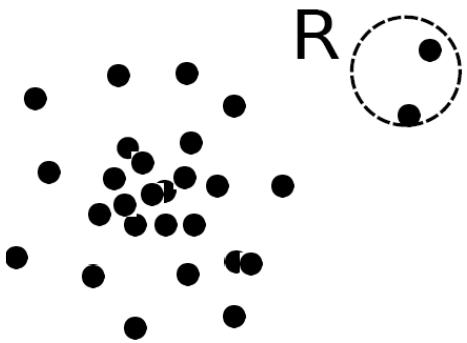
An object is an outlier if the nearest neighbors of the object are far away, i.e., the **proximity** of the object is **significantly deviates** from the proximity of most of the other objects in the same data set

- Example (right figure): 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.
- In some applications, proximity or distance measures cannot be obtained easily.
- 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

Outlier Detection (3): Clustering-Based Methods

Normal data belong to large and dense clusters, whereas outliers belong to small or sparse clusters, or do not belong to any clusters

- Example (right figure): two clusters
  - All points not in R form a large cluster
  - The two points in R form a tiny cluster, thus are outliers



- Since there are many clustering methods, there are many clustering-based outlier detection methods as well
- Clustering is expensive: straightforward adaption of a clustering method for outlier detection can be costly and does not scale up well for large data sets

Chapter 12. Outlier Analysis

Outlier and Outlier Analysis

Outlier Detection Methods

Statistical Approaches

Proximity-Base Approaches

Clustering-Base Approaches

Classification Approaches

Mining Contextual and Collective Outliers

Outlier Detection in High Dimensional Data

Summary

Statistical Approaches

Statistical approaches assume that the objects in a data set are generated by a stochastic process (a generative model)

Idea: learn a generative model fitting the given data set, and then identify the objects in low probability regions of the model as outliers

Methods are divided into two categories: *parametric* vs. *non-parametric*

Parametric method

- Assumes that the normal data is generated by a parametric distribution with parameter θ
- The probability density function of the parametric distribution $f(x, \vartheta)$ gives the probability that object x is generated by the distribution
- The smaller this value, the more likely x is an outlier

Non-parametric method

- Not assume an a-priori statistical model and determine the model from the input data
- Not completely parameter free but consider the number and nature of the parameters are flexible and not fixed in advance
- Examples: histogram and kernel density estimation

Parametric Methods I: Detection Univariate Outliers Based on Normal Distribution

Univariate data: A data set involving only one attribute or variable

Often assume that data are generated from a normal distribution, learn the parameters from the input data, and identify the points with low probability as outliers

Ex: Avg. temp.: {24.0, 28.9, 28.9, 29.0, 29.1, 29.1, 29.2, 29.2, 29.3, 29.4}

- Use the maximum likelihood method to estimate μ and σ

$$\ln \mathcal{L}(\mu, \sigma^2) = \sum_{i=1}^{n} \ln f(x_i|(\mu, \sigma^2)) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

  - Taking derivatives with respect to μ and σ^2, we derive the following maximum likelihood estimates

$$\hat{\mu} = \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$$

  - For the above data with n = 10, we have $\hat{\mu} = 28.61 \;\; \hat{\sigma} = \sqrt{2.29} = 1.51$
  - Then (24 – 28.61) /1.51 = – 3.04 < –3, 24 is an outlier since

$$\mu \pm 3\sigma \text{ region contains } 99.7\% \text{ data}$$

Parametric Methods I: The Grubb's Test

Univariate outlier detection: The Grubb's test (maximum normed residual test) — another statistical method under normal distribution

- For each object x in a data set, compute its z-score: x is an outlier if

$$z \geq \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N-2+t^2_{\alpha/(2N),N-2}}}$$

where $t^2_{\alpha/(2N),N-2}$ is the value taken by a t-distribution at a significance level of α/(2N), and N is the # of objects in the data set

