Outlier Detection using Centrality and Center-Proximity

Sang-Wook Kim

Hanyang University

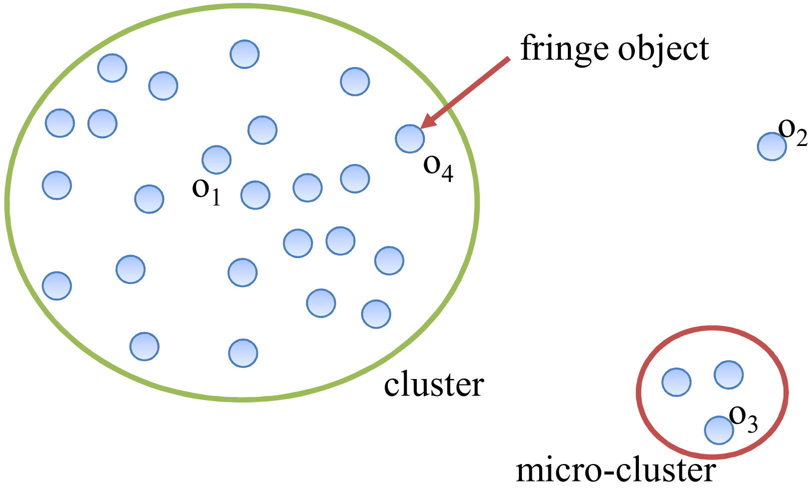
This is a joint work with Duck-Ho Bae, Se-Mi Hwang, and Minsoo Lee, and has been presented in ACM CIKM.

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

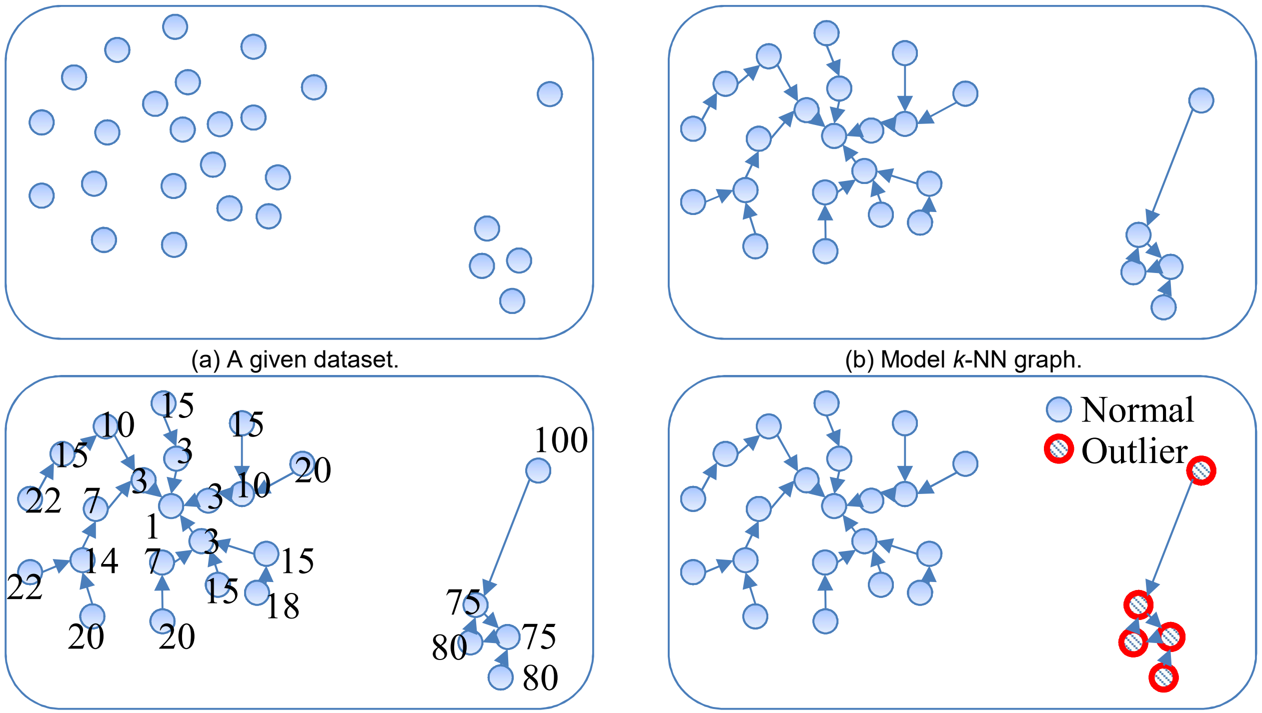
- Types of object



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

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.

