Rank-based outlier detection

Article *in* Journal of Statistical Computation and Simulation · October 2011

DOI: 10.1080/00949655.2011.621124

CITATIONS
17

READS
263

3 authors, including:

Kishan Mehrotra
Syracuse University
112 PUBLICATIONS **1,535** CITATIONS

SEE PROFILE

Chilukuri K. Mohan
Syracuse University
169 PUBLICATIONS **3,645** CITATIONS

SEE PROFILE

Department of Electrical Engineering and Computer Science

Technical Report

SYR-EECS-2011-07 April 6, 2011

Rank-Based Outlier Detection

H. Huang

K. Mehrotra mehrotra@syr.edu

C.K. Mohan ckmohan@syr.edu

ABSTRACT: We propose a new approach for outlier detection, based on a new ranking measure that focuses on the question of whether a point is “important” for its nearest neighbors; using our notations low cumulative rank implies the point is central. For instance, a point centrally located in a cluster has relatively low cumulative sum of ranks because it is among the nearest neighbors of its own nearest neighbors. But a point at the periphery of a cluster has high cumulative sum of ranks because its nearest neighbors are closer to the points. Use of ranks eliminates the problem of density calculation in the neighborhood of the point and this improves performance. Our method performs better than several density-based methods, on some synthetic data sets as well as on some real data sets.

KEYWORDS: Outlier detection, ranking, neighborhood sets

Syracuse University - Department of EECS,
4-206 CST, Syracuse, NY 13244
(P) 315.443.2652 (F) 315.443.2583
http://eecs.syr.edu

Rank-Based Outlier Detection

H. Huang, K. Mehrotra, C. K. Mohan
Department of EECS, Syracuse University

April 6, 2011

Abstract

We propose a new approach for outlier detection, based on a new ranking measure that focuses on the question of whether a point is "important" for its nearest neighbors; using our notations low cumulative rank implies the point is central. For instance, a point centrally located in a cluster has relatively low cummulative sum of ranks because it is among the nearest neighbors of its own nearest neighbors. But a point at the periphery of a cluster has high cummulative sum of ranks because its nearest neighbors are closer to the points. Use of ranks eliminates the problem of density calculation in the neighborhood of the point and this improves performance. Our method performs better than several density-based methods, on some synthetic data sets as well as on some real data sets.

Keywords: Oulier detection, ranking, neighborhood sets

1 Introduction

Outlier detection algorithms attempt to find data points that are "different" from the rest of the data points in a given data set. The problem is of considerable importance, arising frequently in many real-world applications, for data mining researchers. Many practical applications concerning outlier detection occur in different domains such as fraud detection, cyber-intrusion detection, medical anomaly detection, image processing and textual anomaly detection [1].

Statistics-based approaches (see [2,3]) were first used for outlier detection based on an assumption that the distributions of datasets are known. A data point was defined as an outlier if it deviates from the existing distribution. With sufficient knowledge about the dataset, statistics-based methods work effectively. But in real-world, unfortunately, distributions of datasets are unknown, significantly impacting the performances of these methods. To overcome this obstacle clustering-based algorithms have been proposed to detect outliers [4, 5]. The basic idea is that a data point is an outlier if it does not belong to any cluster. Outliers can be found by removing

all points that belong to clusters. The effectiveness of this approach depends on the clustering algorithm. Knorr and Ng [6] propose to detect an outlier based on its distances from neighboring data points, many other variations of distance-based approaches have been discussed in the literature [7–9].

Breunig *et al.* [10] proposed that each data point of the given data set should be assigned a degree of outlierness. In their view, as in other recent studies, a data point's degree of outlierness should be measured relative to its neighbors; hence they refer to it as the "local outlier factor" (LOF) of the data point. Tang *et al.* [11] argued that an outlier doesn't always have to be of lower density and lower density is not a necessary condition to be an outlier. They modified LOF to obtain the connectivity-based outlier factor (COF) which they argued is more effective when a cluster and a neighboring outlier have similar neighborhood densities. Local density of a is generally measured in terms of k-nearest neighbors; LOF and COF both exploit properties associated with k-nearest neighbors of a given object in the data set. However, it is possible that an outlier lies in a location between objects from a sparse and a denser cluster. To account for such possibilities, Jin *et al.* [12] proposed another modification, called INFLO, which is based on a symmetric neighborhood relationship, i.e., the proposed modification considers neighbors and 'reverse neighbors' of a data point when estimating its density distribution. Detailed descriptions of the LOF, COF and INFLO algorithms are presented in the next section.

Analysis of these density-based outlier detection algorithms reveals that they use the following methodology:

- Define the concept of density;
- Use the notion of neighborhood (or some variation); and
- Calculate the "outlierness" of an object; ususlly defined as the ratio of a data points density with the density in the region surrounding the data point.

A methodology that exploits k-neighborhood of a data point has many good features. For instance, it is independent of the distribution of the data and is capable of detecting isolated objects. But it also has some shortcomings:

- Density-based algorithms assume that neighbors of a data point have similar density. If some neighbors located in one cluster and the other neighbors located in another cluster have different densities, then comparing the density of the data point with all of its neighbors may lead to a wrong conclusion and recognition of real outliers may fail.
- The notion of density does not work well for special datasets, for example, if all data points lie on a single straight line. Even if each instance of dataset has equal distances between itself with its closest neighbors, it may still have different density depending on its placement in the dataset.

To overcome the weaknesses mentioned above we defien and use the notion of of the rank of an object with respect to its neighbor. Thus, we propose an outlier detection methodology based on the mutuality of the relationship between a data point and its neighbors. A detailed description of these concepts, and an analysis for the reasons why it is better, is presented in section 3. In section 4, experimental results are discussed.

2 Detailed Description of LOF, COF, and INFLO Algorithms

In the following, D denotes the given data set of all observations, k is a positive integer, and $d(p, q)$ denotes the distance between two points $p, q \in D$. This distance measure could be any reasonable measure but for concreteness we use the Euclidean distance.

2.1 LOF (Local Outlier Factor) approach

Breunig *et al.* [10] proposed the following approach to find an outlier.

1. Find the distance, $d_k(p)$, between p and its kth nearest neighbor. Denote the set of k nearest neighbors of p by $\mathcal{N}_k(p) = \{q \in D - \{p\} : \ d(p, q) \leq d_k(p)\}$.

2. Define the reachability distance of a point q from p, as $\mathcal{R}_k(p, q) = \max\{d_k(q), d(p, q)\}$.

3. The local reachability density of a point is defined as the inverse of the average reachability distance. Specifically it is
$$\ell_k(p) = [\frac{\sum_{q \in \mathcal{N}_k(p)} \mathcal{R}_k(p, q)}{|\mathcal{N}_k(p)|}]^{-1}.$$

4. LOF (local outlier factor) of a point p is defined as:
$$\mathcal{L}_k(p) = [\frac{\sum_{o \in \mathcal{N}_k(p)} \frac{\ell_k(o)}{\ell_k(p)}}{|\mathcal{N}_k(p)|}].$$

5. The LOF of each point is calculated, and points are sorted in decreasing order of $\mathcal{L}_k(p)$. If the LOF values are 'large', the corresponding points are declared as outliers.

