DSAA 5002: Knowledge Discovery and Data Mining in Data Science

Acknowledgement: Slides modified by Dr. Lei Chen based on the slides provided by Pang-Ning Tan,

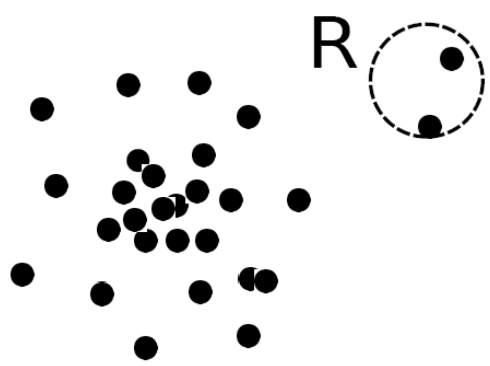
Michael Steinbach,Vipin Kumar, Jiawei Han, Micheline Kamber, and Jian Pei

Chapter 12. Outlier Analysis

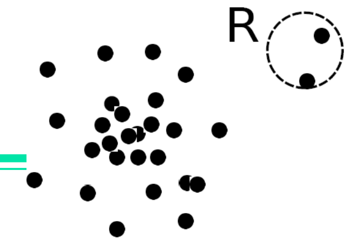
- Outlier and Outlier Analysis
- Outlier Detection Methods
- Statistical Approaches
- Proximity-Base Approaches
- Clustering-Base Approaches
- Classification Approaches
- Outlier Detection in High Dimensional Data
- Summary

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)

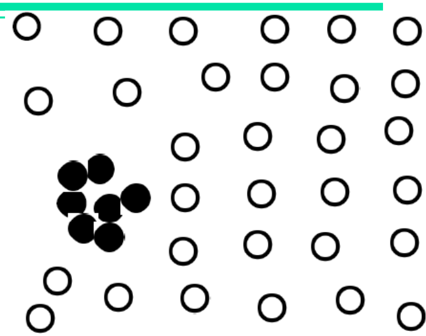


Global Outlier

- 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 Outliers** 集体异常
  - 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



Collective 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

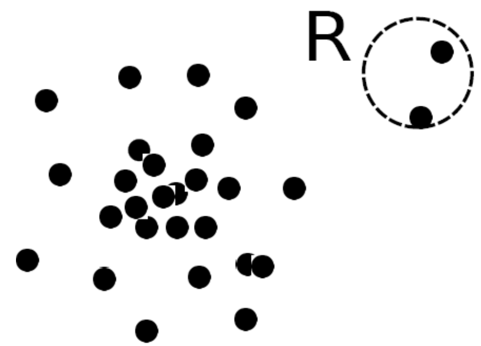
- Assume the normal objects are somewhat ``clustered'' into multiple groups, each having some distinct features
- An outlier is expected to be far away from any groups of normal objects
- Weakness: Cannot detect collective outlier effectively
  - Normal objects may not share any strong patterns, but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

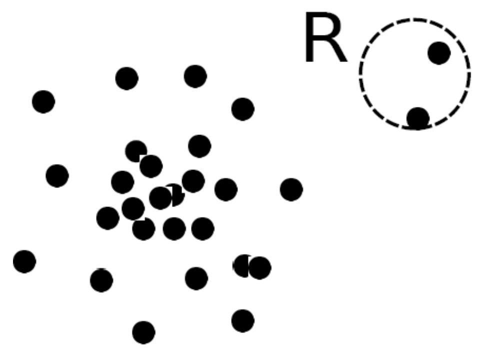
- Statistical methods (also known as model-based methods) assume that the normal data follow some statistical model (a stochastic model)
  - The data not following the model are outliers.
- Example (right figure): First use Gaussian distribution to model the normal data
  - For each object y in region 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

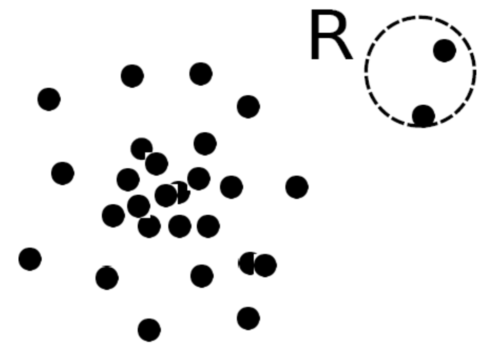
- An object is an outlier if the nearest neighbors of the object are far away, i.e., the **proximity** of the object is **significantly deviates** from the proximity of most of the other objects in the same data set
- Example (right figure): Model the proximity of an object using its 3 nearest neighbors
  - 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
- 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 $f(x, \theta)$ 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

