Outlier Detection using Centrality and Center-Proximity

Sang-Wook Kim

Hanyang University

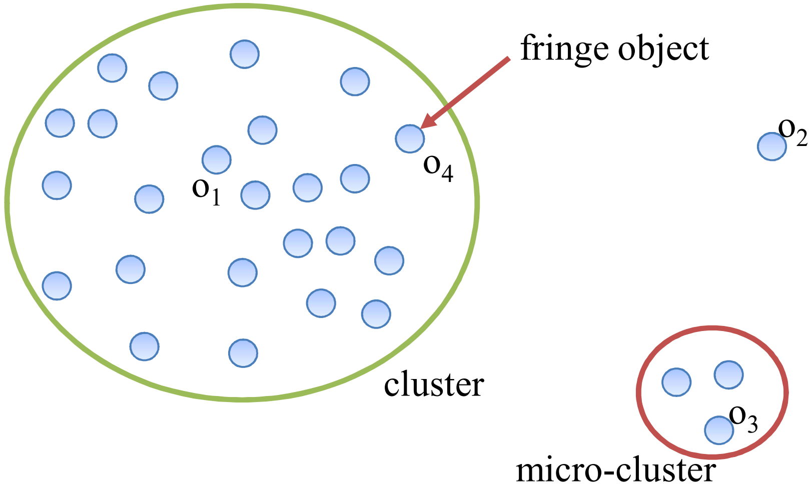
This is a joint work with Duck-Ho Bae, Se-Mi Hwang, and Minsoo Lee, and has been presented in ACM CIKM.

Outlier

- Definition
  - An object that is relatively dissimilar to other normal objects in the dataset
- Applications
  - Detecting network intrusions
    - Identify such packets that are generated intentionally in order to perform harmful operations on the system
  - Detecting misuse of medicines
  - Detecting financial frauds

Outlier

- Types of object



  - O_1: Normal object
  - O_2: Outlier
  - O_3: Outlier belonging to a micro-cluster
  - O_4: Normal object (especially, fringe object)

Previous Methods

- Use their own object location features to detect outliers
  - Object location features reflect the relative characteristics of each object over the distribution of whole objects in the dataset

- Procedures
  1. Compute location features of each object
  2. Assign an outlierness score to the object based on location features
  3. Consider the top *m* objects as outliers

Previous Methods

- Statistics-based outlier detection
- Distance-based outlier detection
- Density-based outlier detection
- RWR-based outlier detection

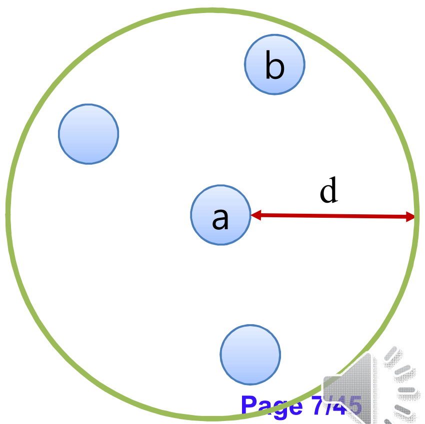
Statistics-based Outlier Detection

- Finds the most suitable statistical distribution model (SDM) for the distribution of objects in the given dataset
- Detects objects that deviate from the SDM as outliers
- Drawbacks
  - Most real-world data is not generated from a specific SDM
  - Difficult to find an SDM for multi-dimensional datasets

Distance-based Outlier Detection

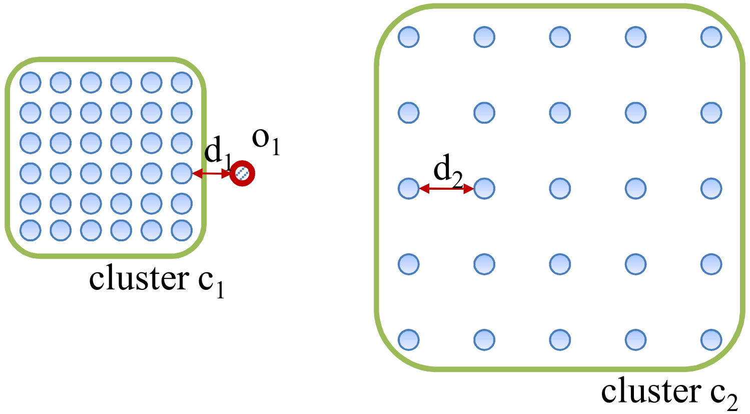
- Uses the distance among objects as a location feature
- Detects objects whose distance to other objects exceeds a specific threshold as outliers

