Kuliah 5
Deteksi Anomali

Referensi :

1. Tan P., Michael S., & Vipin K. 2006. *Introduction to Data mining*. Pearson Education, Inc. – Chapter 10
2. Han J & Kamber M. 2006. *Data mining – Concept and Techniques*. 2nd Edition, Morgan-Kauffman, San Diego - Chapter 7

Anomaly/Outlier Detection

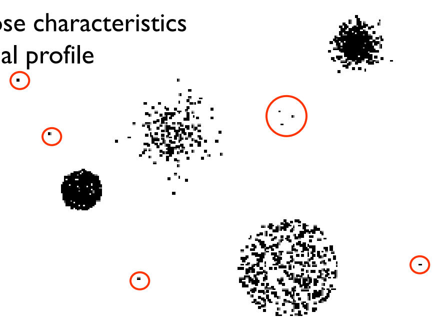
- What are anomalies/outliers?
  - The set of data points that are considerably different (Tan) / dissimilar (Han) than the remainder of the data
- Variants of Anomaly/Outlier Detection Problems
  - Given a database D, find all the data points **x** ∈ D with anomaly scores greater than some threshold t
  - Given a database D, find all the data points **x** ∈ D having the top-n largest anomaly scores f(**x**)
  - Given a database D, containing mostly normal (but unlabeled) data points, and a test point **x**, compute the anomaly score of **x** with respect to D
- Applications:
  - Credit card fraud detection, telecommunication fraud detection, network intrusion detection, fault detection

Anomaly Detection

- Challenges
  - How many outliers are there in the data?
  - Method is unsupervised
    - Validation can be quite challenging (just like for clustering)
  - Finding needle in a haystack
- Working assumption:
  - There are considerably more "normal" observations than "abnormal" observations (outliers/anomalies) in the data

Anomaly Detection Schemes

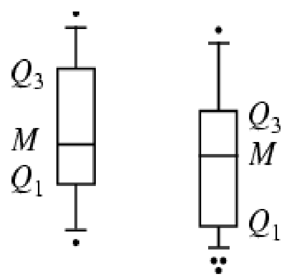
- General Steps
  - Build a profile of the "normal" behavior
    - Profile can be patterns or summary statistics for the overall population
  - Use the "normal" profile to detect anomalies
    - Anomalies are observations whose characteristics differ significantly from the normal profile
- Types of anomaly detection schemes
  - Graphical & Statistical-based
  - Distance-based
  - Model-based

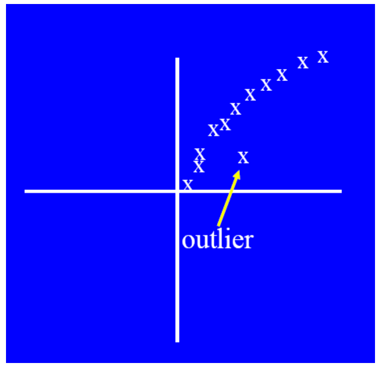


Graphical Approaches

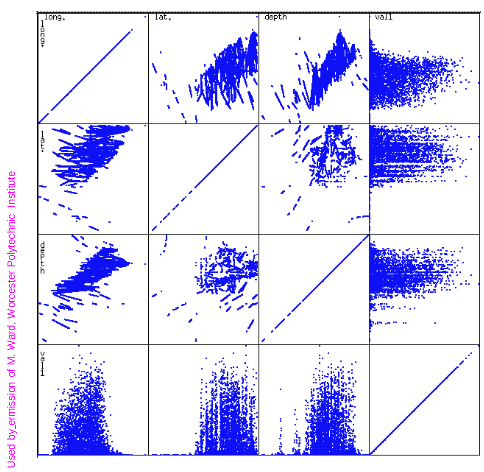
- Boxplot (1-D), Scatter plot (2-D), Spin plot (3-D)

- Limitations
  - Time consuming
  - Subjective





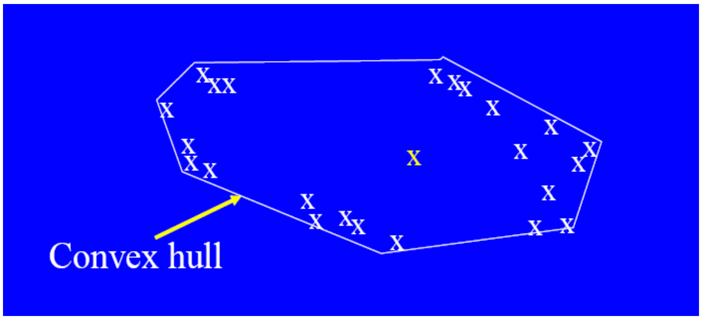
Scatterplot-Matrices [Cleveland 93]



matrix of scatterplots (x-y-diagrams) of the k-dimensional data [total of (k2/2-k) scatterplots]

