Outlier analysis of categorical data using FuzzyAVF

Conference Paper · March 2013

DOI: 10.1109/ICCPCT.2013.6529023

CITATIONS
11

READS
103

2 authors:

Lakshmi Sreenivasa Reddy Dirisinapu
Chaitanya Bharathi Institute of Technology
25 PUBLICATIONS **54** CITATIONS
SEE PROFILE

Raveendra Babu Bhogapathi
Sri Indu College of Engineering & Technology
40 PUBLICATIONS **112** CITATIONS
SEE PROFILE

Outlier Analysis of Categorical Data using FuzzyAVF

Lakshmi Sreenivasa Reddy.D
Department of CSE
Rise Gandhi Group of institutions
Ongole, India
urdlsreddy@yahoo.com

Dr B.Raveendra Babu
VNRVJIET,
Hyderabad, India
rboghapathi@yahoo.com

***Abstract*— Outlier mining is an important task to discover 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. But for categorical dataset there are limited approaches. We propose an algorithm FuzzyAVF to detect outliers in categorical data. This algorithm utilizes the frequent pattern data mining method. It avoids problem of giving k-outliers to get optimal accuracy in any classification models in previous work like Greedy, AVF, FPOF, and FDOD while finding outliers. The algorithm is applied on UCI ML Repository datasets like Nursery, Breast cancer mushroom and bank dataset by excluding numerical attributes. The experimental results show that it is efficient for outlier detection in categorical dataset.**

***Keywords*—Outliers, Categorical, AVF, fuzzyAVF, FPOF, FDOD.**

I. INTRODUCTION

Outlier analysis is an important research field in many applications like credit card fraud, intrusion detection in networks, medical field .This analysis concentrate on detecting infrequent data records in dataset.

Most of the existing systems are concentrated on numerical attributes or ordinal attributes .Sometimes categorical attribute values can be converted into numerical values. This process is not always preferable. In this paper we discuss a simple method for categorical data is presented.

AVF method is one of the efficient methods to detect outliers in categorical data. The mechanism in this method is that, it calculates frequency of each value in each data attribute and finds their probability, and then it finds the attribute value frequency for each record by averaging probabilities and selects top k- outliers based on the least AVF score. The parameter used in this method is only "k", the no. of outliers. FPOF is based on frequent patterns which are adopted from Appriori algorithm [1]. This calculates frequent patterns item sets from each object. From these frequencies it calculates FPOF score and finds the least k- outliers as the least FPOF scores. This method takes more time to detect outliers comparing with AVF. The parameters used in it are σ, a threshold value to decide frequent sub sets in each data object. The next method is based on Entropy score. Greedy [2] is another method to detect outliers from categorical data. The previous approaches used to detect outliers were

II. EXISTING APPROACHES

A. Statistical based

This method adopted a parametric model that describes the distribution of the data and the data was mostly univariate [3, 4]. The main drawbacks of this method are difficulty of finding a correct model for different datasets and their efficiency decreases as the no. of dimensions increases [4]. To rectify this problem the Principle component method can be used. Another method to handle high dimensional datasets is to convert the data records in layers however; these ideas are not practical for more than or equal to three dimensions.

B. Distance-Based

Distance based methods do not make any assumptions about the distribution of the data records because they must compute the distances between records. But these make a high complexity. So these methods are not useful for large datasets. There are some improvements exist in the distance-based algorithms, such as Knorr's et al. [5], they have explained that apart of dataset records belong to each outlier must be less than some threshold value. Still it is an exponential on the number of nearest neighbors.

