Fast Entropy Attribute Value Frequency Algorithm to Detect Outliers for Categorical Data

Kang-Mo Jung
Kunsan National University
Department of Statistics & Computer Science, Korea
+82-63-469-4616
kmjung@kunsan.ac.kr

ABSTRACT

Outliers are extreme observations which is far away from other observations. Outlier detection becomes a significant procedure for many applications such as detecting insurance fraud or industrial damage. Most outlier detection techniques work on numerical data, that is, continuous attributes. However, there are few research works on outlier detection for categorical data. AEVF(Automated Entropy Value Frequency) is a measure to detect outliers for categorical data. AEVF has complexity $O(qn^2)$, and it cannot be applied to large number n of observations and the number q of attributes. We propose a fast entropy attribute value frequency(FEAVF) having complexity $O(qn)$. Furthermore, we propose a fast algorithm for multiple records deletion as well as single record deletion. The performance of FEAVF can be effectively illustrated for UCI machine learning datasets.

CCS Concepts

• Information systems → Information systems applications → Data mining → Data cleaning • Mathematics of computing → Probability and statistics → Statistical paradigms → Exploratory data analysis.

Keywords

Attribute entropy value frequency; Big data; Categorical data; Complexity; Multiple records deletion; Outlier detection.

1. INTRODUCTION

Information driven model is the heart of the era of big data. Information is the condensed result of data with appropriate analytics. Even single outlier can distort the result of analysis and it led to unreasonable decision. Outlier detection is a fundamental step in data processing. Outlier detection can be used to many applications in intrusion detection, mobile phone and insurance claim fraud detection, medical and public health outlier detection and industrial damage detection [1]. Hawkins [2] defined "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism".

ICBDE '18, March 9–11, 2018, Honolulu, HI, USA

ACM ISBN 978-1-4503-6358-7/18/03…$15.00
https://doi.org/10.1145/3206157.3206172

Outlier detection has a long history. However, many diagnostic measures have been proposed for numerical data. For example, there are distance based method using k-nearest neighbor [3], density-based method using low-density region [4] and clustering method using isolated cluster [5].

A few works that treat outlier detection for categorical data are found. The proposed methods for numerical data cannot be directly applied to categorical data, because it needs mappings of categorical attributes into numerical attributes. [6] proposed AVF(Attribute Value Frequency) method uses frequency data and it is similar to distance based method for numerical data. AEVF(Automated Entropy Value Frequency) used entropy change to determine the degree of outliers [7], [8]. Outliers make large change of entropy. Thus AEVF presents good performance on detecting outliers for categorical data. However, they assume the independence between attributes, and also its speed is not fast and its complexity is $O(qn^2)$, where q is the number of attributes and n is the number of observations.

In this paper, we introduce a fast outlier detection procedure for categorical data using entropy attribute value frequency(FEAVF) irrelevant of dependence between attributes. We obtain the formula for direct computation of EAVF instead of reconstruction of AVF matrix. Furthermore, we compute the impact of multiple records on AEVF as well as single record. Therefore, we consider the joint impact of multiple records. Any authors have not studied the joint impact.

The organization of this paper is as follows. In Section 2, we provide a thorough review of conventional EAVF. In Section 3, we propose FEAVF algorithm for single and multiple records. Section 4 contains our experiments and results. Finally Section 5, we summarize our work and provide future works.

2. ENTROPY ATTRIBUTE VALUE FREQUENCY ALGORITHM

For a random variable X the entropy $E(X)$ can be defined as

$$E(X) = -\sum_{x \in S(X)} p(x)\log(p(x)),$$

where $S(X)$ is range of X and $p(x)$ is he probability function of X. In case the multiple random variables $X_1, \cdots, X_q$ in the same manner the entropy is defined as

$$E(X_1, \cdots, X_q) = -\sum_{x_1 \in S(X_1)} \cdots \sum_{x_q \in S(X_q)} p(x_1, \cdots, x_q) \log\left(p(x_1, \cdots, x_q)\right). \quad (1)$$

Suppose that random variables $X_1, \cdots, X_q$ are mutually independent, then the computation of $E(X_1, \cdots, X_q)$ is easily

obtained by summing each entropy $E(X_j), j = 1, \cdots, q$ of random variable X_j.