Compute Two Scores

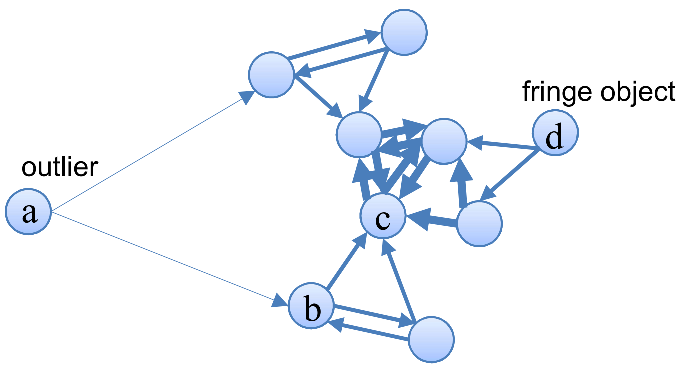
Hanyang University

- Procedures

```
DO Assign initial value '1' to the two scores for all objects
FOR i from 0 to 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
```

Outlierness Score

- 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**

Graph Modeling

1. Graph modeling schemes
   – Edges indicate the neighbor relationships which directly affect the centrality and center-proximity scores of adjacent nodes

2. Weight assignment
   – The centrality and center-proximity scores of an object have influence on its neighboring objects in proportion to the weights on the edges

Graph Modeling Schemes

- We consider three graph modeling schemes

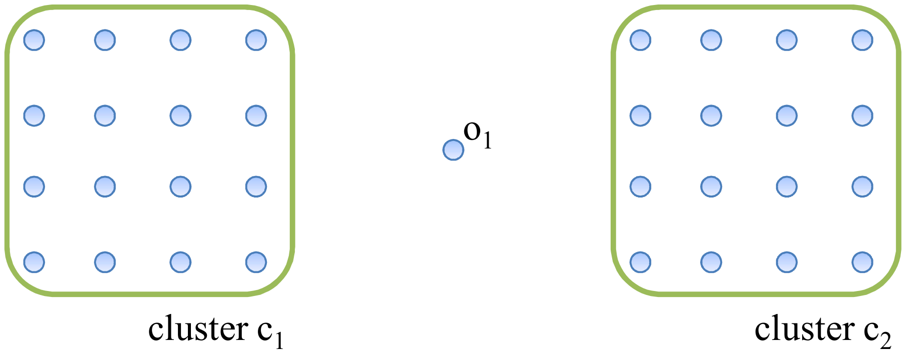
  1. Complete graph
  2. *e*-NN graph
  3. *k*-NN graph