C. Density Based

These methods are based on finding the density of the data and identifying outliers as those lying in regions with low density. Breunig et al.[6], have calculated a local outlier factor (LOF) to identify whether an object contains sufficient neighbor around it or not [6]. They have decided a record as an outlier when the record LOF which is a user defined threshold. Papadimitriou et al. presented a similar technique called Local Correlation Integral, which deals of selecting the minimum points (min pts) in LOF through statistical methods in [7]. The density based methods have some advantages that they can detect outliers that are missed by techniques with single, global criterion methods. The terminology used in this paper is given below

978-1-4673-4920-8/13/$31.00 ©2013 IEEE

TABLE I. TERMINOLOGY

Term	Description
k	Target number of outliers
n	Number of objects in Dataset
m	Number of Attributes in Dataset
d	Domain of distinct values per attribute
xi	i th object in Dataset ranging from 1 to n
Aj	jth Attribute ranging from 1 to m
xij	A value in xi th object which takes from domain dj of j th attribute Aj
D	Dataset
I	Item set
F	Frequent Item set
P(xij)	Frequency of xij value
FS	Set of frequent Item sets
IFSi	Set of infrequent Itemsets of ithobject
Minsup	Minimum support of frequent itemset
Support(I)	Support of Itemset I

III. ALGORITHMS

A. *Greedy algorithm*

If any dataset consists outliers then it deviates 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 attributes A1, A2----- Am and d(Ai) is the domain of distinct values in the variable Ai, then the entropy of single attribute Aj is

$$E(A_j) = -\sum_{x \varepsilon d(A_j)} p(x) \log 2(p(x)). \quad (1)$$

Because of all attributes are independent to each other, Entropy of the entire dataset
D={ A_1, A_2-------- A_m} is equal to the sum of the entropies of each one of the m attributes, and is defined as follows

$$E(A_1, A_2\text{--------} A_m) = E(A_1) + E(A_2) + \text{-------} E(A_m) . \quad (2)$$

When we want to find entropy the Greedy algorithm takes k outliers as input [2]. All records 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 decrease for the entropy of the remaining dataset 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

Algorithm: Greedy
Input: Dataset – D
Target number of outliers – k
Output: k outliers detected
label all data points $x_1, x_2, \text{---} x_n$ as non-outliers
Calculate initial frequency of each attribute value and update hash table in each iteration
calculate initial entropy
counter = 0
while (counter != k) do
counter++
while (not end of database) do
read next record 'xi ' labeled non-outlier
label 'xi' as outlier
calculate decrease in entropy
if (maximal decrease achieved by record 'o')
update hash tables using 'o'
add xi to set of outliers
end if
end while
end while

However entropy needs k as in put and need to find number of outliers more times to get optimal accuracy of any classification model.

B. *AVF algorithm*

The algorithm discussed above is linear with respect to data size and it needs k-scans each time. The other models also exist which are based on frequent item set mining (FIM) need to create a large space to store item sets, and then search for these sets in each and every data point .These techniques can become very slow when we select low threshold value to find frequent item sets from dataset
Another simpler and faster approach to detect outliers that minimizes the scans over the data and does not need to create more space and more
Search for combinations of attribute values or item sets is Attribute Value Frequency (AVF) algorithm. An outlier point xi is defined based on the AVF Score below:

$$\text{AVF Score (xi) is } F(xi) = \frac{1}{m} \sum_{j=1}^{m} f(xi\,j) . \quad (3)$$

In this approach [1] again we need to find k-outliers many times to get optimal accuracy of any classification model. Pseudo code for the AVF Algorithm is as follows

Input : Database D (n points _ m attributes), Target number of outliers - k
Output: k detected outliers
Label all data points as non-outliers;
for each point xi, i = 1 to n do
for each attribute j, j = 1 to m do

```
      Count frequency f(xij)of attribute value   xij;
     end
end
for each point xi, i = 1 to n do
     for each attribute j, j = 1 to m do
       AVF Score(xi) += f(xij);
     end
    AVF Score(xi) /= m;
end
Return k outliers with mini (AVF Score);
```

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. In FPOF [8] this has discussed frequent pattern based outlier detection, in this too k-value and another parameter 'σ 'is required as threshold. This also discussed about frequent pattern based method to find infrequent object, in this too it requires k-value, and another parameter 'σ' as input. In the proposed model(Fuzzy AVF) it has been defined 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 (NAVF) and fuzzy theory. If the frequency is less than "mean-3 S.D" then this model uses fuzzy logic. This method uses AVF score formula, but no k-value is required.