6. To account for k, the final decision is taken as follows: $\mathcal{L}_k(p)$ is calculated for selected values of k in a pre-specified range, max $\mathcal{L}_k(p)$ is retained, and a p with large LOF is declared an outlier.

2.2 COF (Connectivity-based Outlier Factor) approach

Tang *et al.* [13] suggest a new method to calculate the density as described below. Define the distance between two non-empty sets P and Q as $d(P,Q) = \min\{d(p,q) : p \in P, q \in Q\}$. This can be used to find the minimum distance between a point and a set by treating one of the set as a singleton.

1. Given a point p we define set-based path (SBN) of length k as a path $< p \equiv p_1, p_2, \ldots, p_k >$ such that for all $1 \leq i \leq k-1$, p_{i+1} is the nearest neighbor of the set $\{p_1, p_2, \ldots, p_i\}$. In other words, the SBN-path represents the order in which nearest neighbors of p are successively obtained. The set $\mathcal{N}_k(p) = \{p_1, p_2, \ldots, p_k\}$ is the set of k nearest neighbors of p.

2. The Set-based trail (SBT) is an ordered collection of $k-1$ edges associated with a given SBN path $< p \equiv p_1, p_2, \ldots, p_k >$. The ith edge e_i connects a point $o \in \{p_1, \ldots, p_i\}$ to p_{i+1} and is of minimum distance; i.e., length of e_i is equal to $d(o, p_{i+1}) = d(\{p_1, \ldots, p_i\}, \{p_{i+1}, \ldots, p_k\})$. Denote the length of edge e_i as $l(e_i)$.

3. Given p, the associated SBN path $< p \equiv p_1, p_2, \ldots, p_k >$, and the SBT $< e_1, e_2, \ldots, e_{k-1} >$, the average-chaining distance ($\mathcal{A}$) of p is weighted sum of the lengths of the edges, with larger weights assigned to nearest edges, that is:

$$\mathcal{A}_{\mathcal{N}_k(p)}(p) = \frac{2}{k} \sum_{i=1}^{k-1} \frac{k-i}{k-1} l(e_1).$$

4. Finally, the connectivity-based outlier factor (COF) of a point p is defined as

$$\mathrm{COF}_k(p) = [\mathcal{A}_{\mathcal{N}_k(p)}(p)][\frac{\sum_{o \in \mathcal{N}_k(p)} \mathcal{A}_{\mathcal{N}_k(p)}(o)}{|\mathcal{N}_k(p)|}]^{-1}.$$

5. As in COF, larger values of $\mathrm{COF}_k(p)$ denote higher possibility that p is an outlier.

2.3 INFLO (INFLuential measure of Outlierness by symmetric relationship) approach

Proposed by Jin *et al.* [12], in INFLO the k nearest neighbors and reverse nearest neighbors of an object p are used to obtain a measure of outlierness. Recall that given an object p

1. Reverse Nearest Neighborhood (RNN) of p is defined as

$$\mathcal{RN}_k(p) = \{q : \ q \in D \text{ and } p \in \mathcal{N}_k(q)\}.$$

   Note that $\mathcal{N}_k(p)$ has exactly k objects but $\mathcal{RN}_k(p)$ may not have k objects. In some instances, it may be empty, because for all $q \in \mathcal{N}_k(p)$, p may not be in any of the set of $\mathcal{N}_k(q)$.

2. The k-influential space for p, denoted as $\mathrm{IS}_k(p) = \mathcal{N}_k(p) \cup \mathcal{RN}_k(p)$.

3. The influenced outlierness of a point p is defined as

$$\mathrm{INFLO}_k(p) = \frac{1}{\mathrm{den}(p)} \frac{\sum_{o \in \mathrm{IS}_k(p)} \mathrm{den}(o)}{|(\mathrm{IS}_k(p))|}$$

where $\mathrm{den}(p) = \frac{1}{d_k(p)}$.

Thus for any p, INFLO expands $N_k(p)$ to $IS_k(p)$ and compares p's density with average density of objects in $IS_k(p)$.

3 Rank-Based Detection Algorithm (RBDA)

This section presents a new approach to identify outliers based on mutual closeness of a data point and its neighbors. To understand mutual closeness consider a data point $p \in D$ and suppose that $q \in \mathcal{N}_k(p)$. That is, consider a q which is "close" to p because it belongs to k-neighborhood of p. In return, we ask "how close is p to q?". If p and q are 'close' to each other, then we argue that (with respect to each other) p and q are not anomalous data points. This forms the basis of RBDA.
Description of Rank-based Detection Algorithm (RBDA) Algorithm

1. For $p \in D$ let $q \in N_k(p)$. We calculate the rank of p among all neighbors of q; i.e., we calculate the set of $d(q, o)$ for all $o \in D - \{q\}$ and find the rank of $d(q, p)$ in this set. Let this be $r_q(p)$.

2. 'Outlierness' of p, denoted by $O_k(p)$, is defined as:

$$O_k(p) = \frac{\sum_{q \in N_k(p)} r_q(p)}{|N_k(p)|}. \tag{1}$$

If $O_k(p)$ is 'large' then p is considered an outlier.

3. To determine a criterion for 'largeness', let $D_o = \{p \in D \,|\, O_k(p) \leq O_{max}\}$ where O_{max} is chosen such that the size of D_o is 75% of the size of D. We normalize $O_k(p)$ as below:

$$Z_k(p) = \frac{1}{S_k}(O_k(p) - \bar{O}_k) \tag{2}$$

where

$$\bar{O}_k = \frac{1}{|D_o|} \sum_{p \in D} O_k(p) \text{ and } S_k^2 = \frac{1}{|D_o| - 1} \sum_{p \in D} (O_k(p) - \bar{O}_k)^2$$

and if the normalized value $Z_k(p) \geq 2.5$, then we declare that p is an outlier.

Algorithm: Rank-Based Detection Algorithm

Input: k, D

Output: List of RBDA values for each object p∈D

Method:

```
RBDAlist = NULL;                                  /* Initialized output list. */
FOR each object p in D DO
    N(p) = NULL;                                  /* Initialized p's neighborhood. */
    Nk(p) = NULL;
    FOR each object q in D DO
        IF q!=p THEN
            Add q to N(p);                        /* Add q to p's D-neighborhood */
    Sort N(p) by dist(p, q) in ascending order;
    dk(p) = dist(p, qk);                          /* qk is the kth in N(p); */
    tmp = 0; index = 0; rank = 0;                 /* Prepare for assigning rankings */
    FOR each object q in N(p) DO                  /* From the closest neighbor to last neighbor */
        IF dist(p,q) ≤ dk(p) THEN
            Add q to Nk(p);                       /* Add q to p's k-neighborhood. */
        index++;
        IF dist(p,q) != tmp THEN
            rank = index;
        rp(q) = rank; tmp = dist(p,q);            /* Assign rankings */
FOR each object p in D DO
    sumranks = 0;
    FOR each object q in Nk(p) DO
        sumranks += rq(p) ;                       /* Sum up all rankings with respect to neighbors */
    RBDAlist(p) = sumranks / |Nk(p)|;             /* Get p's average rankings */
```

Figure 1: Rank-based Detection Algorithm.

