IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP37-43
IMPACT FACTOR 1.719
www.iord.in

Efficient Anomaly Detection in Network Data Stream(Outlier)

Priti G. Manekar[1] , Prof. Pravin G. Kulurkar[2]

[1]M.Tech CSE,Vidarbha Institute of Engineering, Nagpur
sonu_manekar@rediffmail.com

[2]H.O.D,CSE, Vidarbha Institute of Engineering, Nagpur
pravinkulurkar@gmail.com

Abstract—

Outlier Mining is an important task of discovering the data records which have an exceptional behavior comparing with other records in the remaining dataset. Outliers do not follow with other data objects in the dataset. There are many effective approaches to detect outliers in numerical data. Most of the earliest work on outlier detection was performed by the statistics community on numeric data. But for categorical dataset there are limited approaches by using memory efficient incremental local outlier (MiLOF) detection algorithm and ROAD (Ranking-based Outlier Analysis and Detection algorithm).

Keywords—Outlier detection, Stream data mining, Local outlier, Memory efficiency

1. Introduction

Outlier detection is the process of detecting instances with unusual behavior that occurs in a system. Effective detection of outliers can lead to the discovery of valuable information in the data. Over the years, mining for outliers has received significant attention due to its wide applicability in areas such as detecting fraudulent usage of credit cards, unauthorized access in computer networks, weather prediction and environmental monitoring.

A number of existing methods are designed for detecting outliers in continuous data. Most of these methods use distances between data points to detect outliers. In the case of data with categorical attributes, attempts are often made to map categorical features to numerical values. Such mappings impose arbitrary ordering of categorical values and may cause unreliable result.

Another issue is related to the big data phenomenon. Many systems today are able to generate and capture real-time data continuously. Some examples include real-time

An outlier is a data point which is significantly different from the remaining data. Hawkins formally defined the concept of an outlier as follows:

"An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."

Outliers are also referred to as abnormalities, discordant, deviants, or anomalies in the data mining and statistics literature. In most applications, the data is created by one or more generating processes, which could either reflect activity in the system or observations collected about entities. When the generating process behaves in an unusual way, it results in the creation of outliers. Therefore, an outlier often contains useful information about abnormal characteristics of the systems and entities, which impact the data generation process. The recognition of such unusual characteristics provides useful application-specific insights.

Some examples are as follows:

Intrusion Detection Systems: In many host-based or networked computer systems, different kinds of data are collected about the operating system calls, network traffic, or other activity in the system.

This data may show unusual behavior because of malicious activity. The detection of such activity is referred to as intrusion detection.

OUTLIER detection (also known as rare event or anomaly detection) has received considerable attention in the field of data mining because of the need to detect unusual events in a variety of applications, such as fraud detection, human gait analysis and intrusion detection. A variety of outlier detection algorithms has been proposed for use on static data sets which have a finite number of samples.

However, outlier detection on streaming data is particularly challenging, since the volume of data to be analyzed is effectively unbounded and cannot be stored indefinitely in memory for processing. Data streams are also generated at a high data rate, hence making the task of outlier detection even more challenging. This is a significant problem that arises in many real applications.

For example, an outlier detection system in wireless sensor networks must work with the limited memory in each sensor node in order to detect rare events in near real time. The cost of communication is an important factor in such systems, which limits the scope for downloading data to a central server for archiving and analysis. Hence a memory efficient outlier detection algorithm is needed for streaming data, as it satisfies the memory constraint in the sensor nodes and avoids the communication cost of having to transfer data to external storage.

2. LITERATURE SURVEY

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.

Statistics-based approaches 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, signify all points that belong to clusters. The effectiveness of this approach depends on the clustering algorithm. Knorr and Ng 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.

Breunig et al. 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. 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 etal. 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 impacting the performances of these methods. To overcome this obstacle clustering-based algorithms have been proposed to detect outliers. The basic idea is that a data point is an outlier if it does not belong to any cluster.

An outlier is a data object that deviates significantly from the rest of the data. Detection of outliers is aimed at identifying such rare objects and exceptions in a given data set. It is a popular data mining task with applications in various domains such as fraud detection and intrusion detection in computer networks. Consequently, many methods have been developed for outlier detection employing various detection strategies. According to the distance-based methods, a data object is considered unusual if it has very few neighbors in its proximity. On the other hand, the clustering-based methods try to identify various groups of objects based on their intrinsic similarity there by isolating objects with outlier characteristics.

