Hungarian Association of Agricultural Informatics
European Federation for Information Technology in Agriculture, Food and the Environment

Journal of Agricultural Informatics. Vol. 8, No. 1
journal.magisz.org

Comparison of Chi-square based algorithms for discretization of continuous chicken egg quality traits

Zeynel Cebeci[1], Figen Yildiz[2]

I N F O
Received 13 Nov. 2016
Accepted 21 Feb. 2017
Available on-line 15 Mar. 2017
Responsible Editor: M. Herdon

Keywords:
Data pre-processing, Supervised discretization, ChiMerge, Chi2, Extended Chi2, Modified Chi2.

A B S T R A C T

Discretization is a data pre-processing task for transforming continuous variables into discrete ones. In this study, four Chi-square based supervised discretization algorithms (ChiMerge, Chi2, Extended Chi2 and Modified Chi2) were compared for discretization of the fourteen continuous variables in a chicken egg quality traits dataset. We found that all of the algorithms had similar performances in term of training model accuracies obtained with C5.0 classification tree algorithm whereas ChiMerge and Chi2 were better than the remaining algorithms in term of training error rates. The numbers of intervals obtained with Chi2 tended to be large while they were very small in Extended Chi2 and Modified Chi2. The numbers of intervals from ChiMerge increased as the significance level increases whereas they were the same at all the levels of significance for the remaining algorithms. Consequently, it was revealed that ChiMerge at the significance levels of 0.05 and 0.10 was more efficient than the others and could be a better choice in discretization of the egg quality traits.

1. Introduction

Data mining is the collection of numerous methods and techniques to reveal meaningful patterns, valid and useful information in massive volumes of data. In many data mining applications such as feature selection, classification and association rules extraction, the majority of the algorithms have primarily been developed to run on discrete or categorical variables. On the other hand, the data are generally continuous and/or mixed type in many fields of study. Therefore, a discretization process is needed to turn continuous variables into discrete ones by splitting their range of values into a finite number of subranges called intervals, buckets or bins. As example of a continuous variable, the air temperature (^{o}C) can be transformed into three intervals as: (1) *low ($\leq$15)*, (2) *medium (16-29)*, (3) *high ($\geq$30)*. As in the temperature variable example, continuous variables are divided into finite numbers of intervals that are treated as categories by a discretization algorithm.The number of intervals produced in a discretization process is equal to the number of cut-points plus one. The minimum number of intervals for a continuous variable is equal to 1 while the maximum number of intervals is equal to the number of instances in a dataset.

In broad sense, a typical discretization consists of two stages. The first stage is a four-step task comprising of: (1) sorting values of continuous variables, (2) evaluating a cut-point for splitting or merging adjacent intervals, (3) splitting or merging intervals according to some criterion, and (4) stopping at some point depending on a termination criterion (Dash *et al.* 2011; Hemada & Lakshmi 2013; Kotsiantis & Kanellopoulos 2006; Liu *et al.* 2002). The second stage of discretization includes re-encoding all the values in the intervals. In this stage, each interval is labelled with a discrete value, and then the continuous values within an interval are mapped to the discrete value of corresponding interval. For discretization of continuous variables many discretization methods (or simply discretizers) had been developed. Although they are usually classified as supervised and unsupervised, they can also be classified in many different axes such as: (a) static versus dynamic, (b) local versus global, (c) bottom-

[1] Zeynel Cebeci
Div. of Biometry & Genetics, Faculty of Agriculture, Çukurova University, 01330 Adana - Turkey
zcebeci@cu.edu.tr
[2] Figen Yildiz
Div. of Biometry & Genetics, Faculty of Agriculture, Çukurova University, 01330 Adana - Turkey
yildizf@cu.edu.tr

up versus top-down, and (d) direct versus incremental (Kotsiantis & Kanellopoulos 2006; Ramírez-Gallego *et al.* 2015).

The supervised algorithms use priori known class labels information while unsupervised methods do not use such kind of information. In the static discretization algorithms, number of intervals is determined for each variable independently. Contrarily, the dynamic algorithms determine a possible number of intervals for all variables simultaneously. Since the multivariate algorithms capture interdependencies in discretization an overall improvement is expected in quality of discretization (Tay & Shen 2002). On the other hand, the static algorithms work for one variable at a time and thus they are synonymously called as the univariate algorithms. The dynamic algorithms are multivariate algorithms because they process multiple variables simultaneously. The local discretization algorithms use the local parts of instances space (subsets of instances) while the global algorithms run for the whole instances space (Chmelewski & Grzrmala-Busse 1996). The bottom-up algorithms (merging algorithms) are initialized with a complete list of all values as cut-points, and merge intervals by selecting the best cut-points. The top-down algorithms (splitting algorithms) start an empty list of cut-points and one interval covering all the values of a variable, and then divide this wide interval into smaller intervals with the best cut-points until a determined stopping criterion is reached. The direct algorithms for discretization need a user-defined number of intervals (*k* parameter) in discretization of continuous variables. In contrast to this disadvantage of the direct algorithms, the incremental algorithms do not require users to enter *k*. They start with a frontier discretization step and then search the best intervals in recursive improvements until a stopping criterion is satisfied.