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

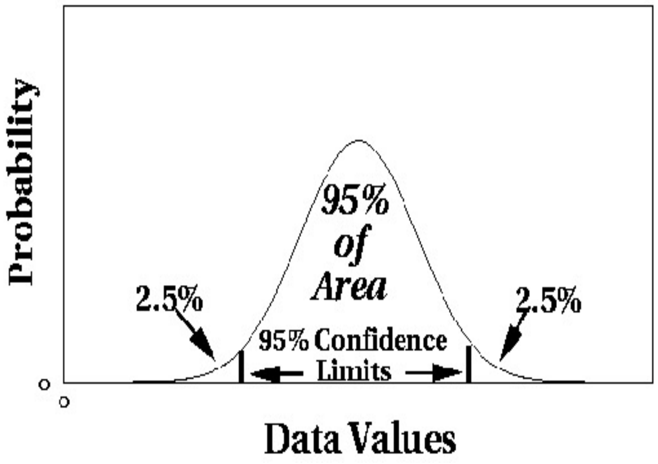
  - Taking derivatives with respect to μ and σ^2, we derive the following maximum likelihood estimates

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

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

$\mu \pm 3\sigma$ region contains 99.7% data

Outlier Discovery: Statistical Approaches



Assume a model underlying distribution that generates data set (e.g. normal distribution)

- Use discordancy tests depending on
  - data distribution
  - distribution parameter (e.g., mean, variance)
  - number of expected outliers
- Drawbacks
  - most tests are for single attribute
  - In many cases, data distribution may not be known

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

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

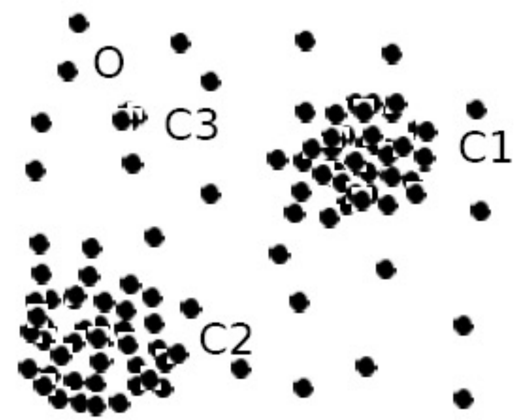
where $t^2_{\alpha/(2N),N-2}$ is the value taken by a t-distribution at a significance level of α/(2N), and N is the # of objects in the data set

Parametric Methods II: Detection of Multivariate Outliers

- Multivariate data: A data set involving two or more attributes or variables 多变量
- 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, ō) = $(o – \bar{o})^T S^{-1}(o – \bar{o})$ where S is the covariance matrix
  - Use the Grubb's test on this measure to detect outliers
- Method 2. Use χ^2 –statistic: $\chi^2 = \sum_{i=1}^{n} \frac{(o_i - E_i)^2}{E_i}$
  - 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

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

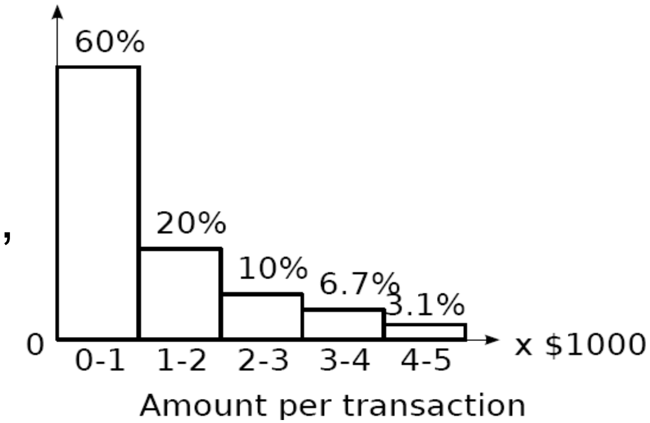
$$Pr(o|\Theta_1, \Theta_2) = f_{\Theta_1}(o) + f_{\Theta_2}(o)$$

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
- An object o is a DB(r, π) outlier if $\frac{\|\{o' | dist(o, o') \leq r\}\|}{\|D\|} \leq \pi$
- 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

Outlier Discovery: Distance-Based Approach

- Introduced to counter the main limitations imposed by statistical methods
  - We need multi-dimensional analysis without knowing data distribution
- Distance-based outlier: A DB(p, D)-outlier is an object O in a dataset T such that at least a fraction p of the objects in T lies at a distance greater than D from O
- Algorithms for mining distance-based outliers [Knorr & Ng, VLDB'98]
  - Index-based algorithm
  - Nested-loop algorithm
  - Cell-based algorithm