Parametric Methods II: Detection of Multivariate Outliers

Multivariate data: A data set involving two or more attributes or variables

Transform the multivariate outlier detection task into a univariate outlier detection problem

Method 1. Compute Mahalaobis distance

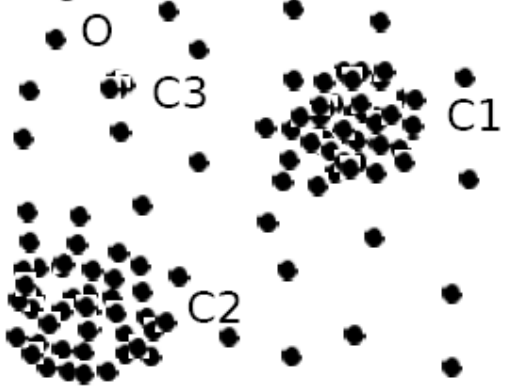
- Let ō be the mean vector for a multivariate data set. Mahalaobis distance for an object o to ō is $MDist(o, \bar{o}) = (o - \bar{o})^T S^{-1}(o - \bar{o})$ where S is the covariance matrix
- Use the Grubb's test on this measure to detect outliers

Method 2. Use χ^2 –statistic:

- where E_i is the mean of the *i*-dimension among all objects, and n is the dimensionality
- If χ^2 –statistic is large, then object o_i is an outlier

$$\chi^2 = \sum_{i=1}^{n} \frac{(o_i - E_i)^2}{E_i}$$

Parametric Methods III: Using Mixture of Parametric Distributions



Assuming data generated by a normal distribution could be sometimes overly simplified

Example (right figure): The objects between the two clusters cannot be captured as outliers since they are close to the estimated mean

- To overcome this problem, assume the normal data is generated by two normal distributions. For any object o in the data set, the probability that o is generated by the mixture of the two distributions is given by

$$Pr(o|\Theta_1, \Theta_2) = f_{\Theta_1}(o) + f_{\Theta_2}(o)$$

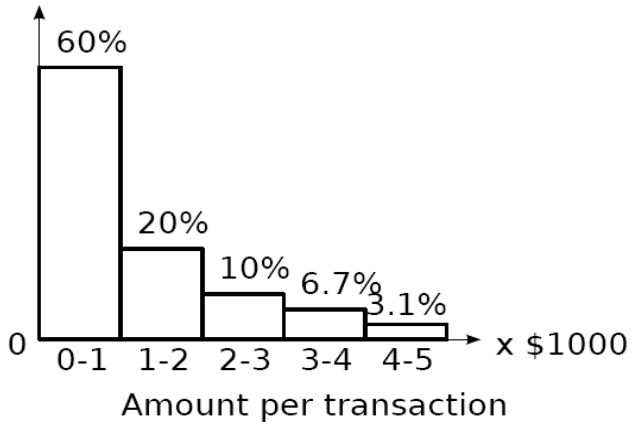
  where $f_{\theta 1}$ and $f_{\theta 2}$ are the probability density functions of θ_1 and θ_2

- Then use EM algorithm to learn the parameters μ_1, σ_1, μ_2, σ_2 from data
- An object o is an outlier if it does not belong to any cluster

Non-Parametric Methods: Detection Using Histogram

The model of normal data is learned from the input data without any *a priori* structure.

Often makes fewer assumptions about the data, and thus can be applicable in more scenarios



Outlier detection using histogram:

- Figure shows the histogram of purchase amounts in transactions
- A transaction in the amount of $7,500 is an outlier, since only 0.2% transactions have an amount higher than $5,000

- Problem: Hard to choose an appropriate bin size for histogram
  - Too small bin size → normal objects in empty/rare bins, false positive
  - Too big bin size → outliers in some frequent bins, false negative
- Solution: Adopt kernel density estimation to estimate the probability density distribution of the data. If the estimated density function is high, the object is likely normal. Otherwise, it is likely an outlier.

