Unmasking Outliers in Large Distributed Databases Using Cluster Based Approach: CluBSOLD

A. Rama Satish[1], P. Bala Krishna Prasad[2], D. Naga Raju[3], Ravi Kumar Saidala[4]

[1] Department of Computer Science & Engg., DVR & Dr HS MIC College of Technology, Kanchickacherla, Krishna District, Andhra Pradesh-521180, India. Ph: 08678-273535 ext: 252, Email: ramsatpm@gmail.com

[2] Principal, Eluru College of Engineering & Technology, Eluru, West Godavari District, Andhra Pradesh-534004, India. Ph: 08812-215655 Email: bk_perali@yahoo.com

[3] Department of Information Technology, LBR College of Engineering, Mylavaram, Krishna District, Andhra Pradesh, India, 521230. Ph: 94411 76913 Email: dnagaraj_dnr@yahoo.co.in

[4] Research Scholar, Department of CSE, Acharya Nagarjuna University, Nagarjuna Nagar, Guntur District, Andhra Pradesh, 522510, Email: saidalaravikumar@gmail.com

***Abstract* - Outliers are dissimilar or inconsistent data objects with respect to the remaining data objects in the data set or which are far away from their cluster centroids. Detecting outliers in data is a very important concept in Knowledge Data Discovery process for finding hidden knowledge. The task of detecting the outliers has been studied in a large number of research areas like Financial Data Analysis, Large Distributed Systems, Biological Data Analysis, Data Mining, Scientific Applications, Health monitoring, etc., Existing research study of outlier detection shows that Density Based outlier detection techniques are robust. Identifying outliers in a distributed environment is not a simple task because processing with a distributed database raises two major issues. First one is rendering massive data which are generated from different databases. And the second is data integration, which may cause data security violation and sensitive information leakage. Handling distributed database is a difficult task. In this paper, we present a cluster based outliers detection to spot outliers in large and vibrant (updated dynamically) distributed database in which cell density based centralized detection is used to succeed in dealing with massive data rendering problem and data integration problem. Experiments are conducted on various datasets and the obtained results clearly shows the robustness of the proposed technique for finding outliers in large distributed database.**

Keywords – Knowledge Discovery, Outliers detection, Clustering, Distributed databases

I. INTRODUCTION

Data Mining is a process of acquiring large amounts of data from different resources and summarizing them into potentially useful information. Data Mining also referred to as surfing or analysing or dredging of data, discovery or extraction of knowledge, pattern analysis, information harvesting, and business intelligence. Data Mining uses Artificial Intelligence(AI) techniques, Neural Networks(NN), and advanced statistical tools to reveal trends, patterns, and relationships in data[1].

Data Mining can be used to address many real world challenges in various domains such as Banking, Marketing, Insurance, Transportation, Health Care and Medicine. Data Mining enables to identify and analyse customer's purchasing activities such as frequency of purchase in a period of time, total monetary value of all purchases and when the last purchase was. The relative measure will be generated for individual customer after analysing the above all. Surfing data enables business persons to understand the hidden information inside historical purchasing transaction data to plan and launch new marketing campaigns in cost effective manner. Retail companies uses Data Mining for basket analysis to avail the information on customer purchased product combinations to promote their business and maximize the profit. Continuous increase of insurance industries, Data Mining techniques can help insurance industries to better predict claims of customers, forecasting customers information who can potentially purchase new policies and helps in detecting fraudulent behaviour of customers. Other Data Mining applications includes Bioinformatics, Security, Privacy Preserving, Web Mining, Eco system disturbances etcetra.

Tremendous algorithms and techniques in Data Mining like Clustering, Classification, Regression, Artificial Intelligence, Decision Trees, Association Rules, Neural Networks, Genetic Algorithms, k-Nearest Neighbour method etc., are used for extracting knowledge from databases. Clustering can be said as grouping of similar data objects that makes meaningful or useful clusters. By using clustering techniques we can further partition dense and sparse regions in the database, thus it is easy to discover the overall distribution pattern and correlations among data attributes. Clustering methods are categorized into Partitioning Methods, Hierarchical methods, Density based methods, Grid-based methods, Model-based methods.

1.1 Clustering

Data Mining deals with the problem of discovering hidden patterns from the data. Cluster analysis or clustering in data mining is the art of finding groups in data for the purpose of summarization or improved understanding. All the members in each set of the group are similar and dissimilar to the members of another group according to metrics [2], [3], [4]. A set of unlabelled samples are to be partitioned into similar groups where all the data objects in a group have similar (homogenous) characteristics defined in given sample space. The similarity of objects can be calculated based on some distance measure or similarity measures. Consider a dataset D featured by P attributes: $\mathrm{D} = [\mathrm{A}_1, \mathrm{A}_2, \dots, \mathrm{A}_\mathrm{p}]$ where $\mathrm{A}_1, \mathrm{A}_2, \dots, \mathrm{A}_\mathrm{p}$ are attributes. Assume D is partitioned into C_i clusters, where $i = 1$ to k, so that for each $\mathrm{D}_\mathrm{n} = [\mathrm{A}_1, \mathrm{A}_2, \dots, \mathrm{A}_\mathrm{p}]$, *n=1 to k* similarity measured by distance ***d*** then

$$d\left(D_j | C_i(A_1, A_2, \dots, A_p)\right) = \max\left\{d\left(D_j | C_l(A_1, A_2, \dots, A_p)\right) \middle| i \neq l\right\}$$

d is the distance between D_j and C_i. C_i Should meet the following conditions:

$$C_i \neq \emptyset \, , C_i \cap C_l \ = \emptyset \ and \ \bigcup_{i=1}^{k} C_i = D \ where \ i, l = 1 \dots k$$

Clustering is an unsupervised learning process, commonly and oftenly used in a variety of domains like scientific applications, medicine, business applications, ecological and biological studies and so on. Clustering will make user to understand the natural grouping in a dataset or the structure of the dataset. In general clustering is considered as an initial step in various data processing methods such as data classification, data compression, indexing, etc. [5].

