ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 11(40), DOI: 10.17485/ijst/2018/v11i40/120367, October 2018*

An Enhanced Low Frequency Discretizer (ELFD) in Data Cleansing Stage

Che Li[1]*, Abeer Alsadoon[1], P.W.C. Prasad[1] and A. Elchouemi[2]

[1]Charles Sturt University, Business, Justice and Behavioural Sciences Faculty, Sydney Campus, Australia; chandanapw@yahoo.co.uk, aalsadoon@studygroup.com, cwithana@studygroup.com
[2]Information Technology Faculty, Ashford University, San Diego, California, USA; aelchoue@yahoo.com

Abstract

Objective: Organizations always use data to help their knowledge discovery by using data mining techniques nowadays. Discretization algorithms are the main techniques to discover knowledge in the data cleansing stage. This study is to develop an enhanced discretization algorithm to investigate the impact of data cleansing on knowledge discovery. **Methodology**: The ELFD algorithm is based on the Low Frequency Discretizer (LFD) which includes four phases: copying dataset, calculating correlation ratio, identifying cut points and discretizing datasets. Using a part of the categorical attributes is to increase the correlation ratio between a numerical attribute and each categorical attribute. We evaluate the new discretization algorithm by using health datasets compared with LFD. The classification accuracy of the discretized dataset is the major criteria for evaluating the ELFD. **Finding**: The classification accuracy of the ELFD is greater than the classification accuracy of the LFD. Accuracy is enhanced by approximately 9% with the use of the ELFD. Considering manual recording errors, the time processing of the ELFD is similar to the LFD algorithm. **Conclusion**: The ELFD adds an additional step by choosing the top 75% categorical attributes for which the correlation ratio values are largest and then calculates the correlation ratio between the numerical attribute and these categorical attributes. Using a part of the categorical attributes increases correlation ratio values so that the ELFD improves knowledge discovery from personal information contained in health records during the stage of data cleansing.

Keywords: Corrupt Data Detection, Data Discretization, Data Cleansing, Data Mining, Data Pre-processing, Missing Value Imputation

1. Introduction

Data are constantly generated and organizations use these data to help their knowledge discovery by using data mining techniques. Discretization, as a data cleansing technique, is to convert numerical values into categorical values[1,2]. Discretization, imputations and corrupt data detection algorithms are prime research areas for data cleansing and knowledge discovery through accurate data. These discretization and Imputation algorithms are the subject of this research with the domain being health data, which consist of patient information and case notes. However, corrupt data detection algorithms are not researched. High dimensional datasets are also not included in the domain of the health datasets.

Correlation is an important variable that impacts the accuracy of discretized or imputed datasets[1]. Discretization algorithms are useful for knowledge discovery by improving data accuracy[1,3] and the imputation

*Author for correspondence

algorithms have the ability to impute missing values into the dataset to increase knowledge discovery.

The purpose of this study is to improve a modified version of the LFD Algorithm. The ELFD contributes a new process of calculating correlation ratio values by using part of the categorical attributes. This process can increase the correlation ratio values between each numerical value and each categorical attribute, thereby enhancing the accuracy of the discretized datasets. The enhanced data quality is directly able to improve knowledge discovery in terms of health data in the data cleansing stage.

This literature review focuses on algorithms of discretization and Imputation which improve data accuracy so that the discretized or imputed datasets can discover knowledge at higher rates of higher accuracy. In this section, the literature review is presented, and each algorithm is described. Also, the first best solution is described with its features, diagram, limitations and its place within the research.

2. Discretization

There is a significant body of research dealing with discretization algorithms. In[1,4] developed a discretization algorithm (LFD) by using low frequency values as cut points to discretize datasets. The discretized datasets provide higher data accuracy than that of data using other pre-processing algorithms. However, the performance of the LFD for higher-dimensional datasets is not as good as that for low dimensional datasets. Furthermore, performance of the LFD decreases if the dataset has no categorical attributes. It provides the current best solution for knowledge discovery[1,5]. Hence it is worth to improve the LFD algorithm.

Also, SMDNS[4] uses the rough set theory to determine the cut points and then uses a sort method based on counting to obtain the partitions of the universe. Some useless cut points are reduced so that the data quality is improved and the processing time is reduced[4,6]. However, SMDNS causes significant information loss and needs the user's pre-defined parameters. Hence, SMDNS is not useful to this project.

Furthermore, ZDISC[7] uses the standard deviation, Z-score, k-equal width number of cut points and C4.5 to improve data quality, and determines the bins automatically so that execution time is reduced as well. Higher accuracy of data is generated by the ZDISC compared to other algorithms[2,7]. However, the ZDISC needs to assume that the widths of bins are the same. Hence, it seems that this discretization algorithm is not suitable for this project.

2.1 Imputation

A different kind of algorithm is the imputation algorithm, of which FEMI[8], is an example. It uses correlation and similarity to impute missing values. The imputation accuracy of the FEMI is greater than that of other imputation algorithms[8,9]. However, the number of clusters is determined by user input, and FEMI is not suitable for time series data. FEMI provides high performance when improving data quality in the data cleansing stage. Hence, the correlation approach is useful for this project.

Similarly, FIMUS[10] uses co-appearances, correlations, and similarity, to impute missing values. FIMUS improves the imputation accuracy of datasets and its performance is better than that of some of the other existing algorithms[10]. However, FIMUS is also not suitable for time series data. And the categorization method in FIMUS causes much information loss which can decrease data quality substantially. Hence, FIMUS is not worth improving.

