Comparative Study of Improving Classifiers Accuracies

Lakshmi Sreenivasa Reddy. D
Department of MCA
Chaitanya Bharathi Institute of Technology
Hyderabad
drreddycsejntuh@cbit.ac.in

Abstract**—** Outlier analysis is an essential task in data science to wipe out inconsistencies from data to build a good model. Finding outliers from categorical data is a tough task. To model a good Classifier, it is necessary to eliminate outliers from data. While modeling categorical data, most infrequent records are treated as outliers. These outliers would disturb the entire data in modeling a good classifier. This paper presents the comparison between classifiers accuracies which are built by normally distributed Outlier factor by infrequency (NOFI) to OFI with different inputs. In modeling a classifier for categorical data, high frequent records are most useful and most infrequent records are most useless. So the infrequent records are obstacles in modeling the classifiers. The experiments are conducted for this comparison on bank dataset with 45000 records and Nursery dataset with 14000 records approximately, which are taken from UCI ML Repository. For normally distributed OFI, the inputs are not needed. It generates the number of outliers automatically. In OFI it is needed to give the inputs. However the threshold value is needed to generate infrequent itemsets for both methods.

Keywords — *Outlier analysis, Categorical datasets, OFI score, NOFI score.*

*****

I. Introduction

Outlier analysis is an important task in data mining. Without deleting outliers a correct classifier cannot be built. Applications related to outlier analysis are like credit card fraud detection, intrusion detection in networks, medical treatment analysis, and decision making analysis in business. This paper presents how the outliers found by NOFI are reliable when compared with OFI method for different inputs. AVF and OFI methods are checked for different input values [4]. These input values (number of outliers) are given by user manually. But arriving at the number of outliers to be eliminated is a problem. To overcome this problem, NOFI is designed to find number of outliers automatically. Even though the number of outliers is found by NOFI automatically, fixing up whether these outliers are reliable or not are another problem. The remedy for this problem is to check the reliability by modeling a classifier. AVF [1] method is one of the good methods to find outliers in a categorical dataset. This method calculates frequency of a value in each attribute for each data point. Then it finds their average AVF score for each record. Here the problem is how many outliers need to be selected from the dataset. In this method, an input is to be given for selecting the number of outliers and then the question is how far they are reliable outliers. NAVF [7] method overcomes this problem. After deleting these outliers automatically by NAVF, the classifier has been built on the remaining data. The other approach FAVF [2] has also been attempted and this method also finds the number of outliers automatically. However the reliability of outliers found by FAVF is less when compared to NAVF. The next approach FPOF [15] for categorical data is also used to find the outliers based on frequent pattern concept generated by Apriori algorithm [18]. FPOF calculates frequent pattern item sets for each record in the data set based on a threshold value given by the user. FPOF score is calculated for each record and from these scores, k outliers are found as the k-records with the least k-FPOF scores. All these methods are based on the concept of average frequency of each attribute value. The complexity of this FPOF is high, because it needs time and space to generate frequent patterns of different levels and also needs a threshold value 'σ' and input 'k' as the number of k outliers need to be eliminated.

II. Existing Methods for categorical data

A. *Attribute value frequency(AVF) Algorithm*

AVF approach is less complex and is a faster approach to find outliers in categorical data. It scans entire dataset only once and it does not require more space. The AVF method is defined as follows.

Let "x_i" is an object in a categorical dataset. AVF score of this object is defined as below.

$$AVF(x_i) = \sum_{j=1}^{m} f(x_{ij}) \qquad (1)$$

This method also need input ‘k’ as the number of outliers to be eliminated. This approach gives us more accuracy with low complexity.