C. FAVF algorithm

Algorithm Detecting Outliers in Categorical Data Sets:
Let D be the Categorical dataset, containes n data points, xi , i= 1…n. If each datapoint has m attributes, we can write xi = [xi1, …..xil,…..xim], where xil is the value of the lth attribute of xi.

Input: Dataset – D, outlier type 1- for low frequency, 2 – for high frequency and 3 – for both low and high frequency.
Output: K detected outliers.

Step 1: Read data set D
Step 2: Lable all Data points as non-outliers
Step 3: calculate normalised frequency of each attribute value for each point x
Step 4: calculate the frequency score of each record xi as,
Frequency average of xi, is

$$Fi = \frac{1}{m}\sum_{k=0}^{m} x_{kl}$$

Step 5: compute fuzzy seed values a,b and c as

b = mean (Fi),

$$a = \begin{cases} b - 3 * std(Fi) & \text{if } max(Fi) > 3 * std(Fi) \\ b - 2 * std(Fi) & \text{if } max(Fi) > 2 * std(Fi) \\ b - std(F:) & \text{otherwise} \end{cases}$$

$$c = \begin{cases} b + 3 * std(Fi) & \text{if } max(Fi) > 3 * std(Fi) \\ b + 2 * std(Fi) & \text{if } max(Fi) > 2 * std(Fi) \\ b & \text{otherwise} \end{cases}$$

Step6: Apply fuzzy S-function to detect outlier points with less frequency.
Step 7: return K detected outliers.

IV. EXPERIMENTAL RESULTS

In this paper this model has used Breast Cancer, Nursery data and Bank marketing data from UCI Machine repository [10]. This method has implemented the approach of using MATLAB tool. We ran our experiments on a workstation with a Pentium(R) D, 2.80 GHz Processor and 1 .24 GB of RAM. Nursery data consists of nine attributes and 6236 records. This data divided into two parts based on parent attribute, first part contains 4320 records with usual parent type, and second part contain 1916 records with pretentious parent type which is used as outliers in our experiment. In first iteration 956 sample records are selected randomly using Clementine tool; from each two records one is selected. These 956 records are mixed up with part one and applied normally distributed AVF and Fuzzy AVF to get outliers. The found outliers are given in table2. Similarly in the next iteration 382 records are selected randomly as one record from each five records and mixed up with first part and applied the same process .The results are given in the below Table2. Similarly one record is selected from each eight records and ten records and repeated the same process. This method has been implemented on Nursery dataset, Breast cancer and Bank dataset which are taken from UCI Machine learning repository. This method compared the models with each sample. Comparison graph is given in Figure 1.

TABLE II. COMPARISION-NURSERY

Sample method	Actual outliers	Total records	NAVF		Fuzzy AVF	
			True positives	*False positive*	*True positives*	*False positives*
1-in-2	956	5276	44	1	56	1
1-in-5	382	4702	132	1	162	1
1-in-8	238	4558	238	0	238	1
1-in-10	190	4570	190	0	190	1

In the first sample from nursery the Fuzzy AVF model found out only 5.85% of outliers from 956 outliers which are mixed up with 4320 records which totals to 5276 records, while

4.60% correct outliers are found by NAVF from the same mixture. The difference between these two is 1.25%. In the next sample of 382 records, 42.67% of correct outliers are found by FuzzyAVF, while 34.8% of correct outliers are found by NAVF .For the sample of 238 records FuzzyAVF found 239 outliers in which 238 are correct and one is false negative while NAVF found 238 correct outliers, which means that both models found 100% outliers correctly. Similarly these two models found 100% outliers in the sample of 190 records (as outliers) mixed up with 4320 records in part one.