Index-based Algorithm [KN98]

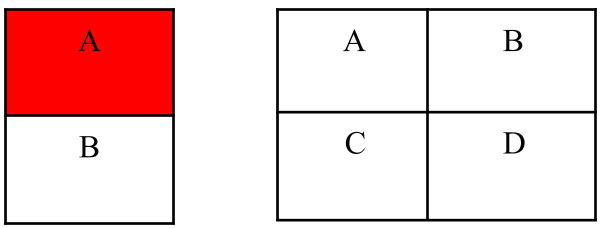
- Indexing Structures such as R-tree (R+-tree), K-D (K-D-B) tree are built for the multi-dimensional database
- The index is used to search for neighbors of each object O within radius D around that object.
- Once K (K = N(1-p)) neighbors of object O are found, O is not an outlier.
- Worst-case computation complexity is $O(K*n^2)$, K is the dimensionality and n is the number of objects in the dataset.
- Pros: scale well with K
- Cons: the index construction process may cost much time

Nested-loop Algorithm [KN98]

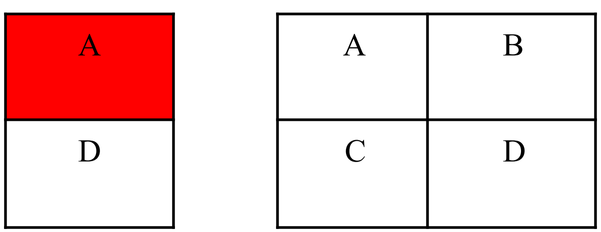
- Divides the buffer space into two halves (first and second arrays)
- Break data into blocks and then feed two blocks into the arrays.
- Directly computes the distance between each pair of objects, inside the array or between arrays
- Decide the outlier.
- Here comes an example:…
- Same computational complexity as the index-based algorithm
- Pros: Avoid index structure construction
- Try to minimize the I/Os

Example – stage 1

Buffer DB



Starting Point of Stage 1



End Point of Stage 1

A is the target block on stage 1

Load A into the first array (1R)

Load B into the second array (1R)

Load C into the second array (1R)

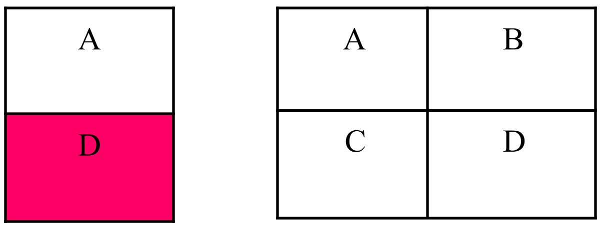
Load D into the second array (1R)

Total: 4 Reads

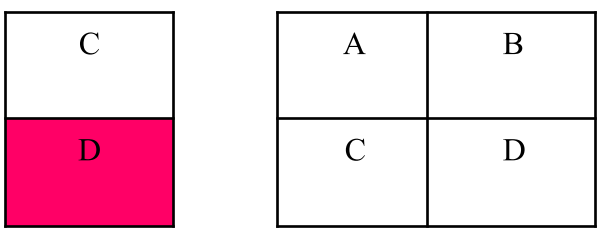
Example – stage 2

Example

Buffer DB



Starting Point of Stage 2



End Point of Stage 2

D is the target block on stage 2

D is already in the buffer (no R)

A is already in the buffer (no R)

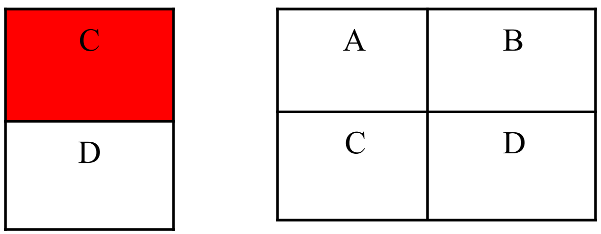
Load B into the first array (1R)

Load C into the first array (1R)

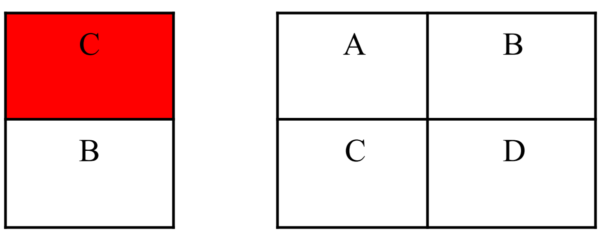
Total: 2 Reads

Example – stage 3

Buffer DB



Starting Point of Stage 3



End Point of Stage 3

C is the target block on stage 3

C is already in the buffer (no R)