Additionally, many application settings like fraud detection and anomaly detection lend themselves to be posed as unsupervised problems due to lack of prior knowledge about the nature of various outlier objects. This emphasizes the need for developing efficient unsupervised methods for outlier detection.

In many data mining applications, the data objects are described using qualitative (categorical) attributes. The acceptable values of such a qualitative attribute are represented by various categories. The information on the occurrence frequencies of various categories of a categorical attribute in a given data set is very useful for many data-dependent tasks such as outlier detection.

Though there exist a number of methods for outlier detection in numerical data, the problem of outlier detection in categorical data is still evolving. The fundamental challenge in solving this problem is the difficulty in defining a suitable similarity measure over the categorical values. This is due to

the fact that the various values that a categorical variable can assume are not inherently ordered. As a result, many data mining related tasks such as determining the nearest neighbor of a categorical object turn out to be non-trivial. Some research efforts in this direction are indicative of the importance of this issue.

3. PROPOSED METHODOLGY

Proposed work includes new hybrid approach for outlier detection analysis for Categorical dataset by (MiLOF) and Ranking algorithm.

Data

A large problem when evaluating outlier detection methods is that there are very few real world data sets where it is exactly known which objects are really behaving differently due to belonging to a different mechanism. Though there exist multiple case studies on outlier detection, the question whether an object is an outlier or not is often depending on the point of view. Another problem is that the list of possible outliers is often incomplete making it hard to evaluate whether the algorithm ranked all outliers in the database properly. Therefore, we decided to evaluate our methods on artificially generated data. Thus, we can generate outliers and ordinary data points with respect to the initial definition, i.e. an outlier is a point being generated by a different mechanism than the majority of data objects. To exactly evaluate the behavior of our new method for different dimensionalities and database sizes, we generated multiple data sets having 25, 50 and 100 dimensions. As database sizes (dbsize) we selected 500, 1,000, 5,000 and 10,000 data objects.

In order to find data sets having well-defined but not obvious outliers, we proceeded as follows. First of all, we randomly generated a Gaussian mixture model consisting of equally weighted processes having random mean and variance values. This mixture model now describes the ordinary data points, i.e. the none-outlier data points. To build the outliers corresponding to another mechanism that does not assign all the outliers to an additional cluster, we employed a uniform distribution on the complete data space.

This way we generated 10 outliers for each data set which are totally independent on the mixture model describing the general data distribution. Let us note that it is possible that some outliers might be generated in an area being populated by none-outlier objects drawn from the Gaussian mixture model. Thus, even if an outlier detection mechanism works well, it does not necessarily have to rank all outliers into top positions.

Data Processing

1) Extract nominal feature attribute values from data files

Protocol_type=icmp,udp,tcp

Attack=phf,buffer_overflow,teardrop,guess_passwd,multihop,loadmodule,smurf,spy,normal,land,back,portsweep,warezclient,ftp_write,nmap,satan,rootkit,perl,imap,neptune,warezmaster,ipsweep,pod

Flag=RSTR,S3,SF,RSTO,SH,OTH,S2,RSTOS0,S1,S0,REJ

Service=vmnet,smtp,ntp_u,shell,kshell,aol,imap4,urh_i,netbios_ssn,tftp_u,mtp,uucp,nnsp,echo,tim_i,ssh,iso_tsap,time,netbios_ns,systat,hostnames,login,efs,supdup,http_8001,courier,ctf,finger,nntp,ftp_data,red_i,ldap,http,ftp,pm_dump,exec,klogin,auth,netbios_dgm,other,link,X11,discard,private,remote_job,IRC,daytime,pop_3,pop_2,gopher,sunrpc,name,rje,domain,uucp_path,http_2784,Z39_50,domain_u,csnet_ns,whois,eco_i,bgp,sql_net,printer,telnet,ecr_i,urp_i,netstat,http_443,harvest

This has been done once and won't be required to be done again.

2) Transformations of the nominal values to the numeric value according to as explained in paper A Practical Guide to Support Vector Classification

"We recommend using m numbers to represent an m-category attribute. Only one of the m numbers is one, and others are zero. For example, a three-category attribute such as fred, green, blueg can be represented as (0,0,1), (0,1,0), and (1,0,0). Our experience indicates that if the number of values in an attribute is not too large, this coding might be more stable than using a single number."