Furthermore[10] developed DMI and SiMI by using only a group of records to improve data accuracy. The main component of DMI and SiMI are decision trees or decision forests and an EMI algorithm. DMI and SiMI provide high data accuracy using high correlation values. In addition, the performance of the DMI and SiMI are better than that of the EMI and IBLLS[9–11]. However, if all records contain the same value in one numerical attribute or all numerical values are missing, DMI and SiMI will not work. Furthermore, the imputation accuracy of DMI and SiMI is lower than that of some of the other algorithms, so DMI and SiMI are not useful for this project. Nevertheless, the idea of using a group of records to increase correlation values is a vital point to enhance the data accuracy.

In[10] designed a classification model, and a decision tree model, to improve calculating classification accuracy and predict the accuracy of the tree to solve the issues with the effectiveness of the DMI. The performance of the DMI is better than the performance of the SRD and the EMI[10]. The main limitation is the number of imputation algorithms and natural datasets. Only two natural datasets and two imputation algorithms are compared, so the effectiveness of DMI is not categorically proven. This evaluation model is useful because it demonstrates the process of evaluation of effectiveness in terms of classification accuracy which is the main factor in this project.

Another imputation algorithm[12] was developed to impute ICU datasets by using aligned sample values and classifications of missing data. It identifies which values are imputed and which values are deleted, which increases the overall data quality[12,13]. However, the argument of not-recoverable true missing data is not always justifiable as the deletion process causes significant bias. Hence, this algorithm is not useful for this project.

COIM[14] uses a combination of collaborative imputation strategies based on Bayesian Principal Component Analysis (BPCA) and Local Least Squares (LLS) to impute missing values in a large missing data set with missing rates. COIM improves the imputation accuracy by using both global information and local structure[14,15]. However, COIM needs to assume the rank of the model is known and the noise measurement is Gaussian. Hence, it seems that COIM is not suitable for this project.

2.2 Noisy Record Detection

Several algorithms focus on noise detection. CAIRAD[16] uses a co-appearances matrix generator to detect noisy records. CAIRAD provides higher precision and recall which in turn enhances data accuracy. However, if the original datasets have a high amount of noise values, the values of natural co-appearances are not accurate. Hence, it seems that this algorithm is not worth improving.

Also, RBRP[17] uses Produce Bins and a k-nearest neighbors algorithm to detect outliers. RBRP detects the data outliers in less processing time compared with other existing algorithms, especially in high dimensional datasets[17]. However, if the datasets have a significant number of missing values, RBRP is not suitable for processing the dataset. Hence, RBRP is not useful for this project.

2.3 State of Art

As shown in Figure 1, the LFD[1] uses a correlation ratio to rank the numerical attributes and then determines the

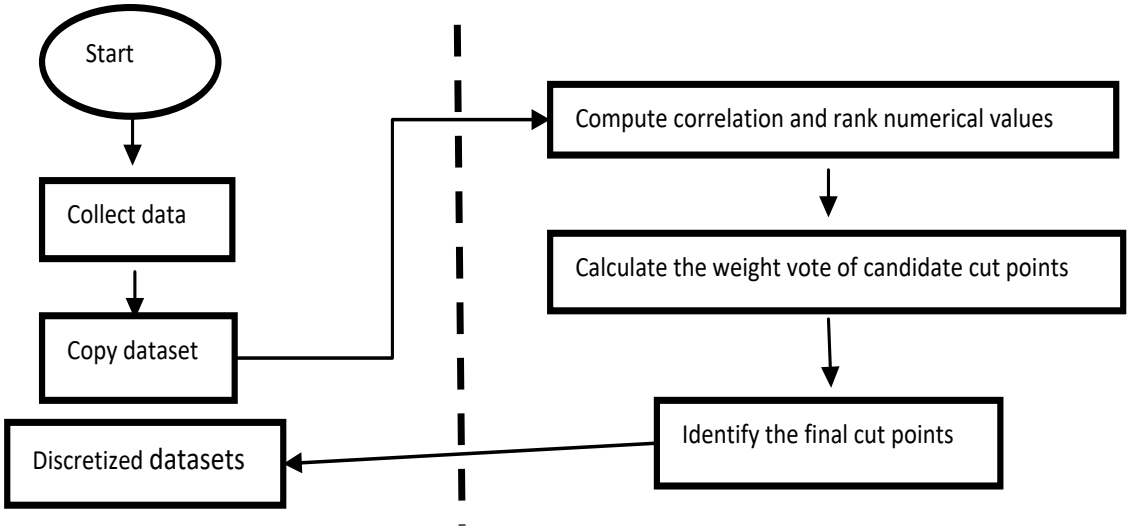

Figure 1. The LFD process.

low frequency values as cut points by correlation ratio, followed by using the cut points to discretize datasets and then return the discretized datasets.

LFD uses low frequency values as cut points because it has reduced the number of cut points and information loss in each number of the interval. When the numerical value converts into the categorical interval, each interval reduces the number of values of the cut point that is smallest. Hence, using the low frequency values as the cut point avoids data loss. This is unlike the other algorithms which assume each interval has the same width and, therefore, does not justify this argument. LFD reduces the processing time compared to other discretization algorithms because the latter need to predefine parameters to determine the cut points.