D is already in the buffer (no R)

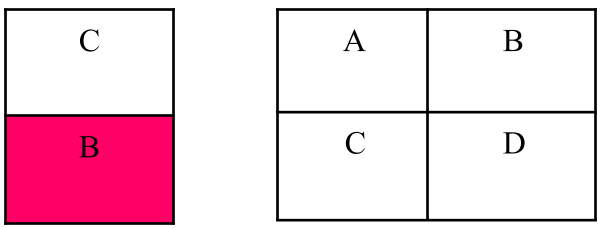
Load A into the second array (1R)

Load B into the second array (1R)

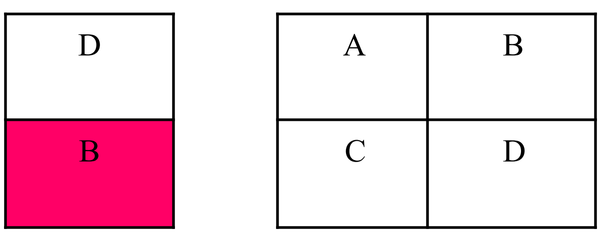
Example – stage 4

Example

Buffer DB



Starting Point of Stage 4



End Point of Stage 4

B is the target block on stage 4

B is already in the buffer (no R)

C is already in the buffer (no R)

Load A into the first array (1R)

Load D into the first array (1R)

Total: 2 Reads

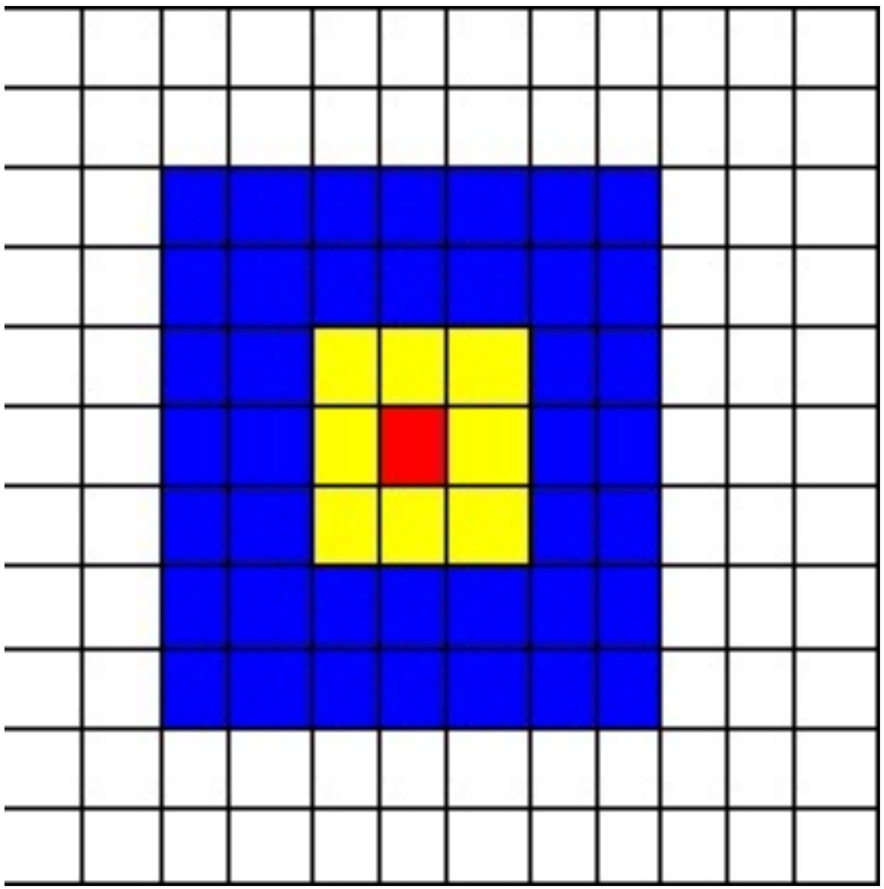
Cell-Based Algorithm [KN98]