Consider data set in Figure 2. Consider $k = 6$, which is chosen to be ten percent of the size of the dataset and the data point A.Six closests neighbors of A are all six points of its closest cluster.

Since A is farest away form these six points, each of these six point contributes $O_6(A) = 6$. In contrast, $O_6(.)$ values for other ponis of the cluster vary from 1 to 6. Thus, it can be easily seen that A will be identified as an outlier.

In this example, LOF, COF and INFLO algorithms assign a higher outlier value to data point B than to A; which is wrong. The reason that data point B gets a higher outlier value is that some data points from a neighboring clusters are its 6-neighbors and density-based algorithms fail to identify the true outlier A. In RBDA algorithm, data point A gets high $O_k(A)$ values with respect to all its k-neighbors. In the final analysis RBDA algorithm identiies A as an outlier and RBDA is the only algorithm that ranks A as most likely outlier data point for $k = 7$.

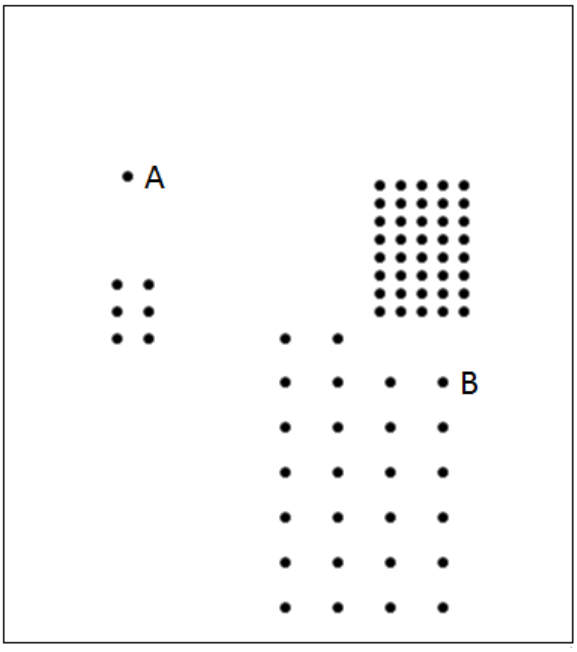

Figure 2: Evaluation of outlierness of a data point, an example.

4 Experiments:

We use two synthetic and three real datasets to compare the performance of RBDA with LOF, COF and INFLO. Metrics to compare the algorithms are described below.

4.1 Metrics for Measurement

To evaluate the performance of the algorithms, three metrics were selected – precision, recall, and Rank-Power [14–17].

Suppose, using a given outlier detection algorithm, we identify m most suspicious instances in D which contains d_t true outliers and let m_t be the number of true outliers among m instances. Then *Precision* which measures the proportion of true outliers in top m suspicious instances, is:

$$\text{Precision} = \frac{m_t}{m},$$

and *Recal* which measure the accuracy of an algorithm is:

$$\text{Recall} = \frac{|m_t|}{|d_t|}.$$

Precision and recall don't capture the effectiveness completely. One algorithm may identify an outlier as the most suspicious while another algorithm mayidentify it as the least suspicious. Yet the above two measures remain the same. Ideally, an algorithm will be considered more effective if it true outliers occupy top positions and non-outliers in the bottom of the m suspicious instances. Rank-Power was proposed by Tang *et al.* [13] to capture this notion. Let n denote the number of outliers found within top m instances and R_i denote the rank of the ith *true* outlier. Then,

$$\text{RankPower} = \frac{n(n+1)}{2\sum_{i=1}^{n} R_i}.$$

Rank-Power takes maximum value 1 when all n true outliers are in top n positions.

For a fixed value of m, larger values of these metrics imply better performance.

4.2 Synthetic Datasets

Two synthetic datasets, shown in Figures 3 and 4, are used to evaluate the outlier detection algorithms. In each dataset, there are multiple clusters with different densities. In each dataset we have placed six additional objects, (a, b, c, d, e, and f) in the vicinities of the clusters to evaluate their''outlierness' by LOF, COF, INFLO, and our proposed algorithm RBDA.

In tables below, that summarize the performances of the algorithms, Nrc denotes the number of outliers within top m instances, Pr represents precision, Re represents recall, and RP represents Rank-Power.

4.2.1 Synthetic Dataset 1

Synthetic dataset 1 contains 74 instances, including six planted outliers; has four clusters of different densities consisting of 36, 8, 8, 16 instances. Four different values of k and four values of m are used; results are shown in Table 1.

Table 1: Comparison of LOF, COF, INFLO and RBDA for k = 5, 7, 10 and 12 respectively for synthetic dataset 1. Maximum values are marked as bold.

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	**5**	**1.00**	**0.83**	**1.00**	**5**	**1.00**	**0.83**	**1.00**	**5**	**1.00**	**0.83**	0.882	**5**	**1.00**	**0.83**	**1.00**
10	**6**	**0.60**	**1.00**	0.95	**6**	**0.60**	**1.00**	0.95	**6**	0.40	**1.00**	0.875	**6**	**0.60**	**1.00**	**1.00**
15	**6**	**0.40**	**1.00**	0.95	**6**	**0.40**	**1.00**	0.95	**6**	**0.40**	**1.00**	0.875	**6**	**0.40**	**1.00**	**1.00**
30	**6**	**0.20**	**1.00**	0.95	**6**	**0.20**	**1.00**	0.95	**6**	**0.20**	**1.00**	0.875	**6**	**0.20**	**1.00**	**1.00**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	**5**	**1.00**	**0.83**	**1.000**	**5**	**1.00**	**0.83**	**1.000**	**5**	**1.00**	**0.83**	0.882	**5**	**1.00**	**0.83**	**1.00**
10	**6**	**0.60**	**1.00**	0.913	**6**	**0.60**	**1.00**	0.913	**6**	**0.60**	**1.00**	0.955	**6**	**0.60**	**1.00**	**1.00**
15	**6**	**0.40**	**1.00**	0.913	**6**	**0.40**	**1.00**	0.913	**6**	**0.40**	**1.00**	0.955	**6**	**0.40**	**1.00**	**1.00**
30	**6**	**0.20**	**1.00**	0.913	**6**	**0.20**	**1.00**	0.913	**6**	**0.20**	**1.00**	0.955	**6**	**0.20**	**1.00**	**1.00**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	3	0.60	0.50	**1.000**	**4**	**0.80**	**0.67**	**1.000**	3	0.60	0.50	**1.000**	**4**	**0.80**	**0.67**	**1.00**
10	4	0.40	0.67	0.667	**5**	**0.50**	**0.83**	0.789	4	0.40	0.67	0.833	4	0.40	0.67	**1.00**
15	4	0.27	0.67	0.667	**5**	**0.33**	**0.83**	**0.789**	4	0.27	0.67	0.833	**5**	**0.33**	**0.83**	0.63
30	4	0.13	0.67	0.667	5	0.17	0.83	**0.789**	5	0.17	0.83	0.360	**6**	**0.20**	**1.00**	0.51