TABLE I. TERMINOLOGY

Term	Description
DB	Database
K	Target number of outliers
N	Number of objects in Dataset
M	Number of Attributes in Dataset
X_i	i^{th} object in the Dataset ranging from 1 to n
A_j	j^{th} Attribute ranging from 1 to m
$D(A_j)$	Domain of distinct values of j^{th} attribute
X_{ij}	cell value in i^{th} object which takes from domain d_j of j^{th} attribute A_j
D	Dataset
V	Set of all distinct values in Dataset D
P	Set of all combinations of distinct attribute values, where each attribute occurs only once in any combination
I	Item set
F	Frequent Item set
IF	Infrequent item set.
$f(x_{ij})$	Frequency of x_{ij} value
$FS(x_i)$	Set of frequent Item sets of x_i object
$IFS(x_i)$	Set of infrequent Item sets of x_i object
Minsup	Minimum support of frequent item set
Support(I)	Support of Item set I
OFI(x_i)	Outlier Factor by Infrequency score
NOFI(x_i)	Normally distributed Outlier Factor by Infrequency score
FPOF(x_i)	Frequent Pattern Outlier Factor score

A. Frequent Pattern Outlier Factor (FPOF) algorithm

In this algorithm the concept of Apriori algorithm is used as a first step to generate all frequent Item sets. This method needs a threshold value called “minimum support(σ)” as input to generate frequent item sets. Considering this threshold value, it generates all possible combinations of attribute values in each record and compares the frequency of each combination with threshold value to decide whether the item set is frequent or not in each record. To find frequency of each combination, it needs one scan of the dataset. FPOF is defined as below.

$$FPOFScore = \sum_{F\subset x_i \wedge F\in IF(x_i)} \frac{\text{support}(F)}{|FS(x_i)|} \tag{2}$$

Where Dataset D= {A1, A2-------- Am},
Minimum support = ‘σ’,
Number of outliers = ‘k’,
F is the frequent item set satisfying the minimum support,
FS (x_i) is the set of all frequent itemsets which are subsets of the record “x_i”,

This model finds FPOF score for each record and selects k-outliers as least k-scores. If there is no frequent itemset at all in any record, FPOF score becomes infinite. This is one of the drawbacks of the method.

B. Outlier Factor by Infrequency (OFI)

OFI [4] calculates the outlier factor based on infrequency of each infrequent itemsets generated by Apriori algorithm [18] for each record. OFI score is calculated by the below formula.

$$OFI(x_i) = \sum_{j=1}^{m} \frac{|DB|}{1+Frequency(\inf requenct\_Itemset\_of\_record)} \tag{3}$$

Here,
Let “x_i” is the record of a dataset DB,
A_j=Attribute, where j takes the values from 1 to m,
IF= Infrequent Itemset,
IFS (x_i) =Set of infrequent Itemsets of “x_i”,
x_{ij} = i^{th} value in j^{th} attribute.
$|DB|$ is length of Dataset

OFI score of each record is calculated by the above equation (3). K-outliers are selected as k-highest OFI score records. This method is also needed input value “k” to get k- outliers and a threshold value to decide infrequent itemsets.

C. Normally Distributed Outlier Factor by Infrequency (NOFI)

OFI method finds k-number of outliers basing on the input ‘k’. NOFI calculates reliable number of outliers automatically based on the threshold value. This threshold value is calculated as below.

$$\text{Mean}_{(OFI)}=\frac{1}{|DB|}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{|DB|}{1+Frequency(\inf requenct\_Itemsets\_of\_record)}$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{1+Frequency(\inf requenct\_Itemsets\_of\_record)} \tag{4}$$

$$\text{SD}_{\text{OFI}}=\sum_{i=1}^{n}\sqrt{\left(OFI(x_i)-\text{Mean}_{(OFI)}\right)}^{2} \tag{5}$$

$$\text{NOFI}_{\text{tresh}}=\text{Mean}_{(OFI)}+3\text{SD}_{\text{OFI}} \tag{6}$$

If X_i is said to be an outlier in dataset DB, its OFI score must satisfies the below condition.

$$\text{if OFIscore}(X_i) \begin{cases} \geq \text{NOFI}_{tresh} \text{ , } X_i \text{ is called Outlier} & \forall\, i=1 \text{ to } n \\ < \text{NOFI}_{tresh} \text{ , } X_i \text{ is called inlier} & \forall\, i=1 \text{ to } n \end{cases} \quad (7)$$