With the usage of computer technology and developing new trends, computer applications are also increased. Every day, many applications generate too much voluminous data. It is necessary to acquire and store that large amount of data for knowledge discovery. In many cases, data is collected from multiple proprietary or distributed databases. Credit card transaction data are the best example for distributed database. Distributed database has been researched, modelled and developed to detect and prevent frauds occurred in credit card usage. Handling high dimensional data in a distributed environment is not a simple task. Extraction or acquiring of data from multiple proprietary or distributed database are usually confounds two common problems: High dimensionality and Outliers. Subspace algorithms are suitable for handling high dimensional data. Outlier detection in high dimensional data is not a simple task because outlierness is a sensitive information.

1.2 Outliers or Anamolies

Outlier detection is a task that discovers the dissimilar or inconsistent data objects with respect to the remaining data [6]. If cluster based technique is used in outlier detection outliers are generally far away from their cluster centroid. To find them several outlier detection algorithms are developed. Distance and Density based approaches are Proximity-based outlier detection approaches. In *Distance based Approach* [4], a point *k* in a dataset *D* which is considered as an outlier with respective to the parameters *M* and *d*, where *M* is a parameter which has the measure of the number of points and *d* is distance. If there are less than *M* points within the distance *d* from *k* then it is said to be as an outlier. The values of *M* and *d* are selected by the user and selection of these values are difficult. In *Density based Approach*[7], the density around data object is compared with the density around its local neighbouring data objects. The relative density of data object to compare its neighbours is computed as an outlier score. A *Distribution based Approach*[8],is a classical statistical approach. If the distribution is known then find discordance test for the distribution. In *Clustering based Approach*[3], consider clusters of smaller sizes as clustered outliers. The clusters in which the number of cluster members is less than the other then the cluster is considered as Clustered Outlier. Cluster-based approaches are optimal to find distinct clusters, and can be used to detect anomalous objects as by products. Clustered outliers are the data objects which have a large distance to the cluster center. In the clustering processes, anomalies can affect the locations of the cluster centers, even aggregating as a micro cluster. Clustering is an unsupervised algorithm so clustering techniques do not have to be supervised.Clustering based approaches are infeasible to complicated databases [9].

1.3 Distributed Environment

In a distributed environment database consists of two or more individual databases located at different sites on a computer network [10], [11]. For distributed data processing, it is necessary to consider observations in the whole database rather than in any individual distributed database. Figure 1 shows the distributed environment where all the systems are connected through a Computer Network and each system has storage and computational capability. In order to prevent information leakage and security violation centralized setting is used. Thus, one system acts as mediator to all the remaining systems. There is not any information sharing activity among the systems except mediator to other systems.

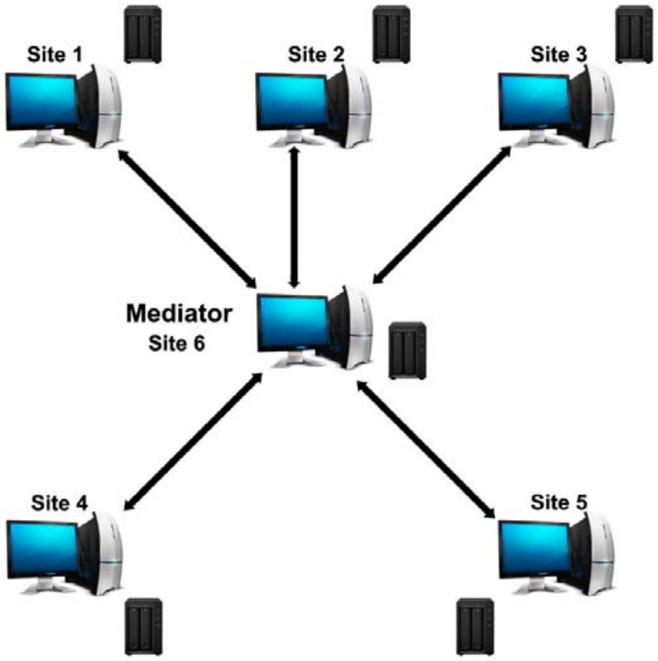

Figure 1 Distributed Environment

Credit card usage is the best example of distributed database. For spotting the suspicious or irregular credit card spending patterns in relation to the whole city, rather than a single community, we must get the distributed observations from all the transactions. Let us consider a simple example of credit card fraud detection with contextual as the location of purchase. The credit card holder usually does his transactions nearest ATMs to his hometown. Suppose the nearest ATMs are Laxmi Nagar, Greater Kailash, Dwaraka Sector, Indraprastha, which are located in New Delhi. A new transaction is done at new location *Thane* which is located in Mumbai will be considered a suspicious activity and consider as contextual outlier. Since, it does not conform to the normal behaviour of the individual in the context of Location. The above locations information will be available on individual database. Finding deviated observations in distributed community data centres is not a simple task. The most straightforward solution to outlier detection in distributed environment is applying centralized setting by integrating individual databases. It leads to two major problems: data security violation and leakage of sensual information. Abnormal data objects in individual database is called Local Outliers and in distributed databases is called Global Outliers. In general distributed databases are updated dynamically. K-medoid clustering algorithm is robust in the presence of noise and outliers or the other extreme values than a mean. In this algorithm it is not necessary to find mean value, so usage of the k-medoid is the best way to handle dynamic data.