  – Same in representing an object as a node

  – Different only in the way they connect nodes with edges

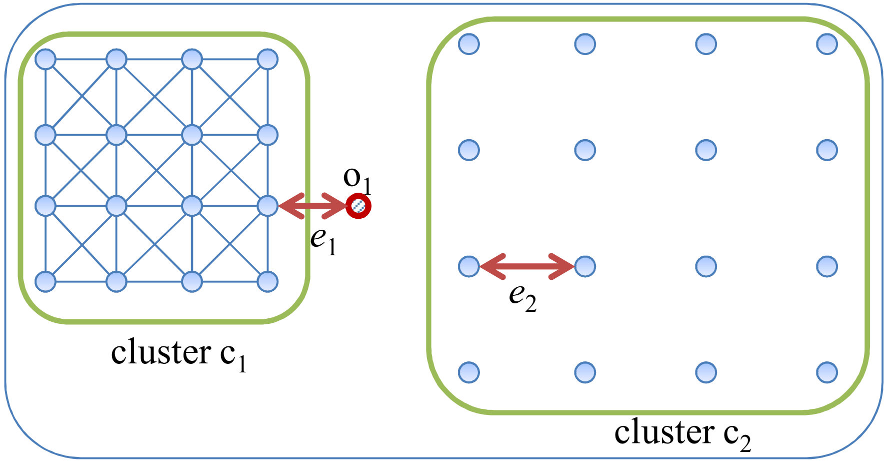
Graph Modeling Schemes

- Complete graph
  - Connects each node to every other node with a directed edge
  - The centrality and center-proximity scores are directly affected by all other objects
    - Two scores show a difference only according to the weight values
    - The objects located at the center of gravity have the highest scores



Graph Modeling Schemes

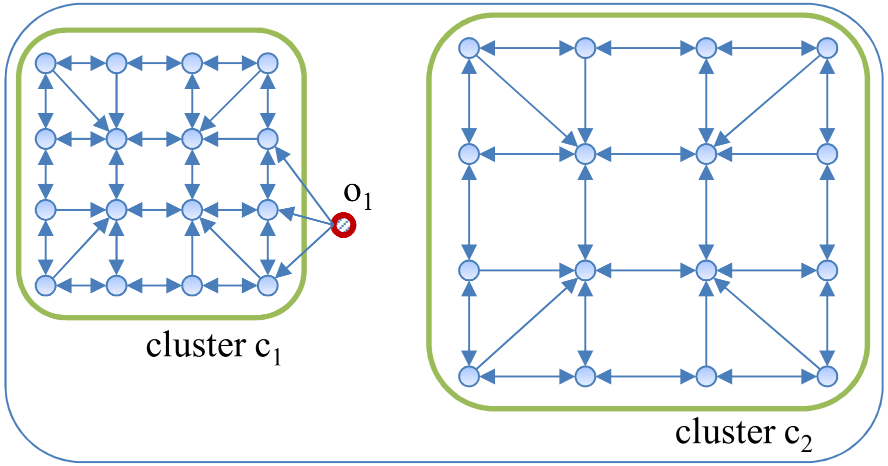
- *e*-NN graph
  - Connects an object with other objects that exist within a specific distance (*e*)
  - Could differ greatly when the value of *e* changes
    - Precision changes considerably when *e* changes



Graph Modeling Schemes

Hanyang University

- *k*-NN graph
  - Connects each object to its *k* nearest objects with a directed edge
  - Out-degrees of all objects are identical
  - In-degrees are different depending on the distribution of neighboring objects
    - Objects located around cluster center: in-degree ↑
    - Objects located at the outside, outliers: in-degree ↓



Graph Modeling Schemes

- *k*-NN graph
  - When *k* is very small,
    - Objects are sparsely connected and may not be able to clearly form a cluster
  - As *k* increases,
    - The clusters are clearly formed
  - When *k* is very large
    - Show a similar result as the complete graph
  - Compared to the *e*-NN graph, *k*-NN shows relatively small fluctuation in precision when *k* changes
    - In *k*-NN graph, it equally connects *k*-NNs for each object

Weight Assignment

- Euclidean similarity
  - Opposite concept of the Euclidean distance

$$Euclidean-Similarity(a,b)=1-\frac{\sqrt{\sum_{i}^{k}(a_i-b_i)^2}-\min}{\max-\min}$$

- Cosine similarity
  - The cosine value between two vectors corresponding two objects
  - $$Cosine-Similarity(a,b)=\frac{\sqrt{\sum_{i}^{k}a_i\times b_i}}{\sqrt{\sum_{i}^{k}a_i^2}\times\sqrt{\sum_{i}^{k}b_i^2}}$$

- Euclidean similarity shows superior precision
  - In case of Cosine similarity, even for two distant objects, a high weight can be assigned

