

# DATA MINING 2

## Anomaly & Outliers Detection

---

Riccardo Guidotti

a.a. 2022/2023

Slides edited from Tan, Steinbach, Kumar, Introduction to Data Mining and from Kriegel, Kröger, Zimek Tutorial on Outlier Detection Techniques



# What is an Outlier?

---

Definition of Hawkins [Hawkins 1980]:

- “An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism”

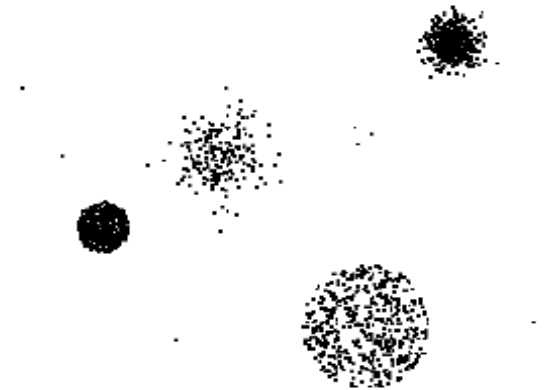
Statistics-based intuition

- Normal data objects follow a “generating mechanism”, e.g. some given statistical process
- Abnormal objects deviate from this generating mechanism

# Anomaly/Outlier Detection

---

- What are anomalies/outliers?
  - The set of data points that are considerably different than the remainder of the data
- Natural implication is that anomalies are relatively rare
  - One in a thousand occurs often if you have lots of data
  - Context is important, e.g., freezing temps in July
- Can be important or a nuisance
  - 10 foot tall 2 year old
  - Unusually high blood pressure



# Applications of Outlier Detection

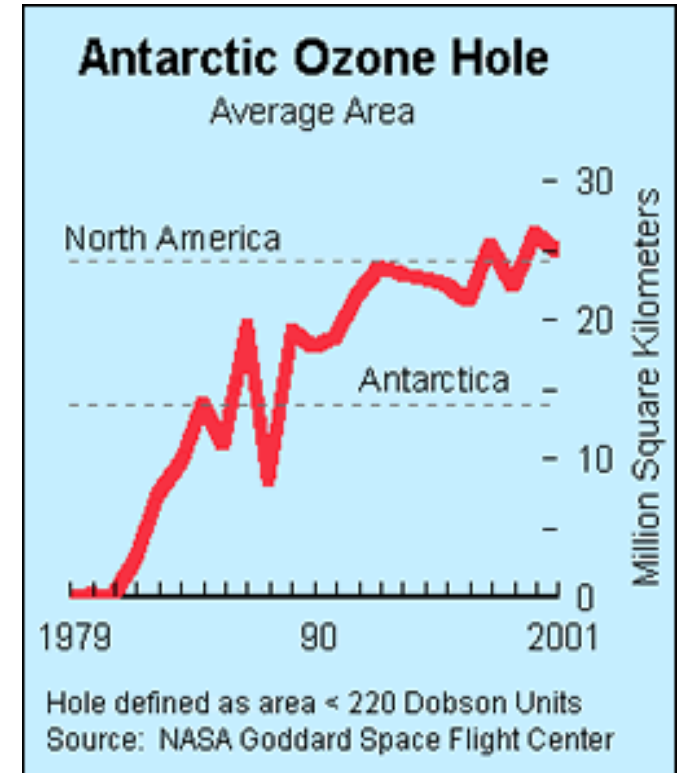
---

- Fraud detection
  - Purchasing behavior of a credit card owner usually changes when the card is stolen
  - Abnormal buying patterns can characterize credit card abuse
- Medicine
  - Unusual symptoms or test results may indicate potential health problems of a patient
  - Whether a particular test result is abnormal may depend on other characteristics of the patients (e.g. gender, age, ...)
- Public health
  - The occurrence of a particular disease, e.g. tetanus, scattered across various hospitals of a city indicate problems with the corresponding vaccination program in that city
  - Whether an occurrence is abnormal depends

# Importance of Anomaly Detection

## Ozone Depletion History

- In 1985 three researchers (Farman, Gardinar and Shanklin) were puzzled by data gathered by the British Antarctic Survey showing that ozone levels for Antarctica had dropped 10% below normal levels
- Why did the Nimbus 7 satellite, which had instruments aboard for recording ozone levels, not record similarly low ozone concentrations?
- The ozone concentrations recorded by the satellite were so low they were being treated as outliers by a computer program and discarded!



# Causes of Anomalies

---

- Data from different classes
  - Measuring the weights of oranges, but a few grapefruit are mixed in
- Natural variation
  - Unusually tall people
- Data errors
  - 200 pound 2 year old

# Distinction Between Noise and Anomalies

---

- Noise is erroneous, perhaps random, values or contaminating objects
  - Weight recorded incorrectly
  - Grapefruit mixed in with the oranges
- Noise doesn't necessarily produce unusual values or objects
- Noise is not interesting
- Anomalies may be interesting if they are not a result of noise
- Noise and anomalies are related but distinct concepts

# General Issues: Number of Attributes

---

- Many anomalies are defined in terms of a single attribute
  - Height
  - Shape
  - Color
- Can be hard to find an anomaly using all attributes
  - Noisy or irrelevant attributes
  - Object is only anomalous with respect to some attributes
- However, an object may not be anomalous in any one attribute



# General Issues: Anomaly Scoring

---

- Many anomaly detection techniques provide only a binary categorization
  - An object is an anomaly or it isn't
  - This is especially true of classification-based approaches
- Other approaches assign a score to all points
  - This score measures the degree to which an object is an anomaly
  - This allows objects to be ranked
- In the end, you often need a binary decision
  - Should this credit card transaction be flagged?
  - Still useful to have a score
- How many anomalies are there?

# Other Issues for Anomaly Detection

---

- Find all anomalies at once or one at a time
  - Swamping
  - Masking
- Evaluation
  - How do you measure performance?
  - Supervised vs. unsupervised situations
- Efficiency
- Context

# Variants of Anomaly Detection Problems

---

- Given a data set  $D$ , find all data points  $\mathbf{x} \in D$  with anomaly scores greater than some threshold  $t$
- Given a data set  $D$ , find all data points  $\mathbf{x} \in D$  having the top- $n$  largest anomaly scores
- Given a data set  $D$ , containing mostly normal (but unlabeled) data points, and a test point  $\mathbf{x}$ , compute the anomaly score of  $\mathbf{x}$  with respect to  $D$

# Model-Based Anomaly Detection

---

Build a model for the data and see

- Unsupervised
  - Anomalies are those points that don't fit well
  - Anomalies are those points that distort the model
  - Examples:
    - Statistical distribution
    - Clusters
    - Regression
    - Geometric
    - Graph
- Supervised
  - Anomalies are regarded as a rare class
  - Need to have training data

# Machine Learning for Outlier Detection

---

- If the ground truth of anomalies is available we can prepare a classification problem to unveil outliers.
- As classifiers we can use all the available machine learning approaches: Ensembles, SVM, DNN.
- The problem is that the dataset would be very unbalanced
- Thus, ad-hoc formulations/implementation should be adopted.

# Unsupervised Anomaly Detection Techniques

---

- **Proximity-based**
  - Anomalies are points far away from other points
  - Can detect this graphically in some cases
- **Density-based**
  - Low density points are outliers
- **Pattern matching**
  - Create profiles or templates of atypical but important events or objects
  - Algorithms to detect these patterns are usually simple and efficient

# Outliers Detection Approaches Taxonomy

---

- **Global vs local** outlier detection
  - Considers the set of reference objects relative to which each point's "outlierness" is judged
- **Labeling vs scoring** outliers
  - Considers the output of an algorithm
- **Modeling properties**
  - Considers the concepts based on which "outlierness" is modeled

# Global versus Local Approaches

---

- Considers the resolution of the reference set w.r.t. which the “outlierness” of a particular data object is determined
- **Global approaches**
  - The reference set contains all other data objects
  - Basic assumption: there is only one normal mechanism
  - Basic problem: other outliers are also in the reference set and may falsify the results
- **Local approaches**
  - The reference contains a (small) subset of data objects
  - No assumption on the number of normal mechanisms
  - Basic problem: how to choose a proper reference set
- Notes
  - Some approaches are somewhat in between
  - The resolution of the reference set is varied e.g. from only a single object (local) to the entire database (global) automatically or by a user-defined input parameter



# Labeling versus Scoring

---

- Considers the output of an outlier detection algorithm
- **Labeling approaches**
  - Binary output
  - Data objects are labeled either as normal or outlier
- **Scoring approaches**
  - Continuous output
  - For each object an outlier score is computed (e.g. the probability for being an outlier)
  - Data objects can be sorted according to their scores
- Notes
  - Many scoring approaches focus on determining the top-n outliers (parameter n is usually given by the user)
  - Scoring approaches can usually also produce binary output if necessary (e.g. by defining a suitable threshold on the scoring values)

# Model-based Approaches

---

## **Approaches classified by the properties of the underlying modeling**

- Rational
  - Apply a model to represent normal data points
  - Outliers are points that do not fit to that model
- Sample approaches
  - Probabilistic tests based on statistical models
  - Depth-based approaches
  - Deviation-based approaches
  - Some subspace outlier detection approaches

# Model-based Approaches

---

## Proximity-based Approaches

- Rational
  - Examine the spatial proximity of each object in the data space
  - If the proximity of an object considerably deviates from the proximity of other objects it is considered an outlier
- Sample approaches
  - Distance-based approaches
  - Density-based approaches
  - Some subspace outlier detection approaches

# Model-based Approaches

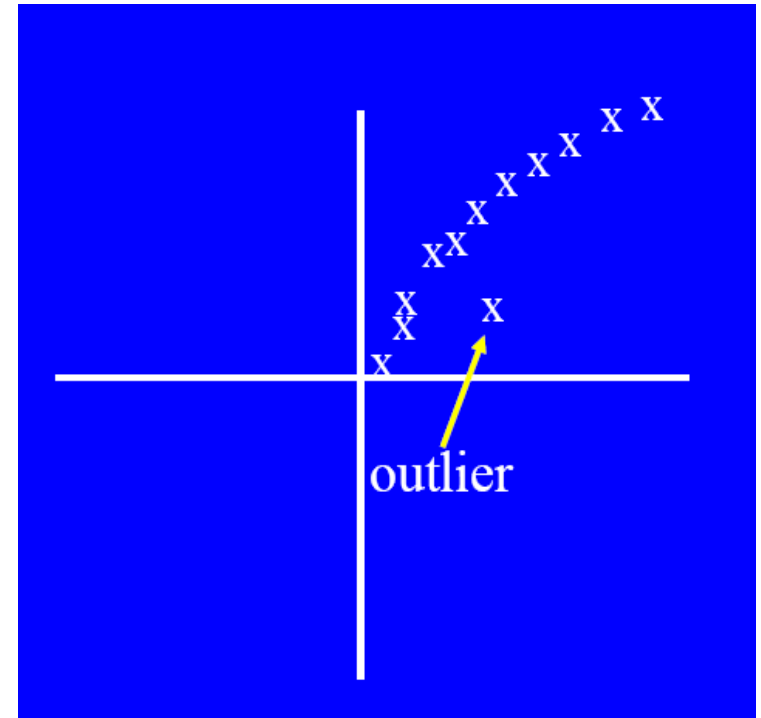
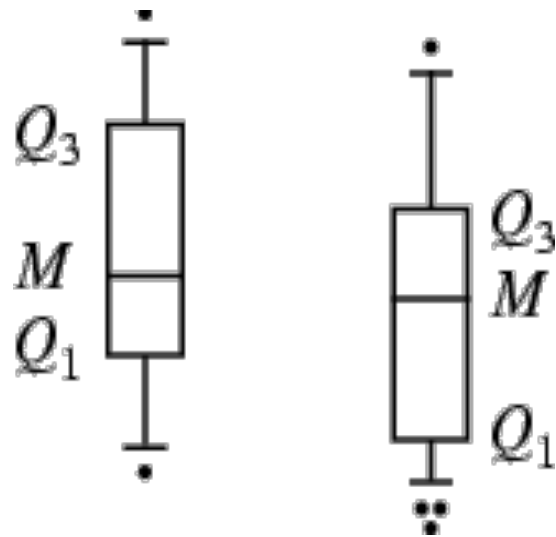
---

## Angle-based approaches

- Rational
  - Examine the spectrum of pairwise angles between a given point and all other points
  - Outliers are points that have a spectrum featuring high fluctuation

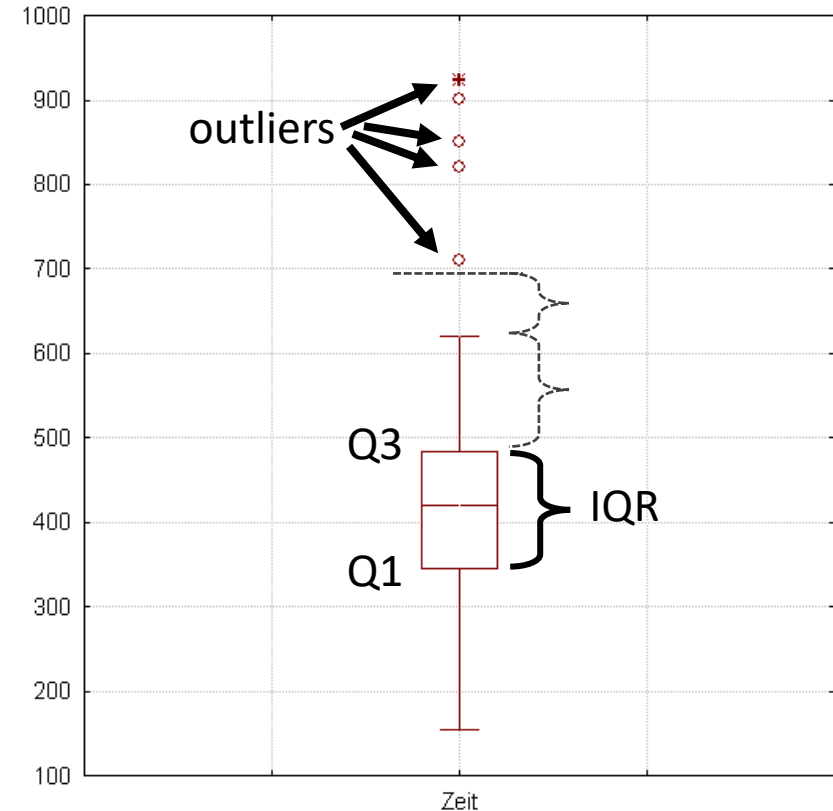
# Visual Approaches

- Boxplots
- Scatter plots
- Limitations
  - Not automatic
  - Subjective



# From Visual Box-plot to Automatic Approach

- The IQR of a set of values is calculated as the difference between the upper and lower quartiles, Q3 and Q1.  $IQR = Q3 - Q1$
- $x$  is an outlier if  $x < Q1 - k IQR$  or  $x > Q3 + k IQR$  (generally  $k=1.5$ )
- In a boxplot, the highest and lowest occurring value within this limit are indicated by *whiskers* of the box and any outliers as individual points.



# HBOS - Histogram-based Outlier Score

---

- *It assumes feature independence* and calculates the outlier scores by building histograms.
- Univariate histogram for each single feature
  - Categorical data: Simple counting
  - Numerical data:
    1. Bin width with  $k$  bins having equal width
    2. Bin width with  $N/k$  instances per bin (equal frequency)
- Frequency (relative amount) of records in a bin is used as density estimation
- Histograms are normalized to  $[0,1]$  for each single feature
- HBOS for each record  $p$  is computed as a product of the inverse of the estimated density:

$$HBOS(p) = \sum_{i=0}^d \log\left(\frac{1}{hist_i(p)}\right)$$

# Statistical Approaches

---



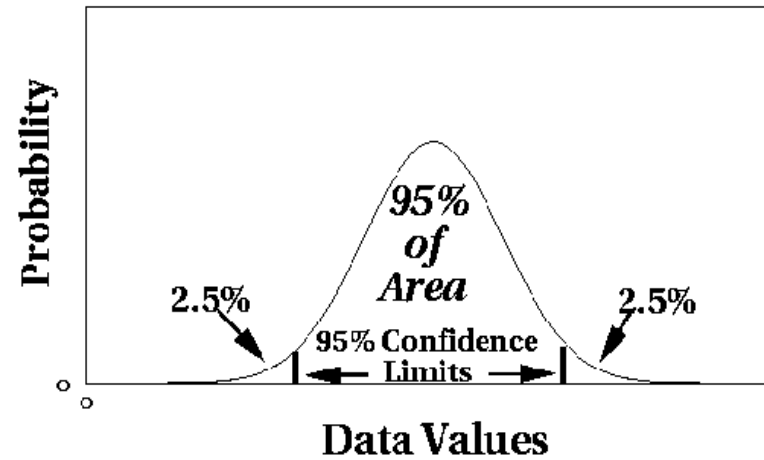
# Statistical Approaches

---

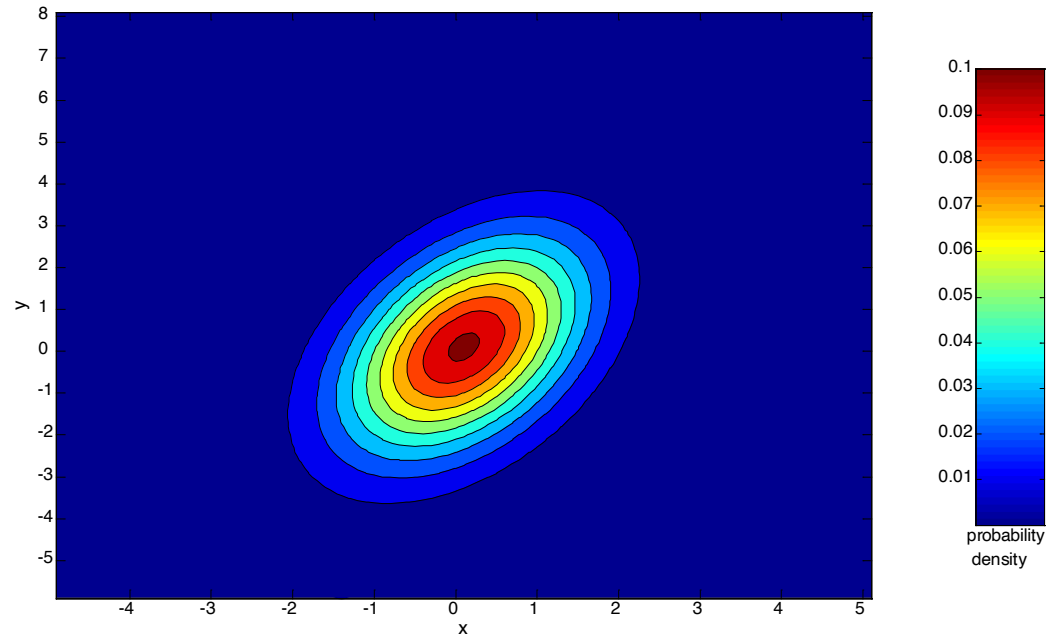
**Probabilistic definition of an outlier:** An outlier is an object that has a low probability with respect to a probability distribution model of the data.

- Usually assume a parametric model describing the distribution of the data (e.g., normal distribution)
- Apply a statistical test that depends on
  - Data distribution
  - Parameters of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)
- Issues
  - Identifying the distribution of a data set
    - Heavy tailed distribution
  - Number of attributes
  - Is the data a mixture of distributions?