The limitation of the LFD lies in the process of calculating the correlation ratio. LFD relies heavily on the value of the correlation ratio to determine the cut points. The correlation ratio refers to the linear relationship with each attribute. The LFD computes the correlation ratio between each numerical attitude and each categorical attribute. This causes that some small absolute values of the correlation ratio are calculated into the average correlation ratio, which means into the loop of computing the correlation ratio, some absolute correlation ratio values which are close to 0 are used in the ranking of numerical attributes. Once this element of the categorical attributes ignored in the process of computing the correlation ratio, the value of the correlation ratio can be improved and the process of ranking numerical attributes becomes more accurate. Therefore, the performance of the Enhanced Low Frequency Discretizer (ELFD) is enhanced.

3. Proposed Enhanced Low Frequancy Discretizer (ELFD)

The proposed solution in this project is the Enhanced Low Frequency Discretizer (ELFD) which is to discretize the numerical attributes into categorical attributes in a dataset. The first part explains the ELFD. The justification of the process of the ELFD is discussed in the second part. The third part demonstrates the main steps of the ELFD. The fourth part in this section highlights the strengths and weaknesses of the ELFD.

3.1 Explaniation of ELFD

The improved process of the ELFD is demonstrated in Figure 2 as follows. The green dotted line illustrates the updated step. A vector of the correlation ratio, η is set, then, like in Step 2 of LFD, we calculate the correlation ratio between A_j and A_k and put into the . The top 75% categorical attributes with the highest correlation ratio value are chosen, and then these top 75% categorical attributes are used to calculate the enhanced correlation ratio between each numerical attribute and the categorical attributes. After calculating the enhanced correlation ratio, we calculate the total correlation ratio for each numerical attribute by using the formula: where is the total correlation ratio and the initial value is 0, and is each correlation ratio value in the vector of correlation ratio, η. Finally, in Step 2, the average correlation ratio values are calculated, which are used to rank each numerical attribute.

3.2 Justification of Improved Process in the ELFD

The idea of improving the LFD is based on an idea from the DMI [10] The researchers suggest improvements through an increase in the correlation to enhance the discretized data quality by calculating a part of the dataset instead of establishing correlations among the whole dataset. As mentioned in Section 2.2.4, the LFD computes the correlation ratio between each numerical attitude and each categorical attribute in the whole dataset; however, some categorical attributes have little relationship with the numerical attributes which means the correlation ratio between these kinds of categorical and numerical attributes are close to 0. Once these correlation ratio values are calculated by the process of the LFD, ranking numerical attributes can yield more bias. The correlation ratio between the numerical values and the categorical attributes is directly related to data accuracy. The higher correlation ratio values are computed by a partition of the categorical attributes, and then the data accuracy can be

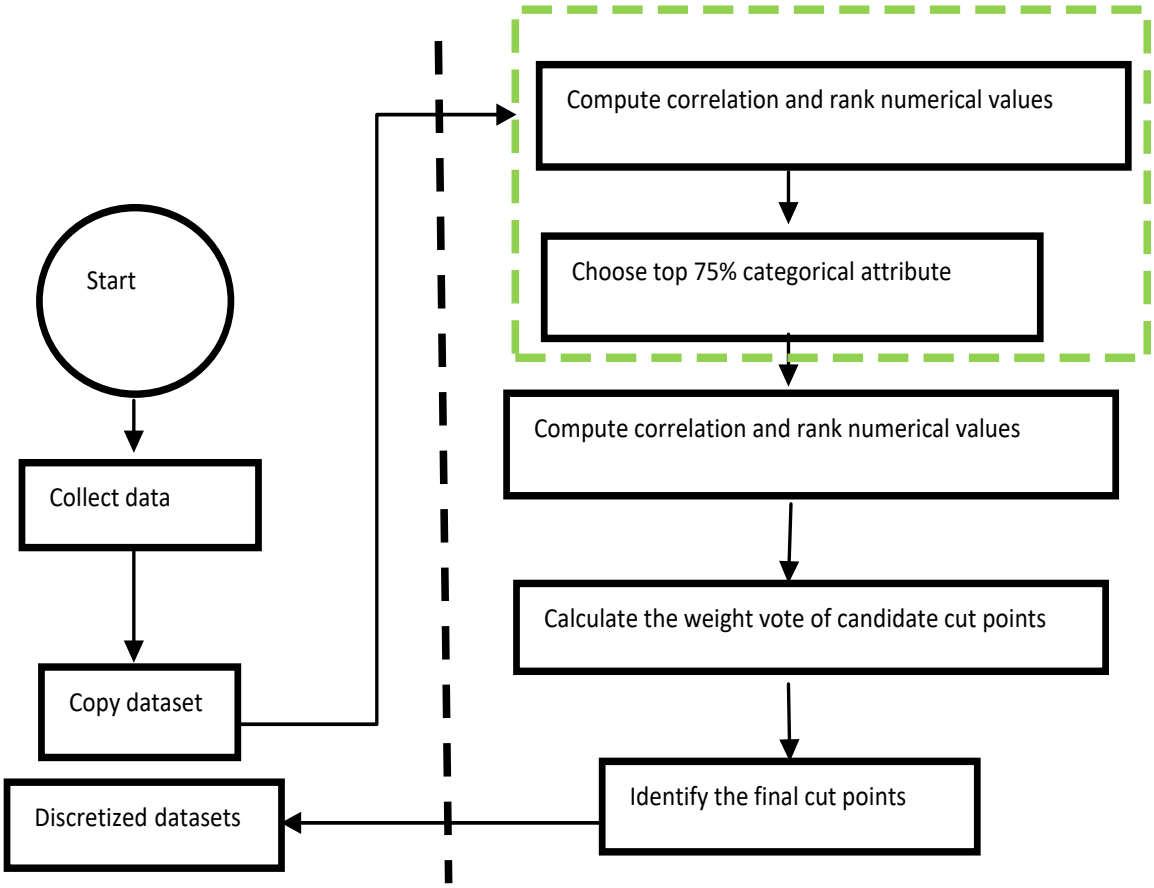

Figure 2. The process of ELFD.

improved. Therefore, the process which is improved is the 2nd step in LFD.