A categorical data can be expressed in table format. The row and column of the table denotes the record and the attribute, respectively. Let the categorical data $x_l = (x_{l1}, \cdots, x_{lq}), l = 1, \cdots, n$, where the size of the *j*th attribute is A_j. For example the data set *D* has five records with three attributes as following. Here $A_1 = 3, A_2 = 2, A_3 = 2$. In Table 1 Attribute1 can be summarized in {2, 1, 2} for category A, B, C from the frequency point of view. Similarly we obtain the frequency of other attributes.

Table 1. An example of categorical data

Record\Attribute	Attribute1	Attribute2	Attribute3
x_1	B	E	F
x_2	A	E	F
x_3	A	E	F
x_4	C	E	G
x_5	C	D	G

The matrix V of AVF is a $p \times q$ matrix of frequency for attributes, where $p = max_{1 \le j \le q} A_j$. The element v_{ij} of V denotes the frequency of the *i*th class for the *j*th attribute, $i = 1, \cdots, p; j = 1, , \cdots, q$. For $i > A_j$, $v_{ij} = 0$. Now we compose the AVF matrix for Table 1 as follows.

$$V = [v_{ij}] = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 4 & 2 \\ 1 & 0 & 0 \end{bmatrix}$$

Suppose that an AVF matrix V is a $p \times q$ matrix of attribute value frequencies which is constructed from n categorical data with q attributes. The entropy of V is defined as [8]

$$E = E(V) = \sum_{i=1}^{p} \sum_{j=1}^{q} -\left(\frac{v_{ij}}{n}\right) \log\left(\frac{v_{ij}}{n}\right), \quad v_{ij} \neq 0 \qquad (2).$$

The joint probability can be estimated by the frequency ratio. The entropy for the above example becomes 3.215.

Outlier detection method using the entropy is based on the phenomenon that outlier or rare records will abruptly change the information content of the data or make the entropy increase [7], [9]. How to measure the abrupt change of information content? The answer is to measure the gap between the entropies on the full data and on the deleted data. Denote the entropy of the data with the deletion of the *l*th data by $E_{(l)} = E(V_{(l)})$. Similarly we define the AVF matrix $V_{(l)}$. After deleting the first record for Table 1 data we obtain the matrix $V_{(1)}$ and the entropy $E_{(1)}$ as following

$$V_{(1)} = \begin{bmatrix} 2 & 1 & 2 \\ 0 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix}, \qquad E_{(1)} = 3.215 = E.$$

In this case the first record does not change the entropy and it can be regard as a normal data. For all data, we can obtain the difference between E and $E_{(l)}$, and we determine the records larger than a cut-off as outliers. According to the AEVF method [7], they set a cut-off the average of the entropy difference which is called the maximal entropy gap. For Table 1 data we obtained the difference $|E_{(l)} - E|$ and the maximal entropy gap 0.362. Therefore records 2, 3, 5 are regarded as outliers.

Table 2. Entropy difference for the example data

Deletion	$E_{(l)}$	$\lvert E_{(l)} - E \rvert$
x_1	3.215	0.000
x_2	3.715	0.500
x_3	3.715	0.500
x_4	3.526	0.311
x_5	2.715	0.500

3. FAST EAVF ALGORITHM

The deleted entropy $E_{(l)}$ can be computed by reconstructing the AVF matrix after deleting the *l*th data x_l. However, it needs the burden of time complexity as the number of records increases, which happens frequently in the era of big data. We compute $E_{(l)}$ by the following formula.

$$E_{(l)} = \sum_{i=1}^{p} \sum_{j=1}^{q} -\left(\frac{v_{ij}}{n-1}\right) \log\left(\frac{v_{ij}}{n-1}\right) + \sum_{j=1}^{q} \frac{v_{l_j,j}}{n-1} \log\left(\frac{v_{l_j,j}}{n-1}\right) - \sum_{j=1}^{q} \frac{v_{l_j,j}-1}{n-1} \log\left(\frac{v_{l_j,j}-1}{n-1}\right), \quad v_{ij} \neq 0, v_{l_j,j} \neq 0$$

$$= \sum_{i=1}^{p} \sum_{j=1}^{q} -\left(\frac{v_{ij}}{n-1}\right) \log\left(\frac{v_{ij}}{n-1}\right) + \sum_{j=1}^{q} \frac{1}{n-1} \log\left[\left(\frac{v_{l_j,j}}{v_{l_j,j}-1}\right)\left(\frac{v_{l_j,j}}{v_{l_j,j}-1}\right)^{v_{l_j,j}} \frac{v_{i_{l_j},j}}{n-1}\right], \quad v_{ij} \neq 0, v_{l_j,j} \neq 0 \qquad (3)$$