# Normal Distributions



**One-dimensional  
Gaussian**



**Two-dimensional  
Gaussian**

# Statistical-based – Grubbs' Test

- Detect outliers in univariate data
- Assume data comes from normal distribution
- Detects one outlier at a time, remove the outlier, and repeat
  - $H_0$ : There is no outlier in data
  - $H_A$ : There is at least one outlier

- Grubbs' test statistic:

one-sided test with  $\alpha/N$   
two-sided test with  $\alpha/2N$

$$G = \frac{\max |X - \bar{X}|}{S}$$

mean
std dev

alpha significance  
t – Student's distribution

- Reject null hypothesis  $H_0$  of no outliers if:

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

degrees of freedom
upper critical value of t-distribution

# Statistical-based – Likelihood Approach

---

- Assume the data set  $D$  contains samples from a mixture of two probability distributions:
  - $M$  (majority distribution)
  - $A$  (anomalous distribution)
- General Approach:
  - Initially, assume all the data points belong to  $M$
  - Let  $L_t(D)$  be the log likelihood of  $D$  at time  $t$
  - For each point  $x_t$  that belongs to  $M$ , move it to  $A$ 
    - Let  $L_{t+1}(D)$  be the new log likelihood.
    - Compute the difference,  $\Delta = L_t(D) - L_{t+1}(D)$
    - If  $\Delta > c$  (some threshold), then  $x_t$  is declared as an anomaly and moved permanently from  $M$  to  $A$

# Statistical-based – Likelihood Approach

---

- Data distribution,  $D = (1 - \lambda) M + \lambda A$
- $M$  is a probability distribution estimated from data
  - Can be based on any modeling method (naïve Bayes, maximum entropy, etc.)
- $A$  is initially assumed to be uniform distribution
- Likelihood at time  $t$ :

$$L_t(D) = \prod_{i=1}^N P_D(x_i) = \left( (1 - \lambda)^{|M_t|} \prod_{x_i \in M_t} P_{M_t}(x_i) \right) \left( \lambda^{|A_t|} \prod_{x_i \in A_t} P_{A_t}(x_i) \right)$$

$$LL_t(D) = |M_t| \log(1 - \lambda) + \sum_{x_i \in M_t} \log P_{M_t}(x_i) + |A_t| \log \lambda + \sum_{x_i \in A_t} \log P_{A_t}(x_i)$$

# Strengths/Weaknesses of Statistical Approaches

---

## Pros

- Firm mathematical foundation
- Can be very efficient
- Good results if distribution is known

## Cons

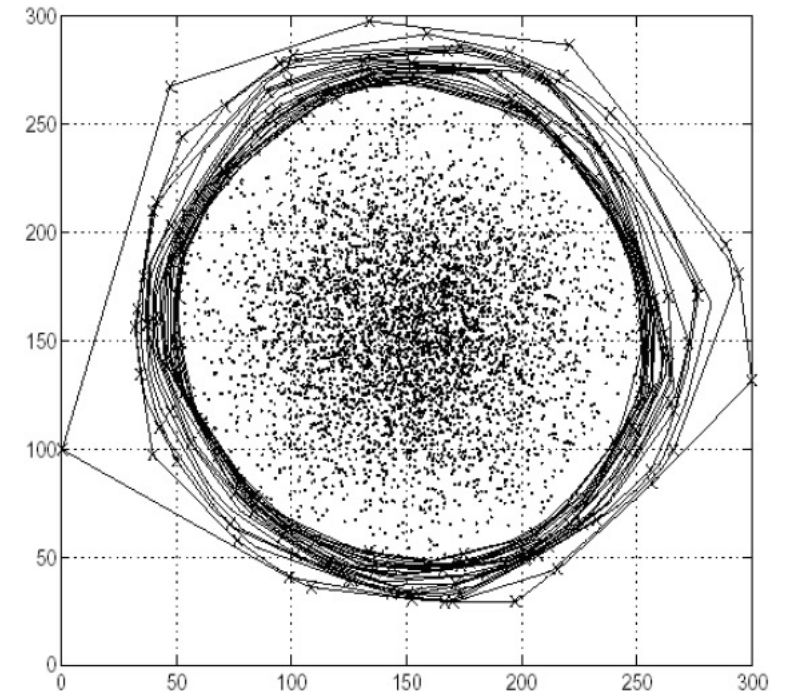
- In many cases, data distribution may not be known
- For high dimensional data, it may be difficult to estimate the true distribution
- Anomalies can distort the parameters of the distribution
  - Mean and standard deviation are very sensitive to outliers

# Depth-based Approaches

---

# Depth-based Approaches

- General idea
  - Search for outliers at the border of the data space but independent of statistical distributions
  - Organize data objects in convex hull layers
  - Outliers are objects on outer layers
- Basic assumption
  - Outliers are located at the border of the data space
  - Normal objects are in the center of the data space

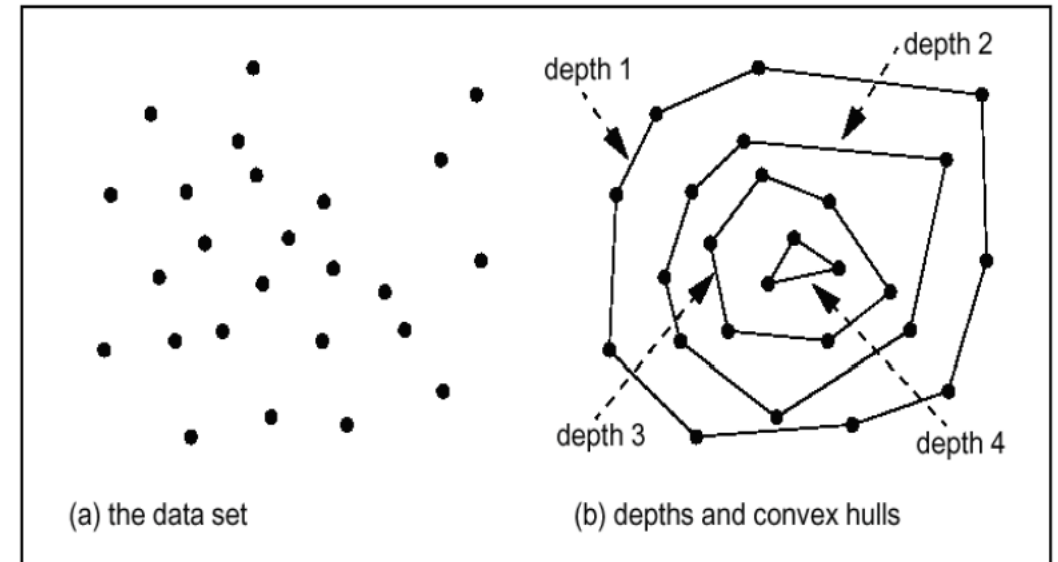




# Depth-based Approaches

Model [Tukey 1977]

- Points on the convex hull of the full data space have depth = 1
- Points on the convex hull of the data set after removing all points with depth = 1 have depth = 2
- — ...
- Points having a depth  $\leq k$  are reported as outliers



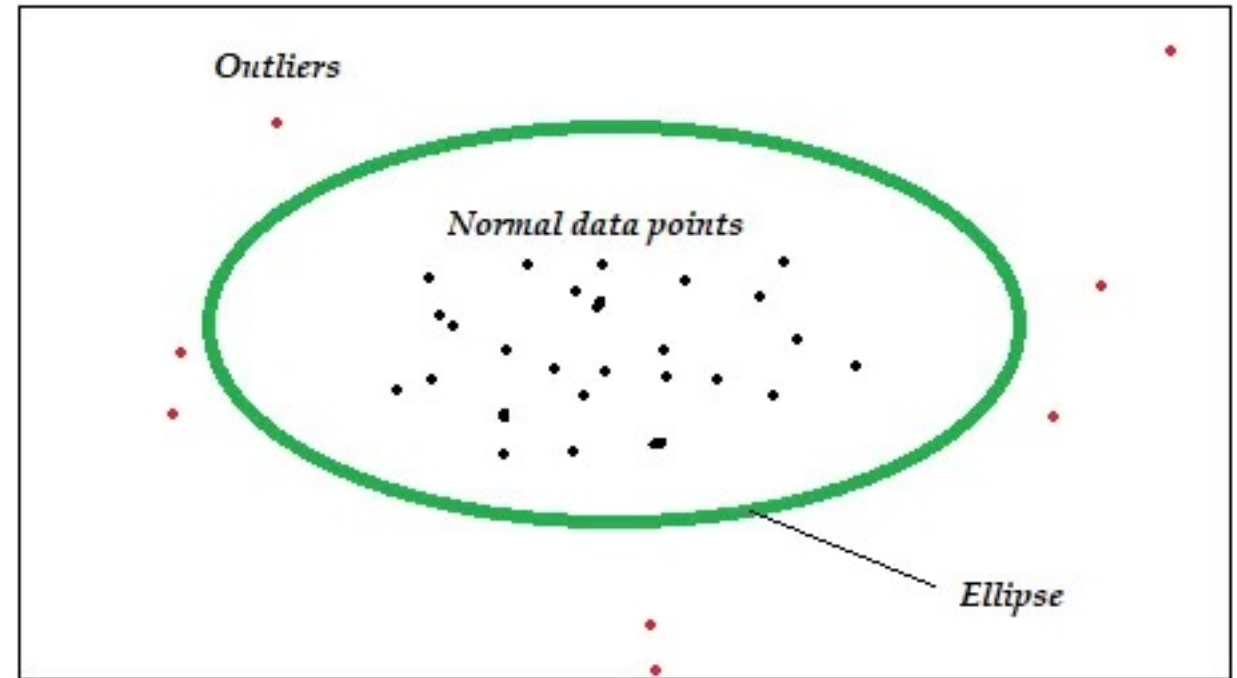
# Depth-based Approaches

---

- Similar idea like classical statistical approaches ( $k = 1$  distributions) but independent from the chosen kind of distribution
- Convex hull computation is usually only efficient in 2D / 3D spaces
- Originally outputs a label but can be extended for scoring easily (take depth as scoring value)
- Uses a global reference set for outlier detection
  
- Sample algorithms
  - ISODEPTH [Ruts and Rousseeuw 1996]
  - FDC [Johnson et al. 1998]

# Elliptic Envelope

- It creates an imaginary elliptical area around a given dataset.
- Values that fall inside the envelope are considered normal data and anything outside the envelope is returned as outliers.
- The algorithm works best if data has a Gaussian distribution.



# Deviation-based Approaches

---

# Deviation-based Approaches

---

- General idea
  - Given a set of data points (local group or global set)
  - Outliers are points that do not fit to the general characteristics of that set, i.e., the variance of the set is minimized when removing the outliers
- Basic assumption
  - Outliers are the outermost points of the data set

# Deviation-based Approaches

---

Model [Arning et al. 1996]

- Given a smoothing factor  $SF(I)$  that computes for each  $I \subseteq DB$  how much the variance of  $DB$  is decreased when  $I$  is removed from  $DB$
- With equal decrease in variance, a smaller exception set  $E$  is better
- The outliers are the elements of  $E \subseteq DB$  for which the following holds:  $SF(E) \geq SF(I)$  for all  $I \subseteq DB$

Discussion:

- Similar idea like classical statistical approaches ( $k = 1$  distributions) but independent from the chosen kind of distribution
- Naïve solution is in  $O(2^n)$  for  $n$  data objects
- Heuristics like random sampling or best first search are applied
- Applicable to any data type (depends on the definition of  $SF$ )
- Originally designed as a global method
- Outputs a labeling

# Distance-based Approaches

---

# Distance-based Approaches

---

- General Idea
  - Judge a point based on the distance(s) to its neighbors
  - Several variants proposed
- Basic Assumption
  - Normal data objects have a dense neighborhood
  - Outliers are far apart from their neighbors, i.e., have a less dense neighborhood



# Distance-based Approaches

---

- Several different techniques
- Approach 1: An object is an outlier if a specified fraction of the objects is more than a specified distance away (Knorr, Ng 1998)
  - Some statistical definitions are special cases of this
- Approach 2: The outlier score of an object is the distance to its  $k$ -th nearest neighbor

# Outlier scoring based on kNN distances

---

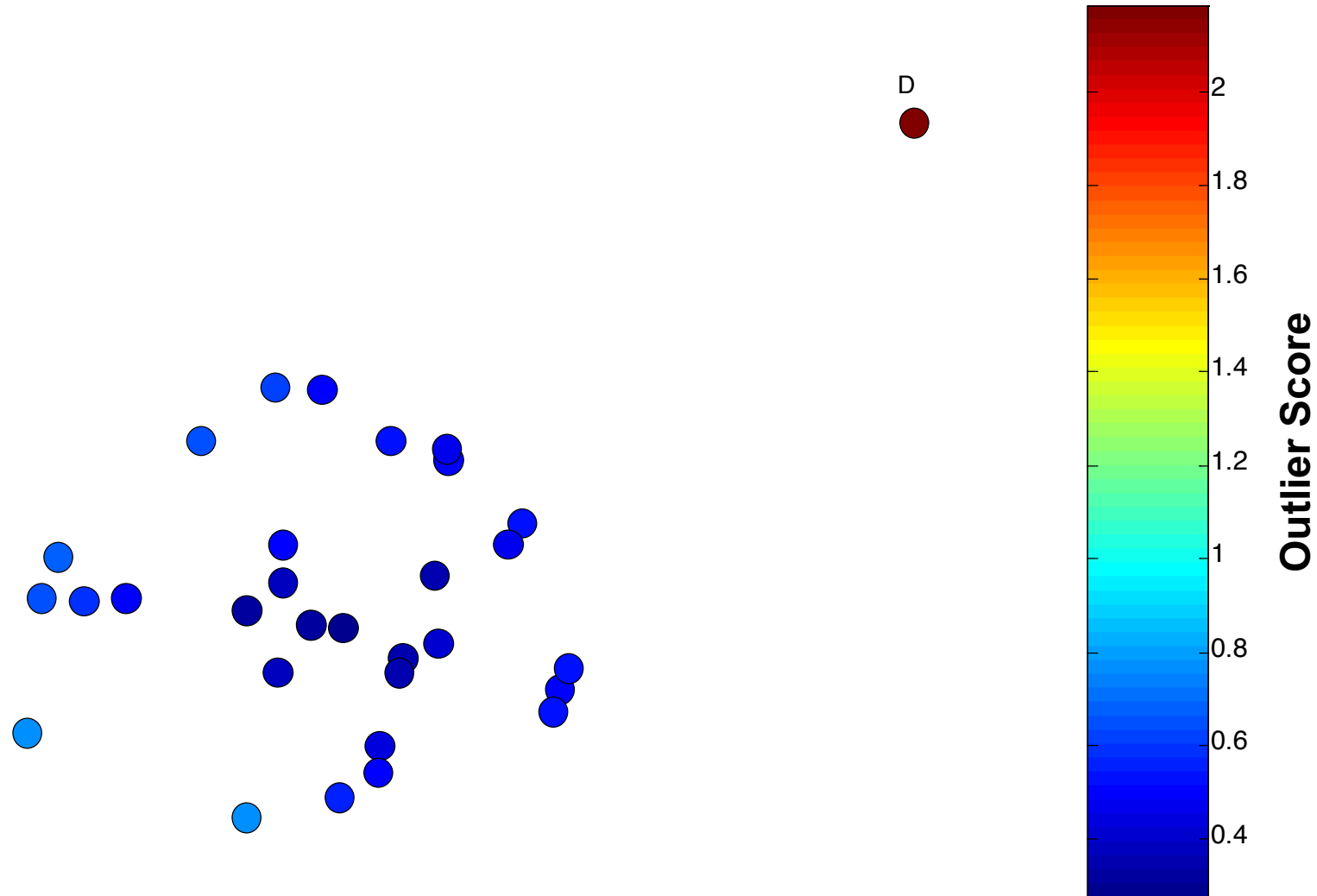
## General models

- Take the kNN distance of a point as its outlier score [Ramaswamy et al 2000]
- Aggregate the distances of a point to all its 1NN, 2NN, ..., kNN as an outlier score [Angiulli and Pizzuti 2002]

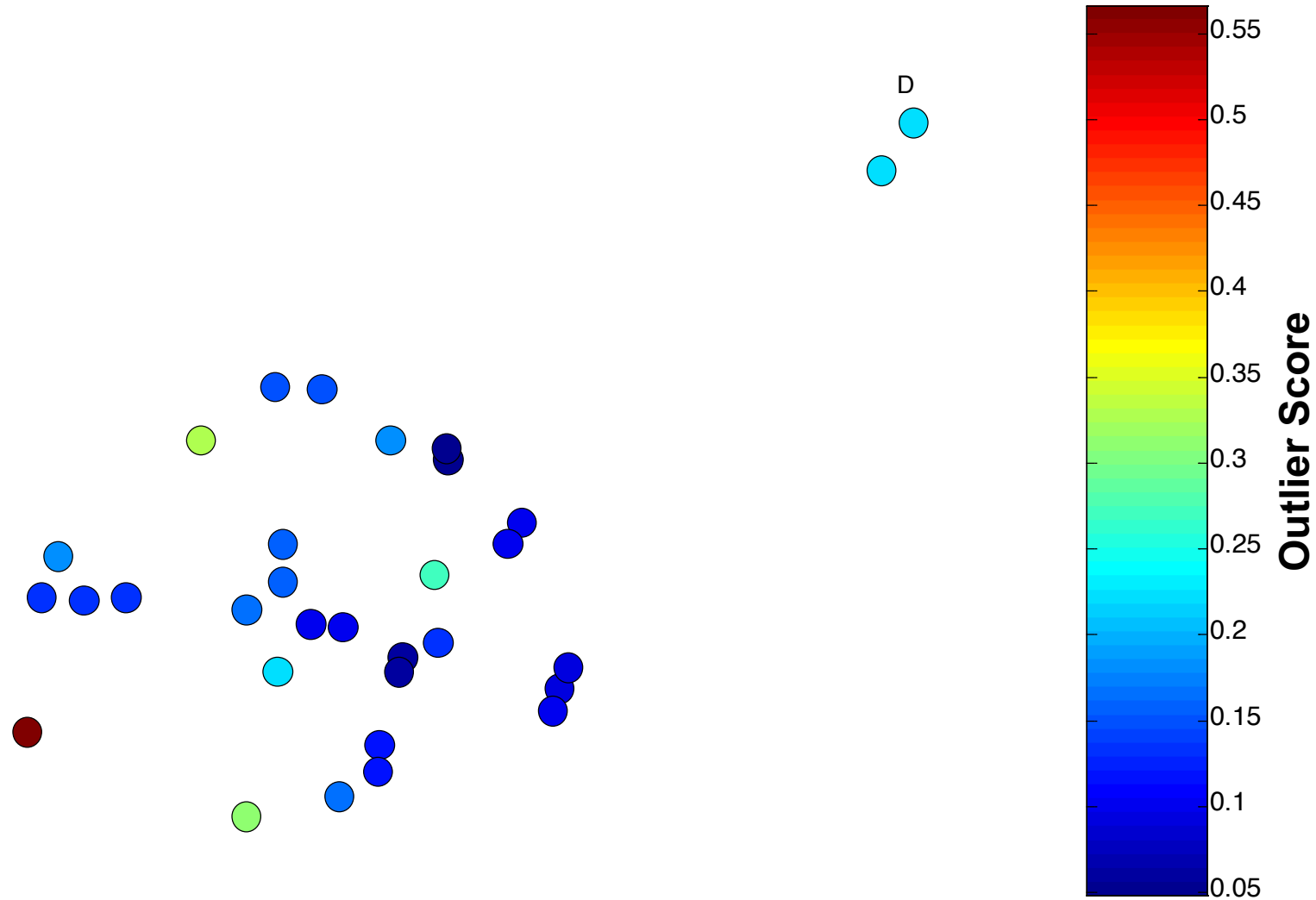
## Algorithms - General approaches

- Nested-Loop
  - Naïve approach: For each object: compute kNNs with a sequential scan
  - Enhancement: use index structures for kNN queries
- Partition-based
  - Partition data into micro clusters
  - Aggregate information for each partition (e.g. minimum bounding rectangles)
  - Allows to prune micro clusters that cannot qualify when searching for the kNNs of a particular point