3.3 Main Steps

There are 4 main steps of the ELFD algorithm.

Step 1: Copy the dataset.

Step 2:Calculate the correlation ratio between each numerical attribute and each categorical attribute and rank the numerical attributes based on the correlation ratios.

Step 3:Calculate the weight, uncertainty, and vote for determining the cut point of each numerical attribute.

Step 4: Return a new discretized dataset.

The details of the above steps as the following:

Step 1:Copy the dataset.

We copy a dataset at initial, and this can protect the reality and accuracy of the original data. The original dataset D_F has N records with x attributes, and the copied dataset is D_F'.

Step 2:Calculate the correlation ratio between each numerical attribute and each categorical attribute and rank the numerical attributes based on the correlation ratios.

In Step 2 of the ELFD, the copied data D_F' is used to calculate the correlation ratio between the numerical

attribute and each categorical attribute for each numerical attribute[18]. The formula of correlation ratio η is

$$\eta^2 = \frac{\sigma_{\bar{y}}^2}{\sigma_y^2} = \frac{\frac{\sum_x n_x(\bar{y}_x - \bar{y})^2}{\sum_x n_x}}{\frac{\sum_{x,i}(y_{xi} - \bar{y})^2}{n}} \quad (1)$$

where, $\sigma_{\bar{y}}$ is the standard deviation of $\bar{y}$. σ_y is the standard deviation of y. x is the observation of the category, n_x is number of observation in x, y_{xi} is the ith observation in x, $\bar{y}_x$ is the mean of the category x, and $\bar{y}$ is the mean of the entire dataset.

Then, we choose the top 75% categorical attributes with the largest , and then use these categorical attributes to again calculate the correlation ratio. After calculating the enhanced correlation ratio, we calculate the total correlation ratio for each numerical attribute and the average correlation ratio of numerical attributes. Finally, we use the average correlation ratio to rank the numerical attributes.

Step 3:Calculate the weight, uncertainty, and vote for determining the cut point of each numerical attribute.

In step 3, we calculate the frequency of each distinct value and the average frequency in each ranked numerical attribute. A candidate cut-point vector, p is built and the values in the set of candidate cut points are the distinct values with frequency less than the average frequency, except the minimum and maximum value of each distinct value. Another set called final cut point set, F is built which includes the minimum and maximum value of each distinct value. For each distinct value in each numerical attribute, the weight, and uncertainty, between each numerical attribute and each categorical attribute are calculated by following formula[1,2] .

$$W_{ji}^k = \frac{\sum_{q=1}^{y} \frac{Max_q^2}{F_{+q}}}{y} \quad (2)$$

$$U_{ji}^k = \sum_{r=1}^{z}\sum_{q=1}^{y} p_{ji}^{qr} \log_2 \frac{1}{p_{ji}^{qr}} \quad (3)$$

where, $p_{ji}^{qr} = \frac{f_{ji}^{qr}}{N}$, f_{ji}^{qr} is the number of records that the category and each the interval of distinct value in numerical attribute co-appears, Max_q is the maximum value of f_{ji}^{qr}, F_{+q} is the total appearance of the interval of distinct value in the numerical attribute.

Subsequently, he weight and uncertainty of a candidate cut point, the weighted vote, of the candidate cut point is calculated by the formula as follows[1, 2] .

$$v_{ji} = \frac{\sum_{k=1}^{c} U_{ji}^k * W_{ji}^k}{\sum_{k=1}^{c} U_{ji}^k} \quad (4)$$

Then, we use the set of weighted votes to find the maximum vote, vmax and use the vamx to determine whether the candidate cut points are the final cut points in F. Once the final cut points are determined in the set F, we use these values in F to discretize the copied dataset D'_F.

Step 4: Return a new discretized dataset.

In Step 4, the discretized dataset of D_F' is returned, and the numerical attributes in the original dataset are converted into categorical attributes.

Algorithm: ELFD

Input: dataset with x attributes

Output: a new discretized dataset with all the numerical attributes converting to categorical attributes.

step1:

dataset1= copy dataset

end

step 2:

loop:

for each numerical attribute, A

set total correlation ratio of a numerical attribute, B=0

count=0

for each categorical attribute, C

calculate correlation ratio between A and C

sort the categorical attribute

set a newser, NewSet

if the categorical attribute is in top 75%:

NewSet = NewSet union categorical attribute, C

end

end

for each categorical attribute in NewSet

calculate correlation ratio between A and C

count=count+1

end

B=B+ each correlation ratio

average correlation ratio value=B/count

end

rank numerical attributes, A based on each average correlation ratio value

end

Step 3:

loop:

for each ranked numerical attributes, A:

build a frequency vector, f for each distinct value,

calculate average frequency vector favg

set a set of cut-points, p

set a Candidate Cut Points vector, p= a set of distinct value, J in f if the frequencies value in f <favg