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	3	0.60	0.50	**1.000**	**4**	**0.80**	**0.67**	**1.000**	2	0.40	0.33	**1.000**	**4**	**0.80**	**0.67**	0.900
10	4	0.40	0.67	0.625	**5**	**0.50**	**0.83**	0.789	4	0.40	0.67	0.526	4	0.40	0.67	**0.909**
15	**5**	**0.33**	**0.83**	0.484	**5**	**0.33**	**0.83**	**0.789**	4	0.27	0.67	0.526	**5**	**0.33**	**0.83**	0.600
30	5	0.17	0.83	0.484	5	0.17	0.83	**0.789**	5	0.13	0.67	0.526	**6**	**0.20**	**1.00**	0.488

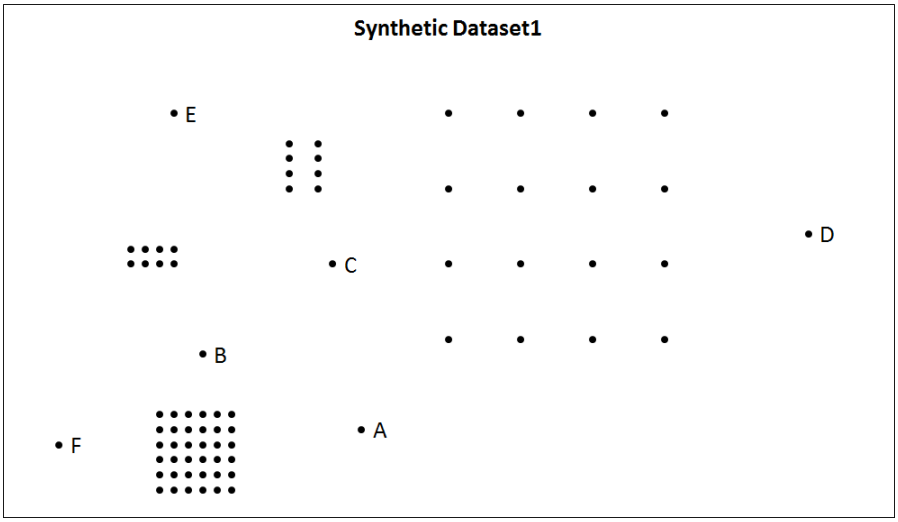

Figure 3: A Synthetic dataset with clusters obtained by placing all points uniformly with varying degrees of densities.

For k equals to 5 and 7, all algorithms find all six outliers within top m ranked instances, but the RankPower of RBDA algorithm is higher than those of LOF and COF algorithms.

For k equals to 10 and 12 and for m smaller than 10, COF has higher precision and recall than the other algorithms, RBDA algorithm comes second. When m is 30, Precision of RBDA is higher than others since all six outliers can be found by RBDA while only 5 outliers can be found in LOF, COF and even less by INFLO.

4.2.2 Synthetic Dataset 2

Synthetic dataset 2 consists of 515 instances including planted six outliers; has one large normally-distributed cluster and two small uniform clusters.

Results are presented in Table 2 for $k = 10, 15$, and 20 and $m = 5, 10, 15, 20$ and 30. It can be seen that when k equals to 10, INFLO has the best Rank-Power and RBDA has the best precision. When k is increased to 15 and 20, RBDA performs better than others and it has the best precision, recall and Rank-Power. Especially for k is 20, RBDA achieves maximum Rank-Power for all values of m from 5 to 30.

In this experiment, RBDA algorithm works better than others in most of the cases, and when $k = 20$, it achieves the best performance.

Table 2: Comparison of LOF, COF, INFLO and RBDA for k = 10, 15 and 20 respectively for synthetic dataset 2. Maximum values are marked as bold.

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	**5**	**1.00**	**0.83**	**1.000**	4	0.80	0.67	0.909	**5**	**1.00**	**0.83**	**1.000**	**5**	**1.00**	**0.83**	**1.000**
10	**5**	**0.50**	**0.83**	**1.000**	**5**	**0.50**	**0.83**	0.882	**5**	**0.50**	**0.83**	**1.000**	**5**	**0.50**	**0.83**	**1.000**
15	**5**	**0.33**	**0.83**	**1.000**	**5**	**0.33**	**0.83**	0.882	**5**	**0.33**	**0.83**	**1.000**	**5**	**0.33**	**0.83**	**1.000**
20	5	0.25	0.83	**1.000**	5	0.25	0.83	0.882	5	0.25	0.83	**1.000**	**6**	**0.30**	**1.00**	0.636
30	5	0.25	0.83	**1.000**	6	0.20	1.00	0.512	5	0.17	0.83	**1.000**	**6**	**0.20**	**1.00**	0.636

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	**5**	**1.00**	**0.83**	**1.000**	4	0.80	0.67	**1.00**	**5**	**1.00**	**0.83**	**1.000**	**5**	**1.00**	**0.83**	**1.000**
10	**6**	**0.60**	**1.00**	**0.955**	5	0.50	0.83	0.938	**6**	**0.60**	**1.00**	**0.955**	**6**	**0.60**	**1.00**	**0.955**
15	**6**	**0.40**	**1.00**	**0.955**	**6**	**0.40**	**1.00**	0.750	**6**	**0.40**	**1.00**	**0.955**	**6**	**0.40**	**1.00**	**0.955**
20	**6**	**0.30**	**1.00**	**0.955**	**6**	**0.30**	**1.00**	0.750	**6**	**0.30**	**1.00**	**0.955**	**6**	**0.30**	**1.00**	**0.955**
30	**6**	**0.20**	**1.00**	**0.955**	**6**	**0.20**	**1.00**	0.750	**6**	**0.20**	**1.00**	**0.955**	**6**	**0.20**	**1.00**	**0.955**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	4	0.80	0.67	**1.000**	4	0.80	0.67	0.909	4	0.80	0.67	**1.000**	**5**	**1.00**	**0.83**	**1.000**
10	**6**	**0.60**	**1.00**	0.913	**6**	**0.60**	**1.00**	0.875	5	0.50	0.83	0.938	**6**	**0.60**	**1.00**	**1.000**
15	**6**	**0.40**	**1.00**	0.913	**6**	**0.40**	**1.00**	0.875	5	0.33	0.83	0.938	**6**	**0.40**	**1.00**	**1.000**
20	**6**	**0.30**	**1.00**	0.913	**6**	**0.30**	**1.00**	0.875	5	0.25	0.83	0.938	**6**	**0.30**	**1.00**	**1.000**
30	**6**	**0.20**	**1.00**	0.913	**6**	**0.20**	**1.00**	0.875	**6**	**0.20**	**1.00**	0.568	**6**	**0.20**	**1.00**	**1.000**