Chapter 12. Outlier Analysis

Outlier and Outlier Analysis

Outlier Detection Methods

Statistical Approaches

Proximity-Base Approaches

Clustering-Base Approaches

Classification Approaches

Mining Contextual and Collective Outliers

Outlier Detection in High Dimensional Data

Summary

Proximity-Based Approaches: Distance-Based vs. Density-Based Outlier Detection

Intuition: Objects that are far away from the others are outliers

Assumption of proximity-based approach: The proximity of an outlier deviates significantly from that of most of the others in the data set

Two types of proximity-based outlier detection methods

- Distance-based outlier detection: An object o is an outlier if its neighborhood does not have enough other points
- 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

For each object o, examine the # of other objects in the *r*-neighborhood of o, where *r* is a user-specified **distance threshold**

An object o is an outlier if most (taking π as a **fraction threshold**) of the objects in D are far away from o, i.e., not in the r-neighborhood of o

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

An object o is a DB(r, π) outlier if

Equivalently, one can check the distance between *o* and its *k*-th nearest neighbor o_k, where $k = \lceil \pi \|D\| \rceil$. *o* is an outlier if dist(*o*, o_k) > r

Efficient computation: Nested loop algorithm

- For any object o_i, calculate its distance from other objects, and count the # of other objects in the r-neighborhood.
- If π·n other objects are within r distance, terminate the inner loop
- Otherwise, o_i is a DB(r, π) outlier

Efficiency: Actually CPU time is not $O(n^2)$ but linear to the data set size since for most non-outlier objects, the inner loop terminates early

Distance-Based Outlier Detection: A Grid-Based Method

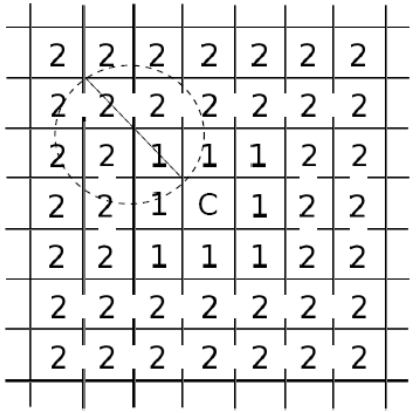
Why efficiency is still a concern? When the complete set of objects cannot be held into main memory, cost I/O swapping

---

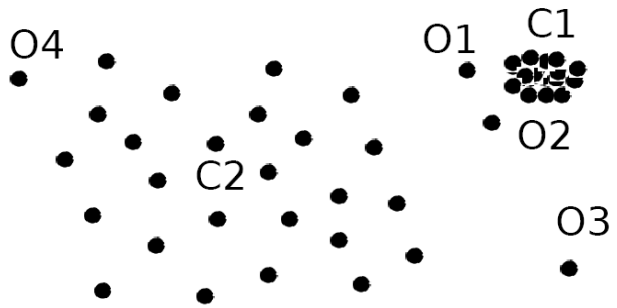
The major cost: (1) each object tests against the whole data set, why not only its close neighbor? (2) check objects one by one, why not group by group?

Grid-based method (CELL): Data space is partitioned into a multi-D grid. Each cell is a hyper cube with diagonal length r/2

- Pruning using the level-1 & level 2 cell properties:
  - For any possible point x in cell C and any possible point y in a level-1 cell, dist(x,y) ≤ r
  - For any possible point x in cell C and any point y such that dist(x,y) ≥ r, y is in a level-2 cell



- Thus we only need to check the objects that cannot be pruned, and even for such an object o, only need to compute the distance between o and the objects in the level-2 cells (since beyond level-2, the distance from o is more than r)