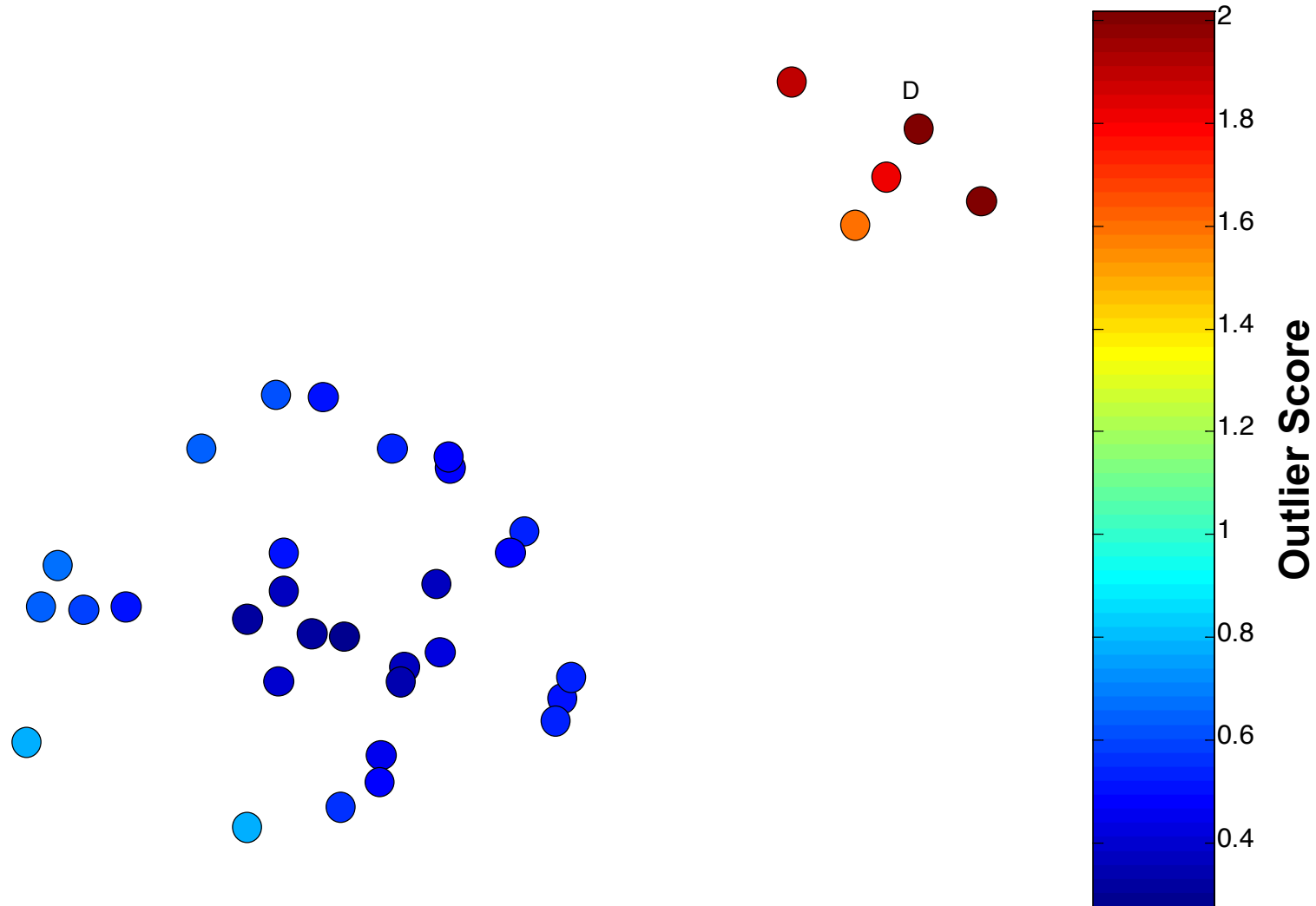
# One Nearest Neighbor - One Outlier



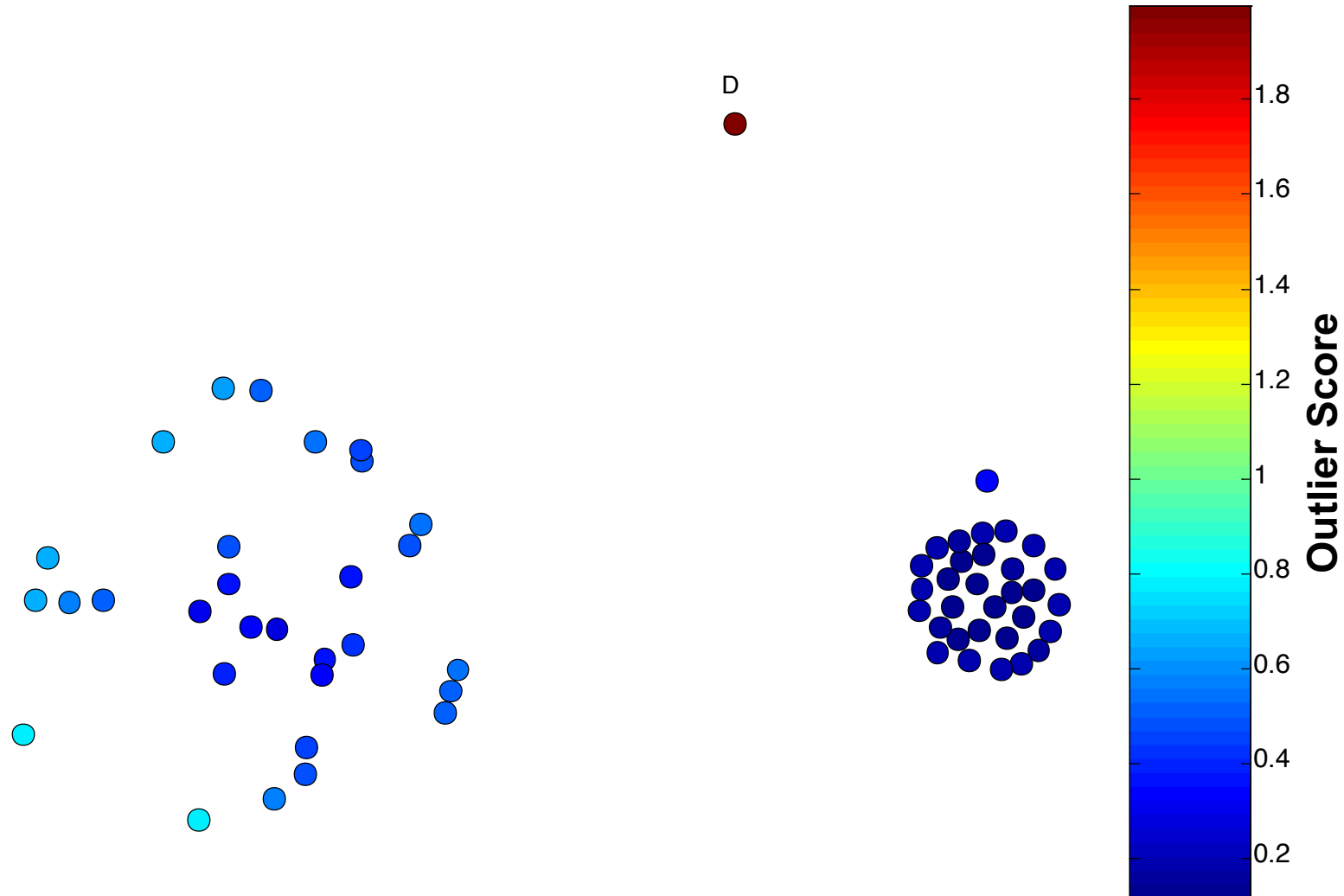
# One Nearest Neighbor - Two Outliers



# Six Nearest Neighbors - Small Cluster



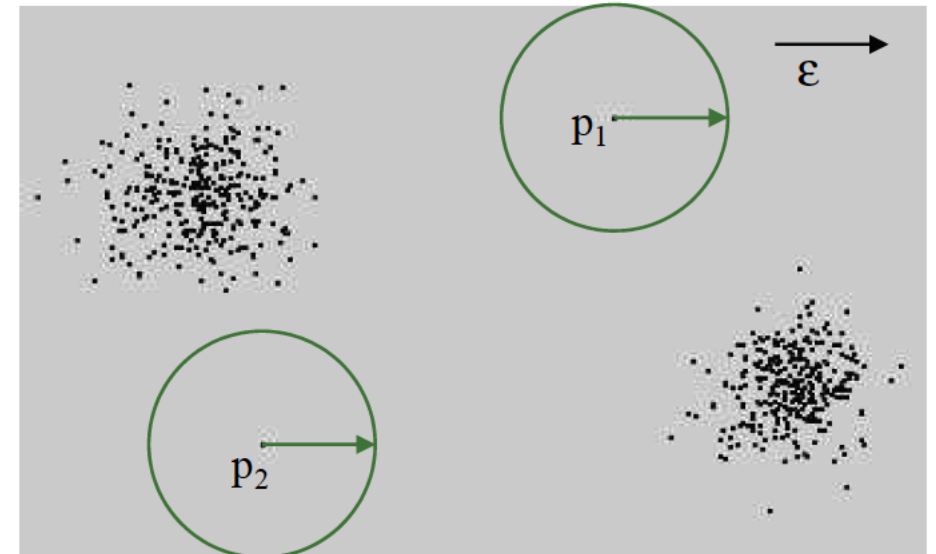
# Five Nearest Neighbors - Differing Density



# Distance-based Approaches

## DB( $\epsilon, \pi$ )-Outliers

- Basic model [Knorr and Ng 1997]
- Given a radius  $\epsilon$  and a percentage  $\pi$
- A point  $p$  is considered an outlier if at most  $\pi$  percent of all other points have a distance to  $p$  less than  $\epsilon$ , *i.e.*, *it is close to few points*



$$OutlierSet(\epsilon, \pi) = \left\{ p \mid \frac{Card(\{q \in DB \mid dist(p, q) < \epsilon\})}{Card(DB)} \leq \pi \right\}$$

range-query with radius  $\epsilon$

# Distance-based Approaches - Algorithms

---

- Index-based [Knorr and Ng 1998]
  - Compute distance range join using spatial index structure
  - Exclude point from further consideration if its  $\varepsilon$ -neighborhood contains more than  $\text{Card}(\text{DB}) \pi$  points
- Nested-loop based [Knorr and Ng 1998]
  - Divide buffer in two parts
  - Use second part to scan/compare all points with the points from the first part
- Grid-based [Knorr and Ng 1998]
  - Build grid such that any two points from the same grid cell have a distance of at most  $\varepsilon$  to each other
  - Points need only compared with points from neighboring cells



# Outlier Detection using In-degree Number

---

- Idea: Construct the kNN graph for a data set
  - Vertices: data points
  - Edge: if  $q \in kNN(p)$  then there is a directed edge from  $p$  to  $q$
  - A vertex that has an indegree less than equal to  $T$  (user threshold) is an outlier
- Discussion
  - The indegree of a vertex in the kNN graph equals to the number of reverse kNNs (RkNN) of the corresponding point
  - The RkNNs of a point  $p$  are those data objects having  $p$  among their kNNs
  - Intuition of the model: outliers are
    - points that are among the kNNs of less than  $T$  other points
    - have less than  $T$  RkNNs
  - Outputs an outlier label
  - Is a local approach (depending on user defined parameter  $k$ )

# Strengths/Weaknesses of Distance-Based Approaches

---

## Pros

- Simple

## Cons

- Expensive –  $O(n^2)$
- Sensitive to parameters
- Sensitive to variations in density
- Distance becomes less meaningful in high-dimensional space

# Density-based Approaches

---

# Density-based Approaches

---

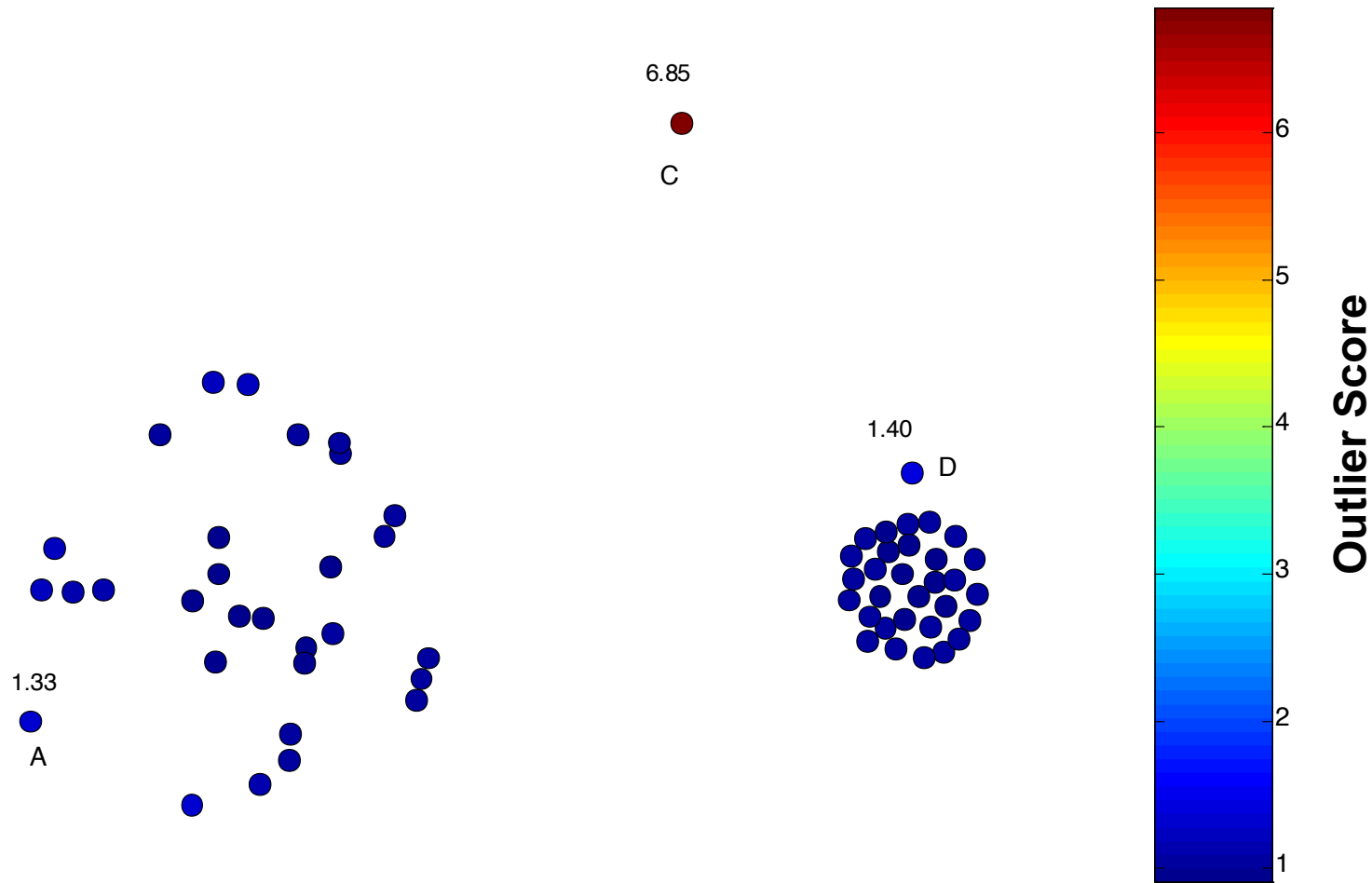
- General idea
  - Compare the density around a point with the density around its local neighbors
  - The relative density of a point compared to its neighbors is computed as an outlier score
  - Approaches differ in how to estimate density
- Basic assumption
  - The density around a normal data object is similar to the density around its neighbors
  - The density around an outlier is considerably different to the density around its neighbors

# Density-based Approaches

---

- **Density-based Outlier:** The outlier score of an object is the inverse of the density around the object.
  - Can be defined in terms of the  $k$  nearest neighbors
  - One definition: Inverse of distance to  $k$ th neighbor
  - Another definition: Inverse of the average distance to  $k$  neighbors
  - DBSCAN definition
- If there are regions of different density, this approach can have problems

# Relative Density Outlier Scores



# Relative Density

---

- Consider the density of a point relative to that of its  $k$  nearest neighbors

$$\text{average relative density}(\mathbf{x}, k) = \frac{\text{density}(\mathbf{x}, k)}{\sum_{\mathbf{y} \in N(\mathbf{x}, k)} \text{density}(\mathbf{y}, k) / |N(\mathbf{x}, k)|}. \quad (10.7)$$

---

**Algorithm 10.2** Relative density outlier score algorithm.

---

- 1:  $\{k$  is the number of nearest neighbors $\}$
  - 2: **for all** objects  $\mathbf{x}$  **do**
  - 3:     Determine  $N(\mathbf{x}, k)$ , the  $k$ -nearest neighbors of  $\mathbf{x}$ .
  - 4:     Determine  $\text{density}(\mathbf{x}, k)$ , the density of  $\mathbf{x}$ , using its nearest neighbors, i.e., the objects in  $N(\mathbf{x}, k)$ .
  - 5: **end for**
  - 6: **for all** objects  $\mathbf{x}$  **do**
  - 7:     Set the *outlier score* $(\mathbf{x}, k) = \text{average relative density}(\mathbf{x}, k)$  from Equation 10.7.
  - 8: **end for**
-

# Local Outlier Factor (LOF) [Breunig et al. 1999], [Breunig et al. 2000]

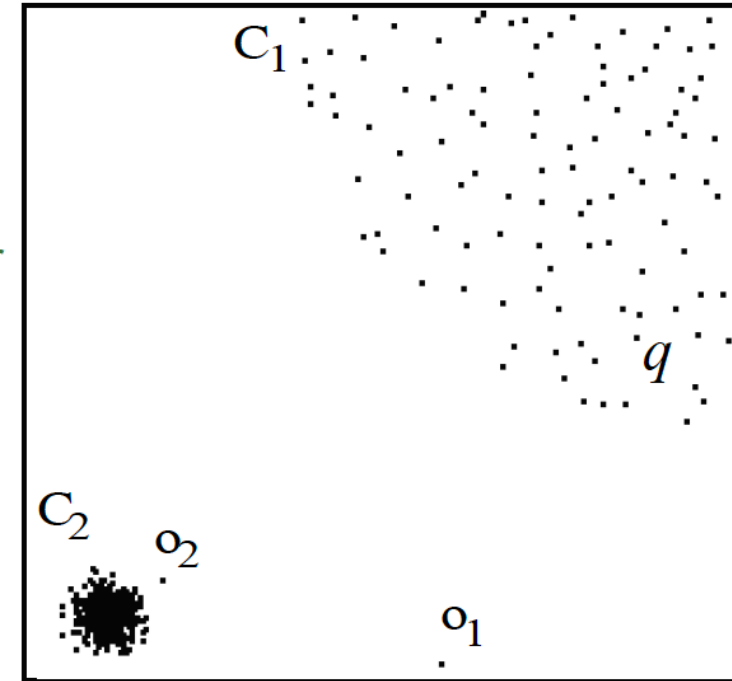
Motivation:

- Distance-based outlier detection models have problems with different densities
- How to compare the neighborhood of points from areas of different densities?

