International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 5 30 – 33

# Efficient approach for Detection of Outlier using Hybrid Algorithm

Ms. Kanchan D. Shastrakar
Department of Computer Science &Engineering
VIT, RTMNU
Nagpur, India
*Kanchan.Shastrakar7@gmail.com*

Prof. Pravin G. Kulurkar
Department of Computer Science &Engineering
VIT, RTMNU
Nagpur, India
*Pravinkulurkar@gmail.com*

***Abstract***— Outlier Mining is an important task of determining the data records which have a remarkable behavior comparing with other records in the remaining dataset. In the dataset outlier do not follow other data objects. There are many real 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. Proposed work includes new hybrid approach for outlier detection analysis for Categorical dataset by merging NAVF (Normally distributed attribute value frequency) and Ranking algorithm.

***Keywords***- *NAVF, ROAD, Outliers, Categorical*

*****

## I. INTRODUCTION

Outlier detection is the process of detecting instances with uncommon behavior that occurs in a system. Effective detection of outliers can lead to the discovery of valued information in the data. Over the years, mining for outliers has received significant attention due to its wide applicability in areas such as detecting untrue usage of credit cards, illegal access in computer networks, Medicals, weather prediction and environmental monitoring. It is often used in preprocessing to remove anomalous data from the dataset.

A number of present 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 defective 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 knowingly different from the lasting data. Hawkins formally defined [205] the concept of an outlier as follows:
"An outlier is a remark which deviates so much from the other remarks as to produce thoughts that it was generated by a different mechanism."

Outliers are also referred to as inconsistencies, conflicting, 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 uncommon 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 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 uncommon behavior because of malicious activity. The detection of such activity is referred to as intrusion detection.

Credit Card Fraud: Credit card fraud is quite prevalent, because of the ease with which sensitive information such as a credit card number may be compromised. This typically leads to unauthorized use of the credit card. In many cases, unauthorized use may show different patterns, such as a buying spree from geographically unclear locations.

Such patterns can be used to detect outliers in credit card transaction data.

Interesting Sensor Events: In many real applications Sensors are used to track various environmental and location parameters. The sudden changes in the underlying patterns may represent events of interest. In the field of sensor network event detection is one of the primary motivating applications.

The analysis of outlier data is denoted to as outlier mining. Most of the existing systems are concentrated on numerical attributes or ordinal attributes. By using NAVF (Normally distributed attribute value frequency) and ROAD (Ranking-based Outlier Analysis and Detection algorithm) new hybrid approach for outlier detection in categorical dataset will be formed.

## II. EXISTING ALGORITHM FOR OUTLIER DETECTION IN CATEGORICAL DATASET:

### *A.* ***Greedy:***

The Greedy algorithm offered 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.

The Greedy algorithm complexity is O(k *n*m*d), where k is the required number of outliers, n is the number 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.

### *B. AVF (attribute value frequency):*

The AVF algorithm complexity is lesser than Greedy algorithm since AVF needs only one scan to detect outliers. The complexity is O (n * m). It needs 'k' value as input.

1.1.3 NAVF (Normally distributed attribute value frequency): This proposed model (NAVF) has been de-fined as an optimal number of outliers in 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.

### *C. ROAD (Ranking-based Outlier Analysis and Detection algorithm):*

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.

## III. PROPOSED METHODOLOGY

### *A. NAVF (Normally distributed attribute value frequency)*

If any dataset consists outliers then it departs from its original behavior and this dataset gives wrong results in any analysis. The Greedy 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 at-tributes 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

$$E(Aj) = -\sum_{x\varepsilon\, d(Aj)} p(x)\log_2(p(x)) \qquad (1)$$

Because of all attributes are independent to each other, Entropy of the entire dataset D= {A1, A2-------- Am} is equal to the sum of the entropies of each one of the m attributes, and is defined as follows

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

When we want to find entropy the Greedy 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, Greedy 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 Greedy algorithm complexity is O(k *n*m*d), where k is the required number of outliers, n is the number 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 Greedy Algorithm is as follows.