Set a final cut point vector, F=the distinct values J of num1 and numN

Set a global vote, V=0

Set L=True (boolean variable)

while loop L=True:

vote, v=empty set

set a vector of temporary interval I={[the distinct values J of num1, the distinct values J of num1 and numN]}

for each distinct value, J in p, but not in F

put J into I

for each categorical attribute, C

Calculate weight, w

Calculate uncertainty, u

end

Calculate vote, v'= Sum(w*u)/sum(u)

vote set, v=v union v'

end

find the maximum vote, vmax

if vmax>= V **then**

Jmax=the relevant distinct value, J (use index)

V=vmax

F=F union Jmax

p=p-Jmax (p and F are vector, vector p removes Jmax)

else

L=False

end

while loop end

new dataset, Output=discretize each numerical attribute A using vector F

Output= Output union C

end

Step 4

return the new dataset, Output

end

3.4 Strengths of The FLFD

The ELFD algorithm applies the correlation ratio twice to increase data accuracy. When the first correlation ratio between the numerical attributes and each categorical attribute had been established, the top 75% categorical attributes with the largest , were chosen. Then these categorical attributes were used to calculate the correlation ratio again. The correlation ratio between the numerical attributes and categorical attributes is proportional to the data accuracy and the correlation ratio among a part of the dataset is greater than that among the whole dataset. Once the top 75% categorical attributes with largest are chosen, the correlation ratio values are enhanced, and then the data accuracy is increased.

4. Implementation of ELFD Model

Implementation of the model consists of deployment of the LFD algorithm and the ELFD algorithms by using the C# language on the Microsoft Visual Studio Community 2017 platform. The 10 sample datasets (Table 1) are applied to test the algorithms, and all datasets come from different online dataset resources. The major impact factor in this project is the data accuracy, which is calculated by XLMiner software.

4.1 Data Set Information

We use the 4 real datasets from the UC Irvine Machine Learning Repository[19]. Samples 1, 2, and 3 are artificial datasets from heart disease dataset; samples 4, 5 and 6 are artificial datasets from hepatitis datasets, samples 7 and 8 come from post-operative datasets, and samples 9 and 10 come from a diabetes dataset. We test the performance of the ELFD based on the number of records, the number of numerical attributes, the number of categorical attributes, whether the dataset has missing values or not, and the number of pure records in Table 1.

4.2 How the FLED Model is Implemented

The performance of the ELFD is based on classification accuracy. In the initial stage, for each real dataset, we artificially change the features of the dataset, so that the features of samples which influence the performance of these algorithms are evaluated. Then, we use the program of the LFD and ELFD to discretize the samples with different features and then record the processing time. After that, we use the Naïve Bayes to calculate the discretized data accuracy of the LFD and ELFD.

Different features of samples are designed to evaluate which data feature influences the performance of the ELFD. Sample 1 and sample 2 demonstrate the influence of the performance of the LFD and ELFD on the number of numerical attributes. Sample 2 and sample 3 demonstrate the influence of the performance of the LFD and ELFD on the number of records. Sample 4 and sample 5 evaluate the influence of the performance of the two algorithms from the aspect of categorical attributes. Sample 7 and sample 8 evaluate the influence of the performance of the two algorithms depending on whether a dataset has missing values or not. Sample 9 and sample 10 are designed to evaluate the performance of the two algorithms when there is only one categorical attribute in the dataset.

The XlMiner can directly calculate the error of the dataset, and then the discretized data accuracy is calculated by the following formula:

$$\text{Accuracy} = 1 - \text{error} \tag{5}$$

The following section (Section 5) demonstrates the result from the LFD algorithm and the ELFD.

Table 1. Dataset features

	Features of samples (different feature of samples)				
Sample	**Records**	**Numerical attribute**	**Categorical attribute**	**Missing**	**Pure records**
1	720	4	6	Yes	307
2	720	6	6	Yes	307
3	500	6	6	Yes	482
4	155	6	10	Yes	150
5	155	6	13	Yes	150
6	134	6	13	No	134
7	90	1	7	Yes	87
8	87	1	7	No	87
9	768	8	1	No	1597
10	768	6	1	Yes	455

5. Experimental Results

Table 2 is the results table which demonstrates the performance of the LFD algorithm and the ELFD algorithm for 10 sample datasets. The classification accuracy is calculated for each discretized dataset which is generated by the LFD algorithm and the ELFD. Each row in Table 2 represents the performance of the LFD algorithm and the ELFD algorithm for each sample.

The classification accuracy of the ELFD for sample 1 is 0.779 which means the discretized dataset which is yielded from the ELFD algorithm achieves 77.9 per cent accuracy from XIMiner software. From Figure 3, the average classification accuracy of the ELFD is around 9% greater than that of the LFD, so that we argue that the performance of the ELFD is better than that of the LFD because of the additional process of correlation ratio calculation in step 2. Figure 3 represents the classification accuracy difference between the ELFD and the LFD, and it also demonstrates that the performance of the ELFD is better than that of the LFD from the aspect of classification accuracy. The correlation ratio is an important variable to increase the data quality. A better quality discretized dataset impacts the classification accuracy of health records and therefore the accuracy of knowledge discovery. Figure 4 depicts a bar chart of the accuracy of the LFD and the ELFD for each sample. The red bar represents the classification accuracy of the ELFD and the blue bar represents the classification accuracy of the LFD. Compared with the classification accuracy of the two algorithms in Figure 5, all the classification accuracies of

Table 2. Result table

	LFD		ELFD	
Sample	**Accuracy**	**Processing time**	**Accuracy**	**Processing time**
1	0.660	9.18	0.779	8.76
2	0.614	27.7	0.725	27.63
3	0.620	22.63	0.732	22.4
4	0.794	25.55	0.884	24.92
5	0.787	34.97	0.869	33.71
6	0.835	33.76	0.902	32.95
7	0.722	1.15	0.814	1.22
8	0.736	0.87	0.827	0.83
9	0.755	42.75	0.853	40.30
10	0.747	37.13	0.838	37.3

the ELFD are greater than the classification accuracies of the LFD.