Example

- DB( $\epsilon, \pi$ )-outlier model
  - Parameters  $\epsilon$  and  $\pi$  cannot be chosen so that  $o_2$  is an outlier but none of the points in cluster  $C_1$  (e.g.  $q$ ) is an outlier
- Outliers based on kNN-distance
  - kNN-distances of objects in  $C_1$  (e.g.  $q$ ) are larger than the kNN-distance of  $o_2$

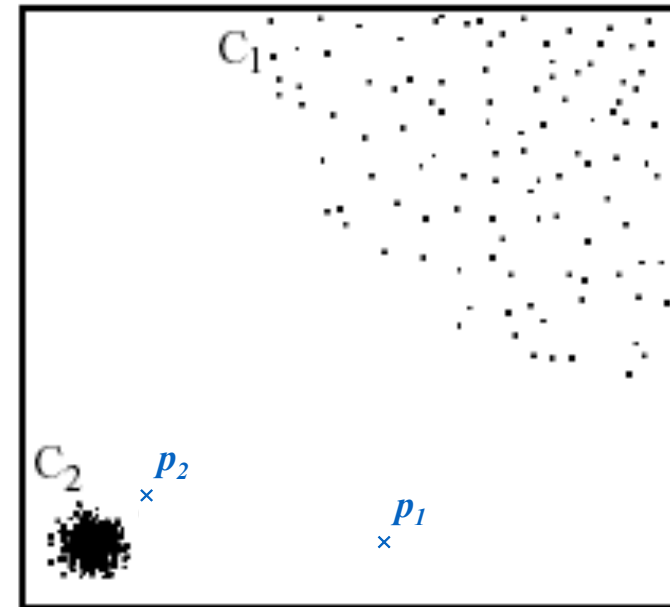
Solution: consider relative density





# Local Outlier Factor (LOF)

- For each point, compute the density of its local neighborhood
- Compute local outlier factor (LOF) of a sample  $p$  as the average of the ratios of the density of sample  $p$  and the density of its nearest neighbors
- Outliers are points with largest LOF value

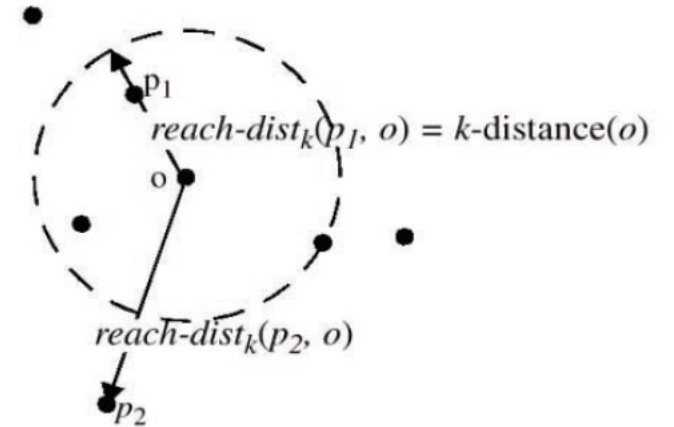


In the NN approach,  $p_2$  is not considered as outlier, while LOF approach find both  $p_1$  and  $p_2$  as outliers

# Local Outlier Factor (LOF)

- Reachability distance
  - Introduces a smoothing factor

$$reach-dist_k(p, o) = \max \{k\text{-distance}(o), dist(p, o)\}$$



- Local reachability distance ( $lrd$ ) of point  $p$ 
  - Inverse of the average reach-dists of the  $kNN$ s of  $p$

$$lrd_k(p) = 1 / \left( \frac{\sum_{o \in kNN(p)} reach-dist_k(p, o)}{Card(kNN(p))} \right)$$

- Local outlier factor (LOF) of point  $p$ 
  - Average ratio of  $lrd$ s of neighbors of  $p$  and  $lrd$  of  $p$

$$LOF_k(p) = \frac{\sum_{o \in kNN(p)} \frac{lrd_k(o)}{lrd_k(p)}}{Card(kNN(p))}$$

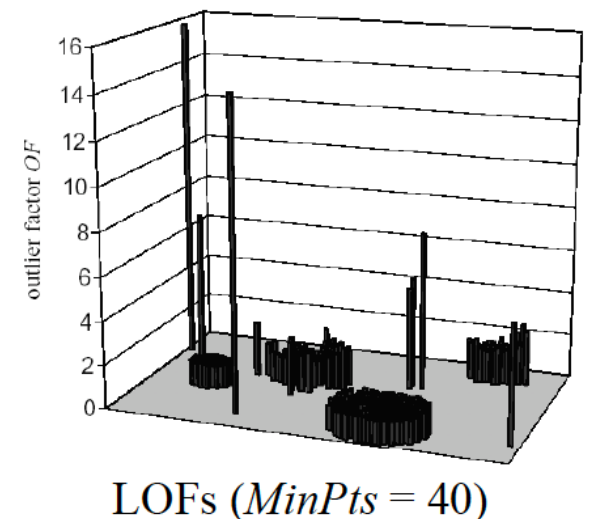
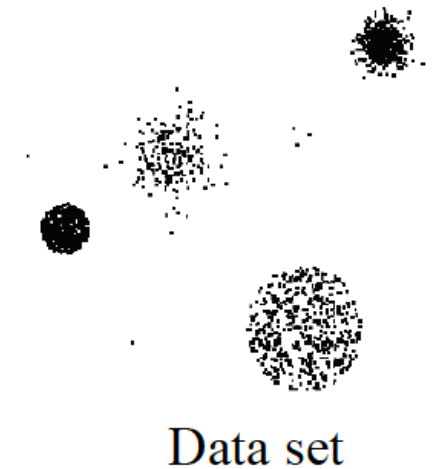
# Local Outlier Factor (LOF)

## Properties

- $LOF \approx 1$ : point is in a cluster (region with homogeneous density around the point and its neighbors)
- $LOF \gg 1$ : point is an outlier

## Discussion

- Choice of  $k$  (MinPts in the original paper) specifies the reference set
- Originally implements a *local* approach (resolution depends on the user's choice for  $k$ )
- Outputs a scoring (assigns an LOF value to each point)



# Mining Top-n Local Outliers [Jin et al. 2001]

---

Idea:

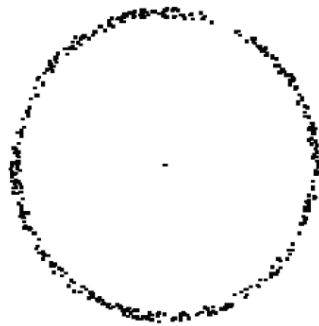
- Usually, a user is only interested in the **top-n** outliers
- Do not compute the LOF for all data objects => save runtime

Method

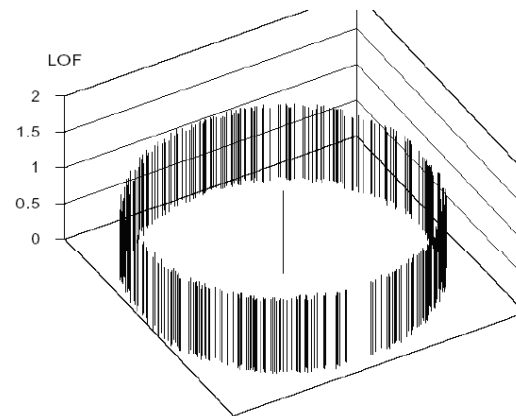
- Compress data points into micro clusters using the CFs of BIRCH [Zhang et al. 1996]
- Derive upper and lower bounds of the reachability distances, lrd-values, and LOF-values for points within a micro clusters
- Compute upper and lower bounds of LOF values for micro clusters and sort results w.r.t. ascending lower bound
- Prune micro clusters that cannot accommodate points among the top-n outliers (n highest LOF values)
- Iteratively refine remaining micro clusters and prune points accordingly

# Connectivity-based outlier factor (COF) [Tang et al. 2002]

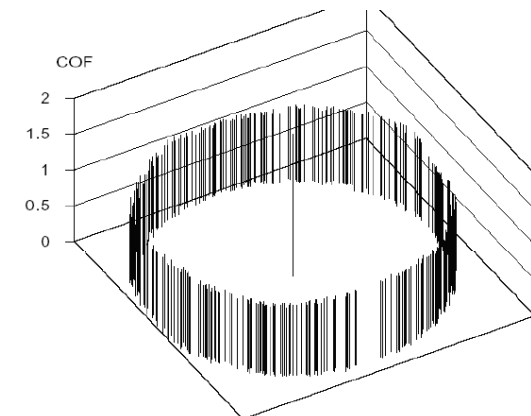
- Motivation
  - In regions of low density, it may be hard to detect outliers
  - Choose a low value for  $k$  is often not appropriate
- Solution
  - Treat “low density” and “isolation” differently
- Example



Data set



LOF



COF

# COF

---

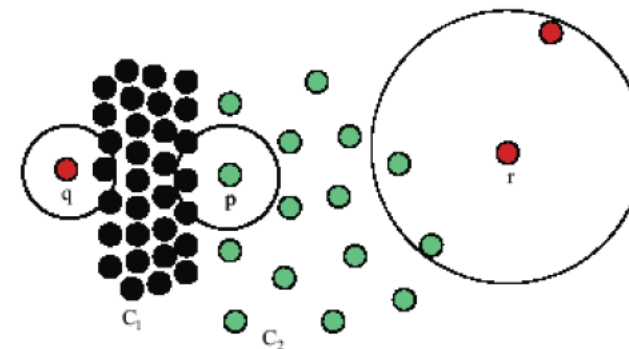
- Introduced because although a high-density set can represent a pattern, not all patterns need to be high-density.
- COF differs from LOF as it uses the chaining distance to calculate the kNN.
- The chaining distances are the minimum of the total sum of the distances linking all neighbors.
- The *connectivity* is then calculated as the ratio between the average chaining distance of the record and the mean average chaining distance of the records in the kNN.

$$COF_k(p) = \frac{|N_k(p)|ac - dist_{N_k(p)}(p)}{\sum_{o \in N_k(p)} ac - dist_{N_k(o)}(o)} \quad ac - dist_{N_k(p_1)}(p_1) = \sum_{i=1}^r \frac{2(r-1+i)}{r(r+1)} \quad r = |N_k(p_1)|$$

# Influenced Outlierness (INFLO) [Jin et al. 2006]

## Motivation

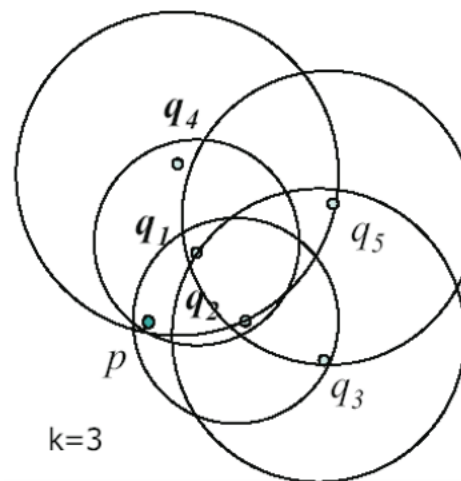
- If clusters of different densities are not clearly separated, LOF will have problems



Point  $p$  will have a higher LOF than points  $q$  or  $r$  which is counter intuitive

## Idea

- Take symmetric neighborhood relationship into account
- Influence space  $kIS(p)$  of a point  $p$  includes its  $k$ NNs ( $kNN(p)$ ) and its reverse  $k$ NNs ( $RkNN(p)$ )



$$\begin{aligned} kIS(p) &= kNN(p) \cup RkNN(p) \\ &= \{q_1, q_2, q_4\} \end{aligned}$$

# Influenced Outlierness (INFLO) [Jin et al. 2006]

## Model

- Density is simply measured by the inverse of the kNN distance, i.e.,
  - $den(p) = 1/k\text{-distance}(p)$

- Influenced outlierness of a point  $p$

$$INFLO_k(p) = \frac{\frac{\sum_{o \in kIS(p)} den(o)}{Card(kIS(p))}}{den(p)}$$

- INFLO takes the ratio of the average density of objects in the neighborhood of a point  $p$  (i.e., in  $kNN(p) \cup RkNN(p)$ ) to  $p$ 's density

## Proposed algorithms for mining top-n outliers

- Index-based
- Two-way approach
- Micro cluster based approach



# Influenced Outlierness (INFLO) [Jin et al. 2006]

---

## Properties

- Similar to LOF
- $\text{INFLO} \approx 1$ : point is in a cluster
- $\text{INFLO} \gg 1$ : point is an outlier

## Discussion

- Outputs an outlier score
- Originally proposed as a *local* approach (resolution of the reference set  $k/S$  can be adjusted by the user setting parameter  $k$ )

# Strengths/Weaknesses of Density-Based Approaches

---

## Pros

- Simple

## Cons

- Expensive –  $O(n^2)$
- Sensitive to parameters
- Density becomes less meaningful in high-dimensional space

# Clustering-based Approaches

---

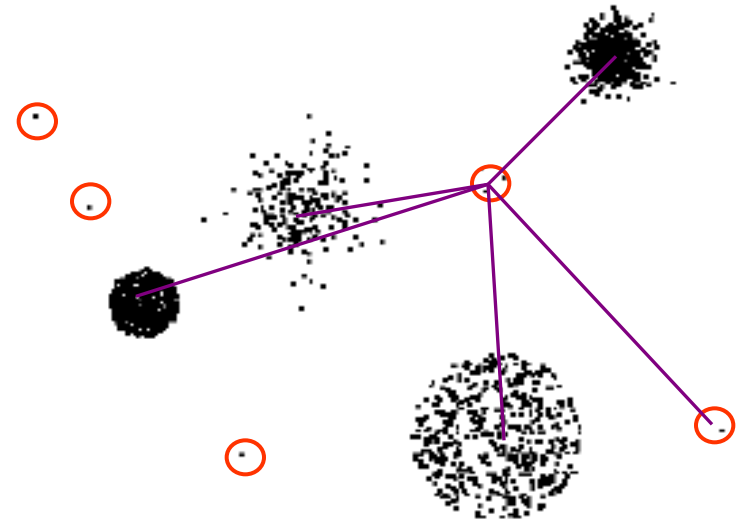
# Clustering and Anomaly Detection

---

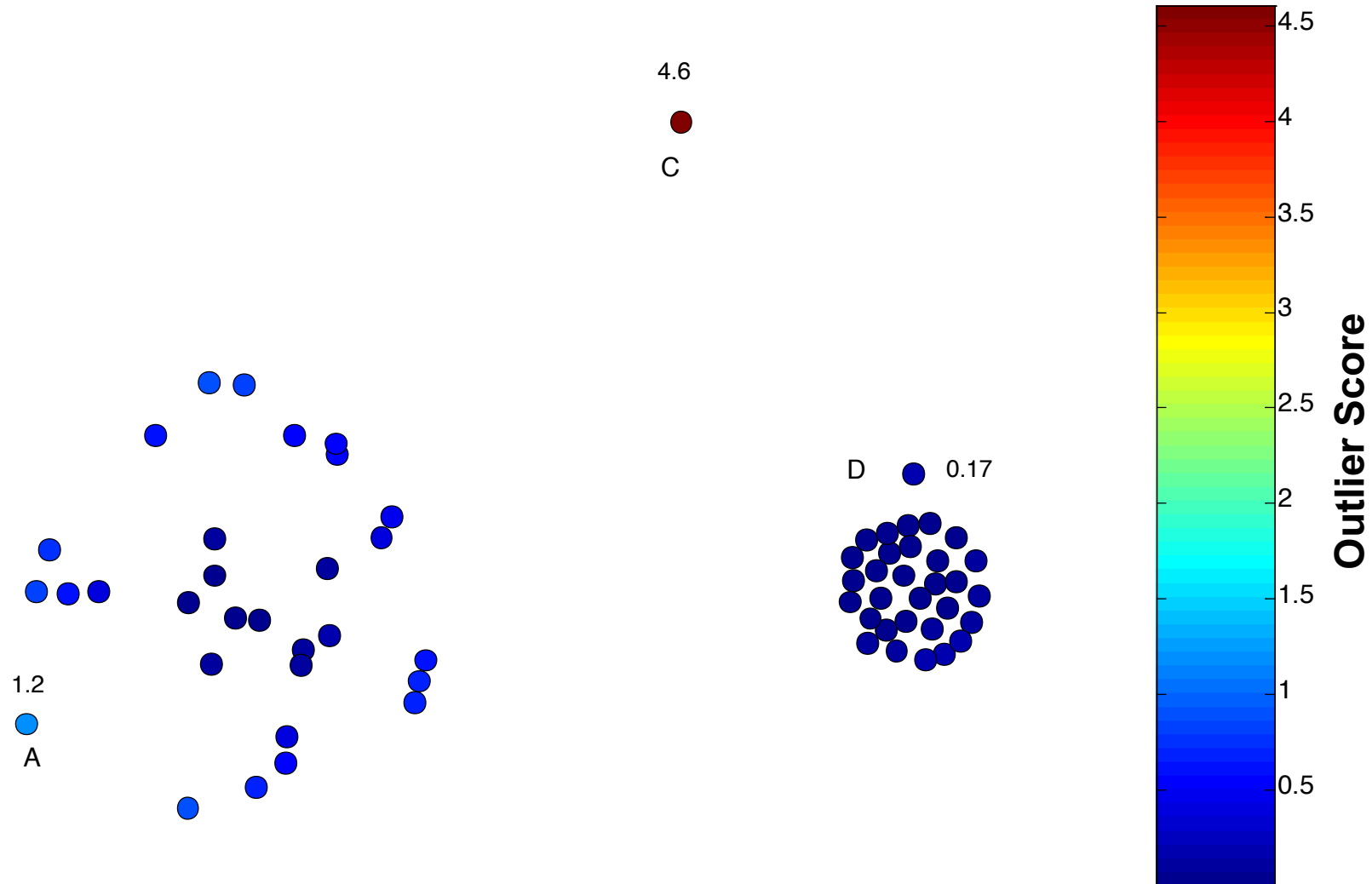
- Are outliers just a side product of some clustering algorithms?
  - Many clustering algorithms do not assign all points to clusters but account for noise objects (e.g. DBSCAN, OPTICS)
  - Look for outliers by applying one algorithm and retrieve the noise set
- Problem:
  - Clustering algorithms are optimized to find clusters rather than outliers
  - Accuracy of outlier detection depends on how good the clustering algorithm captures the structure of clusters
  - A set of many abnormal data objects that are similar to each other would be recognized as a cluster rather than as noise/outliers

# Clustering-Based Approaches

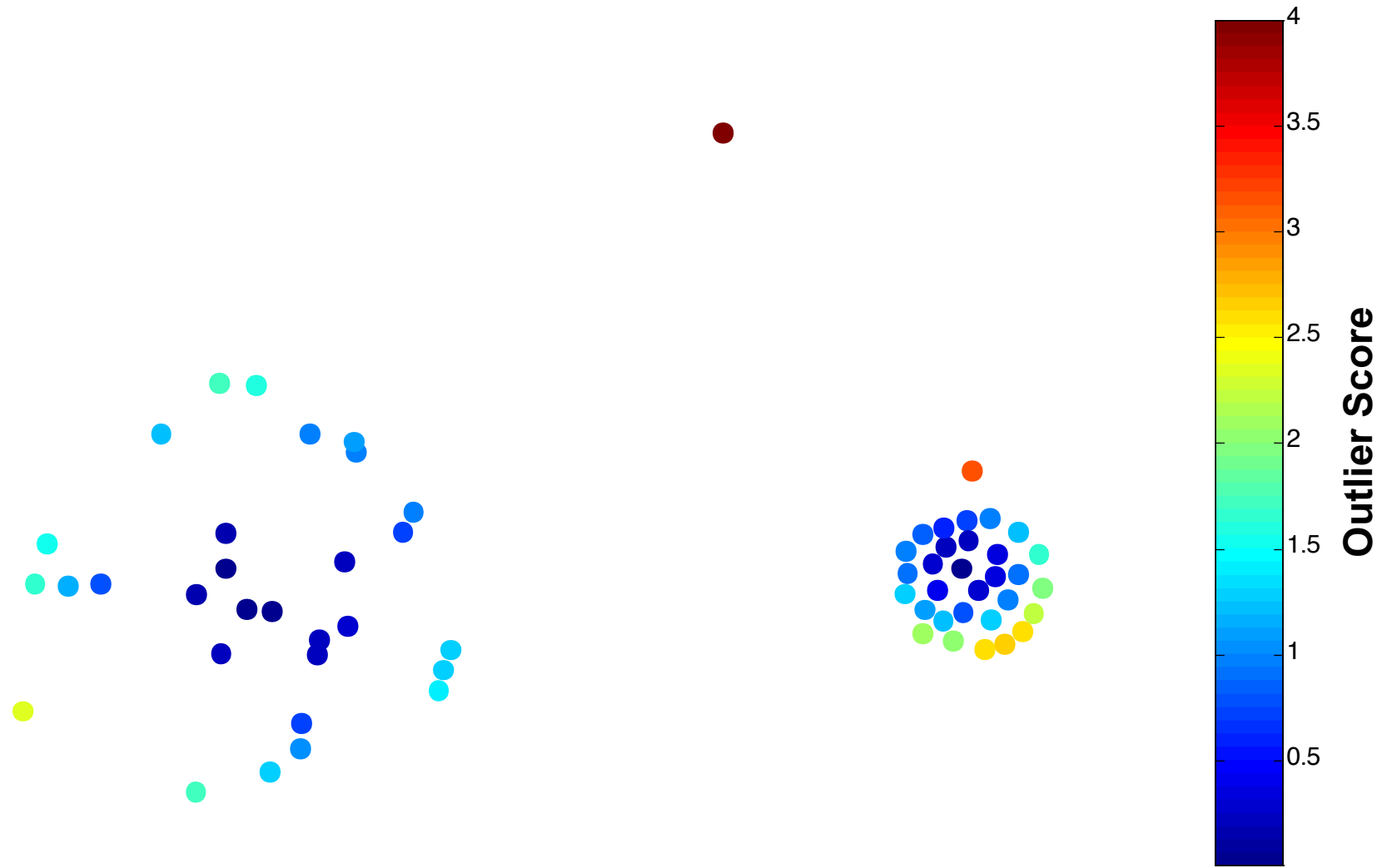
- **Clustering-based Outlier:** An object is a cluster-based outlier if it does not strongly belong to any cluster
  - For prototype-based clusters, an object is an outlier if it is not close enough to a cluster center
  - For density-based clusters, an object is an outlier if its density is too low
  - For graph-based clusters, an object is an outlier if it is not well connected
- Other issues include the impact of outliers on the clusters and the number of clusters