Beyond it is necessarily needed by several data mining algorithms; discretization may also reduce the system memory requirement and shorten the execution time of the algorithms. Additionally, the information explored from discretized variables may be more compact and easily interpretable (Dash *et al.*, 2011; García *et al.* 2013; Gupta *et al*. 2010; Sang *et al*. 2013). In spite of its above mentioned advantages, discretization generally leads to certain level of information loss. Therefore, minimizing such loss is one of the main goals in developing discretization algorithms (García *et al*. 2013).

According to the surveys by Dougherty *et al*. (1995), Liu *et al*. (2002), Kotsiantis & Kanellopoulos (2006), García *et al.* (2013), and finally the advanced review by Ramírez-Gallego *et al*. (2015), many different discretization algorithms have been proposed in the last two decades. García *et al*. (2013) concluded that the most common techniques had been Equal-width Discretization (EWD) and Equal-frequency Discretization (EFD), MDLP, ID3, ChiMerge, 1R, D2, and Chi2. Among these, EWD and EFD are common unsupervised discretization methods due to their simplicity and availability in many data mining applications. However, they are direct algorithms that need an optimal *k* parameter (the number of intervals) for each variable before going to discretization process. Additionally, they have some other disadvantages such as having same values in different intervals and sensitivity to outliers. As an unsupervised alternative, although the K-means clustering algorithm overcomes the same value problem it is still sensitive to outliers. There is no superior algorithm for all of the data types yet the use of supervised algorithms may provide some advantages over unsupervised discretization, for istance they do not require user-defined parameters. On the other hand, the supervised algorithms have some disadvantages such as increase in time complexity which is a most common metric for measuring cost of an algorithm. For example, the time complexity is *O(n)* for EWD whereas it is *O(n log(n))* for ChiM based algorithms (Dash *et al*. 2011).

In the Chi-square based discretization algorithms, χ^2 statistic in Equation 1 is used to test the null hypothesis that two adjacent intervals are similar at a given significance level (α). When the adjacent intervals are independent they are merged, otherwise they are left separate.

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{k}\frac{\left(O_{ij}-E_{ij}\right)^2}{E_{ij}} \qquad (1)$$

Where:

m : Number of intervals to be compared (usually *m=2*)

k : Number of classes

O_{ij}: Observed number of instances in i^{th} interval and j^{th} class

E_{ij}: Expected frequency of instances in i^{th} interval and j^{th} class $(= (r_i * c_j)\ /n)$

r_i : Number of instances in i^{th} interval ($= \sum_{j=1}^{k} O_{ij}$)
c_j : Number of instances in j^{th} class ($= \sum_{i=1}^{m} O_{ij}$)
n : Total number of instances ($= \sum_{j=1}^{k} c_j$)

In general, the Chi-square based algorithms are modifications or improved versions of ChiMerge (ChiM) algorithm. ChiM is a supervised and local merging algorithm applying the χ^2 testing in order to discretize continuous variables (Kerber, 1992). The algorithm performs discretization in two steps that are called the initialization and the bottom-up merging. In the initialization step, each distinct value of a continuous variable is assumed as an independent interval, and then χ^2 statistic is tested for whether the adjacent intervals to be merged or not. When the χ^2 statistic for adjacent intervals is greater than the predefined threshold level (χ^2_α), (if $\chi^2 > \chi^2_\alpha$), adjacent intervals are merged because they are assumed statistically similar. The χ^2 testing can be performed for adjacent interval pairs until a stopping criterion is satisfied.

Chi2 algorithm developed by Liu & Setiono (1995) is an extension of ChiM. This algorithm automates discretization by defining an inconsistency rate as stopping criterion and selects the statistical significance level automatically. It merges more adjacent intervals until the inconsistency criterion is satisfied. Modified Chi2 (mChi2) proposed by Tay & Shen (2002) is an improved modification of ChiM and Chi2. The mChi2 is a completely automatic algorithm fixing the over-merging problem because of the user-defined inconsistency rate which is leading to inaccuracy in Chi2. This algorithm uses a consistency rate from rough set theory to control inconsistency. Extended Chi2 (eChi2) is an extension of Chi2 in which inconsistency checking in Chi2 is replaced with a lowest upper bound (Su & Hsu 2005). In eChi2, two adjacent intervals is merged without considering difference with respect to degree of freedom. For this reason, eChi2 can handle uncertainty data, and obtained discretized data may result in better predictive accuracies when compared to those obtained from Chi2 and mChi2.