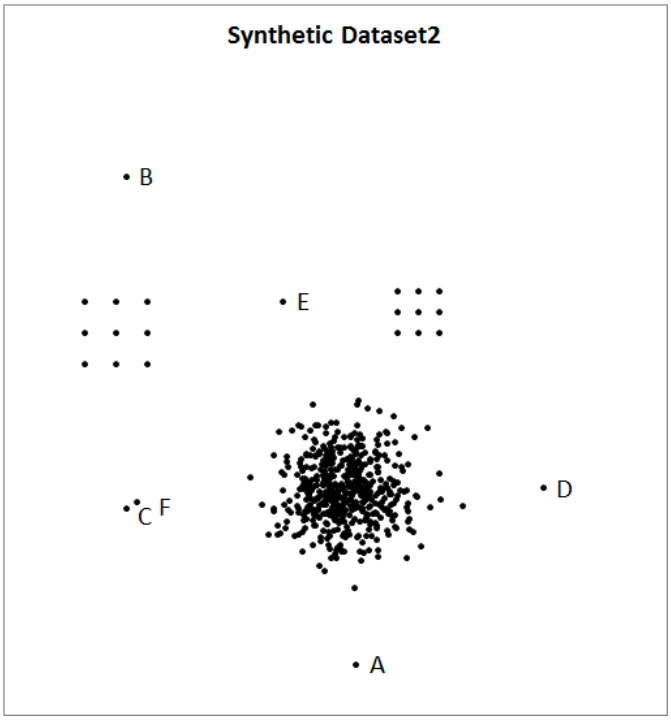

Figure 4: A Synthetic Data set with one cluster obtained using the Gaussian distribution and other clusters by placing all points uniformly.

4.3 Real Datasets:

We have used three well known datasets, namely the Iris, Ionosphere, and Wisconsin breast cancer datasets. We use two ways to evaluate the effectiveness and accuracy of outlier detection algorithms; (i) detect rare classes within the datasets (which has also been used by other researchers such as Feng *et al.*, Orlowska, and Tang *et al.* [**?**, 13, 18]) and (ii) plant outliers into the real datasets (according to datasets' domain knowledge) and expect outlier detection algorithms to identify them.

4.4 Real Datasets with Rare Classes

In this sub-section, we compare the algorithms in detecting rare classes. A class is made 'rare' by removing most of its ovservations. In general, the value of k is chosen between five to ten percentage of the size of the dataset.

4.4.1 Iris Dataset

The dataset is about iris plant and contains three classes: iris setosa, iris versicolour, iris virginica with 50 instances each. The iris setosa class is linearly separable from the other two classes, but the other two classes are not linearly separable from each other. We randomly remove 45 instances from iris-setosa class to make it 'rare'; remaining 105 instances are used in the final dataset. Three selected values of k are 5, 7, 10. Tables 3 summarize our findings.

Table 3: Comparison of LOF, COF, INFLO and RBDA for k = 5, 7 and 10 respectively for the Iris dataset. Maximum values are marked as bold.

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	**2**	**0.200**	**0.4**	**0.150**
15	**5**	**0.333**	**1**	**0.211**	0	0	0	0	1	0.067	0.2	0.067	2	0.133	0.4	0.150
20	**5**	**0.250**	**1**	**0.211**	0	0	0	0	2	0.100	0.4	0.091	**5**	**0.250**	**1**	0.208

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	**4**	**0.800**	**0.8**	0.714	0	0	0	0	**4**	**0.800**	**0.8**	0.714	**4**	**0.800**	**0.8**	**0.769**
10	**5**	**0.500**	**1**	0.750	0	0	0	0	**5**	**0.500**	**1**	0.710	**5**	**0.500**	**1**	**0.790**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	**5**	**1**	**1**	**1**	**5**	**1**	**1**	0.833	**5**	**1**	**1**	**1**	**5**	**1**	**1**	**1**
10	**5**	**0.500**	**1**	**1**	**5**	**0.500**	**1**	0.833	**5**	**0.500**	**1**	**1**	**5**	**0.500**	**1**	**1**

We observe that for $k = 5$ and $m = 15$, LOF has the highest precision value and RBDA comes second. For $m = 20$, LOF and RBDA both have the same performance; COF performs poorly (its precision and recall values are all zero). The reason for COF's poor performance is that instances of rare class are close to each other which decrease average-chaining distance of COF algorithm significantly and thus decrease it's outlierness.

For $k = 7$, RBDA performs better than LOF, INFLO and COF for all values of m. In particular, COF doesn't find any outlier within top 10 ranked instances.

For $k = 10$ and all values of m, LOF, INFLO and RBDA perform well. In general, performance of RBDA algorithm is better than LOF, COF and INFLO algorithms for other values of k. When k increases, precision, recall and Rank-Power of all algorithms improve.

4.4.2 Johns Hopkins University Ionosphere Dataset

The Johns Hopkins University Ionosphere dataset contains 351 instances with 34 attributes; all attributes are normalized in the range of 0 and 1. There are two classes labeled as good and bad with 225 and 126 instances respectively. There is no duplicate instances in the dataset. To form the rare class, 116 instances from the bad class are randomly removed. Final dataset has only 235 instances with 225 good and 10 bad instances. Four values of $k = 8, 11, 13$ and 15 are used and for different value of k the m values also vary. Results are presented in Table 4.

We observe that, for $k = 8$, RBDA has the best Rank-Power among all algorithms, but LOF algorithm achieves the best precision and recall for $m > 85$; and it is also the only algorithm that finds all ten 'bad' class instances. RBDA performs better than COF and INFLO algorithms for all values of m.

For $k = 11$ and $m = 65$, LOF has higher precision (0.15) than RBDA algorithms (0.14). For other values of m, RBDA is the winner and has largest values of precision and recall. COF algorithm has the highest Rank-Power when $m = 85$ but it finds only 8 'bad' class instances instead of 10 found by LOF and RBDA algorithms. For $k = 15$, situation is very similar to previous case. For m from 5 to 85, RBDA consistently does well in precision and recall, but it doesnt achieve the best Rank-Power for all m. When m is 65 or 85, COF algorithm shows the highest Rank-Power and RBDA comes second.

In general, RBDA algorithm shows the best performance compared with other algorithms.

4.4.3 Wisconsin Diagnostic Breast Cancer Dataset

Wisconsin diagnostic breast cancer dataset contains 699 instances with 9 attributes. There are many duplicate instances and instances with missing attribute values. After removing all duplicate and instances with missing attribute values, 236 instances labeled as benign class and 236 instances as malignant were left. Following the method proposed by Cao [17], 226 malignant instances are randomly removed. In our experiments the final dataset consisted 213 benign instances and 10 malignant instances.

Results in Tables 5 clearly shows that RBDA consistently performs better than the other algorithms and for all values of k and m RBDA achieves the best precision, recall and Rank-Power. In addition, RBDA algorithm is the only algorithm that detects all ten rare class instances for all five k values.

4.5 Real Datasets with Planted Outliers

Detecting rare class instances may not be adequate to measure performance of an algorithm designed to find outliers; because it may not be appropriate to declare them as outliers. In experiments described in this subsection we plant some outliers into the real datasets according to datasetss domain knowledge.

Table 4: Comparison of LOF, COF, INFLO and RBDA for k = 8, 11 and 15 respectively for the Ionosphere dataset. Maximum values are marked as bold.