# Distance of Points from Closest Centroids



# Relative Distance of Points from Closest Centroid



# Strengths/Weaknesses of Clustering-Based Approaches

---

## **Pros**

- Simple
- Many clustering techniques can be used

## **Cons**

- Can be difficult to decide on a clustering technique
- Can be difficult to decide on number of clusters
- Outliers can distort the clusters



# High-dimensional Approaches

---

# Challenges

---

## Curse of dimensionality

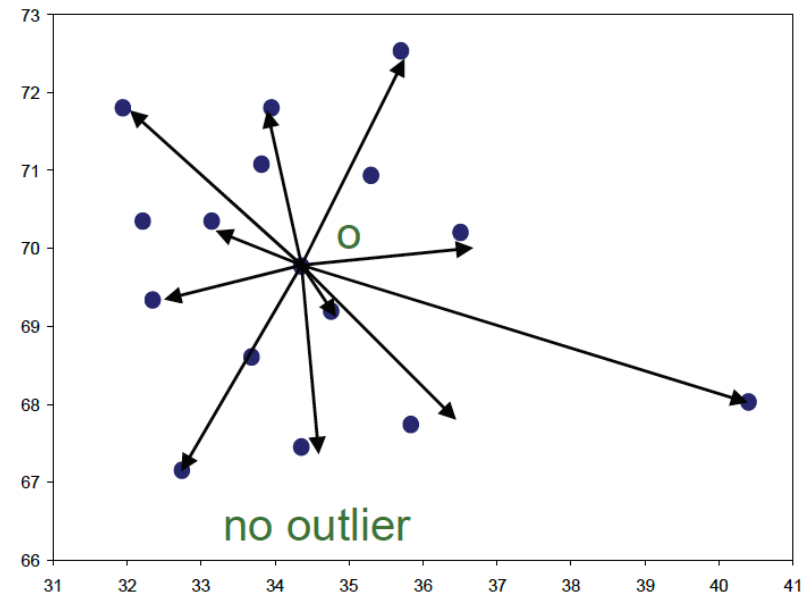
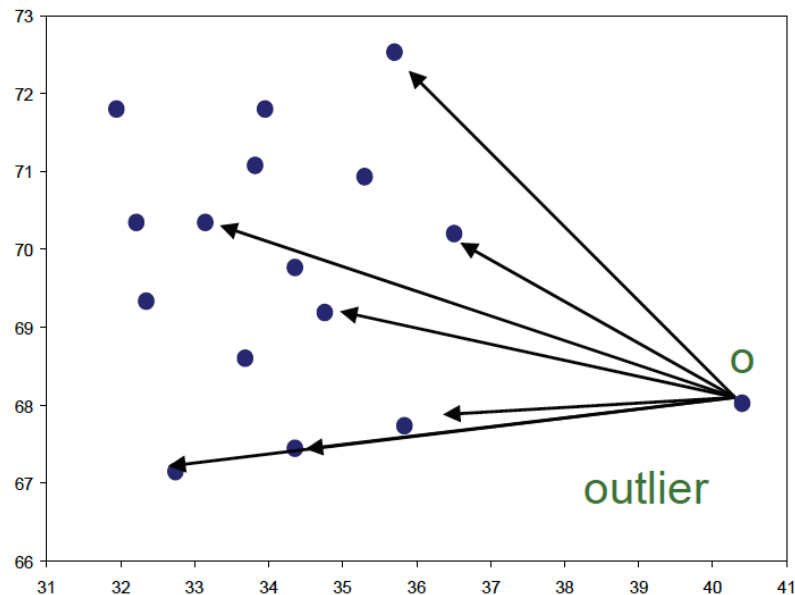
- Relative contrast between distances decreases with increasing dimensionality
- Data is very sparse, almost all points are outliers
- Concept of neighborhood becomes meaningless

## Solutions

- Use more robust distance functions and find full-dimensional outliers
- Find outliers in projections (subspaces) of the original feature space

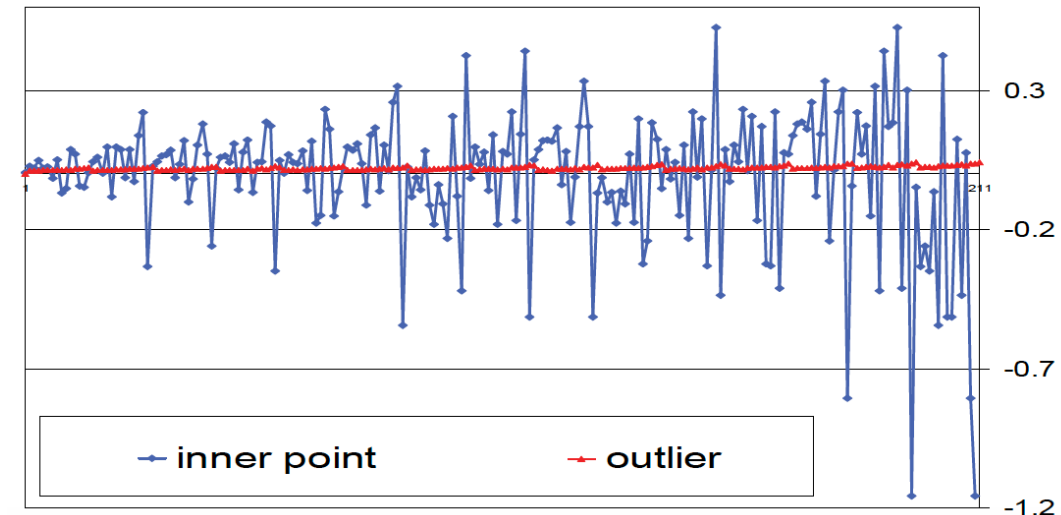
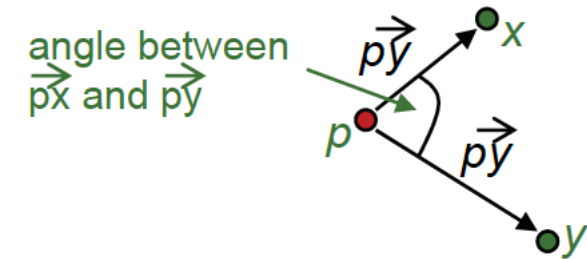
# ABOD – Angle-based Outlier Degree [Kriegel et al. 2008]

- Angles are more stable than distances in high dimensional spaces (e.g. the popularity of cosine-based similarity measures for text data)
- Object  $o$  is an outlier if most other objects are located in similar directions
- Object  $o$  is no outlier if many other objects are located in varying directions



# ABOD – Angle-based Outlier Degree [Kriegel et al. 2008]

- Basic assumption
  - Outliers are at the border of the data distribution
  - Normal points are in the center of the data distribution
- Model
  - Consider for a given point  $p$  the angle between any two instances  $x$  and  $y$
  - Consider the spectrum of all these angles
  - The broadness of this spectrum is a score for the outlierness of a point



# ABOD – Angle-based Outlier Degree [Kriegel et al. 2008]

- Model
  - Measure the variance of the angle spectrum
  - Weighted by the corresponding distances (for lower dimensional data sets where angles are less reliable)

$$ABOD(p) = VAR_{x,y \in DB} \left( \frac{\langle \vec{xp}, \vec{yp} \rangle}{\|\vec{xp}\|^2 \cdot \|\vec{yp}\|^2} \right)$$

- Properties
  - Small ABOD => outlier
  - High ABOD => no outlier

$\vec{xp}$  denotes the difference vector  $x-p$   
 $\langle \vec{xp}, \vec{yp} \rangle$  denotes the scalar product  
scalar product  $\langle a, b \rangle = \sum a_i b_i$

# ABOD – Angle-based Outlier Degree [Kriegel et al. 2008]

---

## Algorithms

- Naïve algorithm is in  $O(n^3)$
- Approximate algorithm based on random sampling for mining top-n outliers
  - Do not consider all pairs of other points  $x, y$  in the database to compute the angles
  - Compute ABOD based on samples => lower bound of the real ABOD
  - Filter out points that have a high lower bound
  - Refine (compute the exact ABOD value) only for a small number of points

## Discussion

- Global approach to outlier detection
- Outputs an outlier score

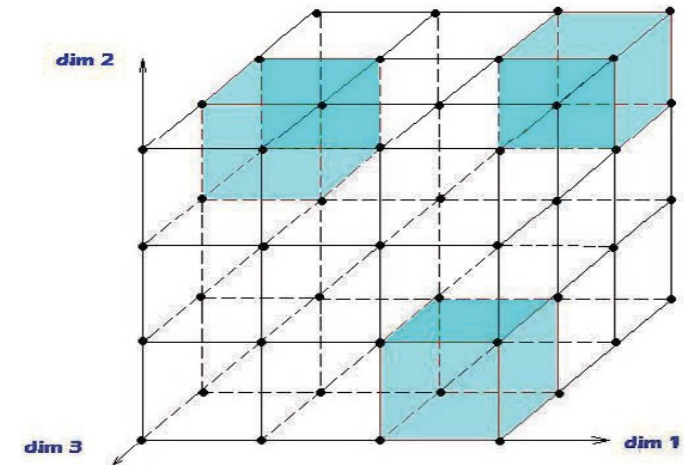
# Grid-based Subspace Outlier Detection [Aggarwal and Yu 2000]

## Model

- Partition data space by an equi-depth grid ( $\Phi$  = number of cells in each dimension)
- Sparsity coefficient  $S(C)$  for a  $k$ -dimensional grid cell  $C$

$$S(C) = \frac{\text{count}(C) - n \cdot \left(\frac{1}{\Phi}\right)^k}{\sqrt{n \cdot \left(\frac{1}{\Phi}\right)^k \cdot \left(1 - \left(\frac{1}{\Phi}\right)^k\right)}}$$

- where  $\text{count}(C)$  is the number of data objects in  $C$
- $S(C) < 0 \Rightarrow \text{count}(C)$  is lower than expected
- Outliers are those objects that are located in lower-dimensional cells with negative sparsity coefficient



$k$  = nbr dimensions (e.g. 3)  
 $\phi$  = nbr of equi-depth ranges (e.g 3)

# Grid-based Subspace Outlier Detection [Aggarwal and Yu 2000]

---

- Algorithm
  - Find the  $m$  grid cells (projections) with the lowest sparsity coefficients
  - Brute-force algorithm is *in*  $O(\Phi d)$
  - Evolutionary algorithm (input:  $m$  and the dimensionality of the cells)
- Discussion
  - Results need not be the points from the optimal cells
  - Very coarse model (all objects that are in cell with less points than to be expected)
  - Quality depends on grid resolution and grid position
  - Outputs a labeling
  - Implements a global approach (key criterion: globally expected number of points within a cell)



# Ensemble-based Approaches

---

# FeaBag - Feature Bagging

---

- FeaBag exploits a set of OD methods, each of them applied on a random set of features selected from the original feature space.
- Each OD method identifies different outliers and assigns to all instances outlier scores that correspond to their probability of being outliers.
- The combination of such scores is returned as the final output.

# LODA - Lightweight On-line Detector of Anomalies

---

- An extension of HBOS is LODA.
- LODA is an ensemble OD method particularly useful in real-time scenarios domains where many records need to be processed.
- LODA approximates the joint probability using a collection of one-dimensional histograms, where every one-dimensional histogram is efficiently constructed on an input space projected onto a randomly generated vector.
- Even though one-dimensional histograms are weak OD methods, their collection yields a strong OD approach.

# Model-based Approaches

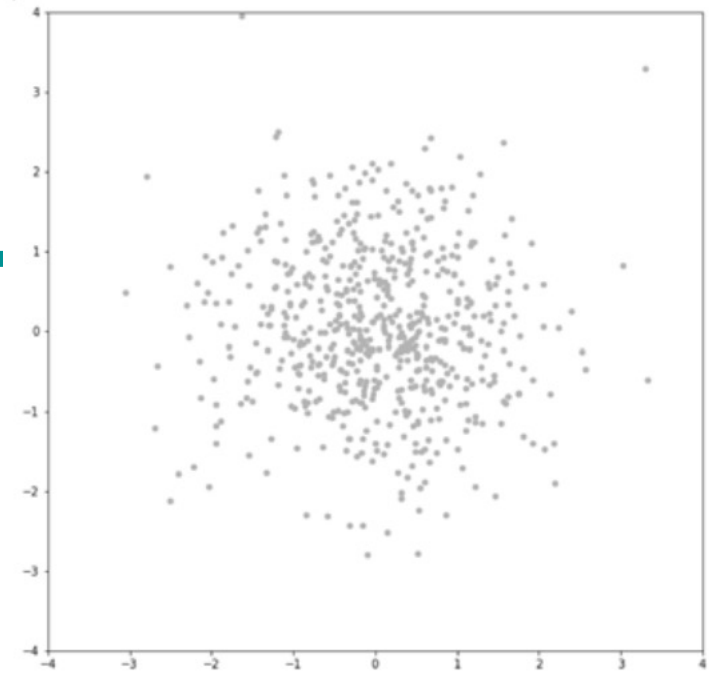
---

Slides revisited from Isolation Forest for Anomaly Detection, Sahand Hariri

# Isolation Forest

---

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.



[https://scikit-learn.org/stable/auto\\_examples/miscellaneous/plot\\_anomaly\\_comparison.html#sphx-glr-auto-examples-miscellaneous-plot-anomaly-comparison-py](https://scikit-learn.org/stable/auto_examples/miscellaneous/plot_anomaly_comparison.html#sphx-glr-auto-examples-miscellaneous-plot-anomaly-comparison-py)

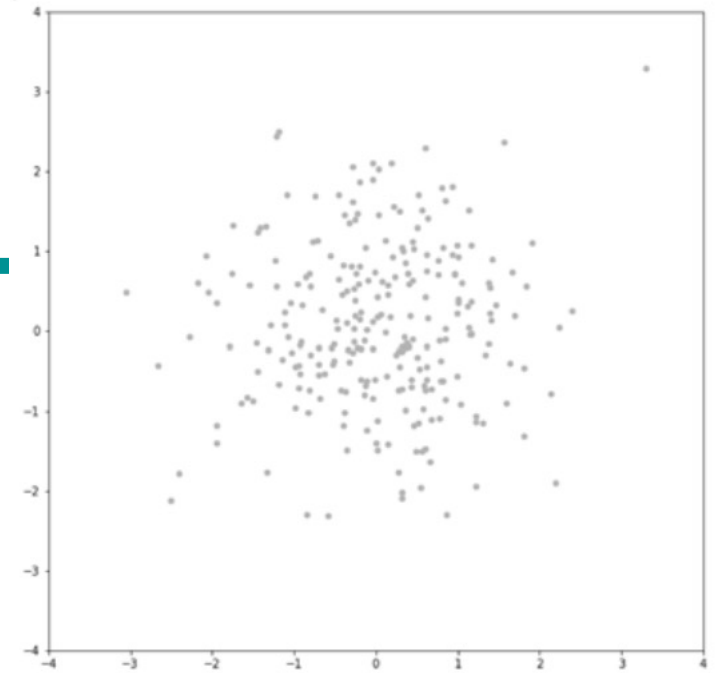
<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html#sklearn.ensemble.IsolationForest>

[https://scikit-learn.org/stable/auto\\_examples/ensemble/plot\\_isolation\\_forest.html](https://scikit-learn.org/stable/auto_examples/ensemble/plot_isolation_forest.html)