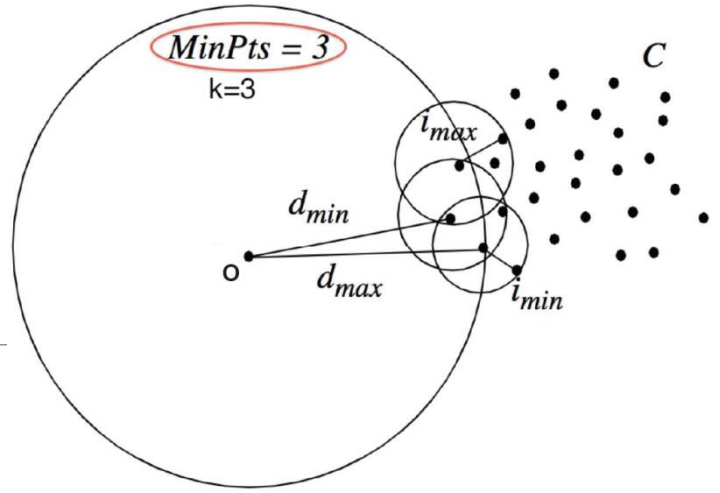
Density-Based Outlier Detection

Local outliers: Outliers comparing to their local neighborhoods, instead of the global data distribution

In Fig., o_1 and o2 are local outliers to C_1, o_3 is a global outlier, but o_4 is not an outlier. However, proximity-based clustering cannot find o_1 and o_2 are outlier (e.g., comparing with O_4).

- 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

Local Outlier Factor: LOF



Reachability distance from *o'* to *o*:

$$reachdist_k(o \leftarrow o') = \max\{dist_k(o), dist(o, o')\}$$

- where k is a user-specified parameter

Local reachability density of *o*: $lrd_k(o) = \dfrac{\|N_k(o)\|}{\sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)}$

- LOF (Local outlier factor) of an object o is the average of the ratio of local reachability of *o* and those of *o*'s k-nearest neighbors

$$LOF_k(o) = \frac{\sum_{o' \in N_k(o)} \frac{lrd_k(o')}{lrd_k(o)}}{\|N_k(o)\|} = \sum_{o' \in N_k(o)} lrd_k(o') \cdot \sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)$$

- The lower the local reachability density of o, and the higher the local reachability density of the kNN of o, the higher LOF
- This captures a local outlier whose local density is relatively low comparing to the local densities of its kNN

Chapter 12. Outlier Analysis

Outlier and Outlier Analysis

Outlier Detection Methods

Statistical Approaches

Proximity-Base Approaches

Clustering-Base Approaches

Classification Approaches

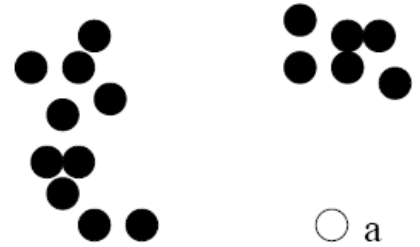
Mining Contextual and Collective Outliers

Outlier Detection in High Dimensional Data

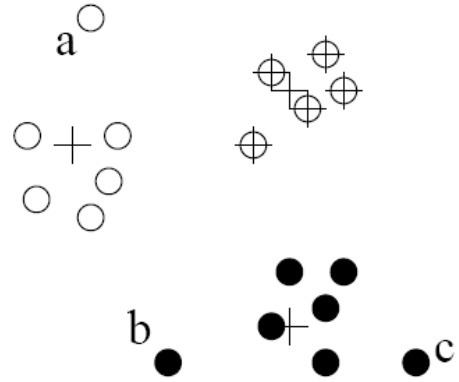
Summary

Clustering-Based Outlier Detection (1 & 2):
Not belong to any cluster, or far from the closest one

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



- Case I: Not belong to any cluster
  - Identify animals not part of a flock: Using a density-based clustering method such as DBSCAN
- Case 2: Far from its closest cluster
  - Using k-means, partition data points of into clusters
  - For each object o, assign an outlier score based on its distance from its closest center
    - If dist(o, c_o)/avg_dist(c_o) is large, likely an outlier