The StatDisc by Richeldi & Rossotto (1995) is an improvement of ChiM generating a discretization interval hierarchy by using the measurement as the interval merging criterion. Risvik (1997) proposed the Interval Merger technique which is a generalization of ChiM algorithm that decreases number of cut-points by removing each cut-point, and merging intervals until an inequivalent threshold value is achieved. The Concurrent Merger technique also uses the χ^2 statistic and inequivalence measures (Wang & Liu 1998). Khiops algorithm proposed by Boulle (2004) includes two steps which are the initialization step and the discretization optimization step. It differs from the previous Chi-square based algorithms with its stopping criterion rule and the use of global domain. Khiops does not require any pre-determined stopping criterion since it optimizes the χ^2 criterion in a global manner on entire instances space. Qu *et al*. (2008) proposed Rectified Chi2 algorithm aiming to fix the issues with mChi2 and eChi2 algorithms. Recently, Bettinger (2011) developed ChiD algorithm based on ChiM and Chi2.

In the literature, the researchers compared and proposed some discretization methods but they mostly worked with the classical benchmark datasets from the UCI Machine Learning Data Repository. So working with the real agricultural datasets is important in order to propose an appropriate method suitable for a specific domain in practice. For this reason, a comparative analysis has been given for discretization of 14 continuous variables in a chicken egg quality traits dataset in this study. Our aim was to generate a discretized dataset in order to use in a further study mining the association rules between these traits. Although there are many discretization algorithms, none of them are optimal for every situation. Based on the comparison of accuracy of PGN-classifier trained with different discretization methods on 8 datasets, Mitov *et al*. (2009) concluded that ChiM discretization method was more efficient for PGN-classifier than other methods. At the same time, as the examples of dynamic algorithms Chi-square based algorithms detect interdependencies between variables and discretize all variables concurrently. They are also known as the non-parametric algorithms which do not require any predefined parameter. On the basis of these advantages, we decided to compare some well-known Chi-square (χ^2) based algorithms for discretization of our dataset. In order to find a good algorithm for our purpose, we compared not only ChiM but also Chi2, eChi2 and mChi2 at the different levels of significance because of their availability in computing environments. In recent years, there are a few number of researches dealing with temporal data discretization for transforming the time series

into timely intervals (Azulay *et al*, 2007; Bakar *et al*. 2010; Acosta-Mesa *et al*. 2014; Chaudhari *et al*. 2014). In this study, we did not consider the temporal order of variables even they were measured weekly since ANOVA analyses showed that the majority of response variables did not differ by the time points of measurement (weeks).

2. Materials and Methods

In this study, we used an egg quality traits dataset containing various quantitative and qualitative variables recorded for totally 4320 eggs from the 3 commercial laying chicken lines (Lines A, D and N). The data was collected from a complete randomized plot design experiment conducted at the Experimental Farm of Faculty of Agriculture in Adana, Turkey. In the experiment, from each line 10 randomly sampled chickens were allocated to totally 18 cages in a three tiered (bottom, middle and top) and two sided (aisle and window) cages system, and raised for 24 weeks in a climate controlled poultry house. At the end of each week the eggs from each cage were collected and labelled for measuring the quality traits listed in Table 1.

In the analyzed dataset, there were 14 continuous variables and 1 class variable (genotype / line) as listed in Table 1. The dataset was checked and cleaned for the missing values and outliers before discretization. Firstly, all the data rows contain the missing values for at least 50% of variables were completely discarded from the dataset. The data size was reduced from 4320 to 4272 after this deletion. PMM (Predictive Mean Matching) imputation method was used in order to impute the remaining missing values in the analyzed dataset. Vink *et al*. (2014) stated that "PMM is very flexible as a method, because of its hot-deck characteristics, and is free of distributional assumptions. Moreover, PMM tends to preserve the distributions in the data, so the imputations remain close to the data". With respect to its above mentioned advantages, we applied PMM to our dataset by using the related functions of the `mice` package (van Buuren & Groothuis-Oudshoorn 2011) in R environment. Following the imputation of missing values, the records having the outliers below `Q1-1.5IQR` and above `Q3-1.5IQR` were successively discarded for each variable. The number of records was totally 3493 (Line A: 1146, Line D: 1187, Line N: 1146) after deletion of the outliers.