- DB-outlier
  - Location feature: # of other objects existing *within* distance *d*
  - Detects as an outlier if there are less than *p* objects



Distance-based Outlier Detection

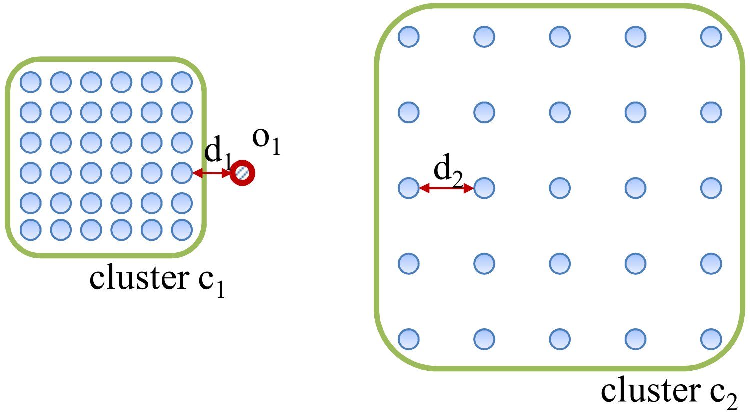
- Drawbacks
  - The location features only consider the characteristics of the object itself
  - Suffer from the local density problem



    - Cannot include object o_1 only as outliers without including all the objects in cluster c_2

Density-based Outlier Detection

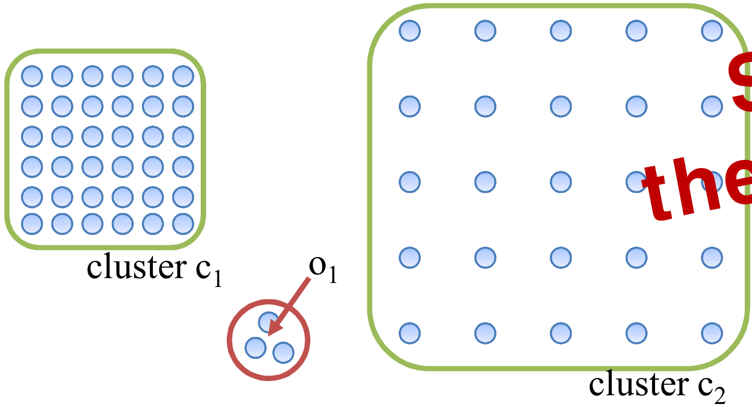
- Detects an object as an outlier if its density is much lower than that of its neighboring objects
  - Density of an object: the number of objects existing within a specific distance



Density-based Outlier Detection

Hanyang University

- Drawbacks
  - The location features only consider the characteristics of the object itself
    - During the calculation of the outlierness score, however, they consider the location features of neighboring objects together
  - Still suffer from the micro-cluster problem



Should consider the characteristics of all the objects in the dataset!

RWR-based Outlier Detection

Hanyang University

- Models a given dataset as an integrated graph
  - Characteristics of all the objects could be considered
- Performs the Random Walk with Restart (RWR)
- Outrank-a
  - Models a dataset as a complete weighted graph
    - Edge weight: similarity between every pair of objects
- Outrank-b
  - Deletes the edges with the similarity lower than a specific threshold

RWR-based Outlier Detection

- Drawbacks (will mention in detail later)
  - Cannot differentiate fringe objects from outlier objects
    - The RWR score is transferred through a directed edge in a single direction
  - Outrank-a
    - Directly considers the characteristics of all the other objects
      - Precision of outlier detection could be low
  - Outrank-b
    - Precision is greatly affected by the user-defined parameter value (threshold)