- Divide the dataset into cells with length
  - K is the dimensionality, D is the distance

$$l = \frac{D}{2\sqrt{k}}$$

- Define Layer-1 neighbors – all the intermediate neighbor cells. The maximum distance between a cell and its neighbor cells is D
- Define Layer-2 neighbors – the cells within 3 cell of a certain cell. The minimum distance between a cell and the cells outside of Layer-2 neighbors is D
- Criteria
  - Search a cell internally. If there are M objects inside, all the objects in this cell are not outlier
  - Search its layer-1 neighbors. If there are M objects inside a cell and its layer-1 neighbors, all the objects in this cell are not outlier
  - Search its layer-2 neighbors. If there are less than M objects inside a cell, its layer-1 neighbor cells, and its layer-2 neighbor cells, all the objects in this cell are outlier
  - Otherwise, the objects in this cell could be outlier, and then need to calculate the distance between the objects in this cell and the objects in the cells in the layer-2 neighbor cells to see whether the total points within D distance is more than M or not.
- An example

Example



Red – A certain cell

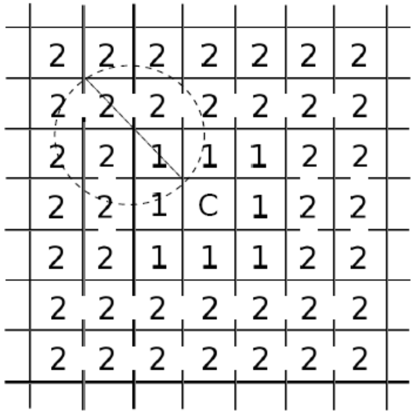
Yellow – Layer-1 Neighbor Cells

Blue – Layer-2 Neighbor Cells

Notes:
The maximum distance between a point in the red cell and a point In its layer-1 neighbor cells is D

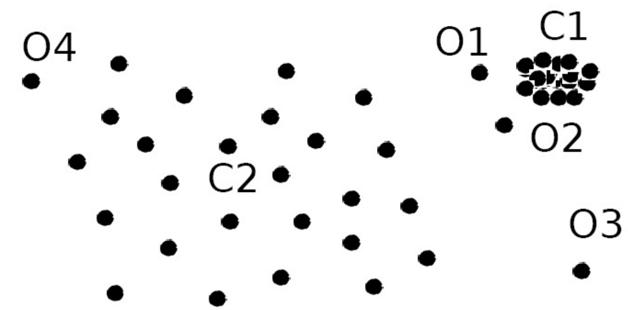
Distance-Based Outlier Detection: A Grid-Based Method

- 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



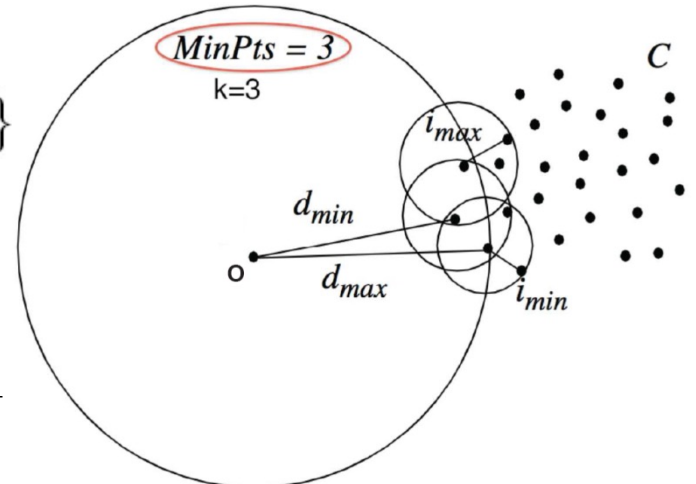
- Local outliers: Outliers comparing to their local neighborhoods, instead of the global data distribution
- In Fig., o_1 and o2 are local outliers to C_1, o_3 is a global outlier, but o_4 is not an outlier. However, proximity-based clustering cannot find o_1 and o_2 are outlier (e.g., comparing with 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, $N_k(o) = \{o' \mid o' \text{ in } D, dist(o, o') \leq dist_k(o)\}$
  - $N_k(o)$ 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*:

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

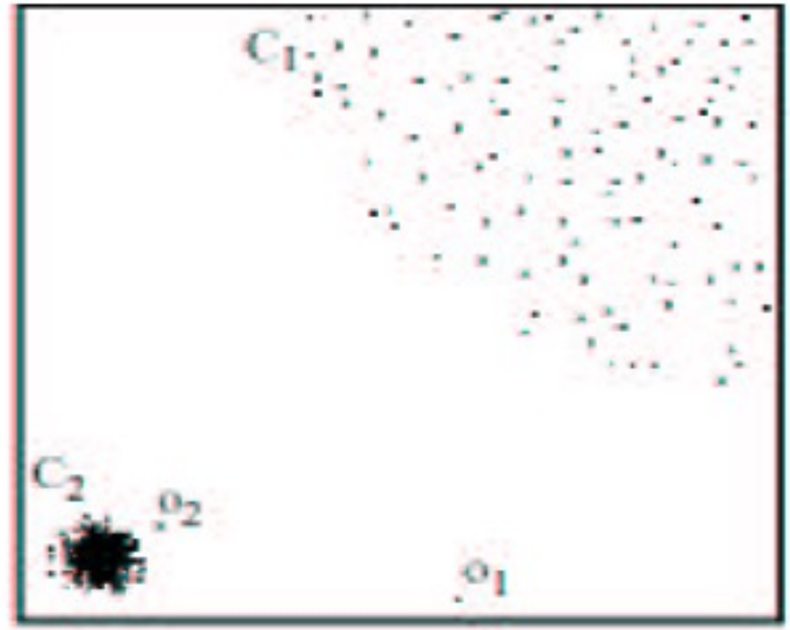
- LOF (Local outlier factor) of an object o is the average of the ratio of local reachability of *o* and those of *o*'s k-nearest neighbors

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

$$= \frac{\sum_{o' \in N_k(o)} lrd_k(o') \cdot \sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)}{\|N_k(o)\|^2}$$

- 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

Density-Based Local Outlier Detection



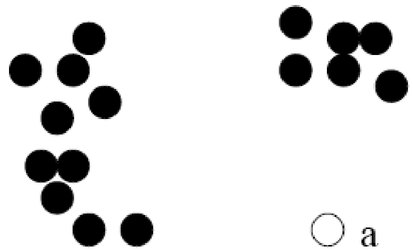
- M. M. Breunig, H.-P. Kriegel, R. Ng, J. Sander. LOF: Identifying Density-Based Local Outliers. SIGMOD 2000.
- Distance-based outlier detection is based on global distance distribution
- It encounters difficulties to identify outliers if data is not uniformly distributed
- Ex. C_1 contains 400 loosely distributed points, C_2 has 100 tightly condensed points, 2 outlier points o_1, o_2
- Distance-based method cannot identify o_2 as an outlier
- Need the concept of local outlier
- Local outlier factor (LOF)
  - Assume outlier is not crisp
  - Each point has a LOF

Chapter 12. Outlier Analysis

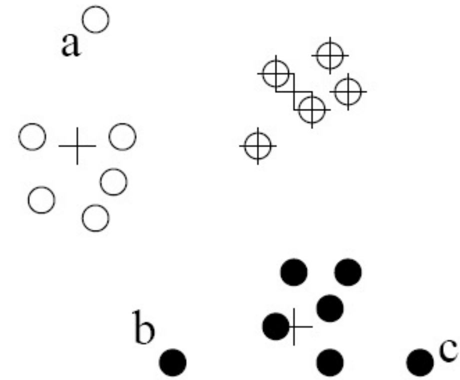
- Outlier and Outlier Analysis
- Outlier Detection Methods
- Statistical Approaches
- Proximity-Base Approaches
- Clustering-Base Approaches
- Classification Approaches
- Mining Contextual and Collective Outliers
- Outlier Detection in High Dimensional Data
- Summary

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

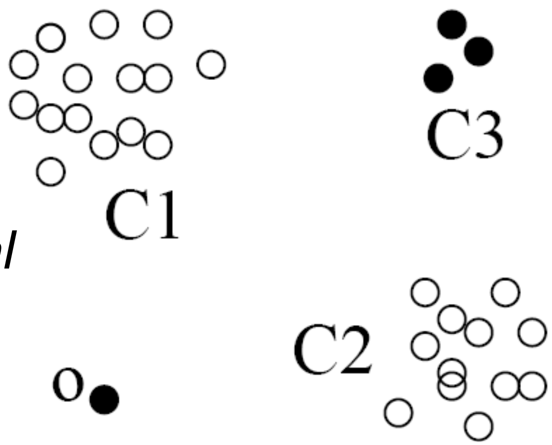


- Case 2: Far from its closest cluster
  - Using k-means, partition data points of into clusters
  - For each object o, assign an outlier score based on its distance from its closest center
    - If $dist(o, c_o)/avg\_dist(c_o)$ is large, likely an outlier