Convex Hull Method

- Extreme points are assumed to be outliers
- Use convex hull method to detect extreme values

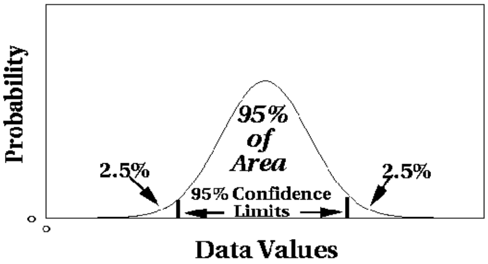


- What if the outlier occurs in the middle of the data?

Statistical Approaches

- Assume a parametric model describing the distribution of the data (e.g., normal distribution)

- Apply a statistical test that depends on
  - Data distribution
  - Parameter of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)



Distance-based Approaches

- 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

Distance-based Approaches

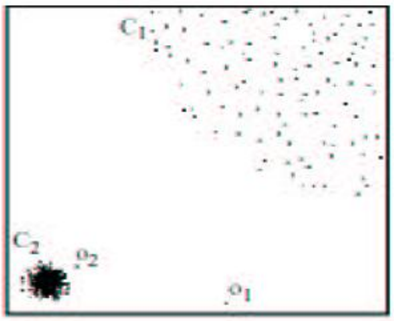
- Data is represented as a vector of features
- Three major approaches
  - Nearest-neighbor based
  - Density based
  - Clustering based

Nearest-Neighbor Based Approach

- Approach:
  - Compute the distance between every pair of data points
  - There are various ways to define outliers:
    - Data points for which there are fewer than *p* neighboring points within a distance *D*
    - The top n data points whose distance to the kth nearest neighbor is greatest
    - The top n data points whose average distance to the k nearest neighbors is greatest

Density-Based Local Outlier Detection

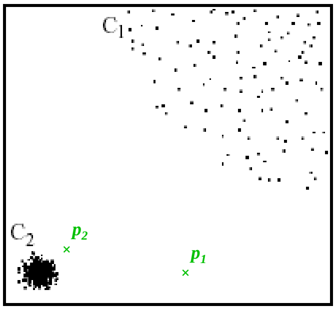
- Distance-based outlier detection is based on global distance distribution
- It encounters difficulties to identify outliers if data is not uniformly distributed
- Ex. C_1 contains 400 loosely distributed points, C_2 has 100 tightly condensed points, 2 outlier points o_1, o_2
- Distance-based method 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

Density-based: LOF approach

- 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

LOF Example

- Consider the following 4 data points:

  a(0,0), b(0,1), c(1,1), d(3,0)

- Calculate the LOF for each point and show the top 1 outlier, set k = 2 and use Manhattan Distance.

Step by Step LOF

Points: a(0,0), b(0,1), c(1,1), d(3,0)

- Step 1: Calculate distance

	a	b	c	d
a	-	1	2	3
b		-	1	4
c			-	3
d				-

- Step 2: Callculate $dist_2$ (o)
  - $dist_2$ (a) = dist(a,c) = 2 (c is the 2nd nearest neighbor)
  - $dist_2$ (b) = dist(b,a) = 1 (a/c is the 2nd nearest neighbor)
  - $dist_2$ (c) = dist(c,a) = 2 (a is the 2nd nearest neighbor)
  - $dist_2$ (d) = dist(d,a) = 3 (a/c is the 2nd nearest neighbor)

Step by Step LOF

- Step 3: Calculate $N_k(o)$

  $N_k(o) = \{o' \mid o' \text{ in } D, dist(o, o') \le dist_k(o)\}$

  - $N_2(a) = \{b,c\}$
  - $N_2(b) = \{a,c\}$
  - $N_2(c) = \{b,a\}$
  - $N_2(d) = \{a,c\}$

- Step 4: Callculate lrd_k (o): Local Reachability Density of o

$$lrd_k(o) = \frac{\|N_k(o)\|}{\sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)} \qquad reachdist_k(o \leftarrow o') = \max\{dist_k(o), dist(o, o')\}$$

$$lrd_k(a) = \frac{\| N_2(a) \|}{reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a)}$$