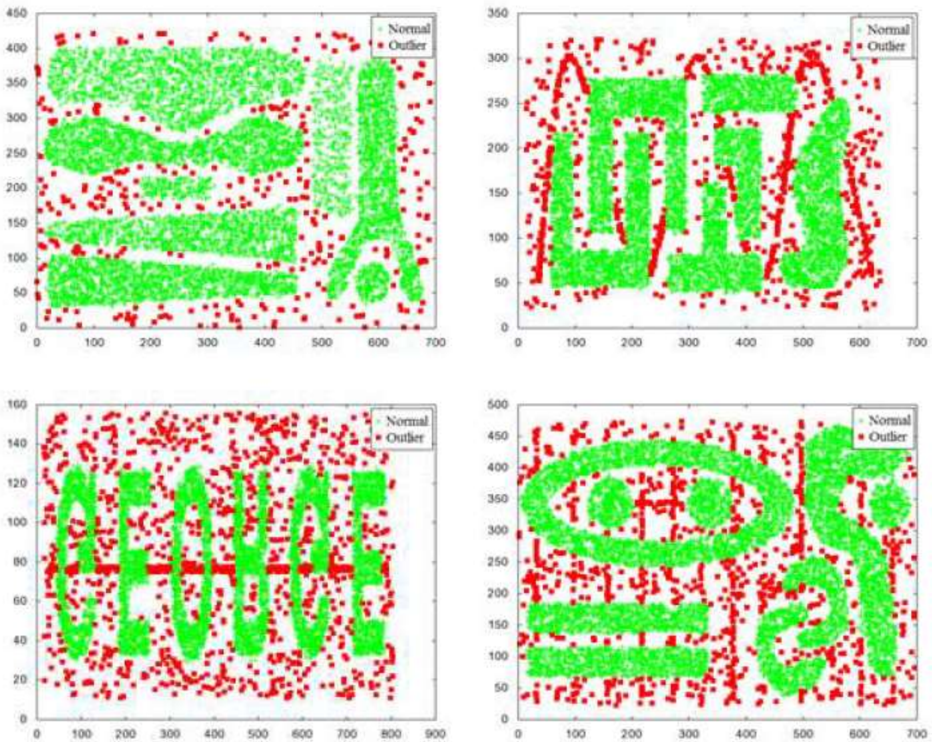
Environment for Experiments

Hanyang University

- Datasets
  - Four 2-dimensional synthetic datasets (Chameleon dataset)
  - One real-world dataset (NBA dataset)
- Evaluation metrics
  - Precision
    - Ground truth was constructed by five human experts
  - Execution time
- Others
  - *i*7 920, 16GB DRAM, Windows 7, C#

Environment for Experiments

- Four Chameleon datasets
  - Composed of 8,000, 8,000, 8,000, and 10,000 objects
    - # of outliers: 328, 803, 1,163, 945



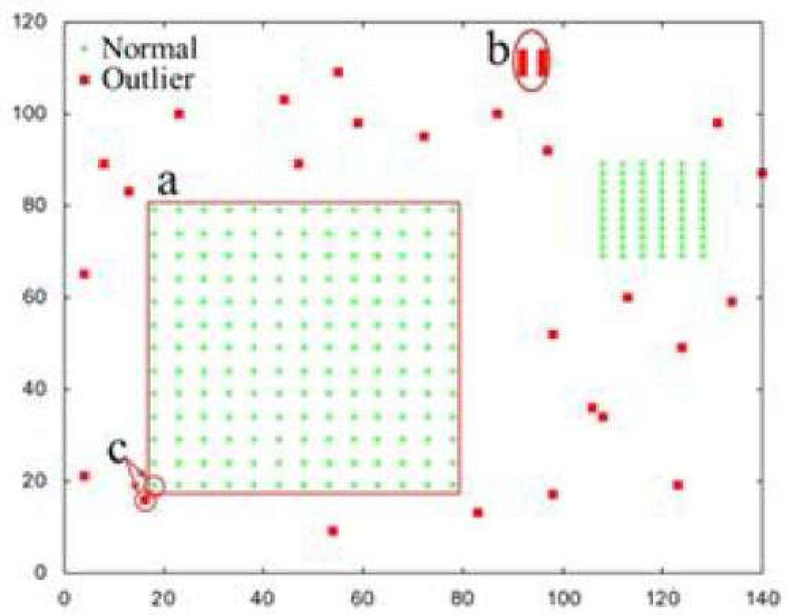
Environment for Experiments

- Types of experiments
  1. Qualitative analysis
  2. Analysis on the proposed method
     - Graph modeling schemes
     - Weight assignment methods
     - The numbers of iterations
  3. Comparison with other methods
     - Precision
     - Execution time
  4. Detecting outliers from a real-world dataset