If input file name is kddcup.data_10_percent_corrected-1000 then result of this step would be saved in kddcup.data_10_percent_corrected-1000-transformed

3) Scaling -as explained in same paper, we scale between 0.0 to 1.0 Result of this step would be saved in kddcup.data_10_percent_corrected-1000-transformed-scaled

Memory efficient incremental local outlier (MiLOF) Algorithm

If any dataset consists outliers then it deviates from its from its original behavior and this dataset gives wrong results in any analysis. The MiLOF algorithm proposed the idea of finding a small subset of the data records that contribute to eliminate the disturbance of the dataset. This disturbance is also called entropy or uncertainty. We can also define it formally as 'let us take a dataset D with m attributes A1, A2--- Am and d(Ai) is the do-main of distinct values in the variable Ai, then the entropy of single attribute Aj is

$$E(Aj) = -\sum_{x\varepsilon\, d(Aj)} p(x)\, \log2(p(x)) \qquad (1)$$

Because of all attributes are independent to each other, Entropy of the entire dataset

IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP37-43
IMPACT FACTOR 1.719
www.iord.in

D={ A1, A2-------- Am} is equal to the sum of the entropies of each one of the m attributes, and is defined as follows

$$E(A1,A2\text{---}Am)=E(A1)+E(A2)+\text{---}E(Am)(2)$$

When we want to find entropy the MiLOF algorithm takes k outliers as input. All re-cords in the set are initially designated as non-outliers. Initially all attribute value's frequencies are computed and using these frequencies the initial entropy of the dataset is calculated.

Then, MiLOF algorithm scans k times over the data to determine the top k outliers keeping aside one non-outlier each time. While scanning each time every single non-outlier is temporarily removed from the dataset once and the total entropy is recalculated for the remaining dataset. For any non-outlier point that results in the maximum de-crease for the entropy of the remaining data-set is the outlier data-point removed by the algorithm. The MiLOF algorithm complexity is O(k *n*m*d), where k is the required number of outliers, n is thenumber of objects in the dataset D, m is the number of attributes in D, and d is the number of distinct attribute values, per attribute. Pseudo code for the MiLOF Algorithm is as follows.

This proposed model has been de-fined as an optimal number of outliers in a single instance to get optimal precision in any classification model with good precision and low recall value. This method calculates 'k' value itself based on the frequency. Let us take the data set 'D' with 'm' attributes A1, A2----- Am and d (Ai) is the domain of distinct values in the variable Ai. kN is the number of outliers which are normally distributed. To get 'kN' this model used Gaussian theory. If any object frequency is less than "mean-3 S.D" then this model treats those objects as outliers. This method uses AVF score formula to find AVF score but no k-value is required. Let D be the Categorical dataset, contains 'n' data points, xi, where i= 1…n. If each data point has 'm' attributes, we can write xi = [xi1, …..xil,…..xim], where xil is the value of the 1th attribute of xi.

Algorithm

Input: Dataset – D.
Output: K detected outliers.

Step 1: Read data set D
Step 2: Label all the Data points as non-outliers
Step 3: calculate normalized frequency of each attribute value for each point x_i
Step 4: calculate the frequency score of each record x_i as, Attribute Value Frequency of x_i is:

$$\text{AVF Score } (x_i) = F_i = \frac{1}{m}\sum_{j=1}^{m} f(xi\,j)$$

Step 5: compute N-seed values a and b as b=mean (x_i), a=b-3*std (x_i), if max (F_i) > 3*std (F_i)
Step 6: If Fi< a, then declare x_i as outlier
Step 7: return K_N detected outliers.

Data point	*Attributes*								
1	1	1	1	1	2	10	3	1	1
2	2	1	1	1	2	1	2	1	1
3	1	1	1	1	2	3	3	1	1
4	4	1	1	1	2	1	2	1	1
5	4	1	1	1	2	1	3	1	1
6	6	1	1	1	2	1	3	1	1
* 7	7	3	2	10	5	10	5	4	4
8	3	1	1	1	2	1	2	1	1
9	1	1	1	1	2	1	3	1	1
10	3	2	1	1	1	1	2	1	1
11	5	1	1	1	2	1	2	1	1
* 12	2	5	3	3	6	7	7	5	1

Example of normal and outlier points from Sensor Network Dataset. Outlier points are denoted by asterisk

Let consider above dataset having 12 data point and every data point is having attribute.
We label all the data point as non-outliner first.
Let D be the Categorical dataset, contains 'n' data points, xi, where i= 1...n. If each data point has 'm' attributes, we can write xi = [xi1, ...xil,.....xim], where xil is the value of the lth attribute of xi.