In case of breast cancer dataset, correct outliers found by both models did not touch 100%. In breast cancer data 119, 48, 29, 23 outliers are selected respectively using 1-in-2, 1 -in-5, 1-in-8, 1-in-10 sampling from benign breast cancer. FAVF model found 43, 11, 11, 15 correct outliers and 1, 1, 1, 2 outliers wrong respectively from 119, 48, 29, 23 outliers. NAVF found 35, 9, 9, and 14 correct and 0, 0, 0, 1 wrong from 119, 48, 29, 23 outliers. The results are given in table 3

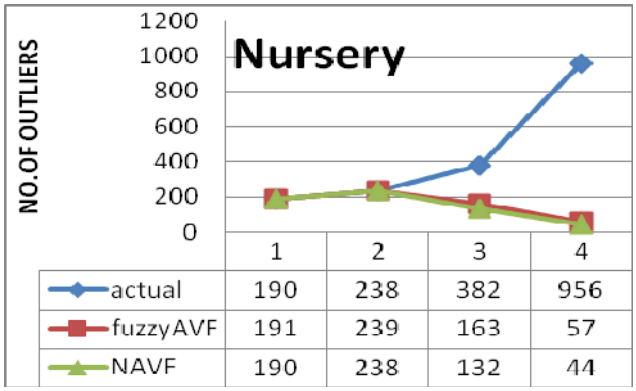

Figure 1. NURSERY-DATA

TABLE III. COMPARISION-BREAST CANCER

Sample method	Actual outliers	Total records	NAVF		Fuzzy AVF	
			True positives	*False positives*	*True positives*	*False positives*
1-in-2	119	577	35	0	43	1
1-in-5	48	506	9	0	11	1
1-in-8	29	487	9	0	11	1
1-in-10	23	481	14	1	15	2

In Bank marketing data, only categorical attributes are selected and 2644, 1027, 661, 528 outliers are selected respectively using 1-in-2, 1 -in-5, 1-in-8, 1-in-10 sampling method from the attribute Y="yes" and applied the same process as above. In this data NAVF found 274, 198, 152, 126 correct outliers and 100, 168, 202, 213 wrong outliers from the random sample of 2644, 1027, 661, 528 outliers taken by 1-in-2, 1 -in-5, 1-in-8, 1-in-10 sampling method. While in FAVF has found 278, 199, 154, 126 correct outliers and 101, 171, 204, 215 wrong outliers from the random sample of 2644, 1027, 661, 528 outliers taken by 1-in-2, 1 -in-5, 1-in-8, 1-in-10 sampling method .Similarly we applied the same process for all samples in banking data. The results are summarized in the table 4 and its graph is given in graph 3. Different classification models are tested for accuracy of the bank dataset after deleting the outliers. Different classification models are tested on 1-in-5 sample data which contain 39922 original records mixed up with 1027 outliers. The NAVF model has found 366 records as outliers in which 198 are correct outliers and 168 are wrong outliers(original records).When Neural network,C5,CRT,QUEST,CHAID Linear Regression ,Decision Logic Classifiers are applied on the above sample data ,the classifiers have given the accuracies as given in the Table 5

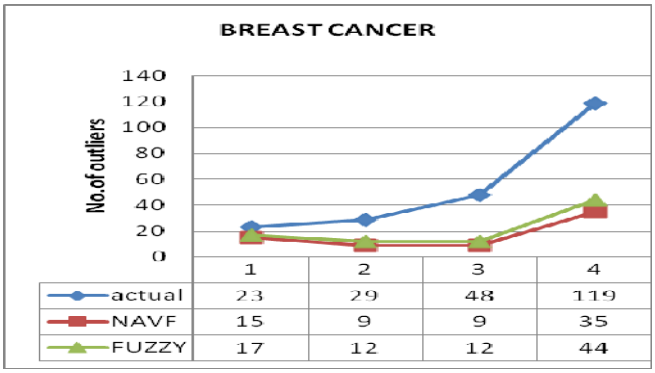

Figure 2. *Breast cancer Data*

TABLE IV. COMPARISION-BANK DATA

Sample method	Actual outliers	Total records	NAVF		FuzzyAVF	
			True positives	*False positives*	*True positives*	*False positives*
1-in-2	2644	39922	274	100	278	101
1-in-5	1027	39922	198	168	199	171
1-in-8	661	39922	152	202	154	204
1-in-10	528	39922	126	213	126	215