Overview

Hanyang University

- Our goals
  1. *Should detect outliers accurately*
     - Can solve (1) local density, (2) micro-cluster, and (3) fringe object problems
  2. *Should provide outlierness scores to the user*
     - User can decide the number of outliers intuitively
     - User can get hints on setting the parameter values
  3. *Should be able to handle data of any types/forms*
  4. *Should be less affected by the parameter values*
     - The number of parameters should be as small as possible
     - The fluctuation of precision by a varying parameter values should be small

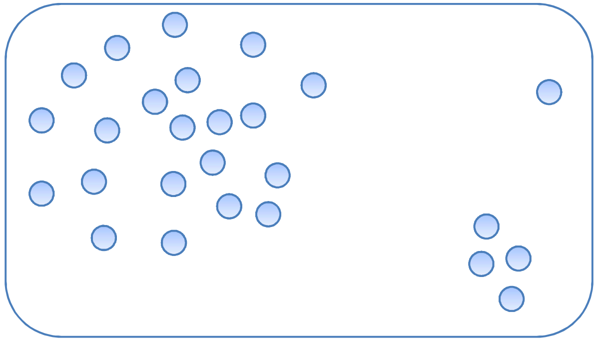
Overview

- Our strategies

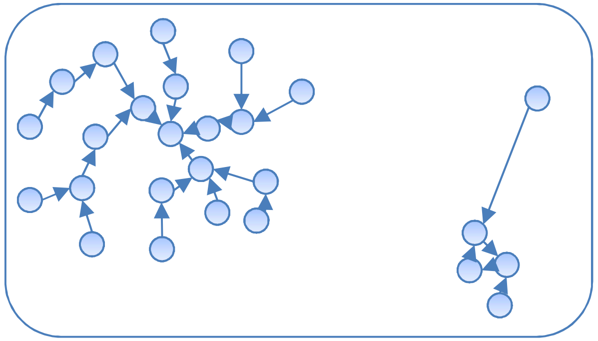
1. Propose two novel location features which can consider the characteristics of all the objects in the dataset
   - Can solve local-density, micro-cluster, fringe object problems (Goal 1)
   - The outlierness score of an individual object not to be seriously affected by parameter values (Goal 4)
2. Build an integrated graph from a given dataset and calculate the outlierness score by analyzing the graph
   - Can provide users with an outlierness score of every object (Goal 2)
   - Can relax the constrains on the input data types/forms (Goal 3)

Overview

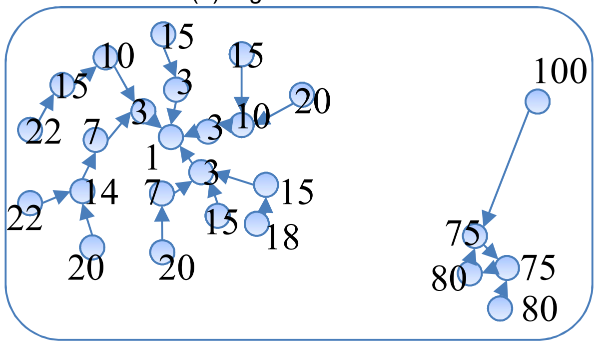
- Procedures
  1. Model a given dataset as a *k*-NN graph
  2. Calculate centrality and center-proximity scores and compute outlierness score using two scores
  3. Detect top *m* objects as outliers



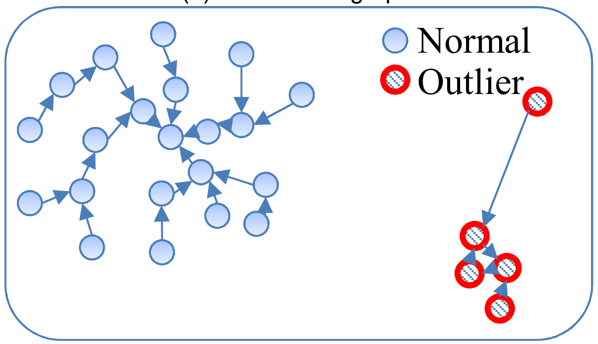
(a) A given dataset.



(b) Model *k*-NN graph.



(c) Calculate outlierness score.



(d) Detect top *m* outliers.