For example from above table for first row we can consider
xi =1
[xi1, ...xil,.....xim]=[1 , 1, 1, 1, 2, 10, 3, 1, 1]

We then calculate normalized frequency for this attribute [1 , 1, 1, 1, 2, 10, 3, 1, 1] that frequency we can denote as Fi.

Normalized frequency is calculated as ((freq of attribute) / (no of attribute)) * 1000
let say it has been calculated a 5 for above attribute.

Then we compute N-seed value as a and b
1) b=mean (xi) = mean(1)
2) a=b-3*std (xi)= mean(1)-3*std (1)
3) if max (Fi) > 3*std (Fi) = max(5) > 3*std(5)
4) Then Fi > a value then xi is outliner.
Step 1: Read data set D
Step 2: Label all the Data points as non-outliers
Step 3: calculate normalized frequency of each attribute value for each point xi

Step 4: calculate the frequency score of each record xi as, Attribute Value Frequency of xi is:
...

Step 5: compute N-seed values a and b as b=mean (xi), a=b-3*std (xi), if max (Fi) > 3*std (Fi)
Step 6: If Fi< a, then declare xi as outlier
Step 7: return KN detected outliers.

Ranking-based Outlier Analysis and Detection Algorithm

Given a data set D consisting of n objects described using m categorical attributes, the aim is to determine the likely set indicating the objects that are most likely outliers.As per the proposed definition for outliers, we propose a two-phase algorithm for unsupervised detection of outliers.The first-phase does the object density computation and also explores a clustering of the given data set. Using the resulting clustering structure, the set of big clustersis identified in order to determine the distance between various data objects and their corresponding nearest big clusters. In the second-phase, the frequency-based rank and the clustering-based rank of each data object are determined.Subsequently, a unified set of the most likely outliers is constructed using these two individual rankings. Therefore,we name the proposed method as Ranking-based Outlier Analysis and Detection (ROAD) algorithm.This algorithm addresses the issue of dealing with categorical data for outlier detection by providing a novel definition for outliers. As per our novel approach, a data object turns out to be an outlier in two scenarios: either the categorical values describing that object are relatively infrequent (hereafterdenoted as Type-1), or the combination of the categoricalvalues describing that object is relatively infrequent, though each one of these values are frequent individually (hereafterdenoted as Type-2). These scenarios can be depicted pictorially as shown in Figure 2, for a simple data set described using two categorical attributes. In this figure, the object O1 turns out to be an outlier of Type-1 as its value corresponding to the second attribute is infrequent.
On the other hand, though both the attribute values of object O2 are frequent individually, their combination is not frequent, making it an outlier of Type-2. For object O3, though it qualifies to be of Type-2, it is primarily of Type-1 due to its infrequentattribute values. Hence, we make no distinction between objects like O1 and O3 in our methodology.As brought out in [9], it is more meaningful to rank the data objects based on their degree of deviation instead of making a binary decision on whether or not an object isan outlier. Also, in many application domains dealing withlarge data, it is more sensible to identify the set of most likely outliers, as it enables in carrying out further analysismore efficiently. Due to this insight, the algorithm proposedhere leverages the ranking concept for determining the setof most likely outliers in a given data set.

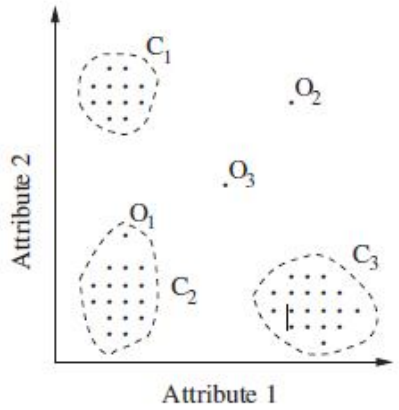

Fig. Various scenarios of outlier occurrence

The computational steps involved in the proposed two phase ROAD algorithm are presented in above Figure.
The computational complexity of the proposed algorithm is mainly contributed by the first three steps. If the maximumnumber of unique values of an attribute is s, then the first step requires O(nms) computations. Typically, s is known to be a small quantity compared to n. The nextstep requires O(nmk2) computations.
The third step is the k-modes algorithm, which requires O(nmkt) computations, where t is the number of iterations required for convergence. As we are using the initializationmethod proposed by [16], the k-modes algorithm needs only a few iterations making t a very small value. Similarly, we propose to work with very small number of clusters (k).
Finally, the ranking phase requires O(nlog(n)) effort. Thus, the computational complexity of the proposed algorithm turns out to be O(nm + nlog(n)). It is important to note that the computational complexity of this algorithm is not affected by the number of outliers to be detected.