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	3	0.600	0.3	0.600	3	0.600	0.3	0.857	3	0.600	0.3	0.600	**5**	**1**	**0.5**	**1**
15	6	0.400	0.6	0.570	6	0.400	0.6	0.600	6	0.400	0.6	0.620	**7**	**0.467**	**0.7**	**0.970**
25	7	0.280	0.7	0.483	7	0.280	0.7	0.549	7	0.280	0.7	0.519	**8**	**0.320**	**0.8**	**0.783**
45	**8**	**0.178**	**0.8**	0.38	**8**	**0.178**	**0.8**	0.440	**8**	**0.178**	**0.8**	0.400	**8**	**0.178**	**0.8**	**0.783**
65	**8**	**0.123**	**0.8**	0.379	**8**	**0.123**	**0.8**	0.440	**8**	**0.123**	**0.8**	0.396	**8**	**0.123**	**0.8**	**0.783**
85	**9**	**0.106**	**0.9**	0.253	8	0.094	0.8	**0.440**	**9**	**0.106**	**0.9**	0.273	**9**	**0.106**	**0.9**	0.391
105	**10**	**0.095**	**1**	0.196	8	0.076	0.8	**0.440**	9	0.086	0.9	0.273	9	0.086	0.9	0.391
130	**10**	**0.077**	**1**	0.196	9	0.069	0.9	0.238	9	0.069	0.9	0.273	9	0.069	0.9	**0.391**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	3	0.600	0.3	0.600	3	0.600	0.3	1	3	0.600	0.3	0.600	**5**	**1**	**0.5**	**1**
15	6	0.400	0.6	0.570	6	0.400	0.6	0.620	**7**	**0.467**	**0.7**	0.600	**7**	**0.467**	**0.7**	**1**
25	7	0.280	0.7	0.528	7	0.280	0.7	0.549	7	0.280	0.7	0.600	**8**	**0.320**	**0.8**	**0.818**
45	8	0.178	0.8	0.420	8	0.178	0.8	0.440	8	0.178	0.8	0.470	**9**	**0.200**	**0.9**	**0.510**
65	**10**	**0.154**	**1**	0.284	8	0.123	0.8	0.440	9	0.138	0.9	0.344	9	0.138	0.9	**0.506**
85	**10**	**0.118**	**1**	0.284	8	0.094	0.8	0.440*	9	0.106	0.9	0.344	**10**	**0.118**	**1**	**0.353**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	4	0.800	0.4	0.769	3	0.600	0.3	1	4	0.800	0.4	0.769	**5**	**1**	**0.5**	**1**
15	7	0.467	0.7	0.651	7	0.467	0.7	0.560	7	0.467	0.7	0.670	**8**	**0.533**	**0.8**	**0.920**
25	7	0.280	0.7	0.651	7	0.280	0.7	0.560	**8**	**0.320**	**0.8**	0.537	**8**	**0.320**	**0.8**	**0.923**
45	**9**	**0.200**	**0.9**	0.430	8	0.178	0.8	0.460	**9**	**0.200**	**0.9**	0.410	**9**	**0.200**	**0.9**	**0.692**
65	**10**	**0.154**	**1**	0.350	8	0.123	0.8	**0.456**	9	0.138	0.9	0.410	**10**	**0.154**	**1**	0.430
85	**10**	**0.118**	**1**	0.350	8	0.094	0.8	**0.456**	**10**	**0.118**	**1**	0.304	**10**	**0.118**	**1**	0.430

Table 5: Comparison of LOF, COF, INFLO and RBDA for k = 8, 11, 13, 15, and 20 respectively for the Wisconsin Breast Cancer data. Maximum values are marked as bold.

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
15	1	0.067	0.1	0.091	0	0	0	0	2	0.133	0.2	0.13	**3**	**0.2**	**0.3**	**0.316**
25	3	0.120	0.3	0.125	1	0.04	0.1	0.059	3	0.120	0.3	0.130	**4**	**0.160**	**0.4**	**0.256**
40	5	0.125	0.5	0.125	3	0.075	0.3	0.07	5	0.125	0.5	0.128	**8**	**0.200**	**0.8**	**0.198**
60	5	0.083	0.5	0.123	5	0.083	0.5	0.083	8	0.133	0.8	0.132	**10**	**0.167**	**1**	**0.190**
80	9	0.113	0.9	0.118	**10**	**0.125**	**1**	0.107	8	0.100	0.8	0.132	**10**	**0.125**	**1**	**0.190**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
15	2	0.133	0.2	0.115	0	0	0	0	2	0.133	0.2	0.120	**4**	**0.267**	**0.4**	**0.313**
25	3	0.120	0.3	0.130	2	0.080	0.2	0.073	3	0.12	0.3	0.125	**5**	**0.200**	**0.5**	**0.263**
40	5	0.125	0.5	0.132	4	0.100	0.4	0.093	5	0.125	0.5	0.144	**8**	**0.200**	**0.8**	**0.228**
60	7	0.117	0.7	0.130	7	0.117	0.7	0.111	7	0.117	0.7	0.138	**10**	**0.167**	**1**	**0.211**
80	**10**	**0.125**	**1**	0.135	**10**	**0.125**	**1**	0.122	8	0.1	0.8	0.136	**10**	**0.125**	**1**	**0.211**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
15	1	0.067	0.1	0.091	1	0.067	0.1	0.067	1	0.067	0.1	0.077	**3**	**0.200**	**0.3**	**0.400**
25	3	0.120	0.3	0.115	3	0.120	0.3	0.100	3	0.120	0.3	0.120	**4**	**0.160**	**0.4**	**0.323**
40	5	0.125	0.5	0.118	5	0.125	0.5	0.120	6	0.150	0.6	0.139	**7**	**0.175**	**0.7**	**0.228**
60	9	0.150	0.9	0.138	10	0.167	1	0.141	7	0.117	0.7	0.142	**10**	**0.167**	**1**	**0.212**
80	**10**	**0.125**	**1**	0.141	**10**	**0.125**	**1**	0.141	9	0.113	0.9	0.134	**10**	**0.125**	**1**	**0.212**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
15	1	0.067	0.1	0.091	0	0	0	0	2	0.133	0.2	0.103	**4**	**0.267**	**0.4**	**0.370**
25	3	0.120	0.3	0.113	4	0.160	0.4	0.128	3	0.120	0.3	0.115	**5**	**0.200**	**0.5**	**0.319**
40	5	0.125	0.5	0.118	6	0.150	0.6	0.146	7	0.175	0.7	0.151	**9**	**0.225**	**0.9**	**0.238**
60	9	0.150	0.9	0.145	**10**	**0.167**	**1**	0.162	8	0.133	0.8	0.157	**10**	**0.167**	**1**	**0.225**
80	**10**	**0.125**	**1**	0.148	**10**	**0.125**	**1**	0.162	8	0.100	0.8	0.157	**10**	**0.125**	**1**	**0.225**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
15	3	0.200	0.3	0.182	3	0.200	0.3	0.200	2	0.133	0.2	0.143	**5**	**0.333**	**0.5**	**0.385**
25	3	0.120	0.3	0.182	6	0.24	0.6	0.210	5	0.200	0.5	0.161	**6**	**0.240**	**0.6**	**0.328**
40	8	0.200	0.8	0.176	8	0.200	0.8	0.220	8	0.200	0.8	0.205	**9**	**0.225**	**0.9**	**0.281**
60	**10**	**0.167**	**1**	0.182	**10**	**0.167**	**1**	0.22	9	0.150	0.9	0.191	**10**	**0.167**	**1**	**0.263**
80	**10**	**0.125**	**1**	0.182	**10**	**0.125**	**1**	0.22	9	0.113	0.9	0.191	**10**	**0.125**	**1**	**0.263**

4.5.1 IRIS with Outliers

We insert three outliers into IRIS dataset, that is, there are three classes with 50 instances each and 3 planted outliers. The first outlier has maximum attribute values, second outlier has minimum attribute values, and the third has two attributes with maximum values and the other two with minimum values. Table 6 contains the results for this setting.