From the result of sample 1 and sample 2, the number of numerical attributes does influence the performance of the ELFD. When the number of numerical attributes

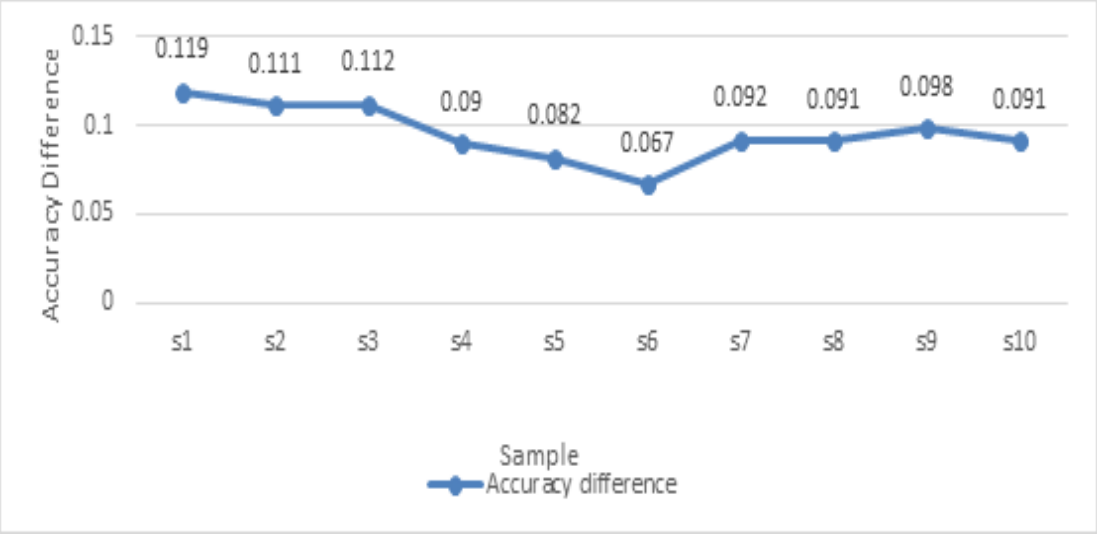

Figure 3. Accuracy difference between ELFD and LFD.

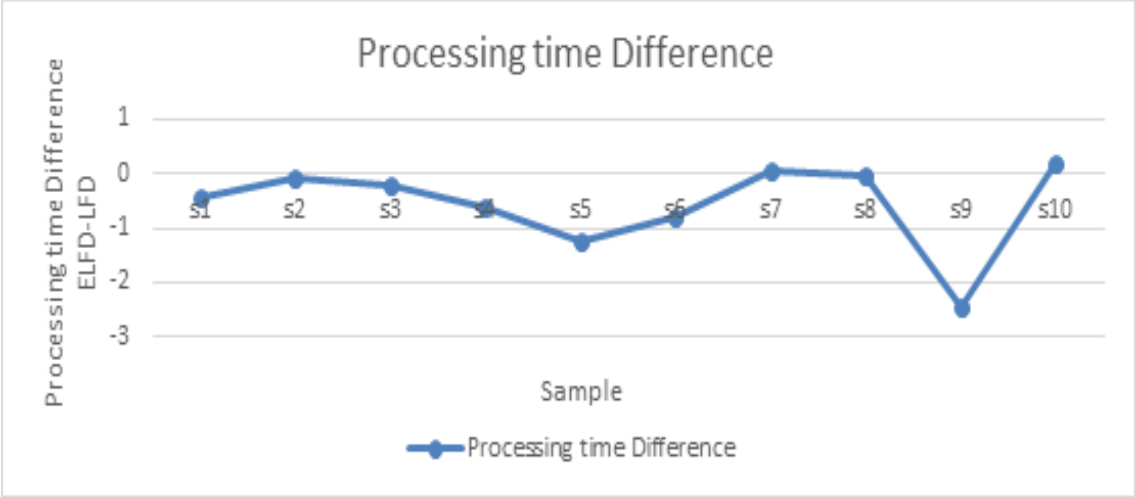

Figure 4. Processing time difference between ELFD and LFD.