Table 1. Descriptive statistics for the continuous variables in the egg quality traits dataset

Vars	Description	Mean	SD	Min	Max	CV (%)	ADT (p)	#Outliers	#Intervals
ewg	Egg weight (g)	66.16	4.91	47.68	74.72	8.19	4.28e-10***	57	30
ewd	Egg width (mm)	43.19	1.22	42.38	46.67	2.83	2.00e-02*	62	33
eln	Egg length (mm)	56.93	2.23	50.43	63.52	3.92	1.04e-14***	44	32
eph	Egg pH	8.46	0.20	7.89	9.04	2.38	8.28e-06***	31	31
sbs	Shell breaking strength	4.68	1.05	1.70	7.65	22.44	1.20e-11***	119	31
sht	Shell thickness (µm)	366.40	22.64	303.33	429.43	6.18	1.40e-03**	42	31
shw	Shell weight (g)	6.80	0.64	4.99	8.66	9.48	1.73e-01^{ns}	45	31
ywg	Yolk weight (g)	16.08	1.90	11.03	21.23	11.80	2.80e-06***	42	31
yht	Yolk height (mm)	18.36	1.07	15.43	21.26	5.84	5.60e-01^{ns}	37	31
ywd	Yolk width (mm)	39.92	2.60	32.55	47.41	6.53	1.04e-06***	62	31
yce	Yolk color index (E)	81.77	5.36	66.29	97.42	6.55	3.00e-03***	74	31
wht	White height (mm)	8.64	1.15	5.32	11.78	13.32	1.10e-03**	39	31
wwd	White width (mm)	64.85	5.53	50.04	80.18	8.53	3.70e-24***	94	31
wln	White length (mm)	85.42	7.03	66.77	104.61	8.23	2.07e-10***	31	29
gen	Genotype of chicken	Class variable has three levels: A, D, N						-	-

* p<0.05, **p<0.01, ***p<0.001, ns: not significant

According to Kaoungku *et al*. (2015) non-normal distribution may have strong effect to discretization although it needs further theoretical and experimental proofs. In order to evaluate the probable effect of the distribution types on discretization performance we tested the normality of variables. Anderson-Darling Test (Anderson & Darling, 1952) is one of the preferred normality tests since it is more sensitive to deviations in the tails of the distribution and applicable for any type of data distribution. We used the `nortest` library (Gross & Ligges 2015) in R environment (R Core Team, 2015) for testing the normality of variables in our dataset. As seen from the ADT (p) column in Table 1, the variables shell weight (*shw*) and yolk height (*yht*) were normal and the remaining variables were non-normal ($p<0.05$). The coefficients of variation (CV% in Table 1) were 22.44%, 13.32% and 11.80% for the variables shell breaking strength (*sbs*), white height (*wht*) and yolk weight (*ywg*) respectively, and were under 10% and close to each other for the remaining variables.

In regard of a rough evaluation of the success of Chi-square based algorithms we aimed to compare the number of intervals generated by the studied algorithms to those of an unsupervised discretization method. For this aim we chose the Equal Width Discretization (EWD) as the representative of the unsupervised methods because of its simplicity and popularity in many studies. For EWD, however several rules such as Sturges', Scott's and Freedman-Diaconis do exist to obtain the interval width of variables ($h(x_i)$) Freedman-Diaconis rule (Freedman & Diaconis 1981) is one of the more informative rules due to inclusion of interquartile range statistic in calculation of interval widths as shown in Equation 2. Thus we generated the reference number of intervals by using EWD with the Freedman-Diaconis rule (EWI-FDR) in our study.

$$h(x_i) = 2\frac{IQR(x_i)}{\sqrt[3]{n}} \; ; k(x_i) = \frac{\max(x_i) - \min(x_i)}{h(x_i)} \qquad (2)$$

Where:
k: Number of intervals
h: Interval width
IQR: Interquartile range (*Q3-Q1*)
n: Number of instances

In this study, the original dataset was discretized by using the functions `chiM`, `chi2`, `extendChi2` and `modChi2` of the `discretization` library (Kim 2012) in the R statistical computing environment. Since the χ^2 test determines similarity of adjacent intervals based on the value of a statistical significance level (α), the levels of this parameter will affect number of intervals calculated. The χ^2 test is more conservative at the smaller significance levels, and thus less number of intervals is generated with the smaller significance levels. In general, researchers have used either the significance level of 0.01 or 0.05. For Chi2 algorithm, Kerber (1992) originally proposed to choose any of the significance levels of 0.01, 0.05 or 0.10. Yang *et al* (2011) used the significance levels of 0.01 and 0.05 for different types of datasets in their experiments. Although there is no a definite rule to choose an appropriate significance level (Liu & Setiono 1995), the use of smaller significance levels can be preferred in order to avoid excessive number of intervals. Hence, in addition to the commonly used significance levels we also included the significance level of 0.001 in order to see how this conservative level will affect the obtained number of intervals in our experiment.

