Proceedings of the World Congress on Engineering and Computer Science 2016 Vol I
WCECS 2016, October 19-21, 2016, San Francisco, USA

Enhancement of Lempel-Ziv Algorithm to Estimate Randomness in a Dataset

K. Koneru, C. Varol

Abstract—**Experts and researchers always refer to the rate of error or accuracy of a database. The percentage of accuracy is determined by the number of correct cells over the total number of cells in a dataset. After all, the detection and processing of errors depend on the randomness of their distribution in the data. Apparently, if the errors are systematic (present in a particular record or column), then they can be fixed readily with minimal changes. As a result, sorting errors would help to address many managerial questions. Enhanced Lempel-Ziv algorithm is reflected as one of the effective ways to differentiate random errors from systematic errors in a dataset. This paper explains Lempel-Ziv algorithm usage in differentiating random errors from systematic ones and proposes its improvement. The experiment spectacles that the Enhanced Lempel-Ziv algorithm successfully differentiates the random errors from the systematic errors for a minimum data size of 5000 and with a minimum error rate of 10%.**

Index Terms—**Data accuracy, Enhanced Lempel-Ziv, Prioritization, Random errors, Systematic errors**

I. Introduction

From the early age of software, data owned by an organization is one of the crucial assets. In order to improve the quality of information, primarily the data quality needs to be measured, to evaluate the value of any information available. Redman et.al, mentioned "the science of data quality has not yet advanced to the point where there are standard measures for any data quality issues" [1]. Considering the quality of data at the database level, the rate of error at the attribute level plays a vital role. The error rate is defined as the number of erroneous cells over the total number of attribute cells available in dataset. Lee et.al, had defined the accuracy rating as *1– (Number of desirable outcomes / total outcomes)* [2]. These definitions ascribe to individual cells which are data attributes for specific records.

Organizations are attentive towards the reliability, correctness and error free data. But the error in the data may not enclose to a particular area. Prioritization of databases plays a critical role when they are suggested to improve their existing quality. The number of errors per dataset or current quality might influence the priority to fix the problems. Hence finding the error relies on the vector quantity known as measure of randomness of error in data.

Manuscript received March 07, 2016. This work was supported in part by the Sam Houston State University.

K. Koneru, Student in Master of Sciences, Department of Computer Science, Sam Houston State University, Huntsville, TX 77341 USA (e-mail: keerthi@ shsu.edu).

C. Varol, Associate Professor in Department of Computer Science, Sam Houston State University, Huntsville, TX 77341 USA.(e-mail: cvarol@shsu.edu).

Distinguishing between dataset with random errors and dataset with systematic errors would help in better assessment of database quality. In this research, the developed method obtains more appropriate complexity metric using Lempel-Ziv algorithm to absolutely state the type of error.

The outcomes are computed by considering a sample dataset with errors (as 1's) and no errors (as 0's) where we could estimate and govern whether the errors are random (or not) by using Enhanced Lempel-Ziv (LZ) complexity measure. The proposed method helps to obtain the dataset with highest percentage of errors. Hence, it will be useful to address the decision-making query such as prioritizing the databases, which should be considered primarily, to fix the issues.

The rest of the paper is organized as follows. Related work in the areas of data quality and studies in randomness of dataset are detailed in section 2. The approach, Enhanced Lempel-Ziv algorithm, is explained in section 3. Section 4 shares the test cases and results that have been used and obtained from the study, and the paper is finalized with conclusion and future work section.

II. Related Work

The definition of randomness has its root from the branch of mathematics which considers the storage and transmission of data [3]. With the same percentage of errors existing in a dataset, the distribution of errors affects the management of a dataset more significantly. Hence the difference in the complexity measure can readily be observed which specifies the distribution of errors.