4.Experimental Results

The experimental evaluation of the proposed method on MiLOFbenchmark data includes six frequently used

IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP37-43
IMPACT FACTOR 1.719
www.iord.in

categorical data sets KDD CUP dataset for Sensor Network. EachRequire: An m-dimensional data set D with n data objects and values for the parameters k and α.
Ensure: List of likely outliers identified.
Phase (1): Computational phase
1: Compute density(Xi) of each data object Xi ∈ D (Equation 3).
2: Determine the initial set {Z1, Z2, . . . ,Zk} of k cluster representatives, using the method described in [16].
3: Perform the k-modes clustering [14] on D using the distance measure given in Equation 2.
4: Determine the set of big clusters BC (Equation 4).
5: For each data object Xi, determine its cluster distance cdist(Xi) (as defined in Equation 5).

Phase (2):
6: Determine the frequency-based rank freq rank(Xi) of each data object Xi ∈ D (Definition 6).
7: Determine the clustering-based rank clust rank(Xi) of each data object Xi ∈ D (Definition 7).
8: Construct the likely set LS using the two ranked sequences, for a given p value (Definition 9).
A novel algorithm for mining categorical outliers through ranking data set consists of labeled instances belonging to two different classes. As per the standard practice in this field,objects with missing attribute values have been removed and objects belonging to small sized class in every data set are considered as outliers. Though these designated outliers are not outliers in real sense, they are considered so for validating the proposed method. In order to imposeimbalance in the number of objects belonging to normal and outlier categories, only a selected sub-set of the objects belonging to outlier class have been considered, by taking every fifth object of the designated outlier class summarizes the description of various benchmark categorical data sets considered in this experimentation. As the proposed algorithm works in unsupervised learning mode, it doesn't require labeled data. However, class labels are used to measure its performance in detecting outliers. The overall resultare shown in following figures that gives time comparison between MiLOF, Rank, HYBRID.

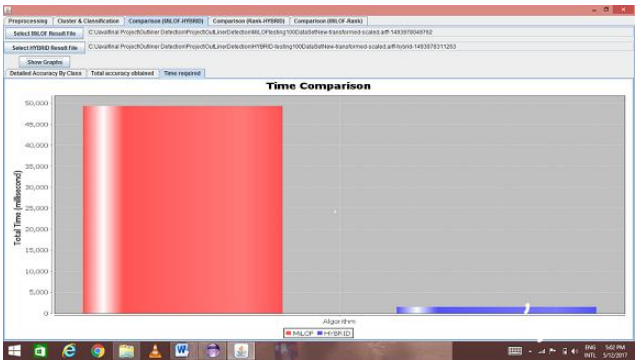

Fig: Time Comparison(MiLOF-HYBRID)

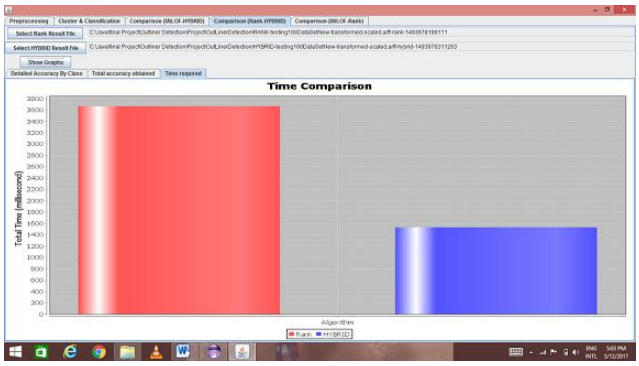

Fig: Time Comparison(Rank-HYBRID)

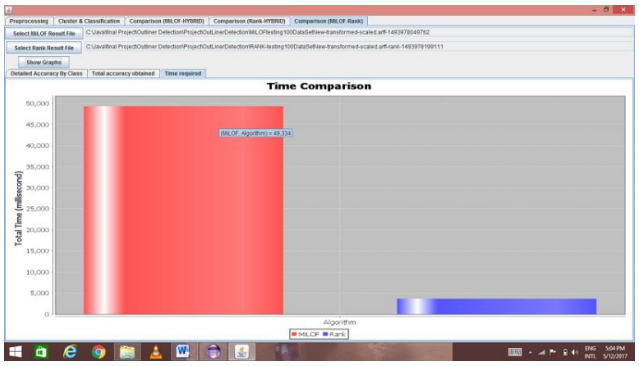

Fig: Time Comparison(MiLOF-Rank)