In order to evaluate the discretization performances of the algorithms, we compared the number of intervals and the execution time required to discrete all of the variables in the analyzed dataset. In addition to these, the classification training error rates and test accuracies calculated with the C5.0 Decision Tree Algorithm were also used for comparing the performances of the algorithms. For calculating these values in the R environment we ran the `C5.0` function of `C50` library (Kuhn *et al.* 2015) on each discretized dataset. We defined the classification tree model by using all of the variables in Table 1 as the predictors (`X`) and the genotype of chickens (*gen*) as the class variable (`Y`), and ran the model for 10 iterations with boosting option. We randomly sampled 80% of the data points (*n*=2750) as the training dataset (`trainY` and `trainX`) and the remaining 20% (*n*=743) as the test dataset (`testY` and `testX`). The applied model was `C5.0` (`trainY ~., data = trainX, trials = 10`). All analysis were done on a PC with i7 processor, 16GB RAM and 1 TB HDD running under an x64 operating system.

3. Results and Discussion

As seen in Table 2, the numbers of intervals varied between 2-254 for ChiM, 17-126 for Chi2, 2-5 for eChi2 and 1-4 for mChi2. The numbers of intervals obtained from Chi2 were relatively larger while they were very small in eChi2 and mChi2. The numbers of intervals obtained with eChi2 and mChi2 were close to each other and smaller than those of EWI-FDR. Although the numbers of intervals from ChiM increased as the significance level increases, they remained the same at all levels of significance for the algorithms Chi2, eChi2 and mChi2. According to these findings, e-Chi2 and mChi2 could be considered not good because they produced few number of intervals which may not be sufficient to keep the information in continuous values of the variables. Moreover, mChi2 produced only 1 interval for the variables *yce* and *wwd*, and it can be considered as inefficient discretization because of production of the low number of intervals for almost all of the variables in the analyzed dataset.

For the normal distributed variables in the analyzed dataset such as *shw* and *yht*, ChiM at the 0.01 and 0.05 significance levels and Chi2 at all significance levels generated closer results to those of EWI-FDR. But similar trends were also observed for some of the non-normal variables such as *shw* and *sbs*. This comparison showed that the algorithms remained insensitive to distribution types for these variables. Similar evaluation was also valid for the variables with higher variation such as *sbs* and *wht* versus the variables with lower variation in the analyzed dataset. In this regard, we need the forthcoming studies to discover the effects of different distribution types and variability levels on discretization since variations of the variables in the analyzed dataset may be not enough for revealing probable effects of different levels of the variations.

Table 2. Number of the intervals by the studied algorithms at different levels of significance

		Significance levels (α)						Significance levels (α)			
Vars	Algorithms	0.001	0.01	0.05	0.10	Vars	Algorithms	0.001	0.01	0.05	0.10
ewg	ChiM	4	17	96	235	**ywg**	ChiM	5	22	64	126
	Chi2	96	96	96	96		Chi2	64	64	64	64
	eChi2	4	4	4	4		eChi2	5	5	4	5
	mChi2	4	4	4	4		mChi2	3	3	2	3
ewd	ChiM	3	6	42	92	**yht**	ChiM	2	9	44	80
	Chi2	42	42	42	42		Chi2	44	44	44	44
	eChi2	3	3	3	3		eChi2	2	2	2	2
	mChi2	3	3	3	3		mChi2	2	2	2	2
eln	ChiM	7	17	65	136	**ywd**	ChiM	4	19	61	138
	Chi2	65	65	65	65		Chi2	61	61	61	61
	eChi2	7	7	5	7		eChi2	4	4	2	4
	mChi2	3	3	3	3		mChi2	2	2	2	2
eph	ChiM	3	6	17	21	**yce**	ChiM	5	16	114	208
	Chi2	17	17	17	17		Chi2	114	114	114	114
	eChi2	3	3	3	3		eChi2	5	5	5	5
	mChi2	3	3	3	3		mChi2	1	1	1	1
sbs	ChiM	3	10	35	73	**wht**	ChiM	4	9	45	90
	Chi2	35	35	35	35		Chi2	45	45	45	45
	eChi2	3	3	3	3		eChi2	4	4	4	4
	mChi2	3	3	3	3		mChi2	3	3	3	3
sht	ChiM	5	11	30	54	**wwd**	ChiM	2	16	112	217
	Chi2	30	30	30	30		Chi2	112	112	112	112
	eChi2	5	5	4	5		eChi2	2	2	2	2
	mChi2	4	4	4	4		mChi2	1	1	1	1
shw	ChiM	5	11	23	48	**wln**	ChiM	2	16	126	254
	Chi2	23	23	23	23		Chi2	126	126	126	126
	eChi2	5	5	5	5		eChi2	2	2	2	2
	mChi2	4	4	4	4		mChi2	2	2	2	2