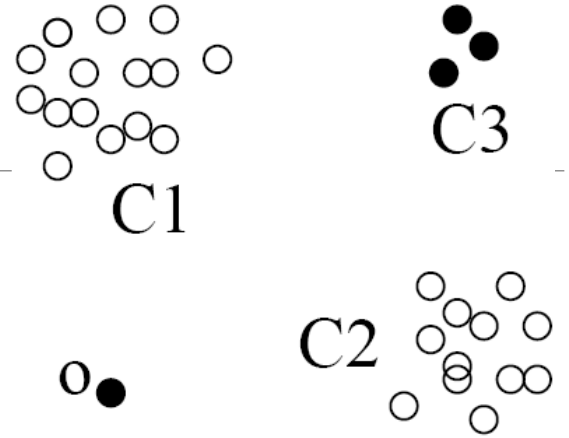


- Ex. Intrusion detection: Consider the similarity between data points and the clusters in a training data set
  - Use a training set to find patterns of “normal” data, e.g., frequent itemsets in each segment, and cluster similar connections into groups
  - Compare new data points with the clusters mined—Outliers are possible attacks

Clustering-Based Outlier Detection (3): Detecting Outliers in Small Clusters

FindCBLOF: Detect outliers in small clusters

- Find clusters, and sort them in decreasing size
- To each data point, assign a *cluster-based local outlier factor* (CBLOF):
- If obj p belongs to a large cluster, CBLOF = cluster_size X similarity between p and cluster
- If p belongs to a small one, CBLOF = cluster size X similarity betw. p and the closest large cluster



- Ex. In the figure, o is outlier since its closest large cluster is C_1, but the similarity between o and C_1 is small. For any point in C_3, its closest large cluster is C_2 but its similarity from C_2 is low, plus $|C_3| = 3$ is small

Clustering-Based Method: Strength and Weakness

Strength

- Detect outliers without requiring any labeled data
- Work for many types of data
- Clusters can be regarded as summaries of the data
- Once the cluster are obtained, need only compare any object against the clusters to determine whether it is an outlier (fast)

Weakness

- Effectiveness depends highly on the clustering method used—they may not be optimized for outlier detection
- High computational cost: Need to first find clusters
- A method to reduce the cost: Fixed-width clustering
  - A point is assigned to a cluster if the center of the cluster is within a pre-defined distance threshold from the point
  - If a point cannot be assigned to any existing cluster, a new cluster is created and the distance threshold may be learned from the training data under certain conditions