Step by Step LOF

- $\text{lrd}_k(a) = \frac{\| N_2(a) \|}{reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a)}$
- $reachdist_2(b \leftarrow a) = max\{dist_2(b), dist(b, a)\}$

  $= \max\{1, 1\} = 1$
- $reachdist_2(c \leftarrow a) = max\{dist_2(c), dist(c, a)\}$

  $= \max\{2, 2\} = 2$
- Thus, $\text{lrd}_2(a) = \frac{\| N_2(a) \|}{reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a)} = 2/(1+2) = 0.667$
- Similarly.. $\text{lrd}_2(b) = \frac{\| N_2(b) \|}{reachdist_2(a \leftarrow b) + reachdist_2(c \leftarrow b)} = 2/(2+2) = 0.5$

  $\text{lrd}_2(c) = \frac{\| N_2(c) \|}{reachdist_2(b \leftarrow c) + reachdist_2(a \leftarrow c)} = 2/(1+2) = 0.667$

  $\text{lrd}_2(d) = \frac{\| N_2(b) \|}{reachdist_2(a \leftarrow d) + reachdist_2(c \leftarrow d)} = 2/(3+3) = 0.33$

Step by Step LOF

- Step 5: Calculate $\text{LOF}_k(o)$

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

$\text{LOF}_2(a) = (\text{lrd}_2(b) + \text{lrd}_2(c)) * (reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a))$

$= (0.5+0.667) * (1+2) = 3.501$

$\text{LOF}_2(b) = (\text{lrd}_2(a) + \text{lrd}_2(c)) * (reachdist_2(a \leftarrow b) + reachdist_2(c \leftarrow b))$

$= (0.667+0.667) * (2+2) = 5.336$

$\text{LOF}_2(c) = (\text{lrd}_2(b) + \text{lrd}_2(a)) * (reachdist_2(b \leftarrow c) + reachdist_2(a \leftarrow c))$

$= (0.5+0.667) * (1+2) = 3.501$

$\text{LOF}_2(d) = (\text{lrd}_2(a) + \text{lrd}_2(c)) * (reachdist_2(a \leftarrow d) + reachdist_2(c \leftarrow d))$

$= (0.667+0.667) * (3+3) = 8.004$

Step by Step LOF

- Step 6: Sort all the $LOF_k(o)$
  - $LOF_2(d) = 8.004$
  - $LOF_2(b) = 5.336$
  - $LOF_2(a) = 3.501$
  - $LOF_2(c) = 3.501$

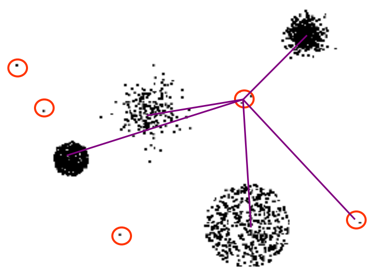
- Obviously, top 1 outlier is point d

Outlier Discovery: Deviation-Based Approach

- Identifies outliers by examining the main characteristics of objects in a group
- Objects that "deviate" from this description are considered outliers
- Sequential exception technique
  - simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects
- OLAP data cube technique
  - uses data cubes to identify regions of anomalies in large multidimensional data

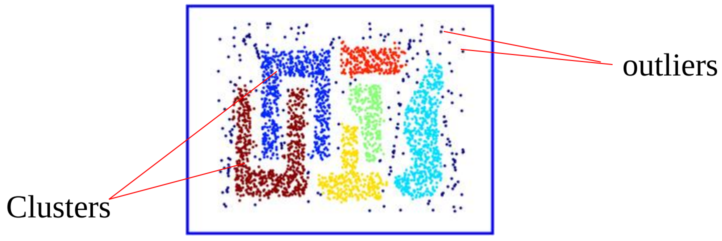
Clustering-Based

- Basic idea:
  - Cluster the data into groups of different density
  - Choose points in small cluster as candidate outliers
  - Compute the distance between candidate points and non-candidate clusters.
    - If candidate points are far from all other non-candidate points, they are outliers



DBSCAN

- Density-based spatial clustering of application with noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996.
- It groups together points that are closely packed together (points with many nearby neighbors)
- Marking as outliers points that lie alone in low-density regions.

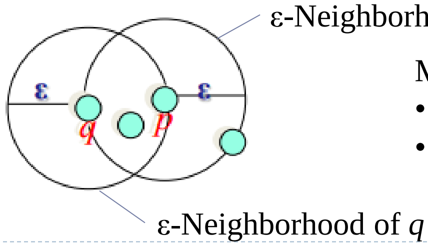


Density Definition

- ε-Neighborhood – Objects within a radius of ε from an object.