Centrality and Center-Proximity

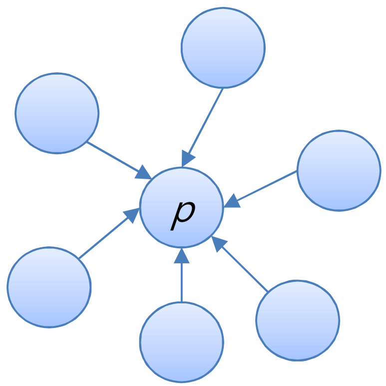
Hanyang University

- Observations
  - An object positioned closer to the cluster center
    - Has many neighbor objects
    - The distances to its neighboring objects are very short
  - An outlier
    - Has very few objects that are close to it
  - In order to quantify such characteristics of objects, we propose two novel location features called centrality and center-proximity

Centrality Score

- The centrality score of object *p* indicates how much other objects recognize *p* as the center of their cluster

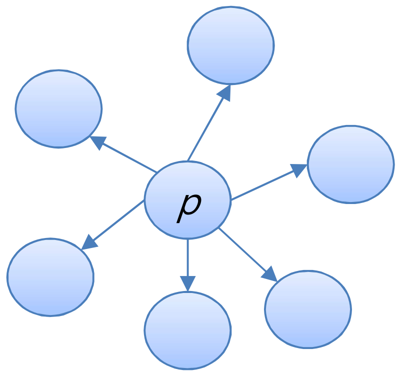
- The centrality score increases when
  1. The number of objects that recognize *p* as their neighbor increases
  2. The center-proximity scores of objects that recognize *p* as their neighbor increase
  3. The distances from *p* to objects that recognize *p* as their neighbor decrease



Center-Proximity Score

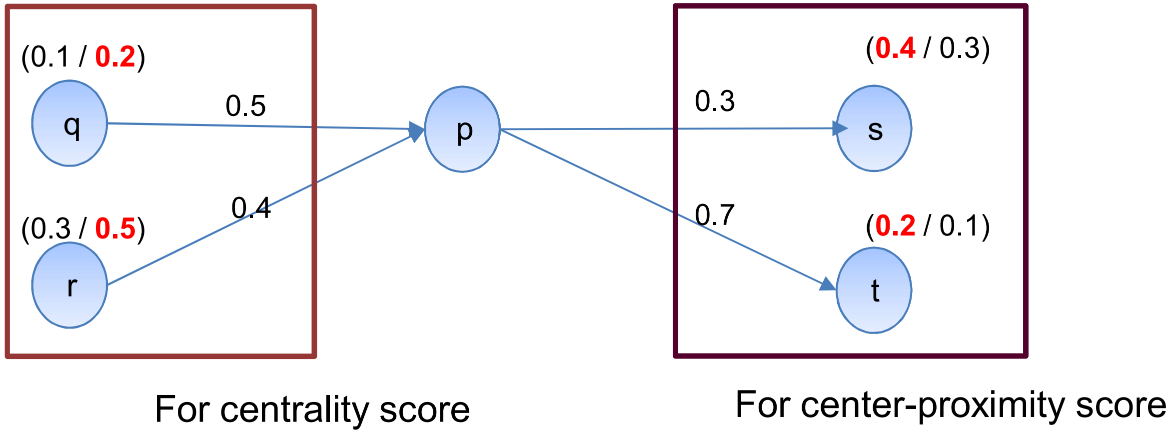
- The center-proximity score of object *p* indicates how close *p* is to objects located in the cluster center

- The center-proximity score increases when
  1. The number of objects that *p* recognizes as its neighbor increases
  2. The centrality scores of objects that *p* recognizes as its neighbor increase
  3. The distances from *p* to the objects that *p* recognizes as its neighbor decrease



Compute Two Scores

- Two scores are computed by referring to each other in an iterative way



  - Centrality score of *p*: 0.2*0.5 + 0.5*0.4
  - Center-Proximity score of *p*: 0.4*0.3 + 0.2*0.7