Chapter 12. Outlier Analysis

Outlier and Outlier Analysis

Outlier Detection Methods

Statistical Approaches

Proximity-Base Approaches

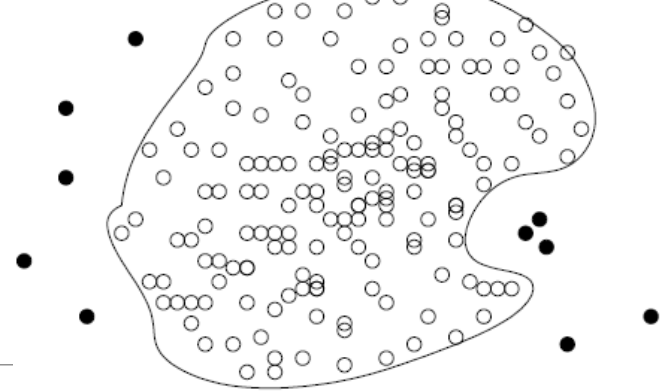
Clustering-Base Approaches

Classification Approaches 

Mining Contextual and Collective Outliers

Outlier Detection in High Dimensional Data

Summary



Classification-Based Method I: One-Class Model

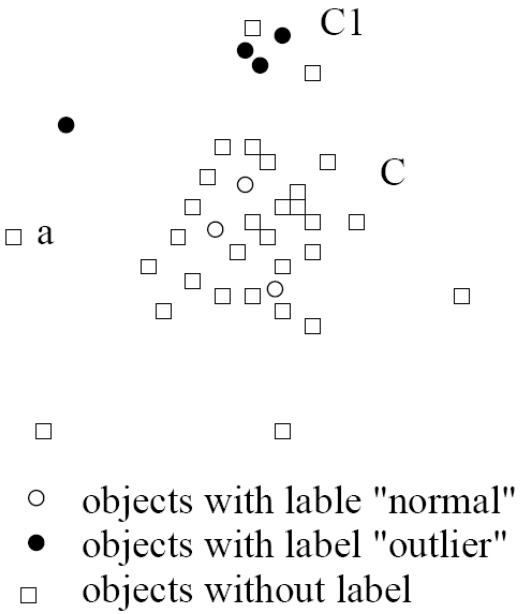
Idea: Train a classification model that can distinguish "normal" data from outliers

A brute-force approach: Consider a training set that contains samples labeled as "normal" and others labeled as "outlier"

- But, the training set is typically heavily biased: # of "normal" samples likely far exceeds # of outlier samples
- Cannot detect unseen anomaly

- One-class model: A classifier is built to describe only the normal class.
  - Learn the decision boundary of the normal class using classification methods such as SVM
  - 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

Classification-Based Method II: Semi-Supervised Learning



Semi-supervised learning: Combining classification-based and clustering-based

Method

- Using a clustering-based approach, find a large cluster, C, and a small cluster, C_1
- Since some objects in C carry the label "normal", treat all objects in C as normal
- Use the one-class model of this cluster to identify normal objects in outlier detection
- Since some objects in cluster C_1 carry the label "outlier", declare all objects in C_1 as outliers
- Any object that does not fall into the model for C (such as *a*) is considered an outlier as well

- Comments on classification-based outlier detection methods
  - Strength: Outlier detection is fast
  - Bottleneck: Quality heavily depends on the availability and quality of the training set, but often difficult to obtain representative and high-quality training data

Chapter 12. Outlier Analysis

Outlier and Outlier Analysis

Outlier Detection Methods

Statistical Approaches

Proximity-Base Approaches

Clustering-Base Approaches

Classification Approaches

Mining Contextual and Collective Outliers

Outlier Detection in High Dimensional Data

Summary

Mining Contextual Outliers I: Transform into Conventional Outlier Detection

If the contexts can be clearly identified, transform it to conventional outlier detection

1. Identify the context of the object using the contextual attributes
2. Calculate the outlier score for the object in the context using a conventional outlier detection method

Ex. Detect outlier customers in the context of customer groups

- Contextual attributes: *age group, postal code*
- Behavioral attributes: *# of trans/yr*, *annual total trans. amount*

Steps: (1) locate c's context, (2) compare c with the other customers in the same group, and (3) use a conventional outlier detection method

If the context contains very few customers, generalize contexts

- Ex. Learn a mixture model U on the contextual attributes, and another mixture model V of the data on the behavior attributes
- Learn a mapping $p(V_i|U_j)$: the probability that a data object o belonging to cluster U_j on the contextual attributes is generated by cluster Vi on the behavior attributes
- Outlier score:

$$S(\boldsymbol{o}) = \sum_{U_j} p(\boldsymbol{o} \in U_j) \sum_{V_i} p(\boldsymbol{o} \in V_i) p(V_i|U_j)$$

Mining Contextual Outliers II: Modeling Normal Behavior with Respect to Contexts

In some applications, one cannot clearly partition the data into contexts

- Ex. if a customer suddenly purchased a product that is unrelated to those she recently browsed, it is unclear how many products browsed earlier should be considered as the context

Model the “normal” behavior with respect to contexts

- Using a training data set, train a model that predicts the expected behavior attribute values with respect to the contextual attribute values
- An object is a contextual outlier if its behavior attribute values significantly deviate from the values predicted by the model

Using a prediction model that links the contexts and behavior, these methods avoid the explicit identification of specific contexts