- 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

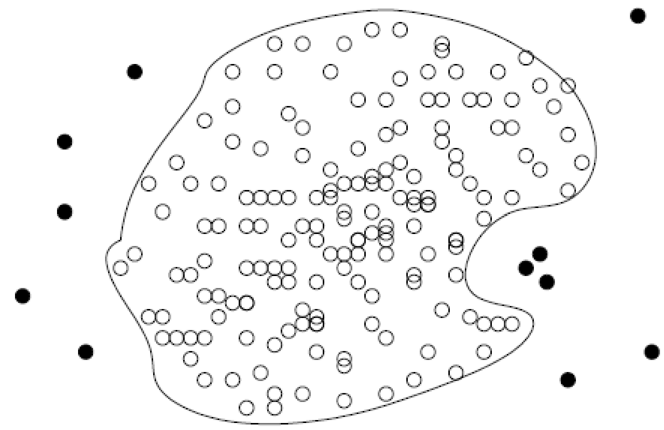
- Strength
  - Detect outliers without requiring any labeled data
  - Work for many types of data
  - Clusters can be regarded as summaries of the data
  - Once the cluster are obtained, need only compare any object against the clusters to determine whether it is an outlier (fast)
- Weakness
  - Effectiveness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

Classification-Based Method I: One-Class Model

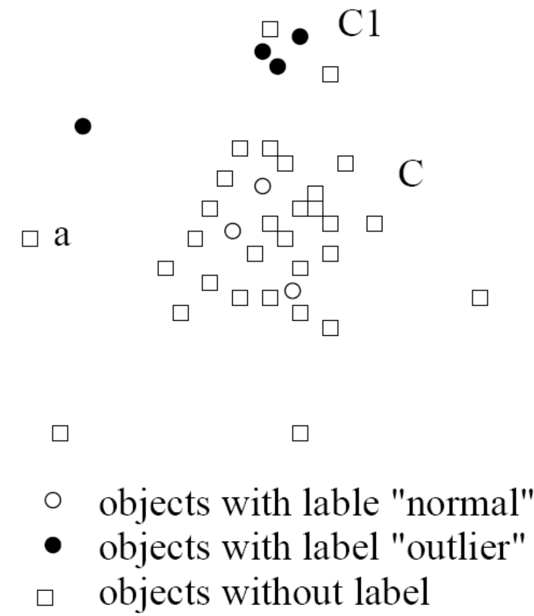
- Idea: Train a classification model that can distinguish “normal” data from outliers
- A brute-force approach: Consider a training set that contains samples labeled as “normal” and others labeled as “outlier”
  - But, the training set is typically heavily biased: # of “normal” samples likely far exceeds # 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 methods
- 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
- 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

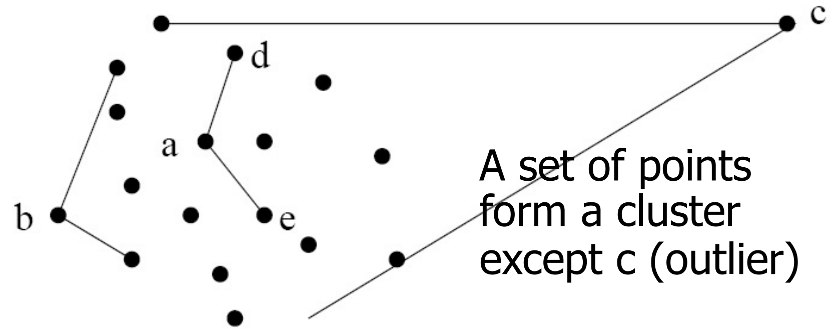
- Method 1: Detect outliers in the full space, e.g., HilOut Algorithm
  - Find distance-based outliers, but use the ranks of distance instead of the absolute distance in outlier detection
  - For each object *o*, find its k-nearest neighbors: $nn_1(o), \ldots, nn_k(o)$
  - The weight of object o: $$w(\boldsymbol{o}) = \sum_{i=1}^{k} dist(\boldsymbol{o}, nn_i(\boldsymbol{o}))$$
  - 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 $(1/\varphi)^k n = f^k n$,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

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
- Outlier detection in high dimensional data

References (I)