Road Map Section 2 describes literature survey, section 3 provides proposed solution. Experimental results of conducted tests on selected databases are reported in section 4 followed by the conclusion in section 5.

II LITERATURE SURVEY

Outlier detection is a broad field having an attractive increasing attention in data mining, machine learning and statistics literature. An outlier is an deviating observation here it deviates so much from all other data objects that was generated by a different mechanism [6][12]. The main objective of outlier detection is uncovering that “different mechanism”. Outlier detection techniques have been studied with long history [9]. Outlier detection techniques can be broadly categorized in to distribution (statistical) - based techniques, clustering-based techniques, density-based techniques and model-based techniques according to [3],[7],[8] and [13].

All techniques mentioned above are developed for handling centralized databases. Knorr and Ng [14] proposed distance based technique for finding anomalies by defining parameters k and R. An object is said to be distance based outlier if and only if less than k objects in the input data set lie within distance R from it. Further extended by ramaswamy going through a new definition [15]: given k and n, an object o is an outlier if no more than $n-1$ other objects in the dataset have higher value for D_k than o, where $D_k(o)$ denotes the distance of the k^{th} nearest neighbour of o. This concept is further extended in [16],[17],[18], where each data point is ranked by the sum of distances from its k- Nearest Neighbours. In the case of clusters having different densities, distance based outliers are not suitable. To overcome the shortcoming of distance based outliers new concept of LOF is proposed by Breunig [7]. Authors Z. He, X. Xu, and S. Deng proposed a cluster-based local outlier detection method to identify the data points based on physical significance [19]. LOF is used to specify the outlier possibility of an object by considering cluster size and that the object belongs to and distance between that object and its closest cluster. This technique is not efficient because it contains high-complex notations.

Authors Ji Zhang, tao, wang proposed DISTORD (Distributed outlier detection) [20] for finding global outliers in large distributed database. They embedded centralized setting into DISTORD to overcome the problems of information leakage and data security violation and implemented optimization enhancement strategies and communication overhead to speed up the outlier detection process. Many research works states that the best solution for analysing distributed data is centralized mechanism. In this one system act as mediator and all systems have computational capability. Jiaogen Zhou et al proposed a distance - based approach for distributed database for finding deviating observations in dataset [21]. The Distance based approach is borrowed from Knorr, E.M., Ng, R.T [14][22]. In each database of distributed site, a data object in a given database of distributed site is identified as a local outlier if it satisfies the condition that is its neighbourhood in the radius of d contains less than $N_i(1 - \rho)$ data points, where d is the scope of the neighbourhood, N_i is the number of data objects at site $S_i$and ρ is a real-valued number that satisfies the condition 0 < ρ <1.The final global outliers are defined as data objects that have been identified as local outliers in database of every distributed site. It can only able to identify a fraction of global outliers that are existing and, quite likely, a significant portion ofglobal outliers will be missed out. This will be happen because the definition of the global outlier is problematic. For example, if a data object is not identified as an outlier object in database of every distributed site, then, this data object is not considered as a global outlier based on the definition of *global outlier* defined in [21]. However, the real situation is that a true global outlier is not necessarily an outlier in each of the distributed sites. This is the main drawback of Distance - based approaches for outlier detection proposed by Knorr, E.M., Ng, R.T. The computational cost of distance-based outlier metric used in their work is inefficient when dealing with large databases, as it will involve pairwise distance calculation for the data if no explicit indexing is built in advance it will make the approaches more complex.

Authors Liang, Han, Yang, and Zou Yan Jia proposed a kernel density estimation technique [23] for detecting outliers from distributed data streams. This approaches become complex because of the existence of kernel density estimation in the kernel density function. It involves calculations of integrals which will take large amount of time and the computation cost of finding kernel density function is very high. The following two are the major drawbacks in this approach, one is, the communication between distributed site and mediator become expensive. Second, it needs to be store and maintained the CF-tree based features. Authors [25] proposed anomaly detection approaches for distributed spatial data. Centralized anomaly detection method is used but it remained as an open question as their evaluation did not allow detailed inspection of individual detections. Inspection of individual detection in the proposed method may perform even better than outcome of the work reported. Kernel density estimation [23] is used in detecting outliers in sensor networks [24], [25], [26] and found the limitations mentioned above. Authors [27], proposed general - purpose and tunable distributed outlier detection algorithm that addresses outliers detection in dataset having mixed attributes, designing outlier detection approaches for a specific application that is sensor networks and the dataset being analysed may be streaming or otherwise dynamic data in nature.