Table 6: Comparison of LOF, COF, INFLO and RBDA for k = 10 and 15, respectively, for the Iris data with planted anomalies. Maximum values are marked as bold.

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
10	**3**	**0.30**	**1**	0.857	**3**	**0.30**	**1**	0.545	**3**	**0.30**	**1**	0.857	**3**	**0.30**	**1**	**1**
15	**3**	**0.20**	**1**	0.857	**3**	**0.20**	**1**	0.545	**3**	**0.20**	**1**	0.857	**3**	**0.20**	**1**	**1**
20	**3**	**0.15**	**1**	0.857	**3**	**0.15**	**1**	0.545	**3**	**0.15**	**1**	0.857	**3**	**0.15**	**1**	**1**
25	**3**	**0.12**	**1**	0.857	**3**	**0.12**	**1**	0.545	**3**	**0.12**	**1**	0.857	**3**	**0.12**	**1**	**1**
30	**3**	**0.10**	**1**	0.857	**3**	**0.10**	**1**	0.545	**3**	**0.10**	**1**	0.857	**3**	**0.10**	**1**	**1**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
5	**3**	**0.60**	**1**	0.857	2	0.4	0.667	**1**	**3**	**0.60**	**1**	**1**	**3**	**0.60**	**1**	**1**
10	**3**	**0.30**	**1**	0.857	**3**	**0.30**	**1**	0.5	**3**	**0.30**	**1**	**1**	**3**	**0.30**	**1**	**1**
15	**3**	**0.20**	**1**	0.857	**3**	**0.20**	**1**	0.5	**3**	**0.20**	**1**	**1**	**3**	**0.20**	**1**	**1**
20	**3**	**0.15**	**1**	0.857	**3**	**0.15**	**1**	0.5	**3**	**0.15**	**1**	**1**	**3**	**0.15**	**1**	**1**
25	**3**	**0.12**	**1**	0.857	**3**	**0.12**	**1**	0.5	**3**	**0.12**	**1**	**1**	**3**	**0.12**	**1**	**1**

For $k = 10$ and $m = 10$, all algorithms found the three outliers, but their Rank-Powers are different. Rank-Power of RBDA is better than all other three algorithms; COF has the lowest value. Only RBDA algorithm ranks three outliers in top 3 positions while no other algorithm do the same.

For $k = 15$, INFLO and RBDA are the best because they all rank three outliers in top three positions which are exactly the expected results that outlier detection algorithms are designed to do. COF has the worst Rank-Power.

4.5.2 Johns Hopkins University Ionosphere Dataset with Outliers

For ionosphere dataset, two classes labeled as good and bad with 225 and 126 instances respectively are kept in resulting dataset. Three outliers are inserted into the dataset; first two outliers have maximum or minimum value in every attribute, and the third has 9 attributes with unexpected

values and 25 attributes with maximum or minimum values. Unexpected value here is the value that is valid between minimum and maximum number but it actually never observed in real datasets. For example, one attribute may have a scope from 0 to 100, but value of 12 never appears in real dataset actually. Since the size of resulting dataset is 354, $k = 15, 20, 25$ were selected for the experiments.

Table 7: Comparison of LOF, COF, INFLO and RBDA for $k = 15, 20$, and 25, respectively, for the Ionosphere data with planted anomalies. Maximum values are marked as bold.

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	1	0.050	0.333	0.091	1	0.05	0.333	0.050	**3**	**0.150**	**1**	**0.143**
30	1	0.033	0.333	0.042	2	0.067	0.667	0.079	2	0.067	0.667	0.064	**3**	**0.100**	**1**	**0.143**
40	**3**	**0.075**	**1**	0.063	**3**	**0.075**	**1**	0.080	**3**	**0.075**	**1**	0.074	**3**	**0.075**	**1**	**0.143**
50	**3**	**0.060**	**1**	0.063	**3**	**0.060**	**1**	0.080	**3**	**0.060**	**1**	0.074	**3**	**0.060**	**1**	**0.143**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
10	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**0.100**	**0.333**	**0.125**
20	0	0	0	0	1	0.050	0.333	0.063	0	0	0	0	**2**	**0.100**	**0.667**	**0.158**
30	1	0.033	0.333	0.034	2	0.067	0.667	0.079	1	0.033	0.333	0.040	**3**	**0.100**	**1**	**0.136**
40	2	0.050	0.667	0.048	2	0.050	0.667	0.079	2	0.050	0.667	0.052	**3**	**0.075**	**1**	**0.136**
50	**3**	**0.060**	**1**	0.056	**3**	**0.060**	**1**	0.071	**3**	**0.060**	**1**	0.060	**3**	**0.060**	**1**	**0.136**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
10	0	0	0	0	0	0	0	0	0	0	0	0	**2**	**0.200**	**0.667**	**0.3**
20	0	0	0	0	0	0	0	0	0	0	0	0	**2**	**0.100**	**0.667**	**0.3**
30	2	0.067	0.667	0.051	2	0.067	0.667	0.060	2	0.067	0.667	0.061	**3**	**0.100**	**1**	**0.182**
40	2	0.050	0.667	0.051	2	0.050	0.667	0.060	**3**	**0.075**	**1**	0.067	**3**	**0.075**	**1**	**0.182**
50	**3**	**0.060**	**1**	0.058	2	0.040	0.667	0.060	**3**	**0.060**	**1**	0.067	**3**	**0.060**	**1**	**0.182**

In Table 7 we observe that, for $k = 15$, and $m = 10$ no algorithm can detect planted outliers. RBDA algorithm is the only algorithm that detects all three outliers for $m = 20$. When $m = 40$ or 50, all algorithms find all outliers but RBDA algorithm has the best Rank-Power value and LOF algorithm has the worst Rank-Power.

For $k = 20$, 25, RBDA is the only algorithm that detects outliers within top 10 ranked objects

and it also has the best Rank-Power for any given m; LOF algorithm is always the worst.

The gap of performance between RBDA and other algorithms is large since RBDAs RankPower is almost double as those of other algorithms for every k and m. Overall performance of RBDA is the best.

4.5.3 Wisconsin Diagnostic Breast Cancer with Outliers

After removal of duplicated instances and instances with missing attribute values, only 449 instances were left with 213 instances labeled as benign and 236 as malignant. Two outliers are planted into dataset. Both outliers have maximum or minimum values for all attributes.

Table 8: Comparison of LOF, COF, INFLO and RBDA for k = 15, 20, and 30, respectively, for the Wisconsin Breast data with planted anomalies. Maximum values are marked as bold.