Results of Experiments

Hanyang University

- Qualitative analysis

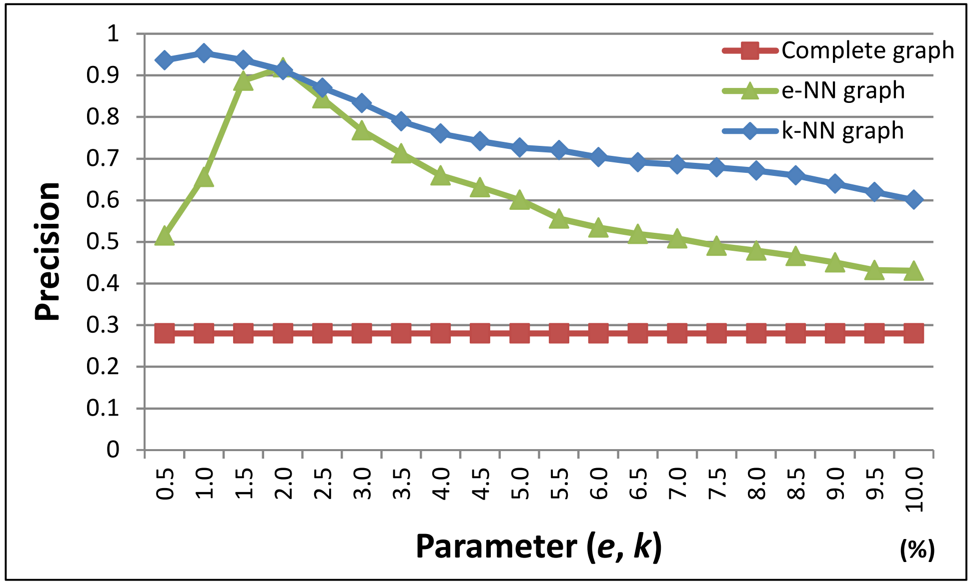


  – The proposed method does not suffer from (a) the local density problem and (b) the micro-cluster problem, and (c) can differentiate between fringe objects and outliers

Results of Experiments

Hanyang University

- Analysis on graph modeling schemes

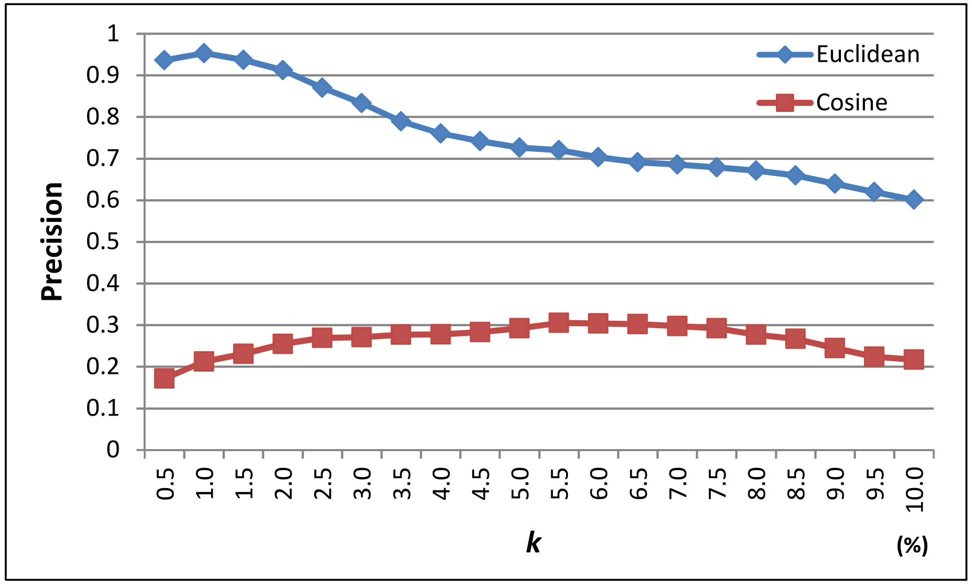


  - *k*-NN graph shows the highest precision in all cases
  - *e*-NN graph shows large fluctuation according to the change of *e*