III PROPOSED WORK

In this section, we present our proposed CluBSOLD (Clustering Based Spotting Outliers in Large Distributed database), for spotting top *n* (*n* is supplied by user) global outliers from large distributed databases. This section mainly contains the following: step wise description, diagrammatic representation and pseudo code of CluBSOLD.

3.1 Step Wise description of CluBSOLD

Step 1 *Clustering dataset in each individual database (for distributed sites)* The incoming data is gathered at specific time interval. Find the cluster index number and cluster density by applying *k-Medoid clustering* to the data of each site. The cluster index increases when a new cluster forms and cluster density increases when a new data object is added to the cluster. The main reason for determination of cluster index and cluster density is to make easier in finding local outliers.

k-Medoid Clustering Algorithm:

```
        Input:   Database of $D$ data objects, initial representative (medoid) objects
        Output:  Cluster index and cluster centroids
        repeat
                associate non medoid data object to the cluster with the nearest medoid
                randomly select non representative object
                compute total points S, if S<0, swap to form new set of $k$ medoid
    until it converges
```

Step 2 *Determining the global data summary (for mediator)* After the completion of step 1 each dataset is partitioned into clusters to determine the global data summary. Only the cluster index and cluster density information are first transmitted from each site to the mediator. This task prevents the data security violation. The mediator aggregates the density information of each cluster. i.e.,

$$\text{density}[cluster] \; = \sum_{s=1}^{T} \text{density}[cluster]^{s}$$

Clusters which are generated in step 1 will be called Global populated clusters if it satisfies the condition below

$$\text{density}[cluster] \; > global\_average\_density$$

where, $global\_average\_density = N/N_{populated\_clusters}$

N is the sum of the number of data objects in each database,

$N_{populated_clusters}$is the sum of number of populated clusters in each database that contain at least one data object.

Step 3 *Determining user-supplied top n local outliers (for distributed sites*) Calculate *k_ODF* (k Outlier Degree Factor) for each data object in each local database. i.e.,

$$k\_ODF(p) \; = \; \sum_{i=1}^{k} Dist(p,\ centroid(C))/k$$

Where, k is the nearest global_dense_cluster to data object p,

k_ODF is used to measure the strength of outlierness of each data point in each global populated cluster. After finding *k_ODF* sort the data objects according their *k_ODF* values. Generate top *n* Local outliers and transmit to the mediator.

Step 4 *Determining user-supplied top n global outliers (for mediator)* At mediator site generates the top *n* global outliers from collected top *n* local outliers by merging them and they are returned to the end user as requested top *n* global outliers.

3.2 Diagrammatic Representation of CluBSOLD

Figure 2 shows the proposed global outlier detection technique in a distributed environment. In a distributed environment, all the systems are connected through a computer network. Each system is having storage facility and computational capability.To ensure data security we prohibited the communication between any two systems but enabled one system as mediator.

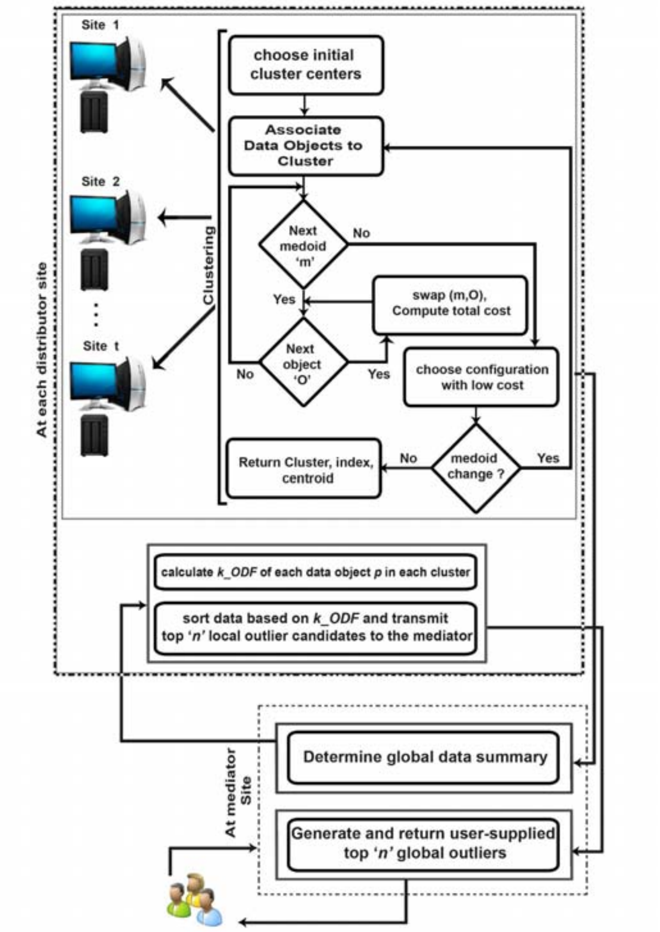
Site 1
Site 2
Site t
At each distributor site
Clustering
choose initial cluster centers
Associate Data Objects to Cluster
Next medoid 'm'
No
Yes
swap (m,O), Compute total cost
Next object 'O'
No
Yes
choose configuration with low cost
Return Cluster, index, centroid
No
medoid change ?
Yes
calculate k_ODF of each data object p in each cluster
sort data based on k_ODF and transmit top 'n' local outlier candidates to the mediator
At mediator Site
Determine global data summary
Generate and return user-supplied top 'n' global outliers