III. Comparison of Experimental Results between OFI and NOFI with discussion

For comparison of these methods, experiments are conducted on bank data with 41512 records taken from UCI ML Repository [17]. Only seven categorical attributes are considered for experiments and is implemented on PL-SQL platform. Bank data records with 7 attributes and 28 distinct attribute values are considered for experiments. The attributes considered for these experiments are “Job”, “Marital status”, “Education”, “loan”, “housing”, “contact”, and a class label attribute “Y”. “Job” attribute consists 12 distinct values, “Marital status” attribute consists 3 distinct values, “Education” attribute consist 4 distinct values, and “loan”, “housing”, and “Y” attributes consist 2 distinct values each. The last attribute “contact” contains 4 distinct values. Bank data has been divided into two parts using Clementine 11.1 tool, first part of dataset considered is with “Yes” Class label and the number of records for “Yes” class are 1590 and second part with “no” class label are 39922 records. The “yes” label records are considered as outliers in this experiment. From the first part, 527 records are selected randomly and mixed up with “no” class label records. The mixed up records are 40499. NOFI method has been applied on these mixtures of records. After eliminating outliers automatically by NOFI, this method has found 39899 inliers. The total outliers are found by NOFI are 600. Of these 600 records, 156 records found with “Yes” class label which are true positives. Similarly 444 false positives are found by NOFI. For NOFI, the threshold value found from OFI scores is 4.9966. OFI method is applied for k= 100, 200, 300, 400, 500, 600, 700, 800 and NOFI is also applied on the entire 40499 records of mixed data. These methods have been found true positives as given below

TABLE II. COMPARISON OF TRUE AND FALSE POSITIVES FOR OFI AND NOFI FOR BANK DATA

TABLE III.

Input (OFI)	**K= 100**	**K= 200**	**K= 300**	**K= 400**	**K= 500**	**K= 600**	**K= 700**	**K= 800**	**(NOFI) 528**
True positives	42	71	96	125	145	176	212	214	156
False Positives	58	129	204	275	355	424	488	586	444

When the outliers are deleted directly by OFI for k=100, OFI found 42 true positives and 58 false positives. Similarly for k=200, true positives are 71 and false positives are129. These true and false positives are found for different inputs of k before and after the threshold value found by NOFI automatically which is k=528.

TABLE IV. COMPARISON OF CLASSIFIERS ACCURACIES MODELED BY OFI AND NOFI FOR BANK DATA

OFI	DL	NN	LR	CHAID
K=100	58.344	98.696	98.696	98.696
K=200	58.039	98.692	98.692	98.692
K=300	58.049	98.691	98.691	98.691
K=400	38.062	98.988	98.691	98.691
K=600	58.804	98.686	98.686	98.686
K=700	57.89	98.697	98.697	98.697
K=800	57.89	98.697	98.697	98.697
NOFI=528	35.559	99.068	99.068	99.068

After eliminating outliers by both these methods, Decision Logic (DL), Linear Regression (LR), Neural Network (NN), and CHAID classifiers are generated. Clementine11.1 tool has been used to model all these classifiers. The classifiers modeled by NOFI show more accuracy than OFI with different inputs. Similarly the results are given in Table III for different classifiers modeled by NOFI. Among all these classifiers LR shows better accuracy.