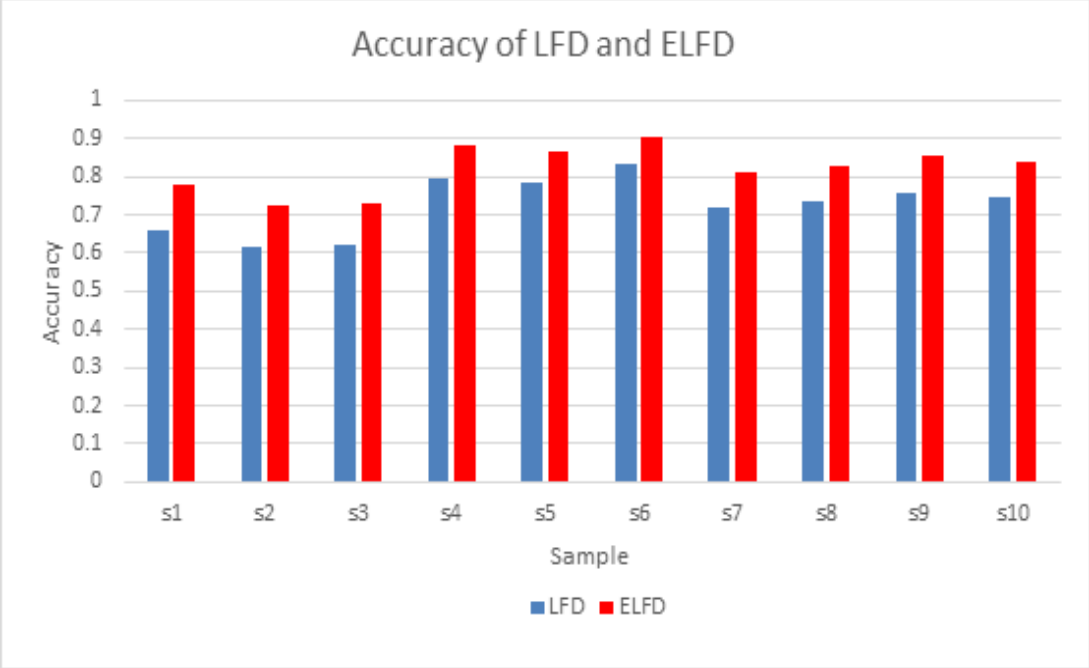

Figure 5. Accuracy of LFD and ELFD.

increases, the classification accuracy decreases and the ELFD needs more time to process. Compared with sample 2 and sample 3, the number of records influences the quality of the discretized dataset by using the ELFD negatively. Since the accuracy of sample 4 is greater than the accuracy of sample 5, it demonstrates that the number of categorical attributes is inversely proportional to the discretized data accuracy, and the increased number

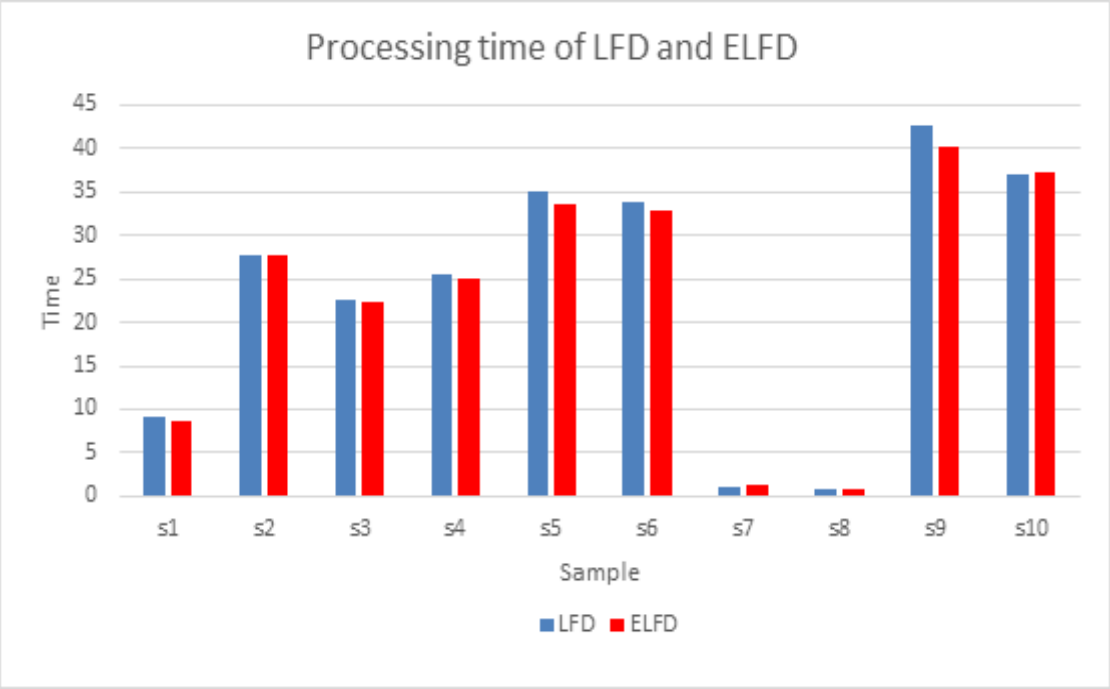

Figure 6. Processing time of the LFD and the ELFD.

of categorical attributes decreases the correlation ratio between each numerical attribute and each categorical attribute and the vote between each numerical attribute and each categorical attribute. Compared with the features and accuracy of sample 7 and sample 8, we argue that the discretized data accuracy of the dataset having no missing value is minimally greater than that of the dataset having missing values when using the ELFD to discretize the dataset. From the features and accuracy of sample 9 and sample 10, we argue that the ELFD achieves satisfactory outcomes when the dataset has only one categorical attribute.

The second major impact factor is processing time. As the result of the processing time established in Figure 4, each dot value of the time-consuming difference of the LFD and the ELFD are close to zero. Figure 6 shows a bar chart of the processing time of the LFD and the ELFD for each sample. The red bar represents the processing time of the ELFD and the blue bar represents the processing time of the LFD. Each pair of bars represents that the processing time of these two algorithms is similar. Considering the manual recording error, we argue that the processing time of the LFD is similar to the ELFD.