Compute Two Scores

Hanyang University

- Equations
  - $Centrality_{i+1}(p) = \sum_{q \in In(p)} w_{q \to p} * \frac{Center{-}Proximity_i(q)}{Z_{Out(q)}}$
  - $Center - Proximity_{i+1}(p) = \sum_{q \in Out(p)} w_{p \to q} * \frac{Centralit\ \ _i(q)}{Z_{In(q)}}$

- $In(p)$: set of objects that point to p
- $Out(p)$: set of objects that p points to
- $w_{p \to q}$: weight assigned to edge from p to q
- $Z_{In(q)}$: Sum of all weights assigned to edges from $In(q)$ to q
- $Z_{Out(q)}$: Sum of all weights assigned to edges from q to $Out(q)$

Properties of Two Scores

1. Have mutual reinforcement relationship

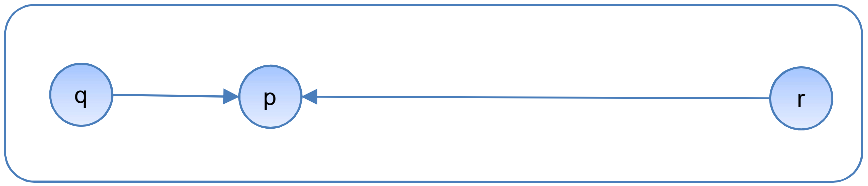
   – The centrality score of an object increases if it is pointed to by many other objects having high center-proximity scores

   – The center-proximity score of an object increases if it points to many objects having high centrality scores

   – Similar to that between the hub and authority scores in HITS

Properties of Two Scores

2. Have influence on its neighboring objects in proportion to the weights on the edges
   - An object has a larger influence on other object that is close to itself



Compute Two Scores

- Procedures

DO Assign initial value '1' to the two scores for all objects
FOR i from 0 to $\boxed{\text{MAX\_ITERATIONS}}$ by 1
{
    FOR j from 1 to NUM_OF_TOTAL_OBJECTS by 1
    {
        DO Calculate the centrality score of node j using Eq. (1)
        DO Calculate the center-proximity score of node j using Eq. (2)
    }
}
DO Normalize the sum of centrality scores of all objects to 1
DO Normalize the sum of center-proximity scores of all objects to 1

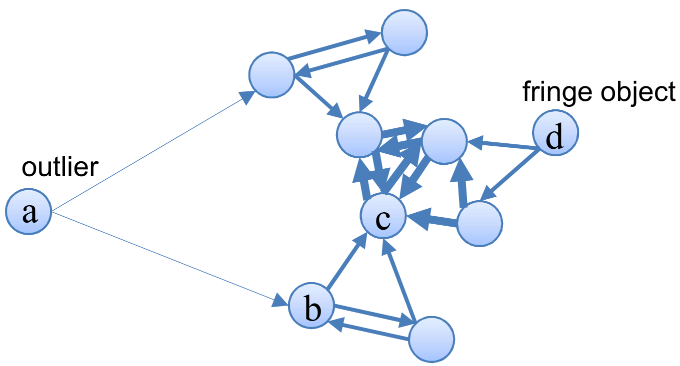
Number of iterations

- Decides how far an object influences other objects in calculating two scores
  - Set MAX_ITERATIONS as 1
    - Consider the influence of only directly connected neighbors
  - Set MAX_ITERATIONS as the diameter of the graph
    - Consider the influence of all the objects in the dataset
  - Compute two scores repetitively until converged - Recommend
    - The mutual reinforcement relationship enables two scores to more clearly differentiate normal objects and outliers

Outlierness Score

Hanyang University

- Uses the inverse of the converged center-proximity score
  - Can differentiate fringe objects and outliers
  - Both are located outside the boundary of the cluster
    - Both have low centrality scores
    - Fringe objects are located closer to the cluster center
      - Have high center-proximity scores compared to outlier objects



Object	Centrality	Center-proximity
a	0.000	**0.128**
b	0.040	0.315
c	0.503	0.341
d	0.000	**0.313**

Compared with RWR

- RWR score
  - Considers (1) how many objects point to an object and (2) how many objects exist around the object
  - Similar in concepts to the centrality score
    - Cannot differentiate fringe objects and outliers

Time Complexity

- $O(E*i)$
  - E: total number of edges in the graph
  - i: number of iterations