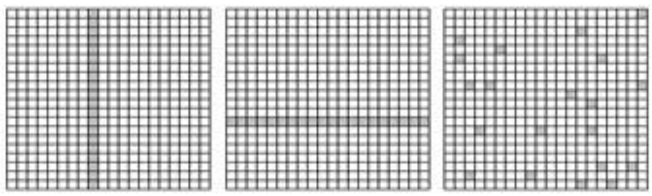

Fig 1.Distribution of Errors [3].
(a)Errors in one column; (b) Errors in one row; (c) Errors randomly distributed throughout the table

Fisher et.al, stated that the database might account for the same percentage of errors but have the errors randomly distributed among many columns and rows, causing both analysis and improvement to be significantly more complicated. Figure 1 depicts the datasets with 5% error rate as Redman's cells with error divided by total number of cells [3].

Sometimes error may be due to a single record or there may be existence of different errors in a single field which is

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

referred as systematic error as shown in Fig 1(a) and Fig 1(b) and can be handled by a single change. On the other hand, there may be existence of different errors in different records which are distributed in the dataset. Though the percentage of errors is same as that of systematic, the dataset is considered to be out of control. These are known as random errors as depicted in Fig 1(c). More effort is required to fix such kind of errors that needs to be taken care based on priority. All three cases could report the same percentage but represent different degrees of quality.

Different algorithms like Kolmogorov-Chaitin complexity, Lempel-Ziv complexity and probabilistic entropy based measures like Shannon entropy, Approximate entropy, Sample entropy etc. [4], are used to measure the degree of randomness in a database. The Lempel-Ziv measure of complexity is one of the most popular methods used to obtain the degree of randomness where the data is considered as binary sequence [5].

The *Kolmogorov-Chaitin (KC) complexity* measure provides the disorder in any sequence. Whenever a sequence is random, it is considered to be incompressible. The length of the minimal description of sequence conveys the complexity, which leads to the measure of randomness of the sequence. Kolmogorov-Chaitin complexity of a sequence of binary digits is defined as "the shortest program that can output the original sequence of bits". It can also be stated that sequence is said to be completely random, if KC complexity is approximately equal to the size of sequence of bits [2].

$$H_{KC}(M_0) = |P_{min}|$$

where H_{KC} is the KC complexity measure, M_0 is the original sequence and P_{min} is the minimal program [2].

This complexity measure not only organizes hierarchy of degrees of randomness but also describes the properties of randomness more precisely than statistical information. It is also used to measure the information content of a sequence.

But, the major problem in the calculation of KC complexity is that there will not be any general algorithm for such a program, and is highly dependent on the data available in the dataset. In such a case, it is hard to estimate the value of time complexity when $n \to \infty$.

Shannon Entropy, defined as the weighted average of the self-information within a message or the amount of randomness that exists in a random event, is another method used to find the random errors in a given dataset. Shannon entropy depends on the probability distribution of the sequence. Let *X* is the random variable of the sequence of binary digits *S* with a probability mass function *p(x)*, and then Shannon Entropy *H(X)* is given as

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

The value of *H(X)* varies from 0 to log (|S|), depicting zero to no uncertainty and log (|S|) when all elements have equal probabilities [6]. As the length of the sequence increases, it underestimates the higher entropies while overestimating lower entropies.

On the other hand, the *Lempel-Ziv algorithm* evaluates the complexity of a system objectively and quantitatively and overcomes the limitation of calculation of complexity statistically. The randomness parameter analyses difference between systematic patterns versus degree of random distribution. Unlike KC complexity measure, in which length of the program plays a major role, LZ complexity measure depends on two operations on the binary digits: copy and insert. It depends on the formation of number of distinct substrings along the length of sequence and the rate of their occurrence [3].

The LZ complexity algorithm and its corollaries are used in the development of application software and dictionary based lossless compressions such as WINZIP etc. It is extensively employed in biomedical applications to estimate the complexity of discrete time signals [3].

Apart from the above applications, Lempel and Ziv in 1976[2] mentioned that the algorithm has overcome the restraint of interpreting the complexity through characteristic quantities of statistical complexity. Simultaneously, as the calculation of characteristic quantities require longer data sequences, other algorithms can only identify whether system is complicated or not, whereas Lempel-Ziv Complexity measure shows the degree of system complexity [2].