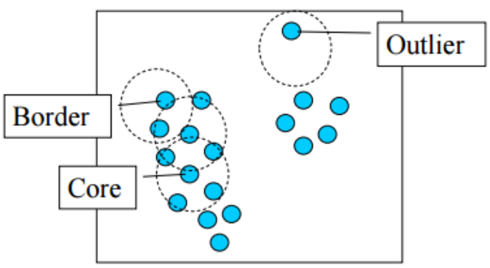
$$N_{\varepsilon}(p):\{q \mid d(p,q) \leq \varepsilon\}$$

- "High density" - ε-Neighborhood of an object contains at least **MinPts** of objects.



MinPts=4
- Density of *p* is "*high*"
- Density of *q* is "*low*"

Core, Border, & Outlier



ε = 1 unit, MinPts =5

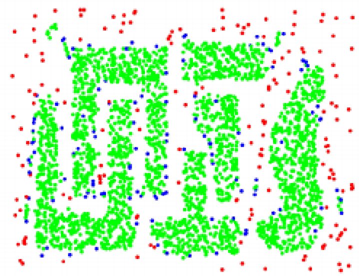
Given ε and MinPts, categorize the objects into three exclusive groups.

A point is a core point if it has more than a specified number of points (MinPts) within Eps—These are points that are at the interior of a cluster.

A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.

A noise point is any point that is not a core point nor a border point.

Core, Border, & Outlier - Example



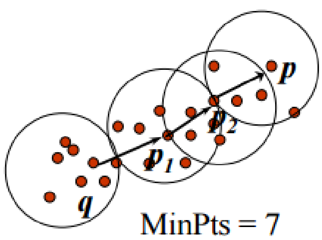
Original Points

Point types: core, border and outliers

ε = 10, MinPts = 4

Density-reachability

- Density-Reachable (directly and indirectly):
  - A point p is directly density-reachable from p_2
  - p_2 is directly density-reachable from p_1
  - p_1 is directly density-reachable from q
  - $p \leftarrow p_2 \leftarrow p_1 \leftarrow q$ form a chain



- p is (indirectly) density-reachable from q
- q is not density-reachable from p

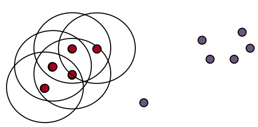
DBSCAN Algorithm

- Arbitrary select a point *p*
- Retrieve all points density-reachable from *p* wrt *Eps* and *MinPts*.
- If *p* is a core point, a cluster is formed.
- If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

DBSCAN, example *)

- Parameter
  - ε = 2 cm
  - *MinPts* = 3

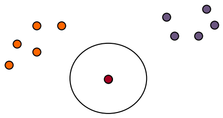


for each $o \in D$ **do**
    if *o* is not yet classified **then**
        if *o* is a core-object **then**
            collect all objects density-reachable from *o*
            and assign them to a new cluster.
        else
            assign *o* to NOISE

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

DBSCAN, example *)

- Parameter
  - ε = 2 cm
  - *MinPts* = 3

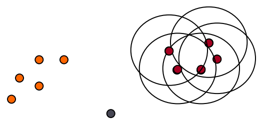


for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

DBSCAN, example *)

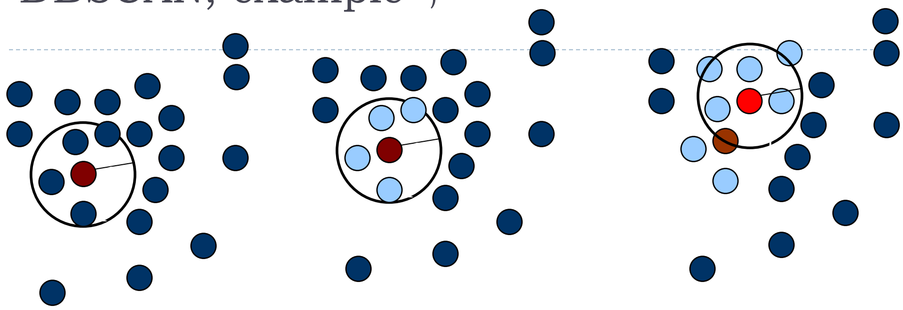
- Parameter
  - ε = 2 cm
  - *MinPts* = 3