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
10	**2**	**0.200**	**1**	0.500	0	0	0	0	**2**	**0.200**	**1**	**1**	**2**	**0.200**	**1**	**1**
20	**2**	**0.100**	**1**	0.500	0	0	0	0	**2**	**0.100**	**1**	**1**	**2**	**0.100**	**1**	**1**
30	**2**	**0.067**	**1**	0.500	0	0	0	0	**2**	**0.067**	**1**	**1**	**2**	**0.067**	**1**	**1**
40	**2**	**0.050**	**1**	0.500	**2**	**0.050**	**1**	0.038	**2**	**0.050**	**1**	**1**	**2**	**0.050**	**1**	**1**
50	**2**	**0.040**	**1**	0.500	**2**	**0.040**	**1**	0.038	**2**	**0.040**	**1**	**1**	**2**	**0.040**	**1**	**1**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
10	**2**	**0.200**	**1**	0.750	0	0	0	0	**2**	**0.200**	**1**	**1**	**2**	**0.200**	**1**	**1**
20	**2**	**0.100**	**1**	0.750	1	0.050	0.500	0.063	**2**	**0.100**	**1**	**1**	**2**	**0.100**	**1**	**1**
30	**2**	**0.067**	**1**	0.750	1	0.033	0.500	0.063	**2**	**0.067**	**1**	**1**	**2**	**0.067**	**1**	**1**
40	**2**	**0.050**	**1**	0.750	1	0.025	0.500	0.063	**2**	**0.050**	**1**	**1**	**2**	**0.050**	**1**	**1**
50	**2**	**0.040**	**1**	0.750	**2**	**0.040**	**1**	0.051	**2**	**0.040**	**1**	**1**	**2**	**0.040**	**1**	**1**

m	LOF				COF				INFLO				RBDA			
	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP	Nrc	Pr	Re	RP
10	**2**	**0.200**	**1**	**1**	1	0.100	0.500	0.200	**2**	**0.200**	**1**	**1**	**2**	**0.200**	**1**	**1**
20	**2**	**0.100**	**1**	**1**	1	0.050	0.500	0.200	**2**	**0.100**	**1**	**1**	**2**	**0.100**	**1**	**1**
30	**2**	**0.067**	**1**	**1**	**2**	**0.067**	**1**	0.094	**2**	**0.067**	**1**	**1**	**2**	**0.067**	**1**	**1**
40	**2**	**0.050**	**1**	**1**	**2**	**0.050**	**1**	0.094	**2**	**0.050**	**1**	**1**	**2**	**0.050**	**1**	**1**
50	**2**	**0.040**	**1**	**1**	**2**	**0.040**	**1**	0.094	**2**	**0.040**	**1**	**1**	**2**	**0.040**	**1**	**1**

In Table 8, for $k = 15$, our algorithm and INFLO algorithm both perform well and rank two

outliers in top 2 positions. LOF algorithm finds the outliers within top 10 instances of its output but doesnt rank the outliers in first two positions. COF algorithm even couldnt find the outliers until top 40 instances, so that its RankPower is very bad compared with others. For $k = 20$, RBDA and INFLO algorithms still perform better than others. With large value of k, both LOF and COF get better result with maximum RankPower, and COF still is the algorithm with the smallest RankPower. When k increases to 30, all algorithms except COF achieve very good results, and find the three outliers in first three positions. COF still looks not good in this experiment.

In general speaking, when k is increasing, most of algorithms can improve their performances. One reason is that when k is larger, more neighbors around a specific instance are involved into the process of evaluating that instance, so that the algorithm holds more information to make an accurate decision about outlier.

In this experiment, RBDA and INFLO algorithms show the same performance for every selected k, and COF algorithm performs badly compared with others especially for its poor RankPower.

5 Conclusion

Outlier detection is an important task for data mining applications. Existing algorithms are effective and have been successfully applied in many real-world applications. But these algorithms, especially density-based algorithms, have low efficiency in datasets with different densities or when datasets consist of clusters with special shapes. In this paper, we introduce a new idea, ranking, to measure an object's outlierness. Sum of ranks of an object is naturally meaningful to measure the degree of isolation of an object. Based on this idea, we propose the Rank-based outlier Detection Algorithm (RBDA) that is effective to solve the problems mentioned above for many situations.

There are two directions for future work. The first one is to improve the performance of RBDA in datasets consisting of clusters with special shapes such as lines or circles. Currently, RBDA doesn't perform as good as COF for this kind of datasets. The second is to further improve the effectiveness of ranking.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, p. ARTICLE 15, July 2009.

[2] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Inlier-based outlier detection via direct density ratio estimation," *In Proceedings of the 2008 Eighth IEEE international conference on data mining, Washington DC, USA*, 2008.

[3] M. E. Otey, A. Ghoting, and S. Parthasarathy, "Fast distributed outlier detection in mixed-attribute data sets.," *Data Mining Knowledge Discovery*, vol. 12, pp. 203–228, 2006.

[4] S. Guha, R. Rastogi, and K. Shim, "An efficient clustering algorithm for large databases," *In Proceedings of the 1998 ACM SIGMOD international conference on management of data, Seattle, Washington, USA*, pp. 73–84, 1998.

[5] D. Yu, G. Sheikholeslami, and A. Zhang, "Findout: Finding out outliers in large datasets," *Knowledge and Information Systems*, vol. 4, no. 4, pp. 387–412, 2002.

[6] E. M. Knorr and R. T. Ng, "Algorithms for mining distanced-based outliers in large datasets.," *In Proceedings of the 24th International Conference on Very Large Data Bases.*, 1998.

[7] A. Fabrizio and P. Clara, "Outlier mining in large high-dimensional data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 203–215, 2005.

[8] A. Fabrizio, B. Stefano, and P. Clara, "Distance-based detection and prediction of outliers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 2, pp. 145–160, 2006.

[9] Y. Zhang, S. Yang, and Y. Wang, "Ldbod: A novel local distribution based outlier detector," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 967–976, 2008.

[10] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *In Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM Press*, pp. 93–104, 2000.

[11] J. Tang, Z. Chen, A. W. chee Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," *In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 535–548, 2002.

[12] W. Jin, A. K. H. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 577–593, 2006.

[13] J. Tang, Z. Chen, A. W. Fu, and D. W. Cheung, "Capabilities of outlier detection schemes in large datasets, framework and methodologies.," *Knowledge and Information Systems*, vol. 11, no. 1, pp. 45–84, 2006.

[14] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval.* Addison-Wesley Longman Publishing Co., Inc. Boston, 1999.

[15] X. Meng and Z. Chen, "On user-oriented measurements of effectiveness of web information retrieval systems," *In Proceeding of the 2004 international conference on internet computing.*, pp. 527–533, 2004.

[16] G. Salton, *Automated text processing: The transformation, analysis, and retrieval of information by computer.* Addison-Wesley Longman Publishing Co., Inc. Boston, 1998.

[17] H. Cao, G. Si, Y. Zhang, and L. Jia, "Enhancing effectiveness of density-based outlier mining scheme with density-similarity-neighbor-based outlier factor," *Expert Systems with Applications: An International Journal*, vol. 37, December 2010.

[18] J. Feng, Y. Sui, and C. Cao, "Some issues about outlier detection in rough set theory," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4680–4687, 2009.