Figure 2 Diagramatic representation of CluBSOLD

Mediator system can access or send information to all remaining systems. That is clearly shown in fig 1. For each individual database applied *k*-Medoid clustering to partition the dataset into clusters. The main objective of CluBSOLD is detecting global outliers. Clustering individual database helps to determine global data summary. In the first step all the databases are clustered individually and obtained the cluster index and cluster centroids. These information is send to the mediator for determining global data summary. This was done at distributed sites. When mediator receives cluster index and centroids it calculates global average density of global database and densities of clusters in all distributed sites for finding global dense clusters. The global dense clusters has the possible global outlying objects. At mediator side find the centroid of global_dense_cluster and transmit it to each distribute site. Each site calculates the k_ODF of each data object by considering the centroid of global dense cluster. Sort the data objects according to their k_ODF values and transmit top *n* data objects which are called as *local outliers* to the mediator for finding global outliers. The mediator merges the received local outliers from all the sites. Sort and find the top *n* global outliers according to the k_ODF values. Finally mediator send the user-supplied top *n* global outliers to the user.

3.3 Notations

The following table shows the notations of the variables and functions.

TABLE 1 LIST OF VARIABLES AND FUNCTIONS

Variable/Function	Description
n	User-supplied number of outliers
N	N is the number of data points in the global database
t	Number of sites
p	Data object
$N_{populated\_cluster}$	The number of the global populated cluasters, here clusters contain at least one data point
global_average_ density	N/ $N_{populated\_cluster}$
global_dense_cluster	A global dense cluster's density is above or equal to the global average density level of all of the populate clusters
density [cluster]	Density of cluster
k_ODF(p)	Outlier Degree Factor of data point p
sort_fun	Sorting function
D_o	nxn all-pairs distance matrix
Randperm()	Function for randomly assigning centroids
kMedoids()	k-Medoid function

3.4 Pseudo Code for CluBSOLD

Below presented pseudo code for CluBSOLD shows the individual tasks which were done at distributed sites side and mediator side separately. Lines 1 and 2 are mediator side to recieve the user request for global outliers. Lines 3-10 represents transmitting cluster index and centroid to the mediator. Lines 12-23 represents generating global data summary for finding local outliers and this was done at mediator side. Lines 24-35 represents generating possible local outliers at mediator side and the remaining lines represents generating user-supplied top *n* global outliers at mediator side.

Algorithm 1 Pseudo code for CluBSOLD

Input	Distributed databases *D1*;*D2*; :::; *Dt* at distributed sites *S1*; *S2*; :::; *St*, respectively. Request for top '*n*' global outliers.
Output	Top 'n' global outliers.

```
disp('top n data objects identified as outliers')
n=str2double(input('input n','s'))
fors=1 :t do {for distributed sites begin}
    load data from site s
    apply k-medoid clustering
    find cluster_index and cluster_centroid
end for
fors=1:t do
    transmit density information to the mediator
end for {for distributor sites}
global_dense_cluster = []          {for mediator begin}
fors=1:tdo
    for each populated cluster [cluste_ index] do
        density [cluster] =∑density [cluster]$^s$
    end for
    $global\_average\_density = N/N_{populated\_cluster}$
    if density [cluster] >global_average_density then
        global_dense_cluster= global_dense_cluster∪cluster
        centroid (cluster_index)$^s$ = mean (cluster(cluster_index))
        transmit centroid of cluster to each site s
    end if
end for {for mediator end}
fors=1:tdo{for distributor sites begin}
    for each data object p∈[cluster_index] do
        temp = []
        forj=1:kdo
            distance = dist (p; centroid ([cluster_index]$^s$))
            temp = [temp; distance]
        end for
        k_ODF(p)=sum(temp)
    end for
    k_ODF(p)=sort_fun(k_ODF(p))
    transmit top n outlier from the sorting list from s to the mediator
end for {for distributor sites end}
    {finally mediator return top n outliers}
merge and generate the top n outliers
return the top n global outliers
```

Below presented code for K-Medoid shows how the clusters are determined. Lines 1 and 2 for for each distributed side selecting number of clusters. Lines 3-4 for randomly assigning centroid. Lines 6-23 for finding the suitable representative objects to the cluster and returns cluster index and centroid.

Algorithm 2 Pseudo code for k-medoid

Input D_o: nxn all-pairs distance matrix
k: number of clusters

Output cluster: *n* X 1 vector of assignments of each sample to a cluster id
cluster index: *k* X 1 vector of sample indices which make up the cluster centers.

```
disp('choose number of clusters k') {for distributor sites begin}
k =str2double(input('input k','s'))
rn = size($D_o$,1);
cluster_centroid= randperm(rn);          {randomly assigning centroids}
sort cluster_centroid
iter = 0
while1            {assign objects to clusters and update cluster objects and centroid}
        cluster  =$D_o$(cluster_centroid,:);
        [vals,cluster] = min(cluster,[],1);
        fori=1:k
                clusteri = find(cluster==i);
                [distfound(i),minind] = min(sum(Do(clusteri,clusteri),2));
                cluster_centroid(i) = clusteri(minind);
        for end
        distfound = nan(k,1);
        ifiter>0 &&distfound_next == distfound
                break;
        end if
        distfound = distfound_next;
        iter = iter+1;
while end
distfound = sum(distfound);
```