6. Conclusion

The use of discretization algorithms is vital to the investigation of the impact of data cleansing on knowledge discovery from personal information in the domain of health records. The current best solution to discover knowledge in the data cleansing stage is the LFD, but it still has limitations. The purpose of this research is to develop an updated version of the LFD algorithm called the ELFD to discover knowledge from health datasets during the data cleansing stage. The limitation of the LFD lies in the process of calculating the correlation ratio. This process causes that some small absolute values of the correlation ratio between a numerical attribute and each categorical attribute are calculated into the average correlation ratio. Such categorical attributes have less relationship with the numerical attribute. Based on this limitation, the ELFD is to add the additional step to enhance the cor-

relation ratio values which is proportional to the data accuracy. The additional step is to choose the top 75% categorical attributes for which the correlation ratio values are largest and then calculate the correlation ratio between the numerical attribute and these categorical attributes.

In this study, 10 sample datasets have been applied to evaluate the performance of the ELFD and the LFD. Based on the results of the experiment, the discretized data accuracy of the ELFD is greater than accuracy produced by the LFD. The ELFD enhances the average classification accuracy by 9%, compared with the LFD. The number of records, number of numerical attributes, and number of categorical attributes and whether the dataset has missing value or not are the dataset features which impact the knowledge discovery in the data cleansing stage. Considering the manual recording error, the time consumption of the ELFD is similar to that of the LFD. However, the ELFD has some limitations. One is that consumption is high when the dataset has an enormous number of records, and high numbers of distinct values in each numerical attribute, or an enormous number of categorical attributes. Another limitation is that the dataset has at least one categorical attribute.

For future research, we plan to reduce the time required for the processing of large datasets. In addition, we plan to discretize datasets without categorical attributes. We also need to add time recording algorithms to account for the processing time more accurately.

7. References

1. Rahman G, Islam Z. Discretization of continuous attributes through low frequency numerical values and attribute interdependency. Expert Systems with Applications Journal. 2016; 16(1):410–23. https://doi.org/10.1016/j.eswa.2015.10.005.
2. Kurgan L, Cios K. CAIM Discretization Algorithm. IEEE Transaction on Knowledge and Data Engineering Journal. 2004; 16(2):145–52. https://doi.org/10.1109/TKDE.2004.1269594.
3. Pyle D. Data preparation for data mining. Morgan Kaufmann; 1999.
4. Jiang F, Sui Y. A novel approach for discretization of continuous attributes. Knowledge-Based Systems Journal. 2014; 73:324–34. https://doi.org/10.1016/j.knosys.2014.10.014.
5. Garcia S, Herrera F. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. IEEE Transactions on Knowledge and Data Engineering Journal. 2013; 25(4):1–25. https://doi.org/10.1109/TKDE.2012.35.
6. Ching J, Wong A, Chan K. Class-dependent discretization for inductive learning from continuous and mixed-mode data. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1995; 17(7):641–51. https://doi.org/10.1109/34.391407.
7. Madhu G, Rajinikanth TV, Govardhan A. Improve the classifier accuracy for continuous attributes in biomedical datasets using a new discretization method. Procedia Computer Science; 2014. p. 671–9. https://doi.org/10.1016/j.procs.2014.05.315.
8. Rahman G, Islam Z. Missing value imputation using a fuzzy clustering-based. Springer London. 2015; 46(2):389–422.
9. Junninen H, Niska H, Tuppurainen K, Ruuskanen J, Kolehmainen M. Methods for imputation of missing values in air quality data sets. Atmospheric Environment. 2004; 38(18):2895–907. https://doi.org/10.1016/j.atmosenv.2004.02.026.
10. Rahman G, Islam Z. Data Quality Improvement by Imputation of Missing Values. CSIT. 2013; 38(18):82–8.
11. Cheng K. Iterative bicluster-based least square framework for estimation of missing values in microarray gene expression data. Pattern Recognition; 2012. p. 1281–9. https://doi.org/10.1016/j.patcog.2011.10.012.
12. Cismondi F, Fialho AS, Vieira SM, Reti SR, Sousa JM, Finkelstein SN. Missing data in medical databases: Impute, delete or classify? Artificial Intelligence in Medicine. 2013; 58(1):63–72. https://doi.org/10.1016/j.artmed.2013.01.003. PMid:23428358 .
13. Allison PD. Missing data. Sage University papers series on quantitative applications. Thousand Oaks: SAGE Publications; 2001.
14. Li H, Shao F, Li G. Semi-supervised imputation for microarray missing value estimation. IEEE International Conference on Bioinformatics and Biomedicine (BIBM); 2014. p. 297–300. https://doi.org/10.1109/BIBM.2014.6999172.
15. Meng F, Cai C, Yan H. A bicluster-based bayesian principal component analysis method for microarray missing value estimation. IEEE journal of Biomedical and Health Informatics; 2014. p. 1–863.

16. Rahman G, Islam Z, Bossomaier T, Gao J. CAIRAD: A co-appearance based analysis for incorrect records and attribute-values detection. IEEE World Congress on Computational Intelligence. 2012:2190–9. https://doi.org/10.1109/IJCNN.2012.6252669.
17. Ghoting A, Parthasarathy S, Otey ME. Fast mining of distance-based outliers in high-dimensional datasets. Data Mining and Knowledge Discovery Journal. 2008; 16(3):349–64. https://doi.org/10.1007/s10618-008-0093-2.
18. Fisher RA. Statistical methods for research workers. Genesis Publishing; 1925. PMid:17246289
19. UC Irvine Machine Learning Repository [Internet]. [cited 2010 Feb]. Available from: http://archive.ics.uci.edu/ml/index.php. Date accessed: 02/2010.