Fisher et.al, had used the Lempel-Ziv complexity measure to obtain the randomness in a dataset [3]. But, the output is unable to provide accurate values of complexity measure for the small values of n, as the parameter, epsilon (ε_n) is disregarded. As a result, it not only overestimates the complexity measure but also impotent to differentiate noticeably between the random errors and systematic errors when the dataset size is less than 10000.

The proposed methodology, in this paper, enhances the Lempel-Ziv algorithm by considering the parameter ε_n and calculates the value of complexity measure appropriately. It differentiates between random and systematic errors for smaller dataset size commencing from 5000. It also determines the dataset with highest percentage of error among the given datasets of particular data size.

III. METHODOLOGY

LZ Complexity measure is a prominent approach used to differentiate the random and systematic errors. The word 'randomness' is used in an instinctual manner in day to day life to define regularity deficit in a pattern. Sequences which are not random will cast a doubt on the random nature of the generating process [7].

A. Lempel-Ziv Algorithm:

The steps for obtaining normalized complexity in the Lempel-Ziv Algorithm are given below.

1. Divide the sequence into consecutive disjoint words such that the next word is the shortest template not seen before.
2. The size of the disjoint sets is considered to be the complexity measure *c(n)* of the sequence, which is also defined as number of steps required to form the disjoint sets in a complete sequence.
3. The asymptotic value of *b(n)* is calculated as

$$b(n) = n/\log_2 n$$

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

4. The Lempel-Ziv complexity measure *C(n)* is evaluated by
5.

$$C(n) = \frac{c(n)}{b(n)}$$

The measure *C(n)* represents the rate of occurrence of new substring, and its value varies between 0 (for systematic sequences) and 1(for totally random sequence). While calculating the randomness in the database files, the measure is easily computable for large values of n [3].

B. Enhanced Lempel-Ziv Algorithm:

The high level architecture of the proposed system is shown in Figure 2. We assume a sample database is given as an input to the system and we already know whether the data available in the sample database is either correct or incorrect. Then, the system samples the data cell values into binary sequence, where 1's indicate the cells with errors whereas 0's indicate the cells without errors. After the binary sequence is generated, the analyzer applies the Enhanced LZ complexity algorithm to generate unique substrings from the sequence. Based on the created substrings, normalized LZ complexity is calculated for each of the datasets, which signifies whether the errors stationed in the data are systematic or random.

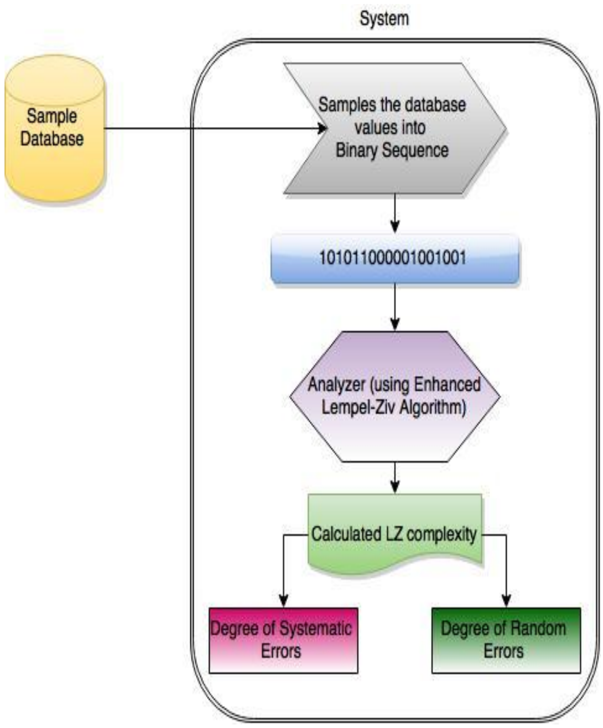

Fig 2. Architecture of Proposed Methodology

Although original LZ complexity objectively and quantitatively estimates the system complexity, there is a drawback of over valuation of normalized complexity for short-series. Fisher et.al, has stated that the *C(n)* value goes close to zero for deterministic sequences and approaches 1 for non-deterministic or random sequences [3]. But it may not be applicable for the finite sequences appropriately to distinguish between random errors and systematic errors. In order to overcome such disadvantages and improve the complexity measure to differentiate the randomness and prioritize the databases which can be considered to obtain the integrity, the Lempel-Ziv algorithm is modified. Yong Tang et.al, mentioned that the asymptotic value of *b(n)* is accurate if the value of $n \to \infty$ [2]. But for every finite sequence, there exists a value ε_n, such that