Here l_j is the index of the *j*th attribute for the record x_l. Then we directly compute $E_{(l)}$ without reconstructing the AVF matrix. The first term of (3) is constant for each single deletion. Therefore, it is enough to compare the second term for each record to investigate the effect of single deletion. By using (3) we got the exact value in Table 2 for the data in Table 1.

Furthermore, we obtain the entropy effect of multiple deletion records as well as single deletion record. Let be *L* the index set of deleted records with size n_L, and let *l* be the element of *L*. The entropy without the index set *L* becomes

$$E_{(L)} = \sum_{i=1}^{p} \sum_{j=1}^{q} -\left(\frac{v_{ij}}{n-n_L}\right) \log\left(\frac{v_{ij}}{n-n_L}\right) + \sum_{l \in L} \sum_{j=1}^{q} \frac{v_{l_j,j}}{n-n_L} \log\left(\frac{v_{l_j,j}}{n-n_L}\right) - \sum_{l \in L} \sum_{j=1}^{q} \frac{v_{l_j,j}-1}{n-n_L} \log\left(\frac{v_{l_j,j}-1}{n-n_L}\right) \quad v_{ij} \neq 0, v_{l_j,j} \neq 0 \qquad (4)$$

4. EXPERIMENTS AND RESULTS

The experiment was done on the personal computer with an Intel Core i7 3.6 GHz processor and 16 GB RAM. The algorithm were implemented in R. We conducted our fast algorithm on the real datasets from the UCI Machine Learning Repository [12], namely *lymphography*, *Wisconsin breast cancer*, and *post-operative*. The *lymphography* dataset contained 148 records with 18 attributes.

There are 4 classes where classes 2, 3 are normal classes and classes 1, 4 are rare classes. The rare classes comprise 2.7% (6 records) of the data, so they can be regarded as outliers. The *Wisconsin breast cancer* dataset has 699 instances with 9 attributes. Each record is labeled as *benign* (458 or 65.5%) or *malignant* (241 or 34.5%). Following the method by Harkins et al. [10] we got a modified dataset with *benign* (444 or 92%) or *malignant* (39 or 8%). The *post-operative* dataset has 8 attributes and 90 records. The purpose to analyze this dataset is to determine where patients should go to after a post-operative unit. It has three classes, *intensive care unit(I), home(S), general hospital floor(A*). We regard the classes *I* or *S* as outliers (26 or 28.8%).

Tables 3 to 5 summarizes the outlier detection performance by each algorithm using the real datasets. Here Fast EAVF denotes the proposed algorithm using equation (3), LSA in the tables denotes the local search algorithm based on the entropy [9], and the AVF means attribute value frequency algorithm that the infrequentness of an attribute value means that the record may be outliers [6]. Fast EAVF, LSA, AVF have time complexity $O(nq), O(n^2), O(nq)$, respectively. We can see that the performance of the proposed algorithm is similar to that of other algorithms. For example Table 4 shows that top-56 outliers by each algorithm converges to 39 outliers. We had similar accuracy in outlier detection for the lymphography, the Wisconsin breast cancer, and the post-operative patient dataset. The results of LSA and AVF methods are reproduced from [6].

Table 3. Outliers detection results on the lymphography data

k	Fast EAVF	LSA	AVF
1.4%(2)	2	2	2
2.7%(4)	3	4	4
4.1%(6)	5	5	4
5.4%(8)	5	6	5
8.1%(12)	5	6	6
10.1%(15)	5	6	6

Table 4. Outliers detection results on the Wisconsin breast cancer data

k	Fast EAVF	LSA	AVF
1%(4)	4	4	4
2%(8)	8	8	7
4%(16)	15	15	14
6%(24)	23	22	21
8%(32)	30	29	28
10%(40)	35	33	32
12%(48)	37	37	36
14%(56)	39	39	39