- B. Abraham and G.E.P. Box. Bayesian analysis of some outlier problems in time series. *Biometrika*, 66:229–248, 1979.
- M. Agyemang, K. Barker, and R. Alhajj. A comprehensive survey of numeric and symbolic outlier mining techniques. *Intell. Data Anal.*, 10:521–538, 2006.
- F. J. Anscombe and I. Guttman. Rejection of outliers. *Technometrics*, 2:123–147, 1960.
- D. Agarwal. Detecting anomalies in cross-classified streams: a bayesian approach. *Knowl. Inf. Syst.*, 11:29–44, 2006.
- F. Angiulli and C. Pizzuti. Outlier mining in large high-dimensional data sets. *TKDE,* 2005.
- C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. *SIGMOD'01*
- R.J. Beckman and R.D. Cook. Outlier...s. *Technometrics*, 25:119–149, 1983.
- I. Ben-Gal. Outlier detection. In *Maimon O. and Rockach L. (eds.) Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, Kluwer Academic, 2005.
- M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. *SIGMOD'00*
- D. Barbar´a, Y. Li, J. Couto, J.-L. Lin, and S. Jajodia. Bootstrapping a data mining intrusion detection system. *SAC'03*
- Z. A. Bakar, R. Mohemad, A. Ahmad, and M. M. Deris. A comparative study for outlier detection techniques in data mining. *IEEE Conf. on Cybernetics and Intelligent Systems*, 2006.
- S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *KDD'03*
- D. Barbara, N. Wu, and S. Jajodia. Detecting novel network intrusion using bayesian estimators. *SDM'01*
- V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41:1–58, 2009.
- D. Dasgupta and N.S. Majumdar. Anomaly detection in multidimensional data using negative selection algorithm. In *CEC'02*

References (2)

- E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Proc. 2002 Int. Conf. of Data Mining for Security Applications*, 2002.
- E. Eskin. Anomaly detection over noisy data using learned probability distributions. *ICML’00*
- T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1:291–316, 1997.
- V. J. Hodge and J. Austin. A survey of outlier detection methdologies. *Artif. Intell. Rev.*, 22:85–126, 2004.
- D. M. Hawkins. *Identification of Outliers*. Chapman and Hall, London, 1980.
- Z. He, X. Xu, and S. Deng. Discovering cluster-based local outliers. *Pattern Recogn. Lett.*, 24, June, 2003.
- W. Jin, K. H. Tung, and J. Han. Mining top-n local outliers in large databases. *KDD’01*
- W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. *PAKDD’06*
- E. Knorr and R. Ng. A unified notion of outliers: Properties and computation. *KDD’97*
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. *VLDB’98*
- E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *VLDB J.*, 8:237–253, 2000.
- H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. *KDD’08*
- M. Markou and S. Singh. Novelty detection: A review—part 1: Statistical approaches. *Signal Process.*, 83:2481–2497, 2003.
- M. Markou and S. Singh. Novelty detection: A review—part 2: Neural network based approaches. *Signal Process.*, 83:2499–2521, 2003.
- C. C. Noble and D. J. Cook. Graph-based anomaly detection. *KDD’03*

References (1)

- B. Abraham and G.E.P. Box. Bayesian analysis of some outlier problems in time series. *Biometrika*, 1979.
- Malik Agyemang, Ken Barker, and Rada Alhajj. A comprehensive survey of numeric and symbolic outlier mining techniques. *Intell. Data Anal.*, 2006.
- Deepak Agarwal. Detecting anomalies in cross-classied streams: a bayesian approach. *Knowl. Inf. Syst.*, 2006.
- C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. *SIGMOD'01*.
- M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Optics-of: Identifying local outliers. *PKDD '99*
- M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. *SIGMOD'00*.
- V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 2009.
- D. Dasgupta and N.S. Majumdar. Anomaly detection in multidimensional data using negative selection algorithm. *Computational Intelligence,* 2002.
- E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Proc. 2002 Int. Conf. of Data Mining for Security Applications*, 2002.
- E. Eskin. Anomaly detection over noisy data using learned probability distributions. *ICML’00*.
- T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1997.
- R. Fujimaki, T. Yairi, and K. Machida. An approach to spacecraft anomaly detection problem using kernel feature space. *KDD '05*
- F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 1969.

References (2)

- V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 2004.
- Douglas M Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.
- P. S. Horn, L. Feng, Y. Li, and A. J. Pesce. Effect of Outliers and Nonhealthy Individuals on Reference Interval Estimation. *Clin Chem*, 2001.
- W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. *PAKDD'06*
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98
- M. Markou and S. Singh.. Novelty detection: a review| part 1: statistical approaches. *Signal Process.*, 83(12), 2003.
- M. Markou and S. Singh. Novelty detection: a review| part 2: neural network based approaches. *Signal Process.*, 83(12), 2003.
- S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. *ICDE'03*.
- A. Patcha and J.-M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Comput. Netw.*, 51(12):3448{3470, 2007.
- W. Stefansky. Rejecting outliers in factorial designs. *Technometrics*, 14(2):469{479, 1972.
- X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE Trans. on Knowl. and Data Eng.*, 19(5):631{645, 2007.
- Y. Tao, X. Xiao, and S. Zhou. Mining distance-based outliers from large databases in any metric space. *KDD '06:*
- N. Ye and Q. Chen. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality and Reliability Engineering International*, 2001.