# Isolation Forest

---

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data

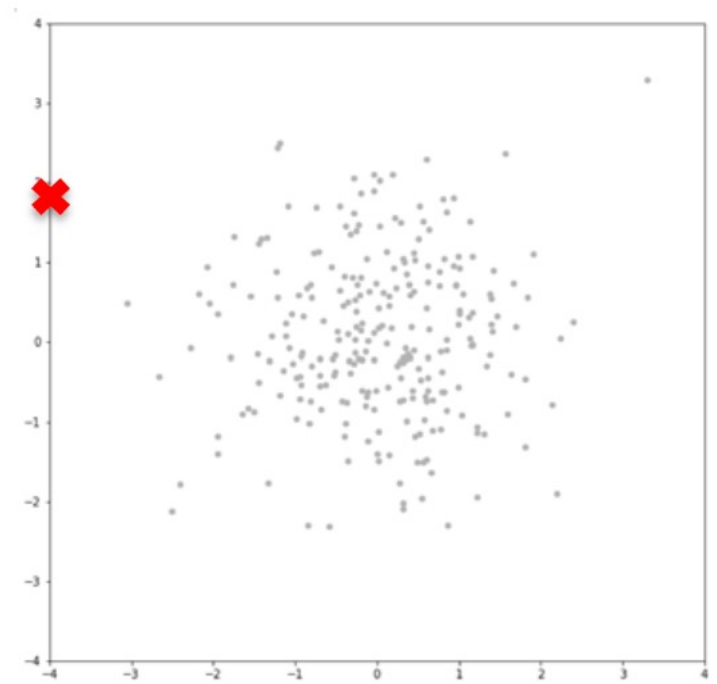


# Isolation Forest

---

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension

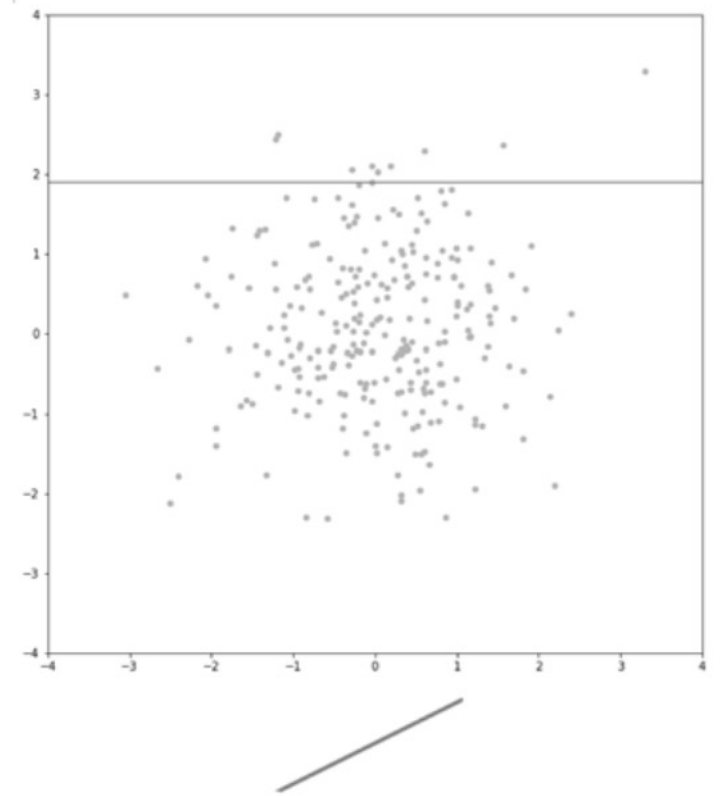
y



# Isolation Forest

---

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data

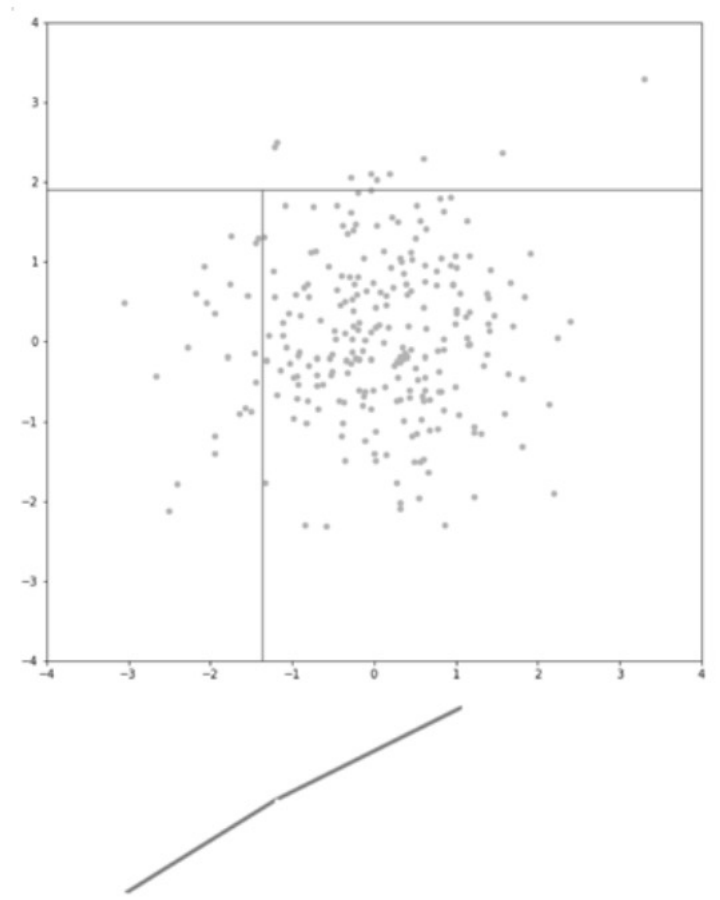




# Isolation Forest

---

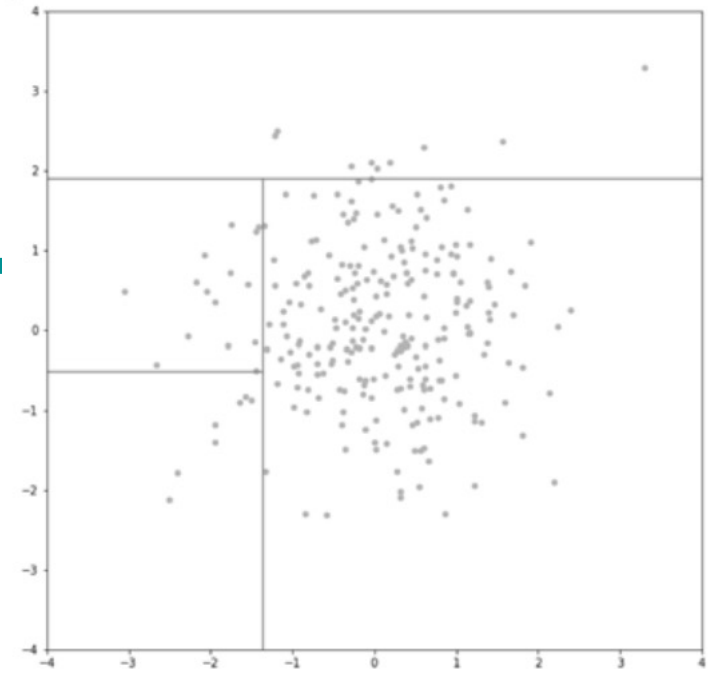
- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete



# Isolation Forest

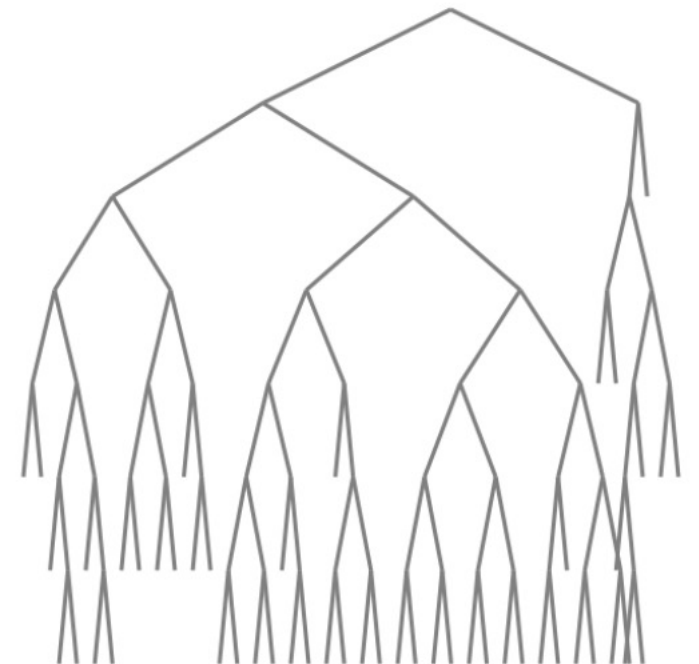
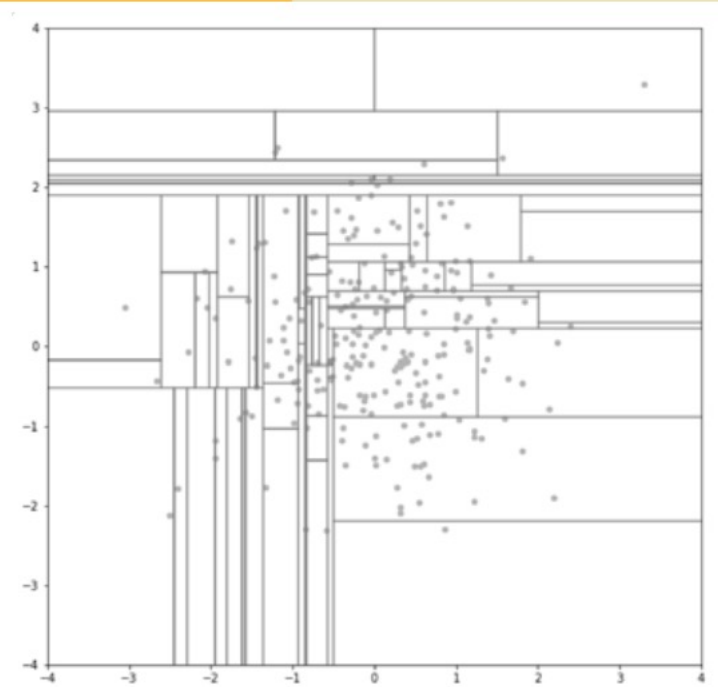
---

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete



# Isolation Forest

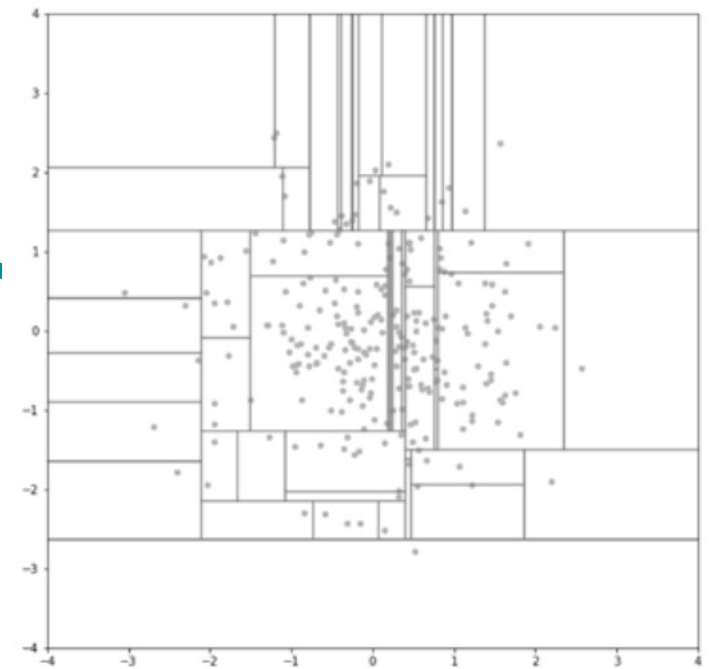
- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete



# Isolation Forest

---

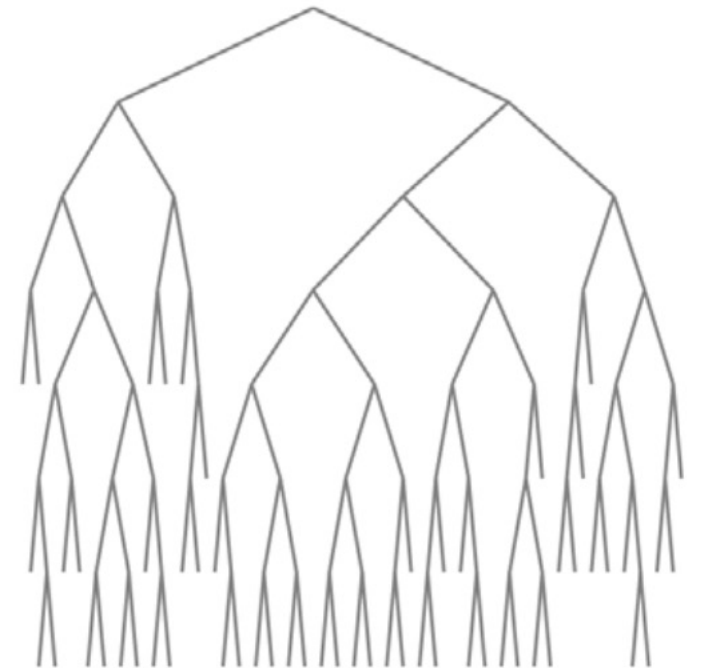
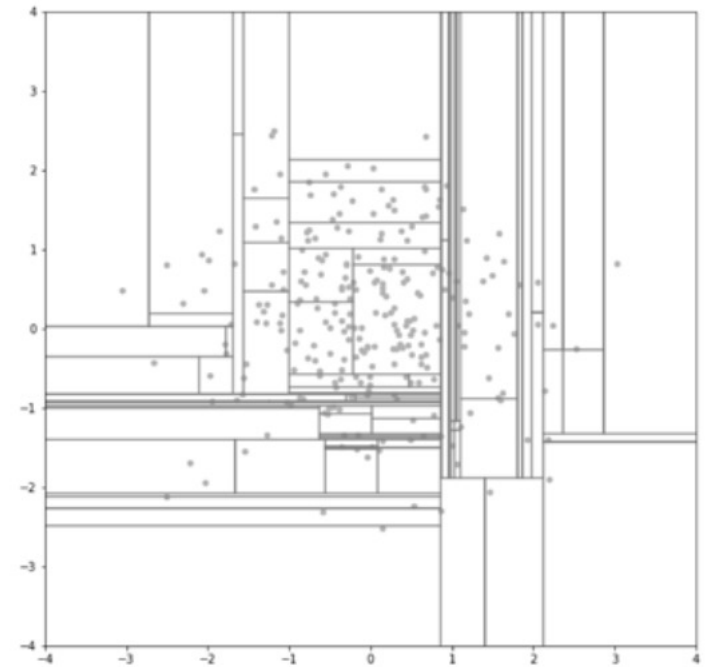
- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees -> forest



# Isolation Forest

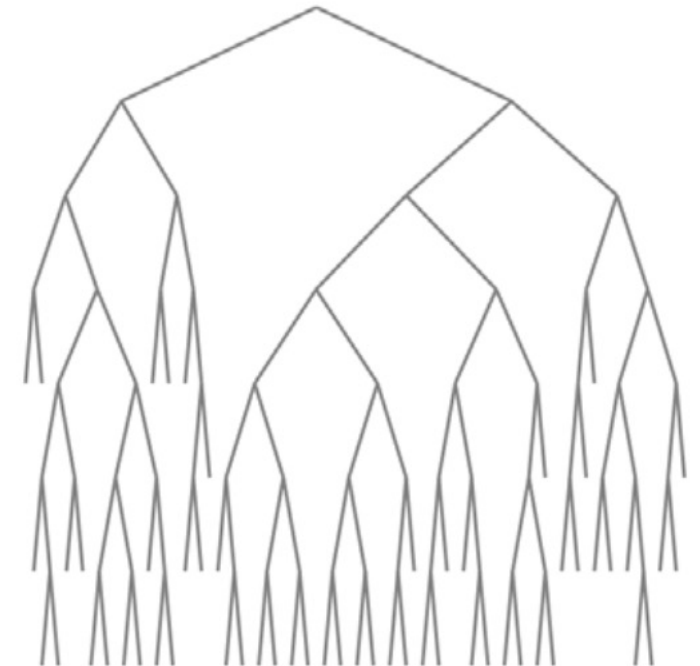
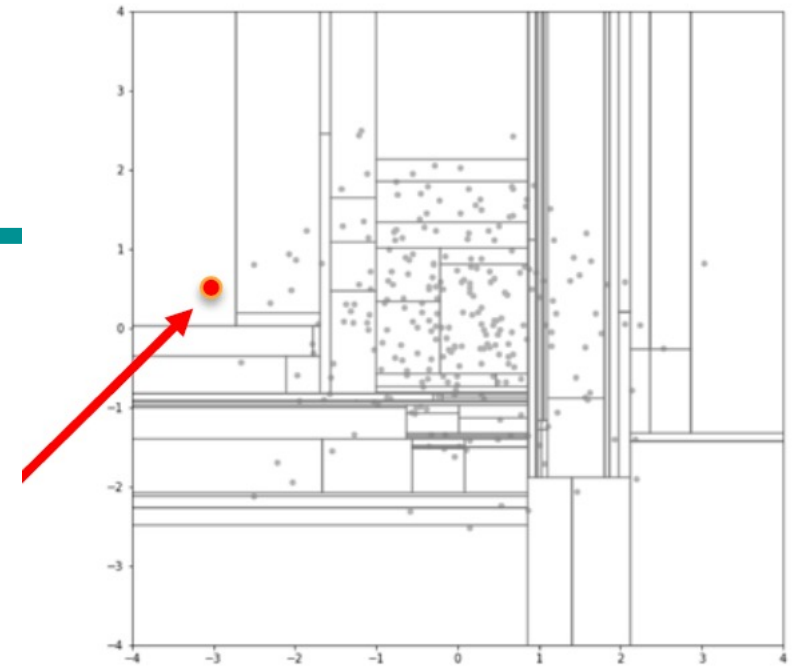
---

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees -> forest



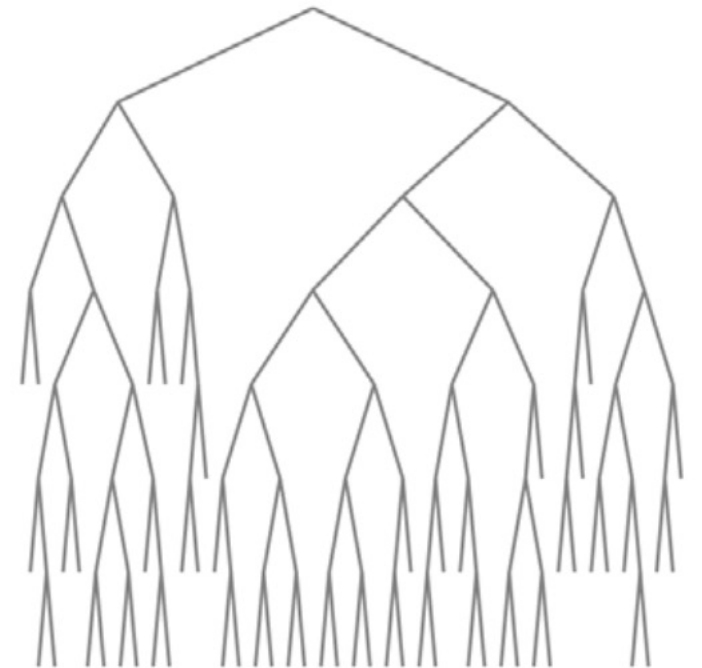
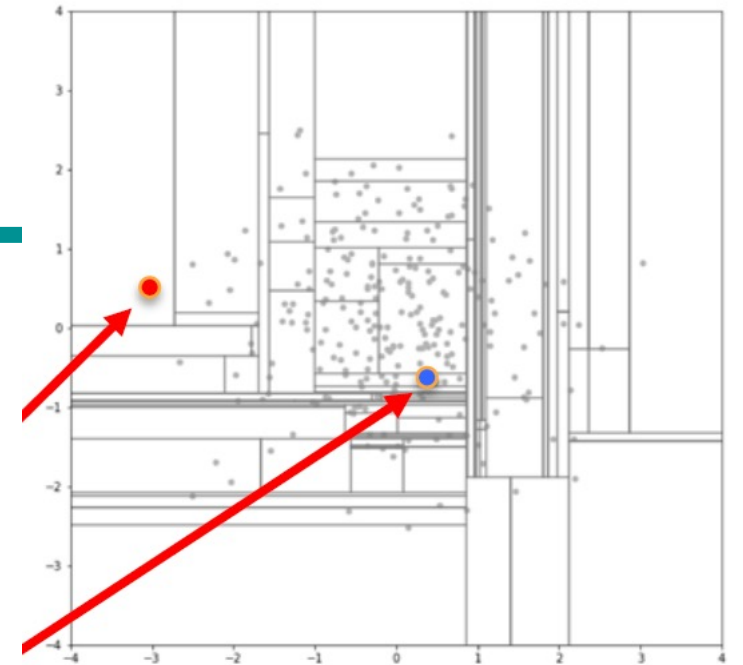
# Isolation Forest

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees -> forest
- Anomalies will be isolated in only few steps



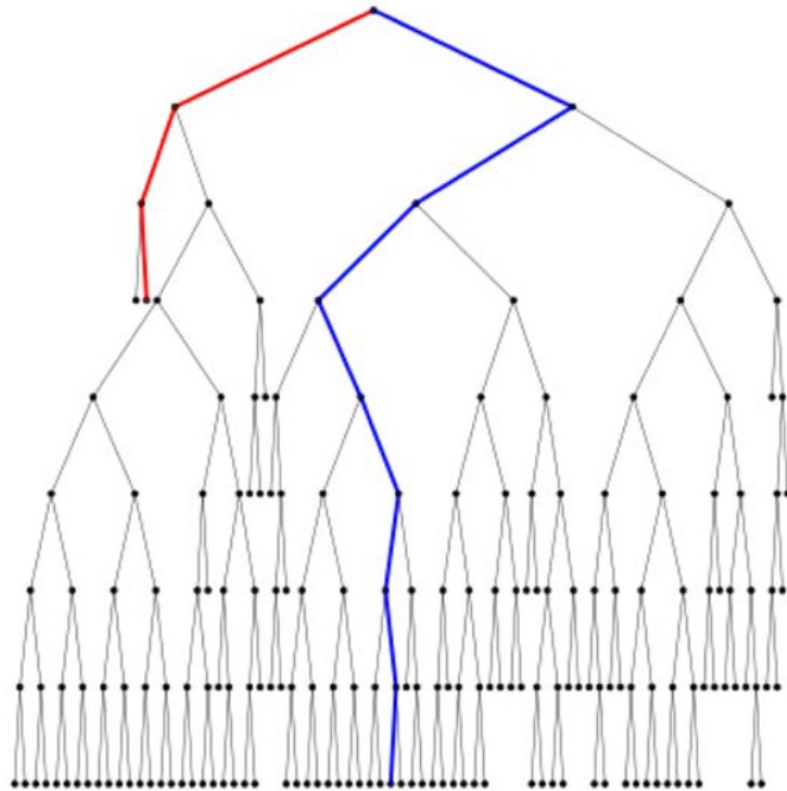
# Isolation Forest

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees -> forest
- Anomalies will be isolated in only few steps
- Nominal points in more