For each variable, the number of intervals from ChiM at the significance level of 0.05 was equal to those obtained from Chi2 at the significance level of 0.001 and the higher levels. This finding showed that ChiM at the significance level of 0.05 produced the same results with Chi2 at all significance levels. It was also interesting that, for all the variables, the numbers of intervals from ChiM at the significance level of 0.001 were equal to those obtained from eChi2 at all significance levels. Similarly the numbers of intervals from ChiM at the significance level of 0.05 were equal to those obtained from eChi2 at all significance levels. These findings showed that ChiM at significance levels of 0.001 and 0.05 produced the same results with eChi2 and Chi2 respectively. This is an important advantage in favor of ChiM when the cost of execution time is taken into account.

In discretized data, the intervals should keep the present information in the continuous values and not produce patterns so different from those in original dataset. Hence, the number of intervals from a discretization algorithm should not be too small or too large. As seen in Table 1, the number of intervals by the variables varied between 29 and 31 with EWD-FDR. Assuming these interval numbers are informative enough and regarding them as reference, ChiM at the significance levels of 0.01 and 0.05 produced the closest results to those from EWD-FDR for the majority of variables.

As seen in Table 3, the error rate of training model which was computed from the continuous values was 5.0%. When this error rate was used as the reference, the results showed that ChiM at the significance level of 0.001, eChi2 and mChi2 at all significance levels did not perform well enough because their training errors were 5-6 times bigger than those computed for continuous values. On the other hand, as seen in Figure 1, the training errors from ChiM at the significance levels of 0.05 and 0.10 were less than those computed for the continuous values in original dataset. Chi2 produced the similar results at all significance levels because it used same discretized datasets in all of them.

Table 3. The training error rates and the test accuracies of the training model by the algorithms

Dataset	Training Error (%)	Test Accuracy (%)
Original (Continuous)	5.0	53.2
ChiM-0.001	24.3	51.8
ChiM-0.01	5.6	49.2
ChiM-0.05	2.7	51.6
ChiM-0.10	2.4	55.0
Chi2-0.001	2.7	51.6
Chi2-0.01	2.7	51.6
Chi2-0.05	2.7	51.6
Chi2-0.10	2.7	51.6
eChi2-0.001	24.3	51.8
eChi2-0.01	24.3	51.8
eChi2-0.05	24.2	50.9
eChi2-0.10	24.3	51.8
mChi2-0.001	32.3	52.5
mChi2-0.01	32.3	52.5
mChi2-0.05	33.0	52.5
mChi2-0.10	32.3	52.5

Figure 1. The training error rates and the test accuracies of the training model by the algorithms

The test accuracy of the training model was 53.2% for the continuous values in original dataset. The training model resulted with a medium level of accuracy for all of the discretized datasets. As seen from Figure 1 and Table 3, the test accuracies computed on discretized datasets were nearly equal to each other and varied between 49.2% and 55.0%. The highest accuracy was obtained as 55.0% for ChiM at the significance level of 0.10. The smallest accuracy was 49.2% and again obtained from ChiM at the significance level of 0.01.

As seen from Table 4, eChi2 and mChi2 required more execution time because these algorithms are based on ChiM and Chi2. The longest execution time of 9.99 minutes was obtained for eChi2 at the significance level of 0.001. This algorithm also required longer execution time at the other significance levels. Excluding ChiM, the discretization time at the significance level of 0.001 was relatively longer when compared to the other levels of significance.

Table 4. Execution time (min) by the algorithms and the significance levels

	Significance levels (α)			
Algorithms	0.001	0.01	0.05	0.10
ChiM	8.56	8.80	8.52	8.44
Chi2	8.96	8.58	8.70	8.62
mChi2	9.34	9.02	9.32	9.33
eChi2	9.99	9.07	9.46	9.34

4. Conclusions

In comparison to the other algorithms, Chi2 generated larger numbers of intervals. Contrarily, eChi2 and mChi2 resulted with very small number of intervals. ChiM at the significance level of 0.01 produced more compatible results compared to those of EWD-FDR which was used as the reference unsupervised method.