This proposed model (NAVF) 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 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 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.

### *B. 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 clusters is 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 (hereafter denoted as Type-1) or the combination of the categorical values describing that object is relatively infrequent, though each one of these values are frequent individually (hereafter denoted 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 infrequent attribute 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 is an outlier. Also, in many application domains dealing with large data, it is more sensible to identify the set of most likely outliers, as it enables in carrying out further analysis more efficiently. Due to this insight, the algorithm proposed here leverages the ranking concept for determining the set of most likely outliers in a given data set.

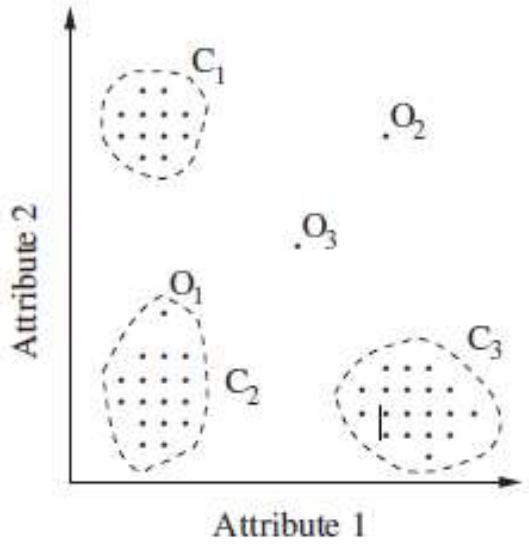


**Fig2 . Various scenarios of outlier occurrence**

The computational steps involved in the proposed two phase ROAD algorithm are presented in Figure 3.

The computational complexity of the proposed algorithm is mainly contributed by the first three steps. If the maximum number 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 next step requires O(nmk2) computations, as reported in [16]. 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 initialization method 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.

**Require:** An $m$-dimensional data set $D$ with $n$ data objects and values for the parameters $k$ and $\alpha$.
**Ensure:** List of likely outliers identified.
Phase (1): Computational phase
1: Compute $density(X_i)$ of each data object $X_i \in D$ (Equation 3).
2: Determine the initial set $\{Z_1, Z_2, \ldots, Z_k\}$ 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 $X_i$, determine its cluster distance $cdist(X_i)$ (as defined in Equation 5).
Phase (2): Ranking phase
6: Determine the frequency-based rank $freq\_rank(X_i)$ of each data object $X_i \in D$ (Definition 6).
7: Determine the clustering-based rank $clust\_rank(X_i)$ of each data object $X_i \in D$ (Definition 7).
8: Construct the likely set $LS$ using the two ranked sequences, for a given $p$ value (Definition 9).

**Figure 3. A novel algorithm for mining categorical outliers through ranking**

## IV. CONCLUSION

Outlier detection is a major task for data mining applications. Present algorithms are real 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 two algorithm i.e. NAVF and RANK to measure an object's outlierness. Sum of two of an object is naturally meaningful to measure the degree of isolation of an object. Based on this idea, we propose the Hybrid Algorithm which is combination of both NAVF and RANK that is effective to solve the problems declared above for many situations.

## REFERENCES

[1] M. E. Otey, A. Ghoting, and A. Parthasarathy, "Fast Distributed Outlier Detection in Mixed-Attribute Data Sets," Data Mining and Knowledge Discovery

[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.

[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)," 2005S. Wu and S. Wang, “Information-Theoretic Outlier Detection for Large-Scale Categorical Data, IEEE Transactions on Knowledge Engineering andData Engineering,2011

[7] 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.

[8] 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.

[9] K. Das and J. Schneider, “Detecting anomalous records in cateorical datasets,” in ACM KDD, San Jose, California, 2007, pp. 220–229.

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

[11] 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.

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

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

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

[15] 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.

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

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