Fig. 1. COMPARISON OF ACCURACY FOR CLASSIFIER **DL** MODELED BY OFI AND NOFI FOR BANK DATA

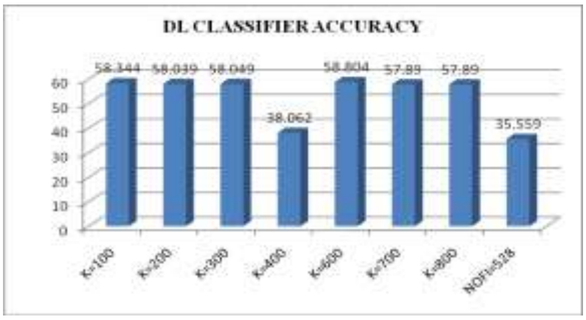


Fig. 2. COMPARISON OF ACCURACY FORCLASSIFIER **NN** MODELED BY OFI AND NOFI FOR BANK DATA

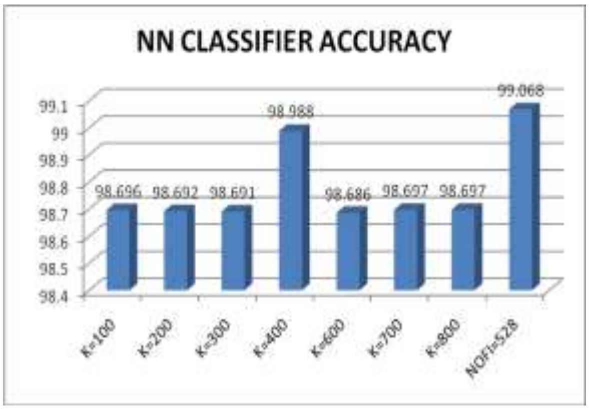


Fig. 3. COMPARISON OF ACCURACY FOR CLASSIFIER **LR** MODELED BY OFI AND NOFI FOR BANK DATA

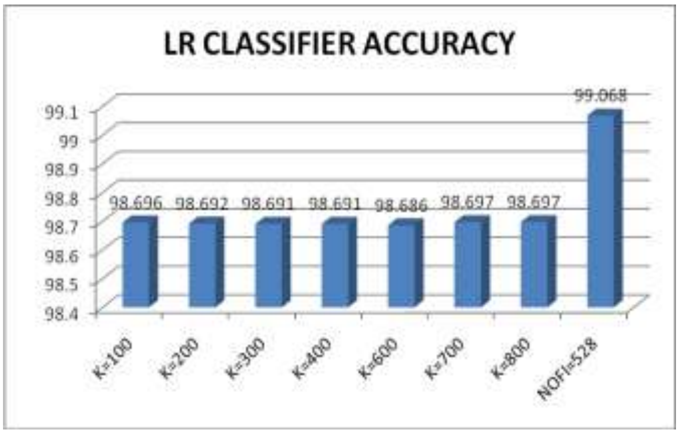
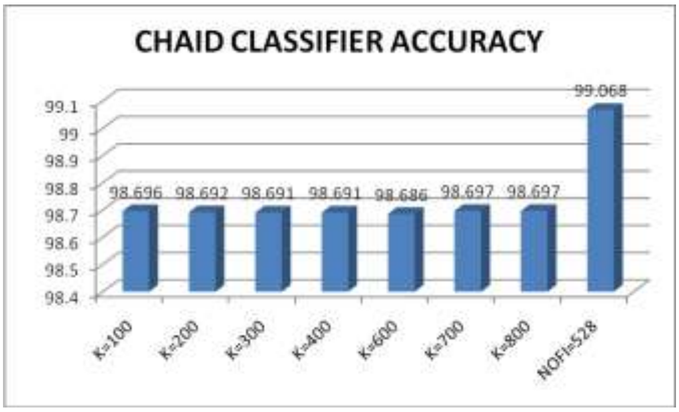


Fig. 4. COMPARISON OF ACCURACY FOR CLASSIFIER **CHAID** MODELED BY OFI AND NOFI FOR BANK DATA



Decision logic (DL) classifier is different and gave less accuracy when compared with, Neural Networks (NN), Linear Regression (LR) and CHAID Classifiers. Among all LR and CHAID gave more accuracy when NOFI found input is used to eliminate outliers. All these classifiers achieved almost 99% accuracy approximately.