Regarding the test accuracy of the training model there were no remarkable differences between the studied algorithms. On the other hand the training error rates were low for ChiM and Chi2 compared to those of eChi2 and mChi2. For the analyzed dataset, this result indicated that ChiM and Chi2 algorithms were better than eChi2 and mChi2. The number of intervals from ChiM at the significance level of 0.05 were equal to those obtained with Chi2 at the significance level of 0.001 for all of the variables. In addition to its acceptable performance in generation of the intervals, ChiM worked faster than Chi2, eChi2 and mChi2. As a consequence of these findings we recommend to work with ChiM at the significance levels of 0.05 or 0.10 for discretization of the chicken egg quality traits when the genotype is used as the class variable.

In this study, even though we compared four Chi-square based algorithms for nonparametric discretization of the continuous egg quality traits, the research still needs to consider with other families of the supervised discretization algorithms as well as the unsupervised methods. In future studies, for comparing the success of the algorithms we also plan to study on the other aspects of discretization such as to use the robustness or the accuracy criterion based on statistical tests.

Acknowledgement

We gratefully thank to Assoc. Prof. Dr. Mikail Baylan and his colleagues at the Çukurova University for their permission to use the dataset analyzed in this study.

References

Acosta-Mesa HG, Rechy-Ramírez F, Mezura-Montes E, Cruz-Ramírez N & Hernández JR (2014), 'Application of time series discretization using evolutionary programming for classification of precancerous cervical lesions', *J Biomedical Informatics*, vol. 49, p. 73-83. doi: 10.1016/j.jbi.2014.03.004

Anderson TW & Darling DA (1952), 'Asymptotic theory of certain 'goodness-of-fit' criteria based on stochastic processes', *Annals of Mathematical Statistics*, 23: 193–212. doi: 10.1214/aoms/1177729437

Azulay R, Moskovitch R, Stopel D, Verduijn M, de Jonge E & Shahar Y (2007), 'Temporal Discretization of medical time series - A comparative study', In *Working Notes of Intelligent Data Analysis in Biomedicine and Pharmacology*, July 8, 2007, p. 73-78.

Bakar AA, Ahmed AM & Hamdan AR (2010), 'Discretization of Time Series Dataset Using Relative Frequency and K-Nearest Neighbor Approach', in Advanced Data Mining and Applications, vol. 6440 of the series Lecture Notes in Computer Science, p. 193-201. doi: 10.1007/978-3-642-17316-5_18

Bettinger R (2011), 'ChiD, A χ^2-based discretization algorithm', *Proc. of Western Users of SAS Software*. San Francisco, California, US, October 12-14, 2011.

Boulle M (2004), 'Khiops: A statistical discretization method of continuous attributes', *Machine Learning*, vol. 55, no. 1, p. 53 – 69. doi: 10.1023/b:mach.0000019804.29836.05

Chaudhari P, Rana DP, Mehta RG, Mistry N & Raghuwanshi M (2014), 'Discretization of temporal data: A survey', *Int. J. of Computer Science and Information Security*, vol. 12, no. 2, p. 66-69.

Dash R, Paramguru RL & Dash R (2011), 'Comparative analysis of supervised and unsupervised discretization techniques', *Int. J. of Advances in Science and Technology*, vol. 2, no. 3, p. 29-37.

Dougherty J, Kohavi R & Sahami M (1995), 'Supervised and unsupervised discretization of continuous feature', In *Proc. of the 12th Int. Conf. on Machine Learning*, p. 194 – 202. doi: 10.1016/b978-1-55860-377-6.50032-3

Freedman D & Diaconis P (1981), 'On the histogram as a density estimator: L2 theory', *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 57, no. 4, p. 453 - 476.

García S, Luengo J, Sáez JA, López V & Herrera F (2013), 'Survey of discretization techniques, Taxonomy and empirical analysis in supervised learning', *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, p. 734 - 750. doi: https://doi.org/10.1109/tkde.2012.35

Gross J & Ligges U (2015), 'Nortest: Tests for normality', R package version 1.0-4, 2015. https://CRAN.R-project.org/package=nortest

Gupta A, Mehrotrab KG & Mohan C (2010), 'A clustering-based discretization for supervised learning', *Statistics and Probability Letters*, vol. 80, p. 816 – 824. doi: https://doi.org/10.1016/j.spl.2010.01.015

Hemada B & Lakshmi KSV (2013), 'A Study on discretization techniques', *Int. J. of Engineering Research and Technology*, vol. 2, no. 8, p. 1887-1892.