IV EXPERIMENAL RESULTS

We conducted all our experiments on a workstation with 3.0 GHz CPU and 4 GB RAM. We implemented all algorithms in MATLAB R2013a in Windows 7 Ultimate OS platform to construct distributed environment initially. We ran our implementation of CluBSOLD algorithms for finding global outliers in sites of the distributed environment. We experimented with real datasets from the UCI Machine Learning repository as well as synthetic datasets. We succeeded in generating synthetic dataset with the help of synthetic data generator software obtained at http://www.cs.umb.edu/~dana/GAClust/index.html. GAClust software enables the user to generate several datasets with selected number of data points and attributes. Our experiments focussed on testing execution time by varying the number of data points and a number of attributes. TABLE 2,3,4 presents the data objects along with their k_ODF values. In TABLE 2 we presented data objects along with their k_ODF values obtained at each distributed site side. At mediator side, it receives the data and k_ODF values and sorts them according to k_ODF values presented in TABLE 3. It is clearly shown that the data objects with high k_ODF values are highlighted. In TABLE 4 we listed the top 5 global outliers. We evaluated both algorithms, CDOD and CluBSOLD, based on two measures: Outlier Detection Accuracy rate - which is the number of outliers correctly identified, and False Positive rate - reflecting the number of normal points erroneously identified as outliers. We also noticed the running time performance of the CluBSOLD algorithm.

With the help of synthetic data generator, we created a dataset with 20 attributes and then varied the number of data points N from 50000 to 500000 with a step of 50000. For conducting experiments with the total number of attributes A_n, we created a dataset with 50000 data points and varied attributes A as 10, 20, 30, 40 and 50. Figure 3, Figure 4 and Figure 5 shows the execution time with a dataset, varying in number of data points, number of attributes and number of outliers respectively.