Results of Experiments

Hanyang University

- Analysis on weight assignment methods

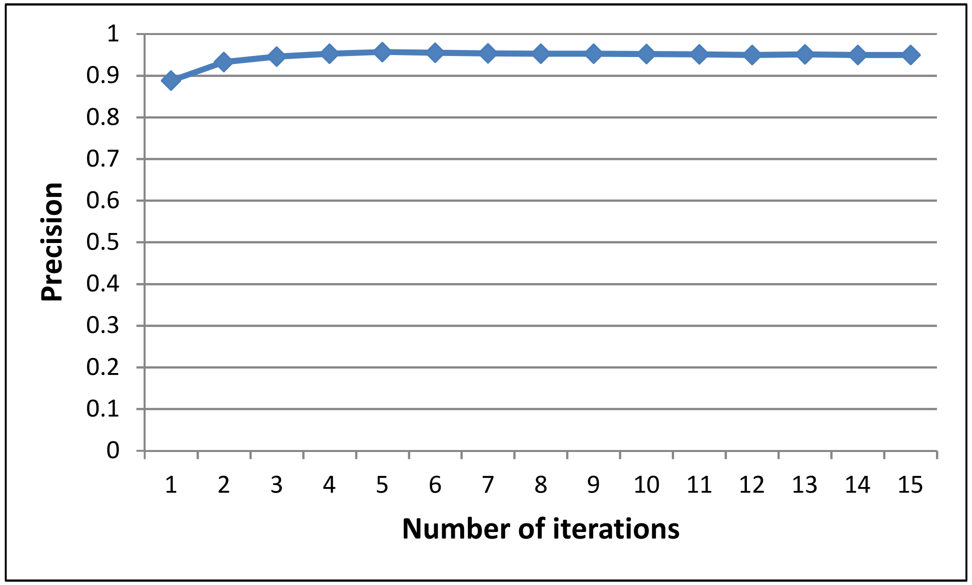


  - Euclidean similarity method shows superior precision

Results of Experiments

Hanyang University

- Analysis on the numbers of iterations



  - The precision increases as the number of iterations increases
  - The number of iterations exceeds 6, the precision does not change
    - Centrality and center-proximity scores are converged

Results of Experiments

- Comparisons with other methods
  - Precision

	k-Dist	LOF	Outrank-a	Outrank-b	Our Method
Average	0.86	0.88	0.13	0.16	**0.90**

- Our method provides the best precision
- Density-based method (LOF) shows a higher precision than distance-based method (k-Dist)
- Outrank methods shows a very low precision
  - They have problems in their location features and in the graph modeling schemes

Results of Experiments

- Comparisons with other methods
  - Execution time (*ms*)

	k-Dist	LOF	Outrank-a	Outrank-b	Our Method
Average	63,847	62,790	472,128	11,162,388	**64,525**

- Our method does not show any big difference
- In case of Outrank methods,
  - Outrank-a method models the dataset as a complete graph, thus, requires a lot of execution time
  - In Outrank-b method, the execution time for weight assignment takes huge amount of time

Results of Experiments

Hanyang University

- Detecting outliers from a real-world dataset
  - Omitted results

Conclusions

- Contributions
  - Have proposed the notions of centrality and center-proximity as novel relative location features
    - Our features consider the characteristics of all the objects in the dataset
  - Have proposed a graph-based outlier detection method
    - Our method solves the local density problem and the micro-cluster problem, and also differentiates the fringe objects and outlier objects
  - Have carefully analyzed the effect of graph modeling schemes on outlier detection
  - Have verified the effectiveness and efficiency of our method through extensive experiments