Kaoungku N, Thinsungnoen T, Durongdumrongchai P, Kerdprasop K & Kerdprasop N (2015), 'Discretization based on Chi2 algorithm and visualize technique for association rule mining', In *Proc. of the 3rd Int. Conf. on Industrial Application Engineering*, p. 254 - 260. doi: https://doi.org/10.12792/iciae2015.047

Kerber R (1992), 'ChiMerge: Discretization of numeric attribute', In *Proc. of the 10th National Conference on Artificial Intelligence*, p. 123 – 128.

Kim HJ (2012), 'Discretization: Data preprocessing, discretization for classification', R package version 1.0-1. (https://CRAN.R-project.org/package=discretization).

Kotsiantis S & Kanellopoulos D (2006), 'Discretization techniques: A recent survey', *GESTS Int. Transactions on Computer Science & Engineering*, vol. 32, no. 1, p. 47-58.

Kuhn M, Weston S, Coulter N & Clup M (2015), 'C50: C5.0 Decision Trees and Rule-Based Models', *R package version 0.1.0-24*. (C code for C5.0 by R. Quinlan License: GPL-3) (https://cran.r-project.org/web/packages/C50/).

Liu H, Hussain F, Tan CL & Dash M (2002), 'Discretization: An enabling technique', *Data Mining and Knowledge Discovery*, vol. 6, no. 4, p. 393 - 423.

Liu H & Setiono R (1995), 'Chi2: Feature selection and discretization of numeric attributes', *IEEE 24th Int. Conf. on Tools with Artificial Intelligence, IEEE Computer Society*, p. 388 – 388. doi: https://doi.org/10.1109/tai.1995.479783

Mitov I, Ivanova K, Markov K, Velychko V, Stanchev P & Vanhoof K (2009), "Comparison of discretization methods for preprocessing data for pyramidal growing network classification method". In *New Trends in Intelligent Technologies*, Int. Book Series Information Science & Computing - Book No: 142009, p. 31-39.

Qu W, Yan D, Sang Y, Liang H, Kitsuregawa M & Li K (2008), 'A novel Chi2 algorithm for discretization of continuous attributes', In *Proc. Progress in WWW Research and Development, 10th Asia-Pacific Web Conference*, China, April 26-28, 2009. p. 560 - 571. doi: https://doi.org/10.1007/978-3-540-78849-2_56

R Core Team (2015), 'R: A language and environment for statistical computing', R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ramírez-Gallego S, García S, Mouriño-Talín H, Martínez-Rego D, Bolón-Canedo V, Alonso-Betanzos A, Benítez JM & Herrera F (2015), 'Data discretization: taxonomy and big data challenge', *WIREs Data Mining Knowledge Discovery*, doi: 10.1002/widm.1173.

Richeldi M & Rossotto M (1995), 'Class-driven statistical discretization of continuous attributes' in *European Conference on Machine Learning*, p. 335 - 338. doi: https://doi.org/10.1007/3-540-59286-5_81

Risvik KM (1997), 'Discretization of numerical attributes: Preprocessing for machine learning', Computer Science Projects #45073, *Knowledge Sys. Grp., Dept. of Comp. and Inf. Sci. at Norwegian Univ. of Sci. and Tech. Trondheim, Norway*.

Sang Y, Zhu P, Li K, Qi H & Zhu Y (2013), 'A local and global discretization method', *Int. J. of Information Engineering*, vol. 3, no. 1, p. 6 – 17.

Su CT & Hsu JH (2005), 'An extended Chi2 algorithm for discretization of real value attributes', *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, p. 437 – 441. doi: https://doi.org/10.1109/icmlc.2012.6359019

Tay FEH & Shen L (2002), 'A modified Chi2 algorithm for discretization', *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 3, p. 666 – 670. doi: https://doi.org/10.1109/tkde.2002.1000349

Van Buuren S & Groothuis-Oudshoorn K (2011), 'mice: Multivariate imputation by chained equations in R', *Journal of Statistical Software*, vol. 45, no. 3, p. 1 - 67. http://www.jstatsoft.org/v45/i03/.

Vink G, Frank LE, Pannekoek J & van Buuren S (2014), 'Predictive mean matching imputation of semicontinuous variables', *Statistica Neerlandica*, vol. 68, no. 1, p. 61–90. doi: 10.1111/stan.12023

Wang K & Liu B (1998), 'Concurrent discretization of multiple attributes', *In the Pacific RIM Int. Conf. on Artificial Intelligence*, p. 250 – 259. doi: https://doi.org/10.1007/bfb0095274

Yang P, Li JS & Huang YX (2011), 'HDD: a hypercube division-based algorithm for discretisation', *Int. J. of Systems Science*, vol. 42, no. 4, p. 557–566. doi: https://doi.org/10.1080/00207720903572455