5. Conclusion

In this project, we introduced a novel, parameter-free approach to outlier detection based on the variance of angles between pairs of data points. This idea alleviates the effects of the curse of dimensionality on mining high-dimensional data where distance-based approaches often fail to offer high quality results. In addition to the basic approach memory efficient incremental local outlier (MiLOF) detection algorithm, we proposed two variants: RANK as an acceleration suitable for low-dimensional but big data sets, and Hybrid ,later-refinement approach as an acceleration suitable also for high-dimensional data. In a thorough evaluation, we demonstrate the ability of our new approach to rank the best candidates for being an outlier with high precision and recall. Furthermore, the evaluation discusses efficiency issues and explains the influence of the sample size to the runtime of the introduced methods.

6.REFERENCES

[1] M. E. Otey, A. Ghoting, and and A. Parthasarathy, "Fast Distributed Outlier Detection in Mixed-Attribute Data Sets," Data Mining and Knowledge Discovery
He, Z., Deng, S., Xu, X., "A Fast Greedy algorithm for outlier mining", Proc. of PAKDD, 2006.

[2] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining: Pearson Addison-Wesley, 2005

[4] E. Knorr, R. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," VLDB Journal, 2000.
M. M. Breunig, H.-P.Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density based local outliers," presented at ACM SIGMOD International Conference on Management of Data, 2000

[5] S. Papadimitriou, H. Kitawaga, P. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," presented at International Conference on Data Engineering, 2003

[6] Z. He, X. Xu, J. Huang, and S. Deng, "FP-Outlier: Frequent Pattern Based Outlier Detection", Computer Science and Information System (ComSIS'05)," 2005

[7] S. Wu and S. Wang, "Information-Theoretic Outlier Detection for Large-Scale Categorical Data, IEEE Transactions on Knowledge Engineering and Data Engineering,2011

[8] A. Frank, & A. Asuncion, (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[9] E. Muller, I. Assent, U. Steinhausen, and T. Seidl, "Outrank: ranking outliers in high dimensional data," in IEEE ICDE Workshop, Cancun, Mexico, 2008, pp. 600–603.

[10] K. Das and J. Schneider, "Detecting anomalous records in categorical datasets," in ACM KDD, San Jose, California, 2007, pp. 220–229.

[11] Z. He, X. Xu, and S. Deng, "A fast greedy algorithm for outlier mining," in PAKDD, Singapore, 2006, pp. 567–576.

[12] A. Koufakou, E. Ortiz, and M. Georgiopoulos, "A scalable and efficient outlier detection strategy for categorical data," in IEEE ICTAI, Patras, Greece, 2007, pp. 210–217.

[13] S. Guha, R. Rastogi, and S. Kyuseok, "ROCK: A robust clustering algorithm for categorical attributes," in ICDE, Sydney, Australia, 1999, pp. 512–521.

[14] Z. Huang, "A fast clustering algorithm to cluster very large categorical data sets in data mining," in SIGMOD DMKD Workshop, 1997, pp. 1–8.

[15] A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters, vol. 31, pp. 651–666, 2010.

[16] F. Cao, J. Liang, and L. Bai, "A new initialization method for categorical data clustering," Expert Systems with Applications, vol. 36, pp. 10 223–10 228, 2009.

[17] A. Asuncion and D. J. Newman.(2007) UCI machine learning repository.[Online]. Available: http://archive.ics.uci.edu/ml

[18] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets.In Proc. VLDB, 1998.

[19] E. M. Knorr and R. T. Ng. Finding intensional knowledge of distance-based outliers.In Proc. VLDB, 1999.

[20] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchthold.
Efficient biased sampling for approximate clustering and outlier detection in large datasets. IEEE TKDE, 15(5):1170{1187, 2003.

[21] D. Newman, S. Hettich, C. Blake, and C. Merz. UCI repository of machine learning databases, 1998.

[22] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In Proc. ICDE, 2003.

[23]s. Ramaswamy, R. Rastogi, and K. Shim. Ealgorithms for mining outliers from large data sets. In Proc. SIGMOD, 2000.

[24]P. Rousseeuw and K. Van Driessen.A fast algorithm for the minimum covariance determinant estimator.Technometrics, 41:212{223, 1999.
..