for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

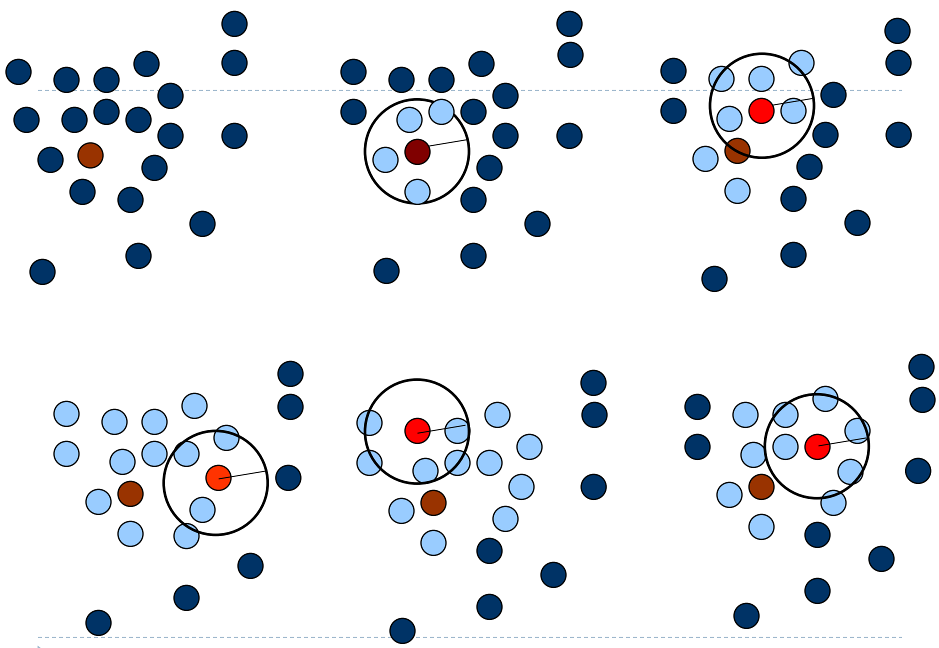
DBSCAN, example *)



1. Check the ε-neighborhood of p;
2. If p has less than MinPts neighbors then mark p as outlier and continue with the next object
3. Otherwise mark p as processed and put all the neighbors in cluster C

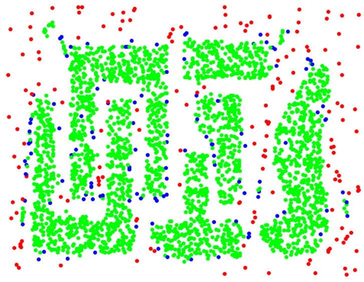
1. Check the unprocessed objects in C
2. If no core object, return C
3. Otherwise, randomly pick up one core object p_1, mark p_1 as processed, and put all unprocessed neighbors of p_1 in cluster C

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt



Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

DBSCAN, example *)



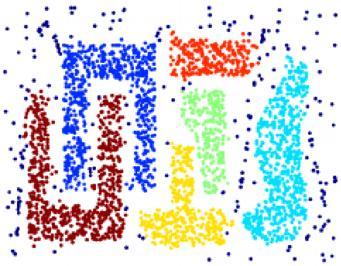
Original Points

ε = 10, MinPts = 4

Point types: core, border and outliers

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

When DBSCAN Works Well

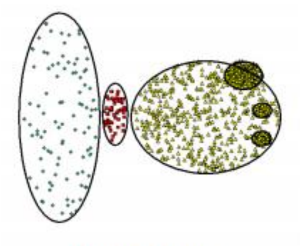


Original Points

Clusters

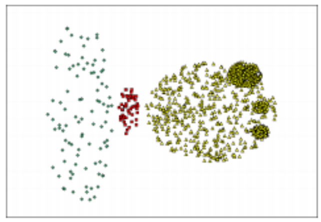
- Resistant to Noise
- Can handle clusters of different shapes and sizes

When DBSCAN Does NOT Work Well

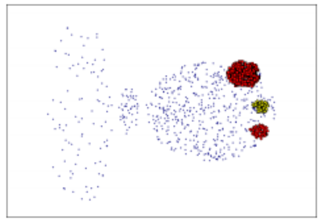


Original Points

- Cannot handle varying densities
- Sensitive to parameters—hard to determine the correct set of parameters



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

Reference

- Tan P., Michael S., & Vipin K. 2006. *Introduction to Data mining*. Pearson Education, Inc. – Chapter 10
- Han J & Kamber M. 2006. *Data mining – Concept and Techniques*. 2nd Edition, Morgan-Kauffman, San Diego - Chapter 7
- www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

Topik selanjutnya:
Teknik klasifikasi