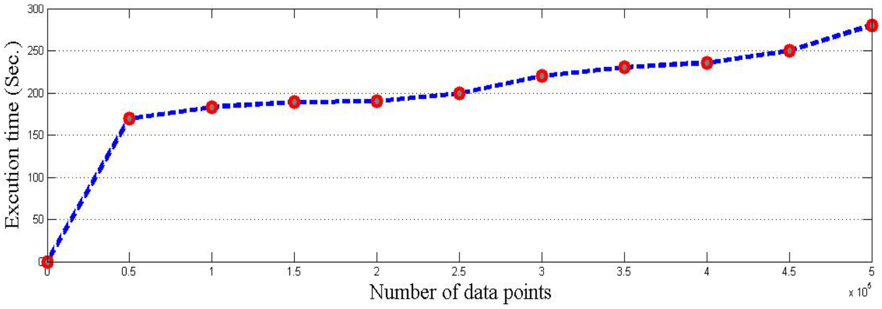

Figure 3 Varying Number of data points in Input

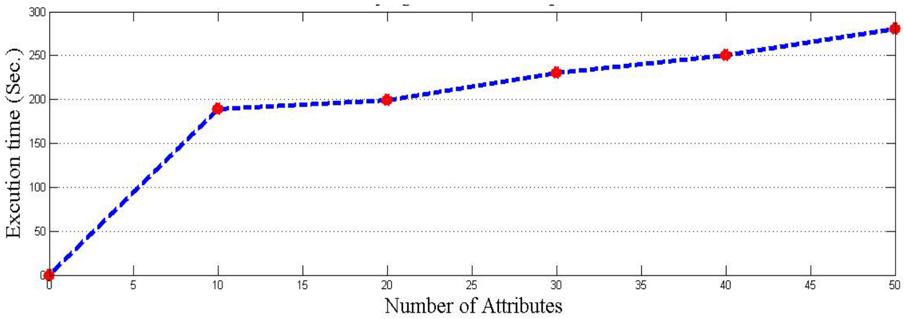

Figure 4 Varying Number of Attributes in Input

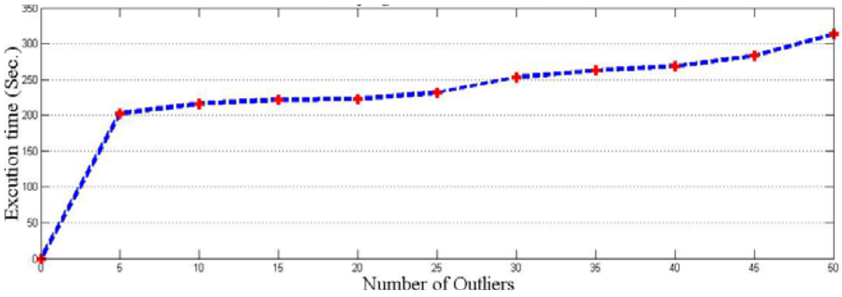

Figure 5 Varying Number of Outliers in Input

TABLE 2 DATA OBJECTS ALONG WITH THEIR K_ODF VALUES

Site 1				...	...	Site t			
A1	k_ODF	A2	k_ODF	...	...	A1	k_ODF	A2	k_ODF
8.6454	1.0496	0.1564	0.3042	...	...	7.9912	1.1013	0.9175	0.9302
8.5376	1.0299	0.1561	0.2999	...	...	8.2376	1.2268	0.1167	0.2999
8.6307	1.0342	0.1550	0.3013	...	...	8.2307	1.0946	0.1178	0.3013
8.6155	1.0899	0.2954	0.5013	...	...	8.2155	1.1638	0.1198	0.3013
8.6605	1.0348	0.1554	0.3005	...	...	8.1905	1.1388	0.1221	0.3005
8.6306	1.0103	0.1546	0.3017	...	...	8.1306	1.1241	0.1250	0.3017
8.6157	1.0215	0.1541	0.3015	...	...	8.1157	1.1095	0.1287	0.3015
8.6306	1.0222	0.1524	0.3005	...	...	8.5306	2.0146	0.1303	0.3005
8.5454	1.0212	0.1510	0.3019	...	...	8.1454	1.1474	0.1300	0.3019
8.0156	1.9979	0.1197	0.3001	...	...	8.1136	1.1249	0.1308	0.3021
...	...	...	...	...	...	...	...	...	...

TABLE 3 SORTED LIST OF TABLE 1 ACCORDING TO K_ODF VALUE

Site 1				...	...	Site t			
A1	**k_ODF**	**A2**	**k_ODF**	...	...	**A1**	**k_ODF**	**A2**	**k_ODF**
8.0156	1.9979	0.2954	0.5013	...	...	8.5306	2.0146	0.9175	0.9302
8.6155	1.0899	0.1564	0.3042	...	...	8.2376	1.2268	0.1309	0.3021
8.6454	1.0496	0.1511	0.3020	...	...	8.2155	1.1638	0.1300	0.3020
8.6605	1.0348	0.1546	0.3017	...	...	8.1454	1.1474	0.1251	0.3017
8.6307	1.0342	0.1541	0.3016	...	...	8.1905	1.1388	0.1287	0.3016
8.5376	1.0299	0.1550	0.3013	...	...	8.1136	1.1249	0.1178	0.3013
8.6306	1.0222	0.1525	0.3005	...	...	8.1306	1.1241	0.1198	0.3013
8.6157	1.0215	0.1555	0.3005	...	...	8.1157	1.1095	0.1221	0.3005
8.5454	1.0212	0.1198	0.3001	...	...	7.9912	1.1013	0.1303	0.3005
8.6306	1.0103	0.1562	0.2999	...	...	8.2307	1.0946	0.1167	0.2999
...	...	...	...	...	...	...	...	...	...

TABLE 4 TOP 5 GLOBAL OUTLIERS ALONG WITH K_ODF VALUES (FOR BETTER VIEW LISTED TOP 5 GLOBAL OUTLIERS)

Site 1				...	...	Site t			
A1	**k_ODF**	**A2**	**k_ODF**	...	...	**A1**	**k_ODF**	**A2**	**k_ODF**
8.0156	1.9979	0.2954	0.5013	...	...	8.5306	2.0146	0.9175	0.9302
8.6155	1.0899	0.1564	0.3042	...	...	8.2376	1.2268	0.1309	0.3021
8.6454	1.0496	0.1511	0.3020	...	...	8.2155	1.1638	0.1300	0.3020
8.6605	1.0348	0.1546	0.3017	...	...	8.1454	1.1474	0.1251	0.3017
8.6307	1.0342	0.1541	0.3016	...	...	8.1905	1.1388	0.1287	0.3016

V CONCLUSION

This research work is focussed on the problem of global outlier detection from large distributed databases. In a distributed environment, the data are generated at various sites which are in a distributed manner. These systems are always needed to be available locally for analyzing distributed data. Centralized setting enables the analysis of distributed data easy for generating global data summary. The proposed technique CluBSOLD is a cluster based outlier spotting technique in a large distributed database.*k-Medoid* clustering algorithm is used to partition the dataset of each site. *k-Medoid* is more robust, because it minimizes a sum of dissimilarities instead of a sum of squared euclidean distances. CluBSOLD is applied to several datasets. The obtained results show the performance of CluBSOLD. We noticed the changes in execution time by varying number of data points, number of attributes and number of outliers in input dataset.

VI REFERENCES

[1] J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2nded. Singapore: Elsevier, 2008.

[2] Ng, R., Han, J.: Efficient and effective clustering methods for spatial data mining. In: VLDB'94, pp. 144–155. Santiago, Chile (1994)

[3] S. Y. Jiang and Q. B. An, "Clustering-based outlier detectionmethod," in Proc. ICFSKD, Shandong, China, 2008, pp. 429–433.

[4] F. Angiulli and F. Fassetti, "Dolphin: An efficient algorithm for mining distance-based outliers in very large datasets," ACMTrans. Knowl. Discov. Data, vol. 3, no. 4, pp. 1–57, 2009.

[5] Cluster Analysis-Wikipedia. http://en.wikipedia.org/wiki/Cluster_analysis

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM CSUR, vol. 41, no. 3, Article 15, 2009.

[7] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in Proc. ACM SIGMOD Int. Conf. Manage. Data, New York, NY, USA, 2000,pp. 93–104.

[8] V. Barnett and T. Lewis, Outliers in Statistical Data. Chichester,U.K.: Wiley, 1994.

[9] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies,"Artif. Intell. Rev., vol. 22, no. 3, pp. 85–126, 2004.