$$b(n) = n/\,(1-\varepsilon_n)\log_2 n.$$

The value of epsilon (ε_n) is given by

$$\varepsilon_n = 2\,\frac{1+\log_2\log_2(2n)}{\log_2 n}$$

The value of ε_n ought to be measured for small values of n, whose value varies between 0 and 1, having its value around 0.5 for n being 1000000. Hence the value of ε_n cannot be ignored for finite sequences, where the results are not accurate in the measure of randomness. It shows that the upper limit is underestimated, which illustrates that the normalized complexity *C(n)* is overestimated [2]. Based on the above analysis, following are the constraints to be considered to evaluate accurate measure of Lempel-Ziv algorithm:

1. The critical value of n when $(1-\varepsilon_n)\log_2 n$ is greater than zero.
2. The assumption of sufficient sequence length.

The new LZ complexity measure is given by,

$$C(n) = \frac{(1-\varepsilon_n)c(n)\log_2 n}{n}$$

Upon the calculation of new complexity measure, the randomness can be determined appropriately even for the short series whose length is significantly much less than ∞.

C. Illustration of obtaining normalized complexity measure C(n):

To illustrate this procedure, the sequence of eleven (n = 11) symbols S = 01010010011 is considered. Enhanced Lempel Ziv's algorithm parses the sequence into six substrings {0, 1, 01, 00, 10, 011} rendering *c(n* = 11) = 6.

The notation S(i) is used to identify the *i*th bit in the string S. The algorithm parses S from left to right looking for substrings that are not present in the superset U. As the algorithm proceeds and the superset is grown, the first substring seen from left to right is S(1) = 0, and **U** = {0}. Then S(2) = 1 is parsed and added to **U**. Hence **U** = {0, 1}. The next bit is S(3) = 0, already present in **U**, S (4) = 1 is appended to S(3), interpreting substring 01. As 01 is not present in the superset, it is included to **U**= {1, 0, 01}. The next bit S (5) = 0 is included in **U**, so S (6) = 0 is appended to it. The resulting substring (00), not present in **U**, is therefore added to the superset: **U**= {1, 0, 01, 00}. As the algorithm proceeds, the next two bits S (7) = 1 and S (8) = 0 are parsed. Similarly S(9) = 0 and S(10) = 1 are parsed. As the resulting substring 01 is part of **U**, S (11) = 1 is appended to it, rendering 011. That value is added to the superset, yielding **U** = {1, 0, 01, 00, 10, 011}. The size of the superset **U** is taken to be the complexity measure *c(n*=11) = 6, *w*hich is also the number of steps required to construct **U**.

Enhanced Lempel-Ziv complexity measure *c(n)* of a binary random sequence is in the limit equal to $b(n) = n/(1-\varepsilon_n)\log_2(n)$. Dividing *c(n)* by *b(n)* gives the normalized Enhanced Lempel-Ziv Complexity measure *C (n).*

$$C(n) = \frac{c(n)}{b(n)}$$

The normalized *C(n)* represents the rate of occurrence of new substrings in the sequence. *C(n)* values go from close to 0 (for deterministic/ periodic sequences) to 0.4 (for totally random sequences).

IV. Test Case and Results

A. Test Data

Three different types of datasets are generated for testing and to obtain the comparative results, specifically, random dataset, dataset with systematic errors in rows, and dataset with systematic errors in columns. All three types of datasets with different percentage of errors are used to obtain the test results respectively. For each of the data type and percentage of errors, 10 different data sets are generated. Particularly, for the size of 5,000 samples, 30 different test datasets are generated for each of different percentage of errors varying from 5% to 20% and another 9 datasets at 50% are considered for random type of errors, systematic errors in rows and systematic errors in columns. Same strategy is applied for 10,000 data sizes.