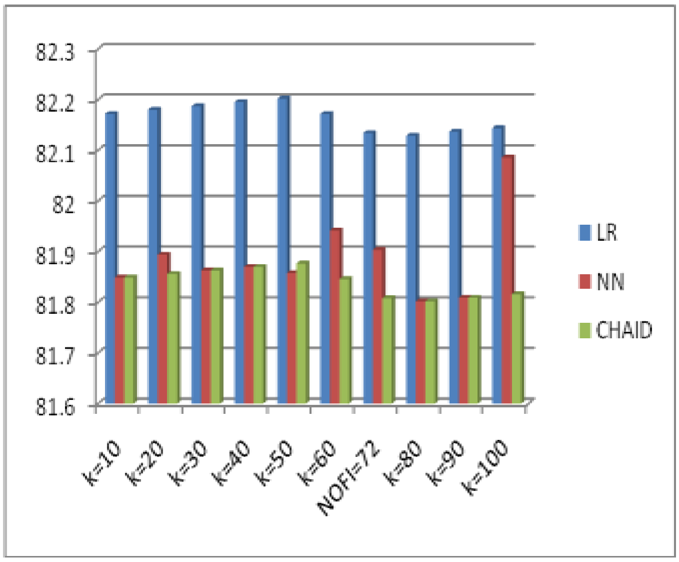
Similarly when the same process is applied on Nursery data with 5275(2-1sample) for different input values of OFI and the input found by NOFI automatically, these models gave the results as below. Nursery data is also taken from UCI ML repository [17]. Nursery data contains 8 attributes including a class label attribute and 27 distinct attribute values. The threshold value for NOFI has been found 17.684724 from OFI scores. The number of records less than 17.684724 is 72. After eliminating these outliers, LR, NN, CHAID classifiers have been built for the remaining pure data and tested them. The results are given in below Table IV.

TABLE V. COMPARISON OF CLASSIFIERS ACCURACIES MODELED BY OFI AND NOFI FOR BANK DATA

OFI	NN	LR	CHAID
K=30	81.863	82.187	81.863
K=40	81.87	82.195	81.87
K=50	81.858	82.202	81.877
K=60	81.942	82.172	81.846
K=80	81.802	82.129	81.802
K=90	81.809	82.137	81.809
K=100	82.086	82.144	81.816
NOFI=72	81.904	82.134	81.808

In Bank data all the classifiers gave high accuracy when NOFI found number of outliers is deleted from the original data. But in nursery data different classifiers gave different accuracies for different number of outliers. Only the CHAID classifier gave highest accuracy for the NOFI found number of outliers. In bank data almost all classifiers except DL gave approximately 99% accuracy. In nursery data all classifiers have reached approximately 82% accuracy.

Fig. 5. COMPARISON OF CLASSIFIERS ACCURACIES MODELED BY OFI AND NOFI FOR NURSERY DATA



IV. CONCLUSION AND FUTURE WORK

NOFI method has achieved good results when compared with direct inputs through OFI and it can be inferred that NOFI is one of the better methods when compared to OFI. In future, NOFI need to be compared to NAVF for more datasets.

References

[1] Anna Koufakou, Michael Georgiopoulos, “A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes” Data mining and Knowledge Discovery , Volume 20, 2010, pp.259-289.

[2] LakshmiSreenivasaReddy.D, B.RaveendraBabu “Outlier Analysis of Categorical Data using FuzzyAVF”, presented at IEEE international conference ICCPCT-2013, pp 1259-1263.

[3] LakshmiSreenivasaReddy.D, B.RaveendraBabu and etc, “Learning Styles Vs Suitable Courses” IEEE international conference -MITE-2013, pp 52-57.

[4] LakshmiSreenivasaReddy.D, B.RaveendraBabu “Efficient Model to Find Outliers in Categorical Data Using Outlier Factor by Infrequency”, presented at IEEE international conference ICCPCT-2014, pp 1324-1328.

[5] LakshmiSreenivasaReddy.D, B.RaveendraBabu and A.Govardhan, “A Novel Approach to Find Outliers in Categorical Dataset” presented at Elsevier - AEMDS-2013 pp 925-932.

[6] LakshmiSreenivasaReddy.D, B.RaveendraBabu and A.Govardhan, “A model for Improving Classifier Accuracy for Categorical data using Outlier Analysis”, International Journal of Computers and Technology” vol 7, 2013. pp 500-509.

[7] LakshmiSreenivasaReddy.D, .B.RaveendraBabu and A.Govardhan, “Outlier Analysis of Categorical Data using NAVF””, Informatica Economica vol 17, Cloud computing issue 1, 2013

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

[9] He, Z., Deng, S., Xu, X., “A Fast Greedy algorithm for Outlier mining” Proc. of PAKDD, 2006.

[10] 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.

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

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

[13] 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.

[14] 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.

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

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

[17] 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.

[18] R. Agrawal and R. Srikant. “Fast Algorithms for Mining Association Rules in Large Databases.” Proceedings International Conference on Very Large Data Bases, 1994, pp. 487-499.