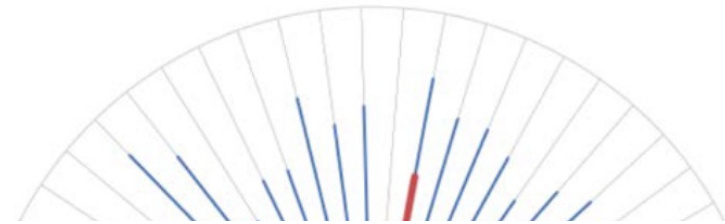


# Isolation Forest

Single Tree scores for  
**anomaly** and **nominal** points



Forest plotted radially.  
Scores for **anomaly** and  
**nominal** shown as lines



$h(x)$  = path length as number of edges from the root to a leaf  
 $E(h(x))$  = average path length (E stands for expectation)  
 $c(m)$  = average  $h(x)$  given  $m$  used to normalize  $h(x)$   
 $H$  = harmonic number estimated as  $H(i) = \ln(i) + \gamma$  with  $\gamma = 0.57$   
 $m$  = size of samples

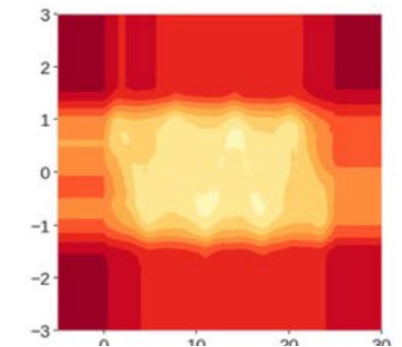
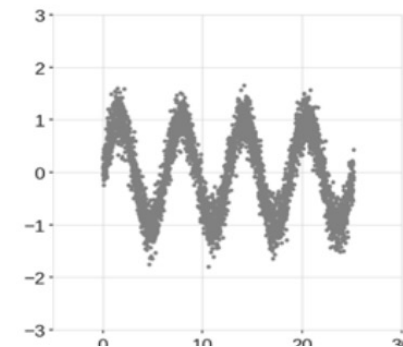
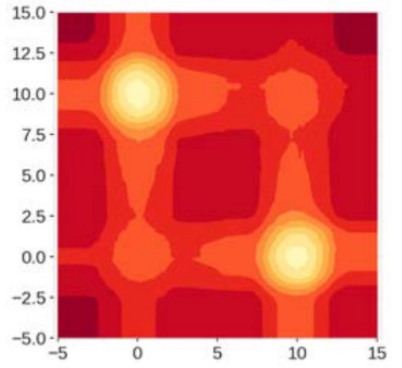
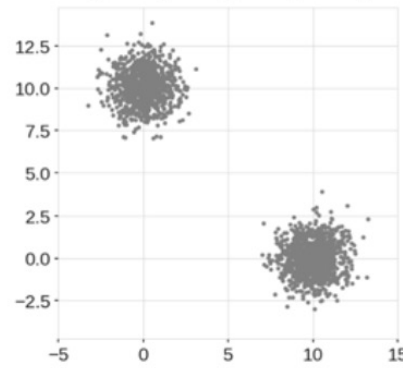
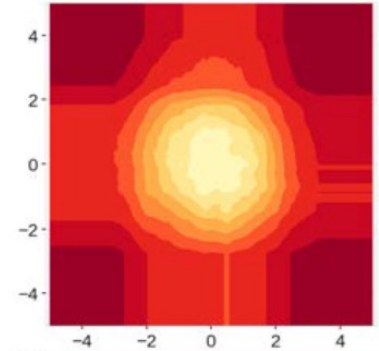
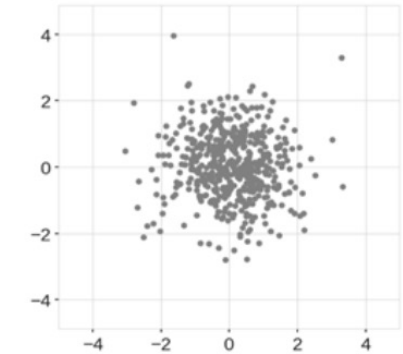
if  $s$  is close to 1 then  $x$  is very likely to be an anomaly  
if  $s$  is smaller than 0.5 then  $x$  is likely to be a normal value

$$s(x, m) = 2 \frac{-E(h(x))}{c(m)} \quad c(m) = \begin{cases} 2H(m-1) - \frac{2^{(m-1)}}{n} & \text{for } m > 2 \\ 1 & \text{for } m = 2 \\ 0 & \text{otherwise} \end{cases}$$



# Anomaly Detection with Isolation Forest

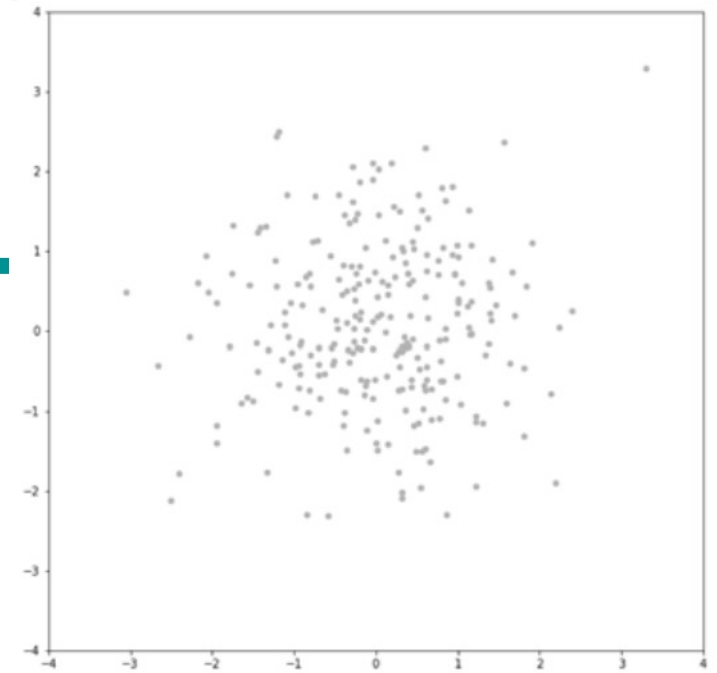
- Isolation Forest
  - Computationally Efficient
  - Parallelizable
  - Handle high dimensional data
  - Inconsistent scoring can be observed



# Extended Isolation Forest

---

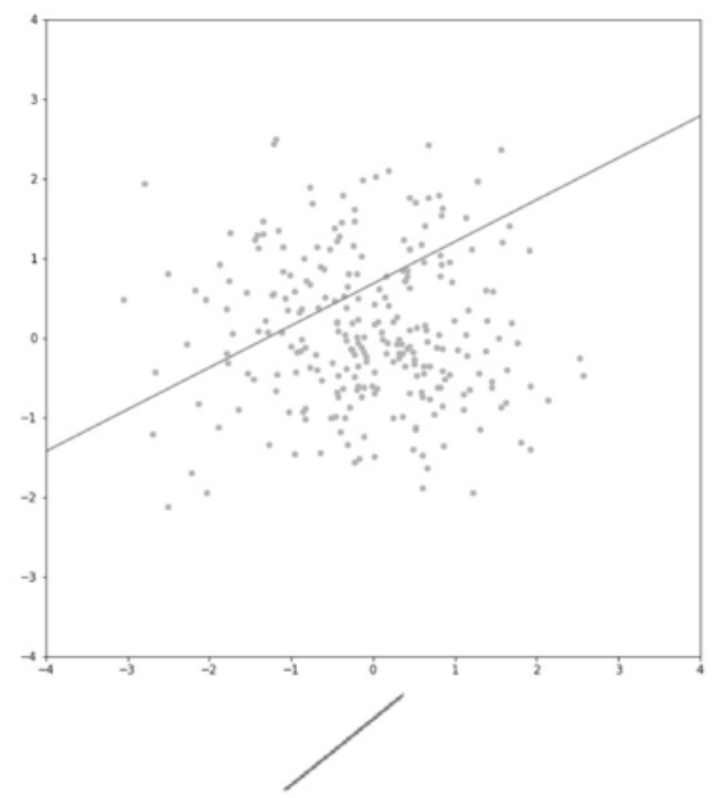
- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a normal vector
  - Randomly select an intercept



# Extended Isolation Forest

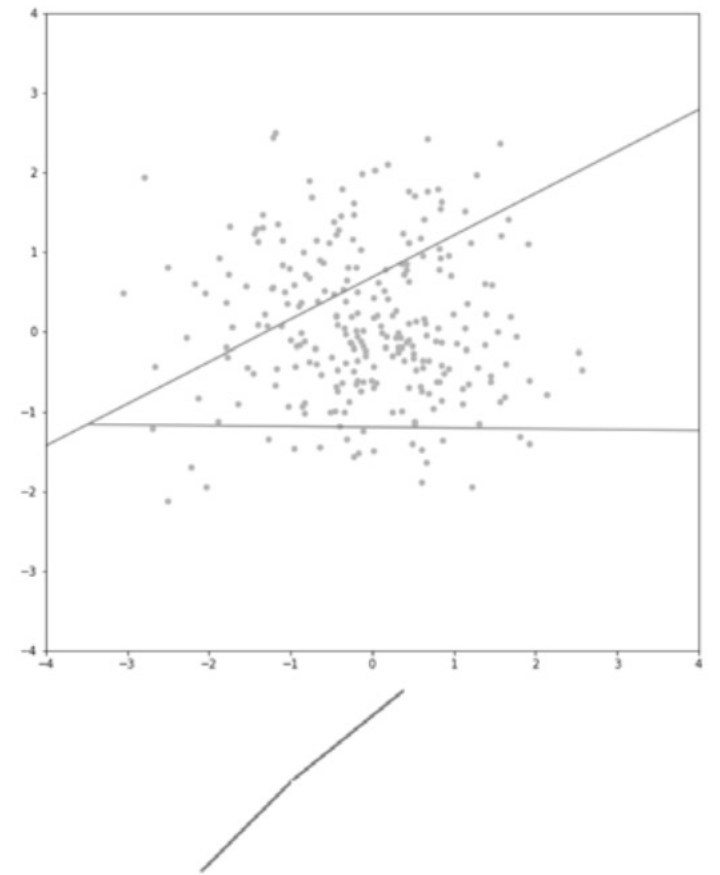
---

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a normal vector
  - Randomly select an intercept
  - Draw a straight line through the data at that value and split data



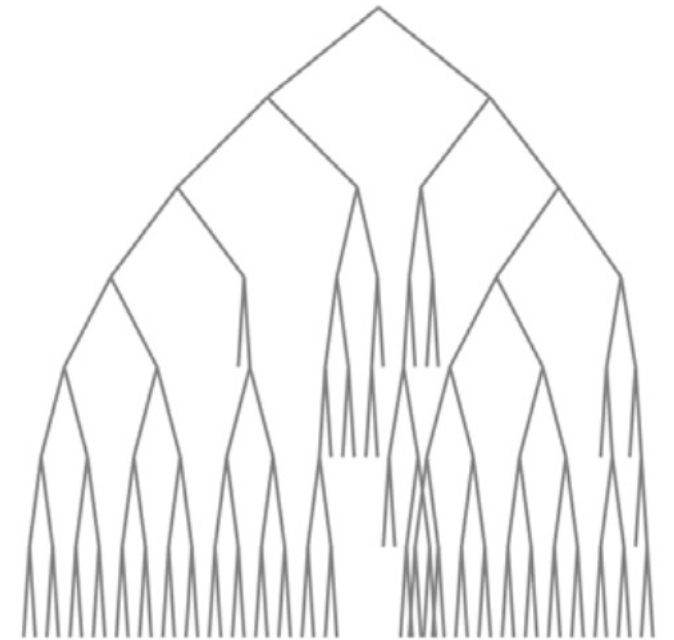
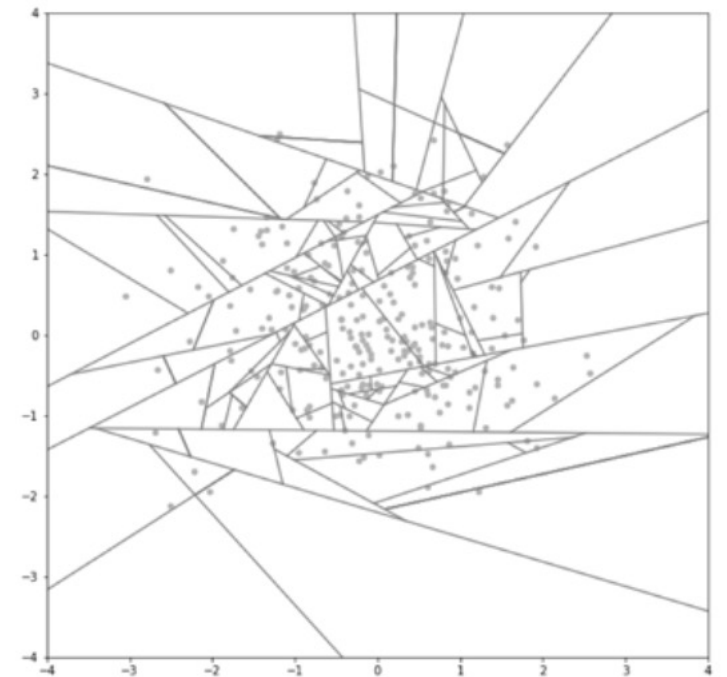
# Extended Isolation Forest

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a normal vector
  - Randomly select an intercept
  - Draw a straight line through the data at that value and split data
  - Repeat until the tree is complete



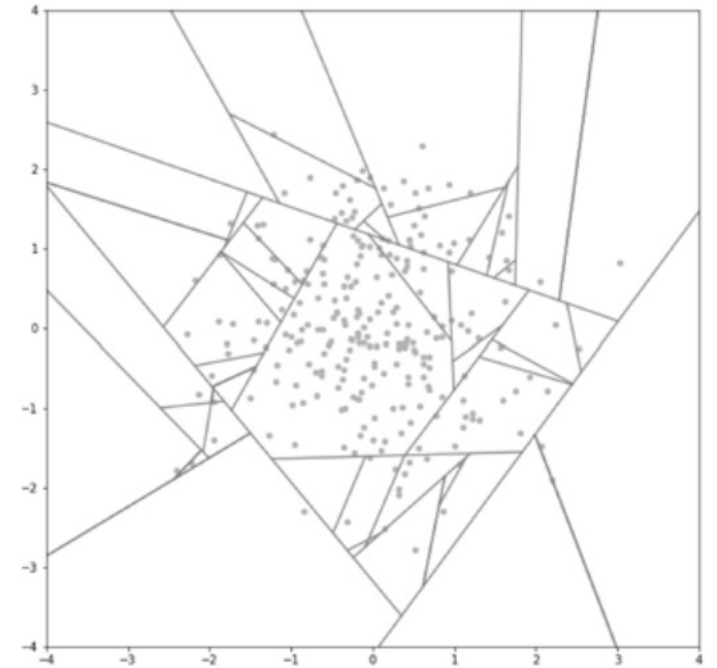
# Extended Isolation Forest

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a normal vector
  - Randomly select an intercept
  - Draw a straight line through the data at that value and split data
  - Repeat until the tree is complete



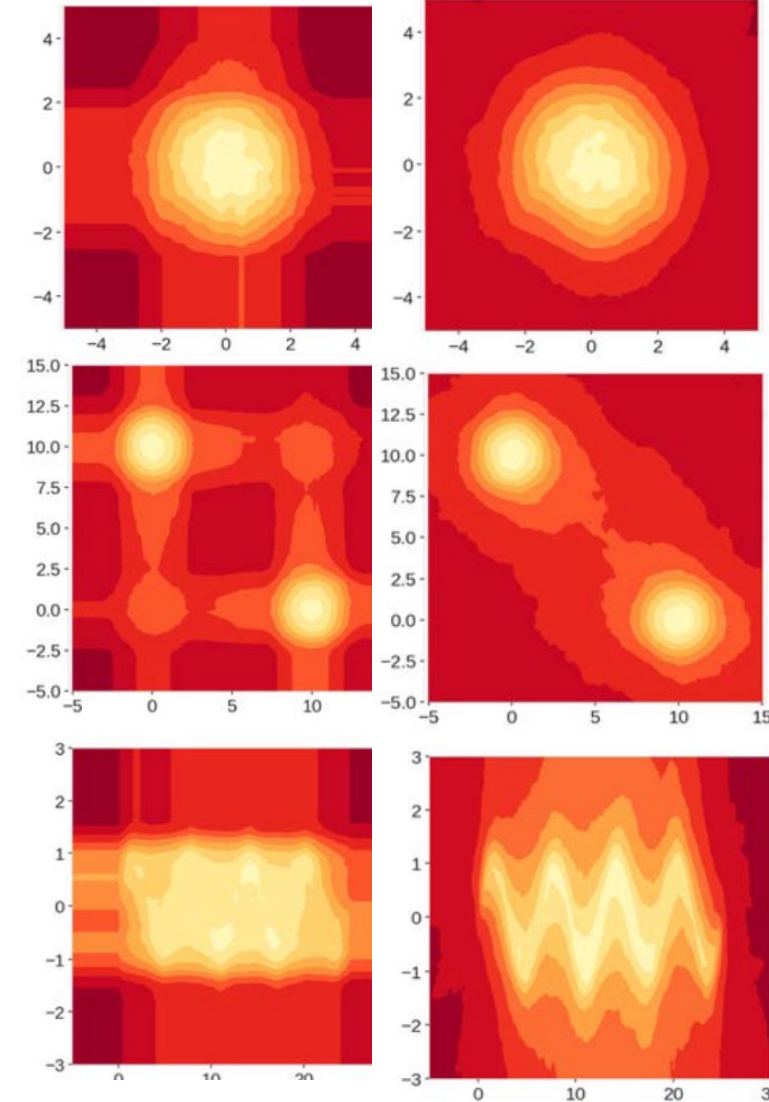
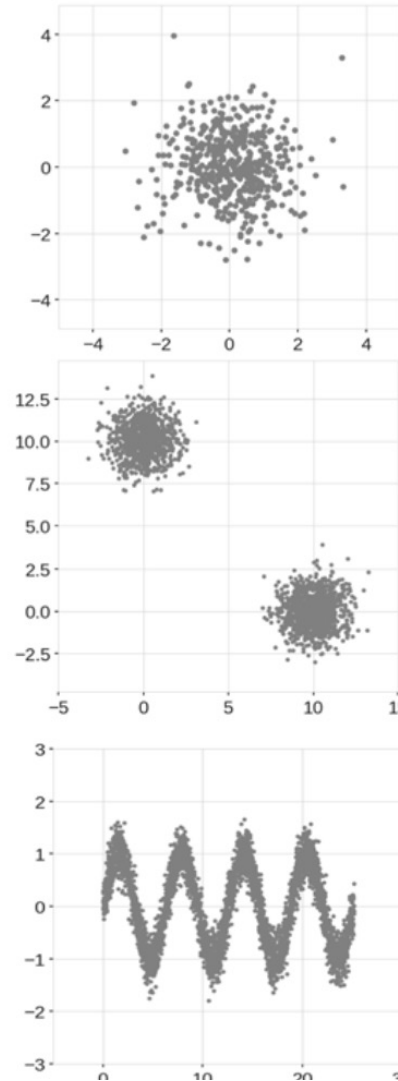
# Extended Isolation Forest

- Idea: Few and different instances can be isolated quicker
- Given the dataset build a forest of trees.
- For each tree:
  - Get a sample of the data
  - Randomly select a normal vector
  - Randomly select an intercept
  - Draw a straight line through the data at that value and split data
  - Repeat until the tree is complete
- Generate multiple trees → forest



# Anomaly Detection with Isolation Forest

- Isolation Forest
  - Computationally Efficient
  - Parallelizable
  - Handle high dimensional data
  - Inconsistent scoring can be observed
- Extended Isolation Forest
  - Computationally Efficient
  - Parallelizable
  - Handle high dimensional data
  - Consistent scoring



# Summary

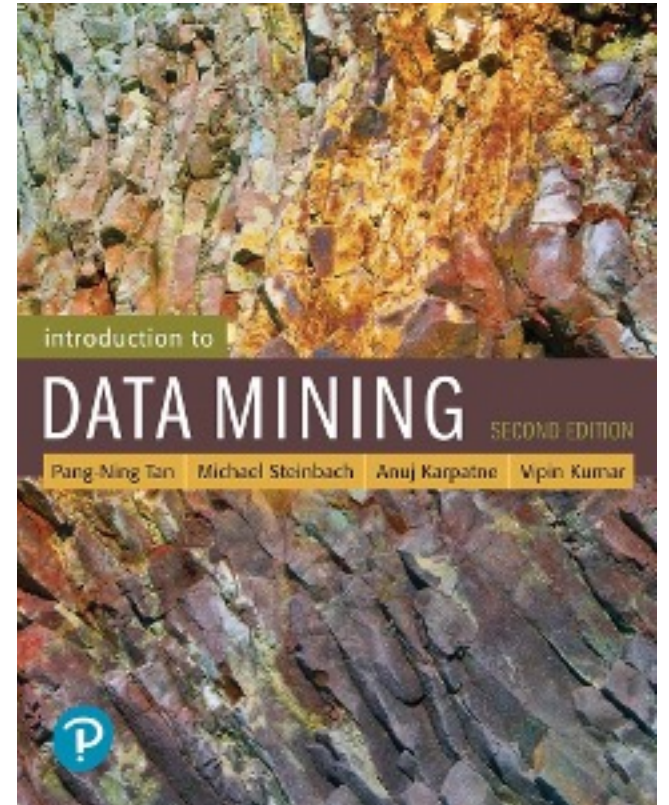
---

- Different models are based on different assumptions
- Different models provide different types of output (labeling/scoring)
- Different models consider outlier at different resolutions (global/local)
- Thus, different models will produce different results
- A thorough and comprehensive comparison between different models and approaches is still missing



# References

- Anomaly Detection. Chapter 10. Introduction to Data Mining.
- Liu, Fei Tony; Ting, Kai Ming; Zhou, Zhi-Hua (December 2008). "Isolation Forest". 2008 Eighth IEEE International Conference on Data Mining: 413–422
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58.