B. Test Results

With the algorithm, a total of more than 120 different datasets is analyzed for each of the sample size. Each type of data is tested by 10 different sets in each sample size with particular percentage of error.

Figure 3 reflects the average *C(n)* scores for different type of datasets in different sample sizes for 10% of errors in each dataset. It can be clearly understood from the figure, that both systematic type of errors (rows and columns) have lower *C(n)* scores related to the random type of errors. From the experiment, it is also clear that the selected sample sizes should have a minimum value of n >= 5000 as the differentiation is quite difficult for lower values of n. The same experiment is conducted on the datasets having 5%, 15% and 20% errors respectively. Unlike in the dataset with 5% errors where the random errors are not clearly differentiated as shown in Figure 4, the algorithm distinguished between random and systematic errors in the datasets with 15% and 20% errors. For the dataset with 15% errors, the average values of *C(n)* are 0.19, 0.13, and 0.25 for systematic errors in rows, columns and random errors respectively. The average values of *C(n)* for the dataset having 20% errors are 0.19, 0.14, and 0.28 for systematic errors in rows, columns and random errors respectively. The standard deviation for these values is in the order of 10^{-4}, showing the algorithm as effective. It is a clear indication that the Enhanced Lempel-Ziv complexity can differentiate the random errors from systematic errors.

As also expected, with the increase of data size, the *C(n)* value of systematic type of data will vary from 0.1 to 0.2 while for random errors the value is greater than 0.23 for different datasets. As the percentage of errors increases to a very high value beyond 50%, the algorithm still distinguishes between random errors and systematic errors but with different threshold values. With the tests on different datasets having 50% errors, the average value of *C(n)* for systematic errors in columns is nearly 0.07 while for systematic errors in rows has an average value of 0.2. For the random errors the average value of *C(n)* is 0.37. Though random errors have highest value it cannot be restrained once the errors increase beyond 50%.

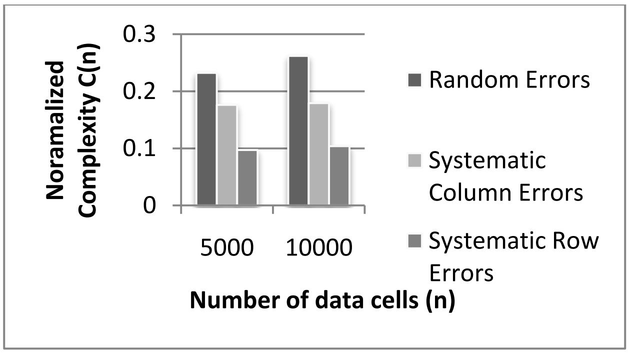

Fig 3. Test Results
(C(n) for the different datasets having 10% of errors)

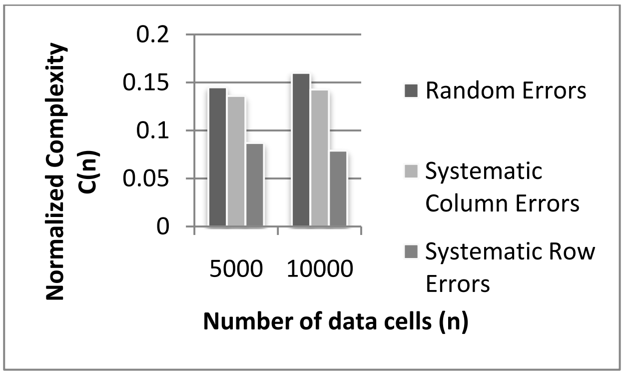

Fig 4. Test Results
(C(n) for the different datasets having 5% of errors)

The Enhanced complexity also differentiates between the percentages of errors by giving highest complexity measure for high percentage of errors. As shown in Figure 5, the databases with high percentage of random errors have high complexity measure than the databases with low percentage of errors for different data set sizes.

Hence, with these results it would be easier to answer the managerial questions to consider the prioritization of databases, which may require the processing proximately.