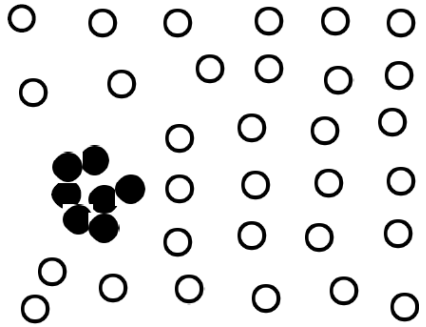
Methods: A number of classification and prediction techniques can be used to build such models, such as regression, Markov Models, and Finite State Automaton

Mining Collective Outliers I: On the Set of "Structured Objects"

Collective outlier if objects as a group deviate significantly from the entire data

Need to examine the *structure* of the data set, i.e, the relationships between multiple data objects

- Each of these structures is inherent to its respective type of data
  - For temporal data (such as time series and sequences), we explore the structures formed by time, which occur in segments of the time series or subsequences
  - For spatial data, explore local areas
  - For graph and network data, we explore subgraphs
- Difference from the contextual outlier detection: the structures are often not explicitly defined, and have to be discovered as part of the outlier detection process.
- Collective outlier detection methods: two categories
  - Reduce the problem to conventional outlier detection
    - Identify *structure units*, treat each structure unit (e.g., subsequence, time series segment, local area, or subgraph) as a data object, and extract features



Mining Collective Outliers II: Direct Modeling of the Expected Behavior of Structure Units

Models the expected behavior of structure units directly

Ex. 1. Detect collective outliers in online social network of customers
- Treat each possible subgraph of the network as a structure unit
- Collective outlier: An *outlier subgraph* in the social network
  - Small subgraphs that are of very low frequency
  - Large subgraphs that are surprisingly frequent

Ex. 2. Detect collective outliers in temporal sequences
- Learn a Markov model from the sequences
- A subsequence can then be declared as a collective outlier if it significantly deviates from the model

Collective outlier detection is subtle due to the challenge of exploring the structures in data
- The exploration typically uses heuristics, and thus may be application dependent
- The computational cost is often high due to the sophisticated mining process

Chapter 12. Outlier Analysis

Outlier and Outlier Analysis

Outlier Detection Methods

Statistical Approaches

Proximity-Base Approaches

Clustering-Base Approaches

Classification Approaches

Mining Contextual and Collective Outliers

Outlier Detection in High Dimensional Data

Summary

Challenges for Outlier Detection in High-Dimensional Data

Interpretation of outliers
- Detecting outliers without saying why they are outliers is not very useful in high-D due to many features (or dimensions) are involved in a high-dimensional data set
- E.g., which subspaces that manifest the outliers or an assessment regarding the “outlier-ness” of the objects

Data sparsity
- Data in high-D spaces are often sparse
- The distance between objects becomes heavily dominated by noise as the dimensionality increases

Data subspaces
- Adaptive to the subspaces signifying the outliers
- Capturing the local behavior of data

Scalable with respect to dimensionality
- # of subspaces increases exponentially

Approach I: Extending Conventional Outlier Detection

Method 1: Detect outliers in the full space, e.g., HilOut Algorithm

- Find distance-based outliers, but use the ranks of distance instead of the absolute distance in outlier detection
- For each object *o*, find its k-nearest neighbors: $nn_1(o), \ldots, nn_k(o)$
- The weight of object o:

$$w(\boldsymbol{o}) = \sum_{i=1}^{k} dist(\boldsymbol{o}, nn_i(\boldsymbol{o}))$$

- All objects are ranked in weight-descending order
- Top-*l* objects in weight are output as outliers (*l*: user-specified parm)
- Employ space-filling curves for approximation: scalable in both time and space w.r.t. data size and dimensionality

Method 2: Dimensionality reduction

- Works only when in lower-dimensionality, normal instances can still be distinguished from outliers
- PCA: Heuristically, the principal components with low variance are preferred because, on such dimensions, normal objects are likely close to each other and outliers often deviate from the majority

Approach II: Finding Outliers in Subspaces

Extending conventional outlier detection: Hard for outlier interpretation

Find outliers in much lower dimensional subspaces: easy to interpret *why* and *to what extent* the object is an outlier
- E.g., find outlier customers in certain subspace: *average transaction amount >> avg.* and *purchase frequency* << avg.