# Exercises – Outlier Detection

---

# Outlier Detection – Exercise 1

Given the dataset of 10 points below, consider the outlier detection problem for points A and B, adopting the following three methods:

a) Distance-based:  $DB(\epsilon, \pi)$  (2 points)

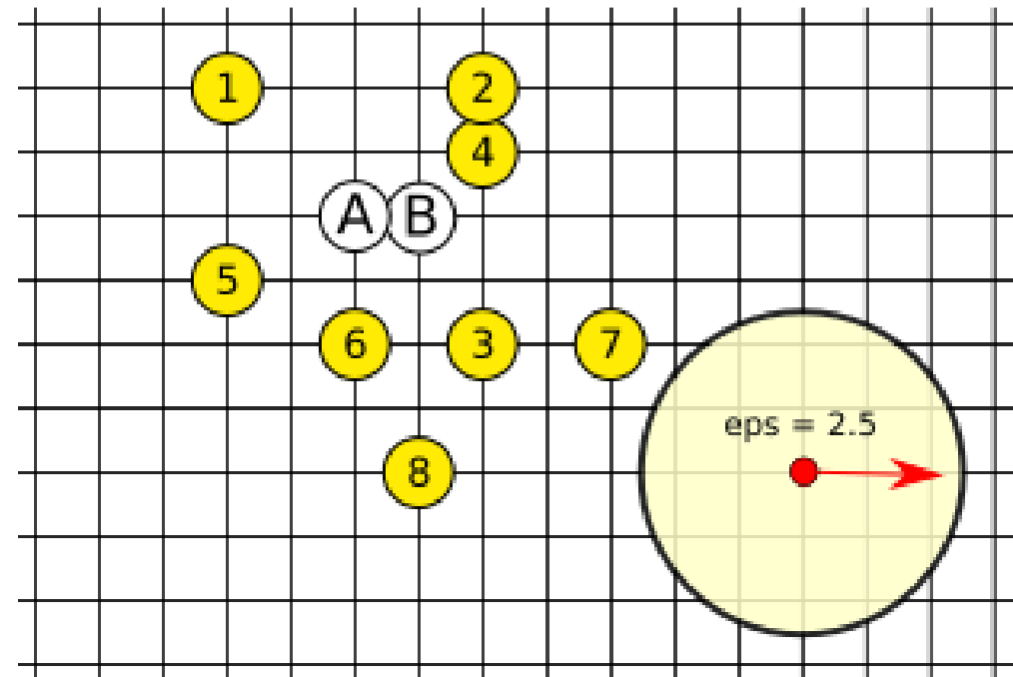
Are A and/or B outliers, if thresholds are forced to  $\epsilon = 2.5$  and  $\pi = 0.15$ ? The point itself should not be counted.

b) Density-based: LOF (2 points)

Compute the LOF score for points A and B by taking  $k=2$ , i.e. comparing each point with its 2 NNs (not counting the point itself). In order to simplify the calculations, the reachability-distance used by LOF can be replaced by the simple Euclidean distance.

c) Depth-based (2 points)

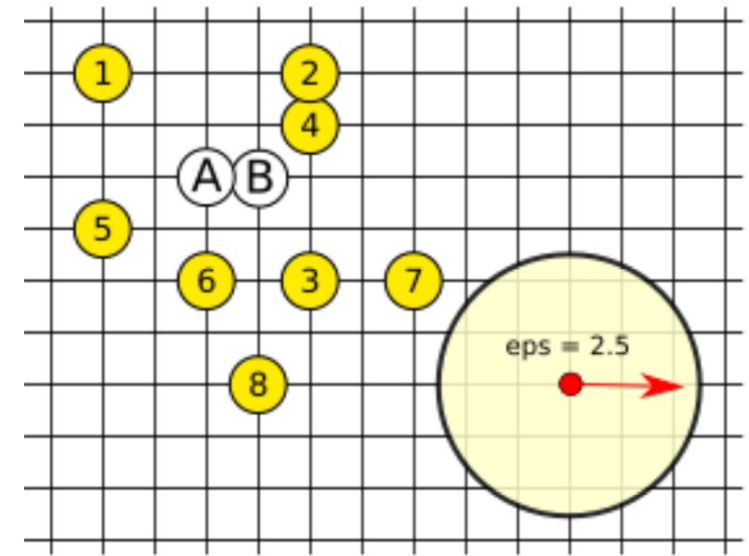
Compute the depth score of all points.



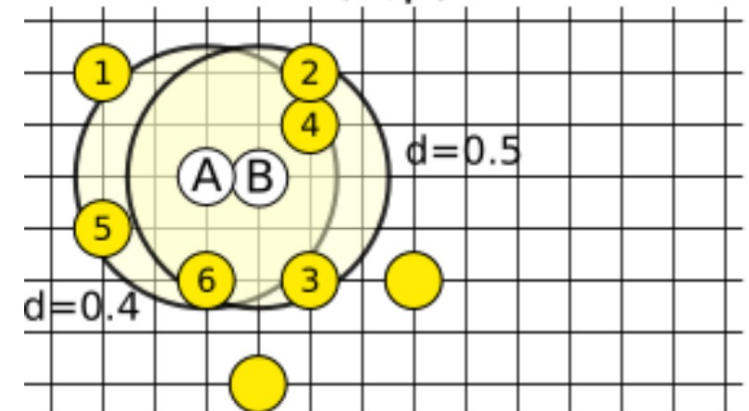
# Outlier Detection – Exercise 1 – Solution

Distance-based

- No outliers because within their radius there are 0.4 and 0.5 points for A and B, respectively



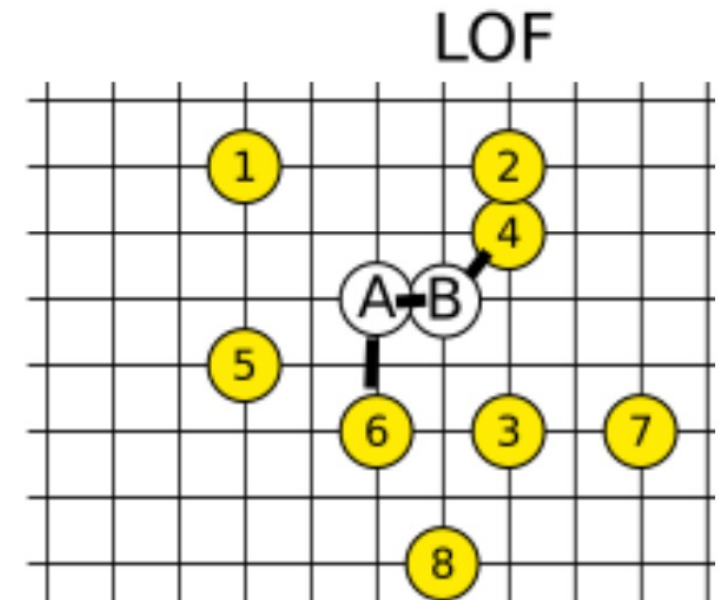
DB(e,p)



# Outlier Detection – Exercise 1 – Solution

Density-based

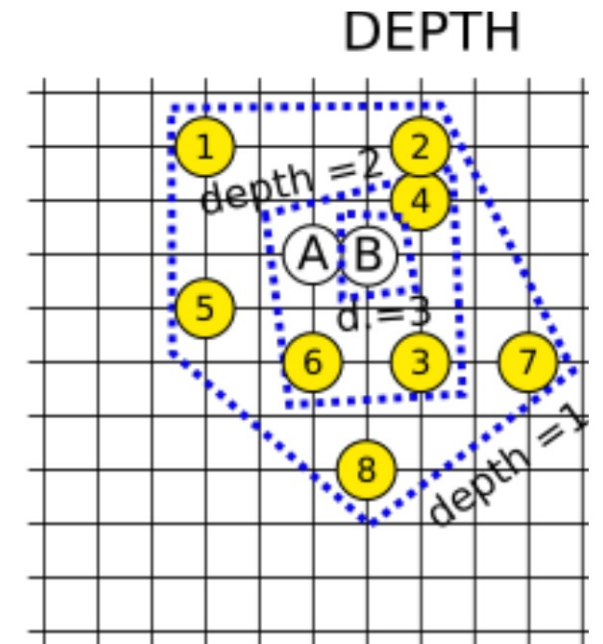
- $LRD(A) = 1 / [ (1 + 2) / 2 ] = 0.666$
- $LRD(B) = 1 / [ (1 + \sqrt{2}) / 2 ] = 0.828$
- $LRD(6) = 1 / [ (2 + 2) / 2 ] = 0.500$
- $LOF(A) = ( [ LRD(B) + LRD(6) ] / 2 ) / LRD(A) = [ (0.828 + 0.500) / 2 ] / 0.666 = 1.003$
- $LRD(4) = 1 / [ (1 + \sqrt{2}) / 2 ] = 0.828$
- $LOF(B) = ( [ LRD(A) + LRD(4) ] / 2 ) / LRD(B) = [ (0.666 + 0.828) / 2 ] / 0.828 = 0.902$
- Both are smaller or very close to 1, so they are most likely no outliers.



# Outlier Detection – Exercise 1 – Solution

Depth-based

- A is an outlier for depth = 2
- For depth  $\leq 1$  neither A or B are outliers



# Outlier Detection – Exercise 2

Given the dataset of 10 points below, consider the outlier detection problem for points A and B, adopting the following three methods:

a) Distance-based:  $DB(\epsilon, \pi)$  (2 points)

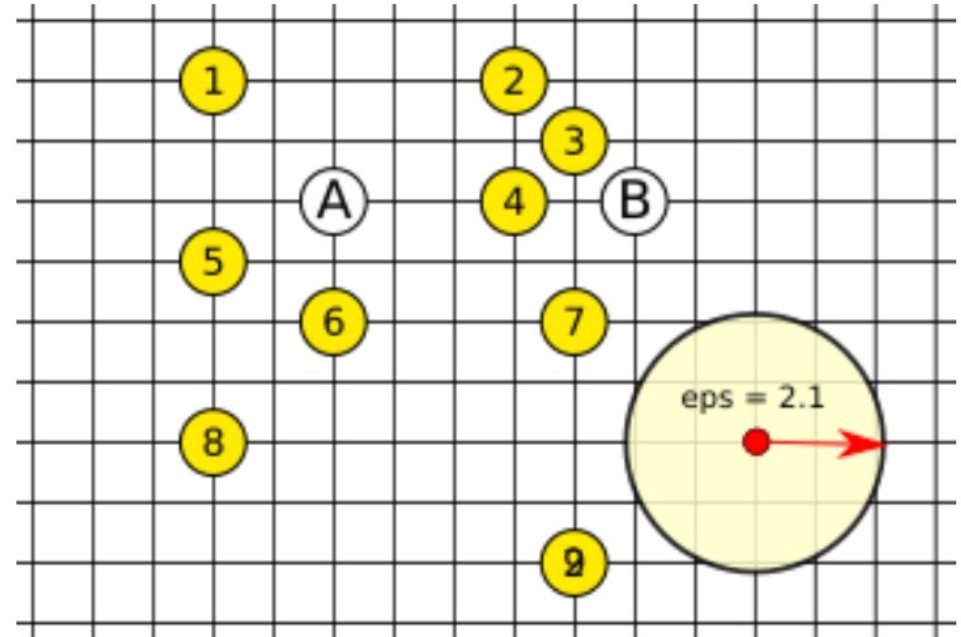
Are A and/or B outliers, if thresholds are forced to  $\epsilon = 2.1$  and  $\pi = 0.15$ ? The point itself should not be counted.

b) Density-based: LOF (2 points)

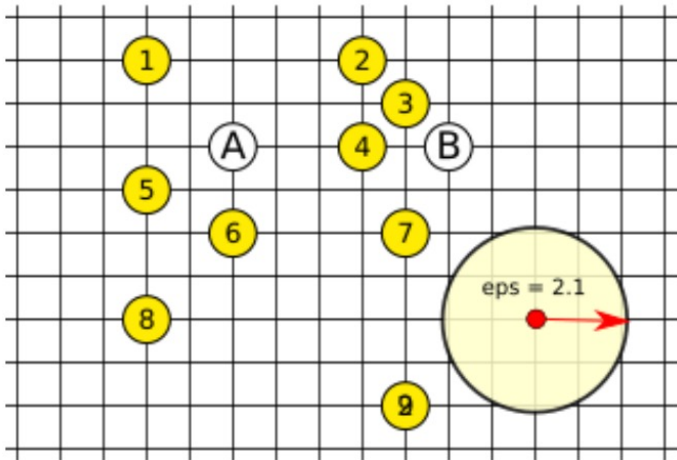
Compute the LOF score for points A and B by taking  $k=2$ , i.e. comparing each point with its 2 NNs (not counting the point itself). In order to simplify the calculations, the reachability-distance used by LOF can be replaced by the simple Euclidean distance.

c) Depth-based (2 points)

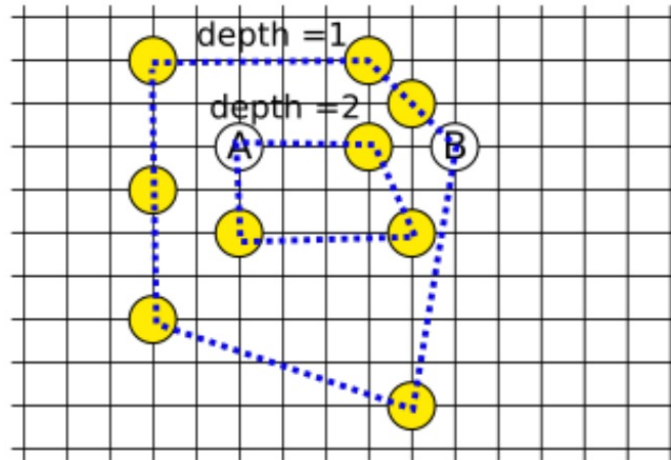
Compute the depth score of all points. Are A and/or B outliers of depth 1?



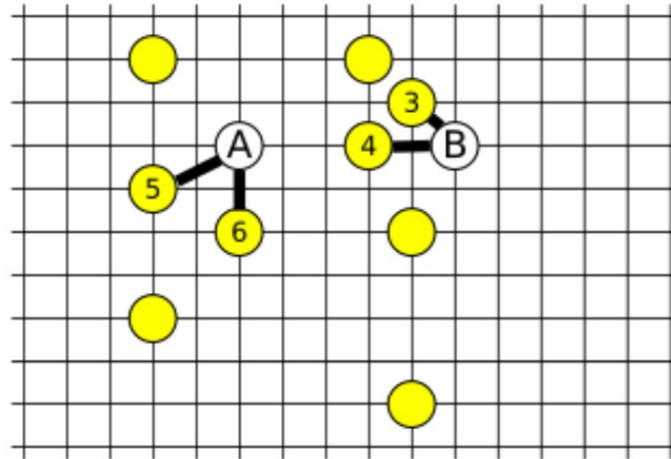
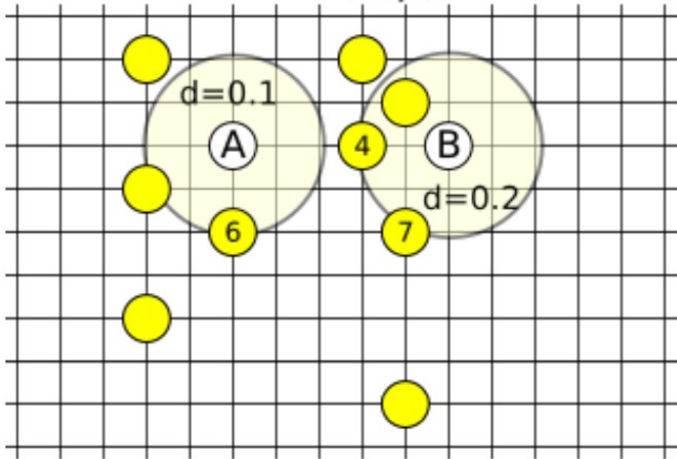
# Outlier Detection – Exercise 2 – Solution



DB(e,p)



LOF



$$\text{LRD}(A) = 1 / [ (2 + \sqrt{5}) / 2 ] = 0.472$$

$$\text{LRD}(5) = 1 / [ (\sqrt{5} + \sqrt{5}) / 2 ] = 0.447$$

$$\text{LRD}(6) = 1 / [ (2 + \sqrt{5}) / 2 ] = 0.472$$

$$\begin{aligned} \text{LOF}(A) &= ( [ \text{LRD}(5) + \text{LRD}(6) ] / 2 ) / \text{LRD}(A) \\ &= [ (0.472 + 0.447) / 2 ] / 0.472 = 0.973 \end{aligned}$$

$$\text{LRD}(B) = 1 / [ (2 + \sqrt{2}) / 2 ] = 0.586$$

$$\text{LRD}(3) = 1 / [ (\sqrt{2} + \sqrt{2} + \sqrt{2}) / 3 ] = 0.707$$

$$\text{LRD}(4) = 1 / [ (2 + 2 + \sqrt{2}) / 3 ] = 0.554$$

$$\begin{aligned} \text{LOF}(B) &= ( [ \text{LRD}(3) + \text{LRD}(4) ] / 2 ) / \text{LRD}(B) \\ &= [ (0.707 + 0.554) / 2 ] / 0.586 = 0.929 \end{aligned}$$



# Outlier Detection – Exercise 3

Given the dataset of 10 points below (A, B, 1, 2, ..., 8), consider the outlier detection problem for points A and B, adopting the following three methods:

**a) Distance-based: DB( $\epsilon, \pi$ ) (2 points)**

Are A and/or B outliers, if thresholds are forced to  $\epsilon = 2.5$  and  $\pi = 0.3$ ? Show the density of the two points. (Notice: in computing the density of a point P, P itself should not be counted as neighbour).

**b) Density-based: LOF (3 points)**

Compute the LOF score for points A and B by taking  $k=2$ , i.e. comparing each point with its 2-NNs (not counting the point itself). In order to simplify the calculations, the reachability-distance used by LOF can be replaced by the simple Euclidean distance.

**c) Depth-based (1 points)**

Compute the depth score of all points.