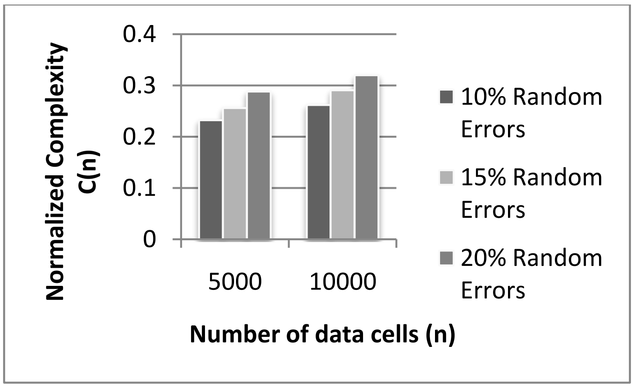

Fig 5. Test Results
(C(n) for the different datasets having different error percentages)

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Figure 6 clearly reflects that at any given percentage of errors the random errors have highest complexity and for any given type of errors the value of complexity measure is high for high percentage of errors in a given dataset having same number of data cells. But as the percentage of errors reach closer to or beyond 50%, due to high errors organization and hence it would be hard to prioritize the datasets in such scenario.

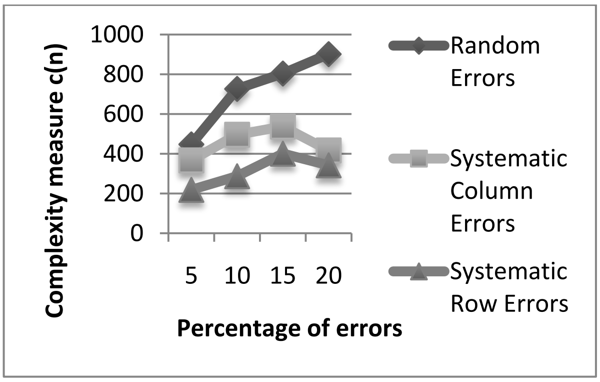

Fig 6. Test Results
(c(n) for the different datasets having 10000 data cells)

V. Conclusion

Enhanced Lempel-Ziv Algorithm is an effective and efficient way to obtain the degree of randomness in a dataset, without over-estimating the normalized complexity, *C(n)*, for moderate data sizes. It is simple and fast algorithm that renders an intuitive measure of the degree of randomness of the data errors. It is well instituted in conceptual principles and has vast applications in practical fields. It is one of the most prominent applications to assess the random number generators by The Computer Security Division of the National Institute of Standards and Technology (NIST), a key component of all modern cryptographic systems.

The tests also have been performed on sample sizes from 500 to 4000, which cannot differentiate the random errors from systematic errors due to data size constraint. The above proposed algorithm predominantly differentiates between the errors when the minimum sample size is 5000.

The proposed method is a significant step in comparing databases. In the near future, a probability distribution function as means of characterizing random distribution errors may help to monitor and benchmark the quality status of datasets and may assist to obtain the prioritization of datasets even for more percentage of errors.

References

[1] T. Redman. "Measuring Data Accuracy, in Information Quality," p. 265. Armonk, NY: R.Y. Wang, et al., , 2005.

[2] Y.W. Lee, L. L. Pipino, J. D. Funk, and R. Y. Wang, “Journey to Data Quality,” Cambridge: MA: MIT Press., 2006.

[3] C.W. Fisher and E. J. M. Lauria, “An Accuracy Metric: Percentages, Randomness and Probabilities,” ACM Journal of Data and Information Quality, Vol. 1, No. 3, Article 16, December 2009.

[4] A.H. Lhadj, “Measuring the Complexity of Traces Using Shannon Entropy,” Fifth International Conference on Information Technology: New Generations, 2006.

[5] F. Liu and Y. Tang, “Improved Lempel-Ziv Algorithm Based on Complexity Measurement of Short Time Series,” Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), 2007 IEEE.

[6] W. Kinser, “Single-Scale Measures for Randomness and Complexity,” Proc. 6th IEEE Int. Conf. on Cognitive Informatics (ICCI’07), 2007 IEEE.

[7] R. Falk and C. Konold. "Making Sense of randomness: Implicit encoding as a bias for judgment." 104: p. 301-318. Psychological Review, 1997.

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)