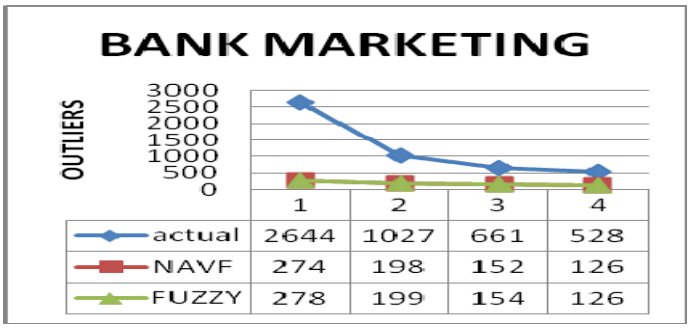

Figure 3. *Bank Data*

TABLE V. CLASSIFIERS RESULTS ON BANK DATA

Classifier	FAVF	NAVF	K=100	K=354	K=370	K=500	K=700
NN	97.887	98.735	97.581	97.878	97.885	97.994	98.135
C5	97.887	98.735	97.581	97.881	97.885	97.994	98.138
CRT	97.887	98.735	97.581	97.881	97.885	97.994	98.138
QUEST	97.887	98.735	97.581	97.881	97.885	97.994	98.138
DL	92.942	37.058	95.445	92.933	92.94	92.339	69.148
CHAID	97.887	98.735	97.581	97.881	97.885	97.994	98.138
LR	97.885	98.735	97.581	97.881	97.882	97.994	98.138

After deleting the outliers and the classifiers are applied on the remaining dataset found by NAVF and FAVF models, all classifiers given good results. Only the decision logic classifier gave very less accuracy (37.058) by NAVF. But by FAVF the classifiers given consistent results

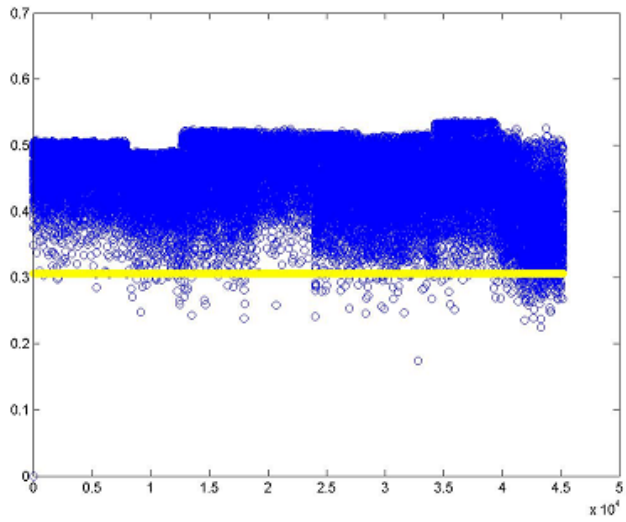

Figure 4. *Bank data by n-1 sample*

This model gives the graph with outliers .In the Figure 5 the outliers can be observed under the yellow line. The outliers are with very less frequency in the above figure.

V. CONCLUSION AND FUTURE WORK

To sum up, this proposed method gives the optimal number of outliers 'K_N '.In existing models it is mandatory to give the number of outliers to find them .While taking the number of outliers sometimes the original data may be missed. If we modeled the classifier this data wrong classifiers may be modeled. In future there is a possibility of checking the precision and recall values of each model with the existing models. The same method can also be applied on mixed type of dataset.

VI. REFERENCE

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

[2] He, Z., Deng, S., Xu, X., "A Fast Greedy algorithm for outlier mining", Proc. of PAKDD, 2006.

[3] 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.

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

[5] E. Knorr, R. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," VLDB Journal, 2000.

[6] 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

[7] 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

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

[9] Shu Wu and Shengrui Wang, "Information-Theoretic Outlier Detaction for Large-Scale Categorical Data, IEEE Transactions on Knowledge Engineering and Data Engineering,2011

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