Ex. A grid-based subspace outlier detection method
- Project data onto various subspaces to find an area whose density is much lower than average
- Discretize the data into a grid with ϕ equi-depth (why?) regions
- Search for regions that are significantly sparse
  - Consider a k-d cube: k ranges on k dimensions, with n objects
  - If objects are independently distributed, the expected number of objects falling into a k-dimensional region is $(1/\phi)^k n = f^k n$, the standard deviation is $\sqrt{f^k(1-f^k)n}$
  - The sparsity coefficient of cube C: $S(C) = \frac{n(C) - f^k n}{\sqrt{f^k(1-f^k)n}}$
  - If S(C) < 0, C contains less objects than expected
  - The more negative, the sparser C is and the more likely the objects in C are outliers in the subspace

Approach III: Modeling High-Dimensional Outliers

- Develop new models for high-dimensional outliers directly
- Avoid proximity measures and adopt new heuristics that do not deteriorate in high-dimensional data

Ex. Angle-based outliers: Kriegel, Schubert, and Zimek [KSZ08]

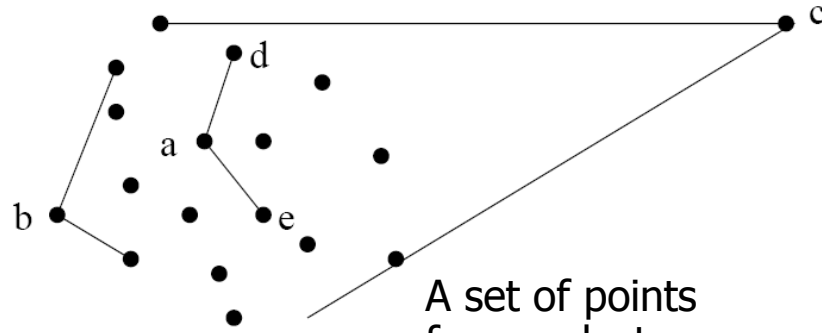
For each point o, examine the angle Δxoy for every pair of points x, y.
- Point in the center (e.g., a), the angles formed differ widely
- An outlier (e.g., c), angle variable is substantially smaller

Use the variance of angles for a point to determine outlier

Combine angles and distance to model outliers
- Use the distance-weighted angle variance as the outlier score
- Angle-based outlier factor (ABOF):

$$ABOF(\boldsymbol{o}) = VAR_{\boldsymbol{x},\boldsymbol{y} \in D, \boldsymbol{x} \neq \boldsymbol{o}, \boldsymbol{y} \neq \boldsymbol{o}} \frac{\langle \overrightarrow{\boldsymbol{ox}}, \overrightarrow{\boldsymbol{oy}} \rangle}{dist(\boldsymbol{o}, \boldsymbol{x})^2 dist(\boldsymbol{o}, \boldsymbol{y})^2}$$

- Efficient approximation computation method is developed
- It can be generalized to handle arbitrary types of data



A set of points form a cluster except c (outlier)

Chapter 12. Outlier Analysis

Outlier and Outlier Analysis

Outlier Detection Methods

Statistical Approaches

Proximity-Base Approaches

Clustering-Base Approaches

Classification Approaches

Mining Contextual and Collective Outliers

Outlier Detection in High Dimensional Data

Summary

Summary

Types of outliers
- global, contextual & collective outliers

Outlier detection
- supervised, semi-supervised, or unsupervised

Statistical (or model-based) approaches

Proximity-base approaches

Clustering-base approaches

Classification approaches

Mining contextual and collective outliers

Outlier detection in high dimensional data