[10] Chhabra, P., Scott, C., Kolaczyk, E.D., Crovella, M.: Distributed spatial anomaly detection. In: INFOCOM'08, pp. 1705–1713. Phoenix, AZ (2008)

[11] Otey, M., Ghoting, A., Parthasarathy, S.: Fast distributed outlier detection in mixed attributedata sets. Data Min. Knowl. Disc. 12(2), 203–228 (2006)

[12] C. C. Aggarwal, Outlier Analysis. New York, NY, USA: Springer, 2013

[13] C. Li and W. H. Wong, "Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection," in Proc. Natl. Acad. Sci. USA, 2001, pp. 31–36.

[14] Knorr, E. M., Ng, R. T. (1998). Algorithms for Mining Distance-Based Outliers in Large Datasets, Proc. 24th VLDB.

[15] Ramaswamy S., Rastogi R., Kyuseok S. (2000). Efficient Algorithms for Mining Outliers from Large Data Sets, Proc. ACM SIDMOD Int. Conf. on Management of Data.

[16] F. Angiulli, S. Basta, and C. Pizzuti. (2006). Distance-based detection and prediction of outliers. IEEE Transaction on Knowledge and Data Engineering, 18(2):145(160, February 2006).

[17] M. F. Jiang, S. S. Tseng, C. M. Su. (2001). Two-phase clustering process for outliers detection. Pattern Recognition Letters, 22(6/7): 691-700.

[18] Charu C. Aggarwal, Philip S. Yu.(2001). Outlier detection for high dimensional data, Proc. of the 2001 ACM SIGMOD int. conf. on Management of data, p.37-46, May 21-24 , Santa Barbara, California, United States.

[19] Z. He, X. Xu, and S. Deng, "Discovering Cluster-Based LocalOutliers," Pattern Recognition Letters, vol. 24, pp. 1641-1650, 2003.

[20] Ji Zhang, Xiaohui Tao, and Hua Wang. Outlier detection from large istributed databases. World Wide Web , 17(4):539–568, 2014

[21] Zhou, J. et al: A novel outlier detection algorithm for distributed databases. In: FSKD'05, pp. 293–297. Shangdong, China (2008)

[22] Knorr, E.M., Ng, R.T.: Finding intentional knowledge of distance-based outliers. In: VLDB'99, pp. 211–222. Edinburgh, Scotland (1999)

[23] Liang Su, Weihong Han, Peng Zou, Yan Jia: Continuous Kernel-Based Outlier Detection over Distributed Data Streams. ISPA Workshops 2007: 305-314

[24] Zhang, Yang, Nirvana Meratnia, and Paul Havinga. "Outlier detection techniques for wireless sensor networks: A survey." Communications Surveys & Tutorials, IEEE 12.2 (2010): 159-170.

[25] Chhabra, Pooja, et al. "Distributed spatial anomaly detection." INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, 2008.

[26] Gogoi, Prasanta, et al. "A survey of outlier detection methods in network anomaly identification." The Computer Journal (2011): bxr026.

[27] Koufakou, Anna, and Michael Georgiopoulos. "A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes." Data Mining and Knowledge Discovery 20.2 (2010): 259-289.

AUTHOR PROFILE

Rama Satish Aravapalli is an Associate Professor in the Department of Computer Science & Engineering, DVR & Dr HS MIC College of Technology, Kanchikacherla. He pursuing his Ph.D in the Computer Science department, Acharya Nagarjuna University. He has finished his M.Tech (CSE) from Jawaharlal Nehru University Kakinada in the year 2006. He completed his B.E (CSE) from University of Madras in the year 2001. He has 15 years of teaching experience. His research interests are Network Security, Data Mining, and Computer Networks.

Dr. P. Bala Krishna Prasad is the Principal of Eluru College of Engineering & Technology, Eluru. He obtained Doctoral Degree in Engineering, Ph.D (CSE) from Acharya Nagarjuna University, in the year 2007. He received his Master's Degree in Engineering, M.Tech (CSE) from Andhra University in the year 2000. He completed his Bachelor's Degree in Engineering, B.Tech (CSE) from IETE, New Delhi. He has a rich experience of 21 years which includes Teaching, Research and Administration. He has published several papers in National and International Journals of repute. He is a renowned author for Computer Science and Information Technology subjects and has authored six text books. His research interests are Computer Networks, Operating Systems, Image Processing, Data Mining, and Pattern Recognition.

Dr. D. Naga Raju is the HOD & Professor of IT Department in Lakireddy Balireddy College of Engineering, Mylavaram. He was awarded his Ph.D in Computer Science & Engineering from the Jawaharlal Nehru Technological University Hyderabad in the year 2014. He completed his master degree M.Tech (CSE) from Jawaharlal Nehru University(JNU), New Delhi in the year 2005. He completed his B.Tech (CSE) from Sri Venkateswara University, Tirupati in the year 2002. He has 16 years of teaching experience and published papers in various International journals, National and International conferences. His research interests are Data Mining, Soft Computing, Machine Learning and Pattern Recognition.

Ravi Kumar Saidala is a research scholar. He pursuing his Ph.D. in Acharya Nagarjuna University (ANU). He completed his master degree in the year 2013 from Computer Science & Engineering with the specialization of Digital Image Processing, ANU. He has worked as a guest faculty in the CSE department, ANU. He has received his B.Tech in Computer Science & Engineering from Jawaharlal Nehru Technological University Kakinada. His research interests are Data Mining, Network Security, Digital Image Processing and Pattern Recognition.