Table 5. Outliers detection results on the post-operative data

k	Fast AVF	LSA	AVF
11.1%(10)	2	4	3
22.2%(20)	4	7	7
33.3%(30)	8	10	10
44.4%(40)	13	11	12
55.5%(50)	16	13	12
66.6%(60)	16	16	16
77.7%(70)	20	21	21
88.8%(80)	23	24	24

The single deletion using equation (3) may cause Fast EAVF to miss a few outliers. Thus we conducted double using equation (4) for three real datasets and we summarized the results of top 5 in Table 6. Outlier index in the table means that the record corresponding to the index satisfying the outlier index condition is outlier. From Table 6 the single and double deletion results for the lymphography data and breast cancer data gives good results for outlier detection, but the method for the post-operative data missed the outliers.

Table 6. Single and double records deletion results.

Dataset	Single	Double	Outlier index
Lympho-graphy	147 143 145 89 144	(143, 147) (145, 147) (143, 145) (89, 147), (144, 147)	> 142
Breast cancer	451 461 456 458 483	(451,461) (451,461) (451,458) (451,483) (451,459)	> 444
Post-operative	13 16 27 79 31	(13, 16) (13, 27) (13, 79) (16, 27) (16, 79)	> 64

Now we compared the proposed algorithm to the EAVF with reconstructing the AVF table. The results in Table 7 shows that the Fast EAVF is very faster than the EAVF, especially for double deletion case the proposed algorithm is about one hundred times faster than the traditional method. It implies that the proposed method has proven very useful in the age of big data.

Table 7. Computing time (Seconds)

Dataset	Single record deletion		Double records deletion	
	Fast EAVF	EAVF	Fast EAVF	EAVF
Lympho-graphy	0.094	0.204	0.096	12.823
Breast cancer	0.180	0.371	0.194	90.341
Post-operative	0.027	0.049	0.028	2.206

5. CONCLUSION

Outlier detection has been an indispensable procedure in data mining. There are various types of attributes like categorical attributes which has increased by automated classifiers. However,

most of the outlier detection methods are focused on numerical data. This paper propose a fast algorithm to outlier detection for categorical data. The experimental results have shown that for real datasets it was able to detect outliers efficiently. Furthermore, the method gives the effect of multiple records deletion as well as single record deletion. In future work we will develop outlier detection methods for mixed type data of very large data size with categorical data and numerical data.

6. ACKNOWLEDGMENTS

This research was supported by Basic Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A 01060528).

7. REFERENCES

[1] Bansal, R., Gaur, N., and Singh, S. N. 2016. Outlier detection: applications and techniques in data mining. In *Proc. Int Conf. – Cloud Systems and Big Data Engineering*.

[2] Hawkins, D. 2000. *Identification of Outliers*, Chapman and Hall, London.

[3] Ramaswamy, S., Rastogi, R. and Shim, K. 2000. Efficient algorithms for mining outliers from large data sets. In *Proc. of ACM SIGMOD.*

[4] Aggarwal, C. C. and Yu, P. S. 2001. Outlier detection for high dimensional data. *In Proc. ACM-SIGMOD Int. Conf. Management of Data*, pp. 37-46.

[5] Jiang, M. F., Tseng, S. S. and Su, C. M. 2001. Two-phase clustering process for outlier detection. *Pattern Recognition Letters*, 22, pp. 691-700.

[6] Koufakou, K., Ortiz, E. G., Georgiopoulos, M., Anagnostopoulos, G. C. and Reynolds, K. M. 2007. A scalable and efficient outlier detection strategy for categorical data. In *Proc. Int. Conf. Tools with Artificial Intelligence*.

[7] Qamar, U. 2013. Automated entropy value frequency algorithm for outlier detection in categorical data, *Recent Advances in Knowledge Engineering and Systems Science*.

[8] Rokhman, N., Subanar and Winarko, E. 2016. WMEVF: an outlier detection methods for categorical data, *Int. Conf. on Informatics and Computing.*

[9] He, Z., Xu, X. and Deng, S. 2005. An optimization model for outlier detection in categorical data, *Lecture Notes in Computer Science*, Volume 3644, pp. 400-409.

[10] Harkins, S., He, H., Williams, G., Baster, R. 2002. Outlier detection using replicator neural networks, In *Proc. Int. Conf. Big Data Analytics and Knowledge Discovery*, pp. 170-180.

[11] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml