Volume 3, Issue 11, November 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Discretization Technique Using Maximum Frequent Values and Entropy Criterion

B.Hemada, K.S.Vijaya Lakshmi
Department of Computer Science and Engineering,
V.R.Siddhartha Engineering College (Autonomous)
Affiliated to JNTUK, Vijayawada, Andhra Pradesh, India.

Abstract- Discretization is a process of dividing a continuous attribute into a finite set of intervals to generate an attribute with small number of distinct values. Discretization not only produce a concise summarization of continuous attributes to help the experts understand the data more easily, but also make learning more accurate and faster. The existing system, EDISC (Entropy-based Discretization Intervals using Scope of Classes) considers the scope of each class, and then calls the standard Ent-MDLP procedure for the scope of each class. If the scope-limited list for each of the k classes is equal to the full list of attribute values, then the search space and time complexity of calculating the cut-points increases as the entropy method is applied k times on the full list of attribute values. This discretization technique is an overlapping one, and it also generates large number of cut-points, which increases the penalty of the model. This paper presents a new discretization technique, which considers the maximum occurring value of each class as initial cut-points and applies the standard Ent-MDLP procedure between the initial cut-points, to find the set of final cut-points. This technique is applicable to any classification algorithm.

Keywords- Discretization, Continuous Data, Discrete Data, Classification, Data Transformation, Data Mining.

I. INTRODUCTION

Real-world data sets predominantly consist of continuous or quantitative attributes, i.e., attributes having numerous numeric values spread over a continuous spectrum. Many algorithms related to data mining require the training examples that contain only discrete values, for such data mining algorithms, the continuous attributes need to be transformed into discrete or qualitative ones. The procedure whereby this transformation is carried out is known as discretization, which has been the focus of active research in the field of data mining for more than a decade. Discretization is the process of dividing the range of the continuous attribute into intervals. Every interval is labeled a discrete value, and then the original data will be mapped to the discrete values.

Discretization can be performed either before learning, referred to as *preprocessing* or *offline discretization* [2], [3], [4], [5], [6], or during the learning process, which is referred to as *online discretization* [7], [8], [9], [10], [11]. The former approach discretizes all the continuous attributes before learning, resulting in transformation of the data set into a discrete one which is then used by the learning algorithm for mining. In the latter approach, the original data set is passed to the learning algorithm which then discretizes its continuous attributes during learning using its built-in discretization procedure. This involves finding suitable upper and lower bounds (cut points) which mark the boundaries of a discrete interval, so that all continuous attribute values which fall within these two bounds are assigned to that particular discrete interval label. There are many advantages of Discretization: (1) Discretization will reduce the number of continuous features' values, which brings smaller demands on system's storage. (2) Discrete features are closer to a knowledge-level representation than continuous ones. (3) Data can also be reduced and simplified through discretization. For both users and experts, discrete features are easier to understand, use, and explain. (4) Discretization makes learning more accurate and faster. (5) In addition to the many advantages of having discrete data over continuous one, a suite of classification learning algorithms can only deal with discrete data. Successful discretization can significantly extend the application range of many learning algorithms.

This paper presents a new discretization technique, which considers the maximum frequent value in each class as initial cut-points and applies the Entropy-MDLP method between the initial cut-points to find the final cut-points. Since the technique is essentially preprocessing (all the cut points are found prior to learning), it does not have to use the binary splitting approach necessary to reduce complexity during learning. As a result, the discretization is multi-interval, which is the optimal choice for maximum possible discrimination between the classes. The proposed technique is capable of minimizing the occurrence of instances of other classes which results in a significant increase in classification accuracy for data sets.

The paper is organized as follows: Section 2 presents a categorization of the major discretization approaches. Section 3 presents a survey of Discretization Techniques. Section 4 represents the proposed Discretization approach. Section 5 presents the empirical evaluation of Proposed Discretization technique on CN2 [17]. Section 6 concludes the paper and proposes directions for further research.

II. CATEGORIZATION OF DISCRETIZATION APPROACHES

Discretization methods have been developed along different needs. They can be categorized as [1]: supervised *vs* unsupervised, dynamic *vs* static, global *vs* local, splitting (top-down) *vs* merging (bottom-up), univariate *vs* multivariate and direct *vs* incremental.

A. Supervised vs. Unsupervised

Unsupervised discretization techniques do not take into account the class labels of the values of the continuous attribute being discretized. Equal Width [2] and Equal Frequency [2] are two representative unsupervised discretization algorithms. Supervised discretization techniques on the other hand take the class labels of the continuous attribute values into account when forming intervals. Compared to supervised discretization, previous research has indicated that unsupervised discretization algorithms have less computational complexity, but may result in much worse classification performance. When classification performance is the main concern, supervised discretization should be adopted.

B. Dynamic vs. Static

A dynamic method would discretize continuous values when a classifier is being built, such as in C4.5 while static discretization is done prior to classification task.

C. Top-down vs. Bottom-up

Top-down methods [1], [4], [13], [14] start with an empty list of cut-points (or split-points) and keep on adding new ones to the list by 'splitting' intervals as the discretization progresses. Bottom-up methods [5], [15] start with the complete list of all the continuous values of the feature as cut-points and gradually remove some of them by 'merging' intervals as the discretization progresses.

D. Local vs. Global

A local method would discretize in a localized region of instance space (i.e., a subset of instances) while a global discretization method uses the entire instance space to discretize.

E. Univariate vs. Multivariate

Univariate discretization quantifies one feature at a time while multivariate discretization considers simultaneously multiple features.

F. Direct vs. Incremental

Direct methods divide the range of *k* intervals simultaneously (i.e., equal-width [2], equal-frequency [2], or K-means), needing an additional input from the user to determine the number of intervals. Incremental methods [4], [13], [14], [15] begin with simple discretization and are followed by an improvement or refinement process, which requires a stopping criterion to halt further discretization.

III. LITERATURE SURVEY

The past few decades have seen many researches on discretization for mining association rules. The discretization algorithms proposed here comes under two categories: Unsupervised discretization methods and Supervised discretization methods. All the techniques presented here are univariate, i.e., they operate on each attribute independently after sorting its values in ascending order.

A. Unsupervised Discretization Methods

Equal-width interval [2] discretization is a simplest discretization method that divides the range of observed values for a feature into k equal sized bins, where k is a parameter provided by the user. The process involves sorting the observed values of a continuous feature and finding the minimum, Vmin and maximum, Vmax, values. The interval can be computed by dividing the range of observed values for the variable into 'k' equally sized bins using the following formula, where 'k' is a parameter supplied by the user:

Interval = $V_{max} – V_{min} / k$

Boundaries = V_{min} + (i*Interval)

The boundaries can be constructed for i = 1...k-1 using the above equation.

The Equal-frequency [2] discretization technique is similar to equal-width with the exception that the number of unique values (frequency) within each of the user-specified n intervals should be equal. The interval frequency is obtained using the following relation

i_f = nb_unique_values / n

where

i_f = interval frequency

nb_unique_values = number of unique values for a continuous attribute.

The two methods are simple but are sensitive to 'n'. The limitation of Equal-Width method is, the uneven distribution of the data points: some intervals may contain much more data points than other. For equal-frequency, for instance, many occurrences of a continuous value could cause the occurrences to be assigned into different bins. This can be handled by adjusting boundaries of neighboring bins so that duplicate values should belong to one bin only. Another problem is the presence of outliers that take extreme values. This can be overcome by removing the outliers using a threshold.

B. Supervised Discretization Methods:

Supervised discretization methods make use of the class label when partitioning the continuous features. There are wide number of supervised discretization algorithms, of those we consider few of them. Lukasz A. Kurgan and Krzyszt of J. Cios proposed a supervised CAIM (Class-Attribute Interdependence Maximization) [13] discretization algorithm that handles continuous and mixed mode attributes. The CAIM algorithm's goal is to find the minimum number of discrete intervals while minimizing the loss of class-attribute interdependency. The algorithm uses class-attribute interdependency information as the criterion for the optimal discretization. The Class-Attribute Interdependency Maximization (CAIM) criterion measures the dependency between the class variable *C* and the discretization variable *D* for attribute *F*, for a given quanta matrix. The CAIM criterion is defined as

$$CAIM\,(C, D|F) = \frac{\sum_{r=1}^{n}\left(max_r^{\,2} / M_{+r} \right)}{n}$$

where,
n is the number of intervals,
r iterates through all intervals, i.e. r = 1,2,...,n,
max_r is the maximum value among all qir values (maximum value within the rth column of the quanta matrix), i=1,2,...,*S*,
M_{+r} is the total number of continuous values of attribute *F* that are within the interval (dr-1, dr].

The pseudocode of the CAIM algorithm is as follows:
Given: Data consisting of M examples, S classes, and continuous attributes Fi
For every Fi do:
Step1.
1.1 find maximum (dn) and minimum (do) values of Fi
1.2 form a set of all distinct values of Fi in ascending order, and initialize all possible interval boundaries B with minimum, maximum and all the midpoints of all the adjacent pairs in the set
1.3 set the initial discretization scheme as D: $\{[d_0 ,d_n]\}$, set GlobalCAIM=0
Step2.
2.1 initialize k=1;
2.2 tentatively add an inner boundary, which is not already in D, from B, and calculate corresponding CAIM value
2.3 after all the tentative additions have been tried accept the one with the highest value of CAIM
2.4 if (CAIM > GlobalCAIM or k<S) then update D with the accepted in step 2.3 boundary and set GlobalCAIM=CAIM, else terminate
2.5 set k=k+1 and go to 2.2
Output: Discretization scheme D

The algorithm starts with a single interval that covers all possible values of a continuous attribute, and divides it iteratively. From all possible division points that are tried it chooses the division boundary that gives the highest value of the CAIM criterion. When the algorithm was tested on datasets and compared with six other state-of the-art discretization algorithms, the comparison showed that the CAIM algorithm generated lowest number of intervals and the highest dependence between class labels and discrete intervals. The drawback of this technique is, it cannot perform well with datasets where each example belongs to more than one of given n classes.

A supervised, static and global discretization method which uses the Gini gain [14] as discretization measure was proposed by Xiao-Hang Zhang, Jun Wu, Ting-Jie Lu and Yuan Jiang. In this discretization method the cut point is chosen based on the criterion, whose Gini gain value is the biggest on attribute A. The Gini gain ΔG is defined as

$$\Delta G(A,b;S)=Gini(S)-\frac{|S_1|}{|S|}Gini(S_1)-\frac{|S_2|}{|S|}Gini(S_2)$$

where
S_1 and S_2 are the subsets of S partitioned by the cut point b
Gini (.) is the Gini measure defined by

$$G\,(\text{interval}) = 1 - \sum_{j=1}^{k}(p_j^{(i)})^2$$

$p_j^{(i)}$ is the *j*th class probability in *i*th interval and satisfies $\sum_{j=1}^{k} p_j^{(i)} =1$.
|.| denotes the number of instances.

The training set is split into two subsets by the cut point which is chosen using Gini measure. Subsequent cut points are selected by recursively applying the same binary discretization method to one of the generated subsets, which has biggest Gini gain value, until the stopping criterion is achieved. The stopping criterion of the discretization algorithm is defined by

$$G_{n+1}\ln(n+1+p) > G_n \ln(n+p) \qquad (1)$$

where
n denotes the current number of intervals,
p is a positive integer determined by the user,

G_n is the Gini value with n intervals, defined by

$$G_n = \sum_{i=1}^{n} \frac{|S1|}{|S|} \text{ Gini (interval i)}$$

To discretize attribute *A*, the pseudocode of Gini based discretization algorithm is:

```
p = GettingParameterValue( )
S = SortingValue(A)
Splitting(S, p) {
B_D = { }
b = GetBestCutPoint(S, B_D )
While not StoppingCriterion(S, B_D, p) {
B_D = B_D + {b}
b = GetBestCutPoint(S, B_D )
}
}
```

Where:

GettingParameterValue – getting the value of p which is defined in the Eq.(1).

SortingValue – sorting all the values of attribute A in ascending order.

GetBestCutPoint – getting the best cut point by searching all the possible cut points in each interval to get the highest Gini gain value within all the intervals.

StoppingCriterion – returning the value TRUE if the stopping criterion is satisfied, else the value FALSE.

The Khiops discretization method [15] proposed by Marc Boulle is a bottom-up method based on the global optimization of chi-square (χ^2) [5]. χ^2 is a statistical measure that conducts a significance test on the relationship between the values of a feature and a class. χ^2 statistic determines the similarity of adjacent intervals based on some significance level. It tests the hypothesis that two adjacent intervals of a feature are independent of the class. If they are independent, they should be merged, otherwise they should remain separate. The formula for computing χ^2 value is

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{p} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

where:

p = number of classes

A_{ij} = number of distinct values in the *i*th interval, *j*th class

R_i = number of examples in *i*th interval = $\sum_{j=1}^{p} A_{ij}$

C_j = number of examples in *j*th class= $\sum_{i=1}^{m} A_{ij}$

N = total number of examples = $\sum_{j=1}^{p} C_j$ and

E_{ij} = expected frequency of $A_{ij} = (R_i * C_j) / N$

The chi-square value is not reliable to test the hypothesis of independence if the expected frequency in any cell of the contingency table is less than some minimum value. This is equivalent to a minimum frequency constraint for each row of the contingency table, i.e. for each interval of the discretization. The algorithm will cope with this constraint.

Algorithm Khiops

1. Initialization

1.1. Sort the explanatory attribute values

1.2. Create an elementary interval for each value

2. Optimization of the discretization

Repeat the following steps:

2.1. Search for the best merge

Search among the merges with at least one interval that does not meet the frequency constraint if anyone exists, among any merge otherwise

Merge that maximizes the chi-square value

2.2. Evaluate the stopping rule

Stop if all constraints are respected and if no further merge decreases the confidence level

2.3. Merge and continue if the stopping rule is not met

The Khiops discretization method starts from the elementary single value intervals and then searches for the best merge between adjacent intervals. Two different types of merges are encountered. First, merges with at least one interval that does not meet the constraint and second, merges with both intervals fulfilling the constraint. The best merge candidate (with the highest chi-square value) is chosen in priority among the first type of merges (in which case the merge is accepted unconditionally), and otherwise, if all minimum frequency constraints are respected, among the second type of merges (in which case the merge is accepted under the condition of improvement of the confidence level). The algorithm is reiterated until both all minimum frequency constraints are respected and no further merge can decrease the confidence level. When compared with other chi-square based methods like ChiMerge and ChiSplit methods, this global evaluation carries some intrinsic benefits. The Khiops automatic stopping rule brings both ease of use and high quality discretizations. Its computational complexity is the same as for the fastest other discretization methods.

Quisha Zhu, Lin Lin, Mei-Ling Shyu and Shu-Ching Chen proposed a novel supervised discretization algorithm based on correlation maximization (CM) [16]. It is proposed by using multiple correspondence analyses (MCA). MCA is an effective technique to capture the correlations between intervals/items and classes. The one that gives the highest correlation with the classes is selected as a cut-point. The geometrical representation of MCA not only visualizes the correlation relationship between intervals/items and classes, but also presents an elegant way to decide the cut-points. The graphical representation of MCA is called symmetric map. It is used to visualize the intervals of a feature and the classes as points in a two-dimensional map. The correlation between an interval and a class can be represented by the cosine angle between these 2 vectors in the first 2 dimensions. The larger the cosine value of the angle is the stronger the correlation between them. For a numeric feature F_i , all values of this feature are sorted to form a set of n+1 distinct values. Candidate cut points are the mid points of all adjacent pairs in the set. The cut point with the largest cosine is selected as the first cut-point T1. Then the same strategy can be carried out separately in the left and right intervals in a binary recursive way. The recursion is terminated if the correlation between current intervals and classes is lower than the correlation between their predecessor and their classes. When compared with some other discretization algorithms, this algorithm produced relatively small number of intervals and also has a low computational complexity. The drawback of this method is, it cannot discretize the datasets containing more than 2 classes.

IV. PROPOSED APPROACH

The Proposed Discretization Technique considers the maximum occurring value of each class as initial cut-points and next the standard Ent-MDLP [4] procedure is applied between the initial cut-points to find the set of final cut-points.

Entropy-MDLP Discretization

Ent-MDLP [4] uses entropy measure from information theory to find a cut-point to split a range of continuous values into two intervals. Class information entropy is a measure of purity and it measures the amount of information which would be needed to specify to which class an instance belongs. The entropy measure is defined as

$$E_e = E_1 + E_2$$

$$E_e = - p_{left} \sum_{i=1}^{k} p_{i,left} \log p_{i,left}$$

$$- p_{right} \sum_{i=1}^{k} p_{i,right} \log p_{i,right}$$

where

Ee = entropy of the cut-point
E1 = entropy to the left of the cut-point
E2 = entropy to the right of the cut-point
k = total number of classes
i = a practical class

pleft = number of instances to the left of cut-point / total number of instances, N
pright = number of instances to the right of cut-point / total number of instances, N
pi,left = num of instances of class i to the left of cut-point / number of instances to the left of cut-point
pi,right = {num of instances of class i to the right of cut-point } / { number of instances to the right of cut-point}

It considers one big interval containing all known values of a feature and then recursively partitions this interval into smaller subintervals until the stopping criterion satisfies. The stopping criterion was based on the MDL (Minimum Description Length) principle [12] which is defined as

$$\text{Gain} > \frac{\log_2 (N-1)}{N} + \frac{\log_2 (3^k-2) - k E + k_1 E_1 + k_2 E}{N}$$

where

$$E = - \sum_{i=1}^{k} p_i \log p_i$$

$$p_i = \frac{\text{Number of instances of Class } i}{N}$$

gain = E – E_e = information gained by splitting at the cut- point
N = total number of instances in the attribute value list at each recursion
k_1 = number of classes to the left of the cut-point
k_2 = number of classes to the right of the cut-point

Ent-MDLP says that a partition induced by a cut-point for a set of instances is accepted if and only if the cost or length of the message required to send before partition is more than the cost or length of the message required to send after the partition.

Proposed Discretization algorithm

Input: A, continuous attributes.
C, class values in training set.

Algorithm:

For each attribute perform {
for each class perform{
→find the no. of occurrences for each distinct value
→find the max occurrence value in the class and consider it as key value
}
→Sort the values of each attribute
→By considering the key values as initial cut-points, form the subsets
→for each subset perform{
→Apply Ent-MDLP criteria and find the final cut-points
}
}

Output*:* Interval values of continuous attributes.

The Proposed Discretization Algorithm initially finds the frequently occurring value in each class and considers them as initial cut-points. By considering the initial cut-points, the instances are formed into subsets. The standard Ent-MDLP criterion is applied to these subsets to get final cut-points. By considering all the cut-points of each attribute, the intervals of each attribute are formed.

V. EMPIRICAL EVALUATION

This section presents the results obtained from the experimental evaluation of Equal-Width, Equal Frequency, 1R, Entropy, EDISC and the Proposed Discretization Technique on CN2 [17] algorithm. To evaluate the performance of the discretization techniques *classification accuracy* criterion is used, which is the prime importance in rule-based classification. Table-I and Table- II represents the classification accuracy of Equal-Width, Equal-Frequency, 1R, Entropy based Discretization, EDISC and Proposed Discretization techniques for blood–transfusion, credit-approval and breast cancer datasets [18].

Table I

PROPERTY	**B-T**	**C-A**	**B-C**
EQUAL-WIDTH	77.03%	75.74%	93.04%
EQUAL-FREQUENCY	76.35%	76.76%	94.49%
1R	76.62%	72.79%	94.93%
ENTROPY	77.03%	76.47%	94.35%
EDISC	77.03%	79.38%	97.49%
PROPOSED DISCRETIZATION	79.70%	86.70%	97.77%

B-T :- Blood Transfusion Dataset
C-A:- Credit Approval Dataset
B-C:- Breast Cancer Dataset

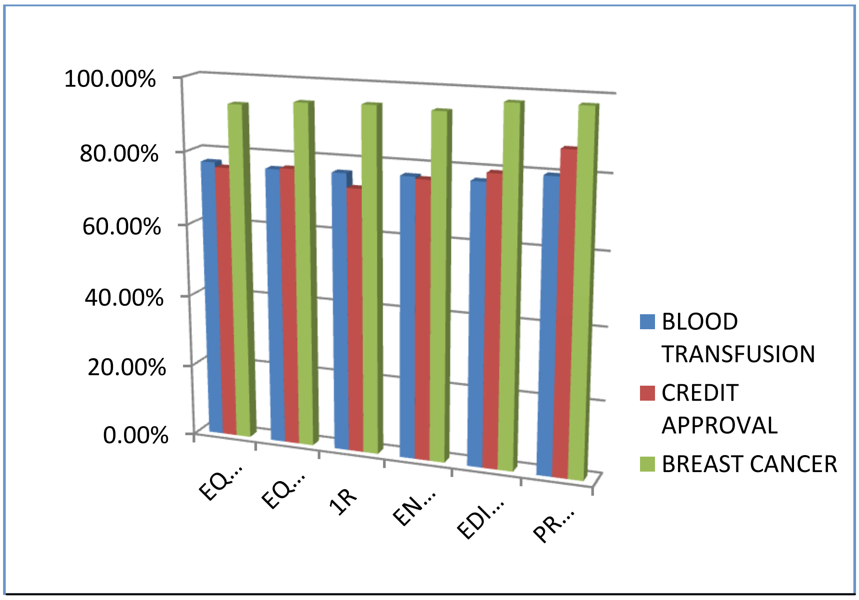

Figure 1: Graphical Representation of Comparison of Results.

VI. CONCLUSION AND FUTURE WORK

Discretization of continuous features plays an important role in data pre-processing. From the past few decades much work has been done in this area resulting in many different discretization methods. The proposed discretization technique is essentially preprocessing since all the cut points are discovered and stored in a discretization table prior to the start of the learning phase. It generates less number of cut-points, and the cut-points generated are best one, minimizing the occurrence of instances of other classes. The discretization is multi-interval, which is the optimal choice for maximum possible discrimination between the classes. It does not require user interaction, and performs automatic selection of the number of discrete intervals, in contrast to some other discretization algorithms. The technique is compared with five other discretization methods, used as preprocessing discretization procedures with CN2 on three data sets. Experimental evaluation has proved that the new technique results in a significant increase in classification accuracy.

Further work can be carried out using the idea of unifying both top-down and bottom up approach to get better results. The idea might be extended to fuzzy discretization [19], [20], where a value may belong to multiple intervals identified for a particular class, each with a certain degree of membership. While a lot of work has been done, there are still many issues that remained unsolved, and new methods are needed to address these issues. In future we expect robust discretization techniques which can overcome the drawbacks of handling huge data and large number of attributes.

REFERENCES

[1] Khurram Shehzad "*EDISC: A Class-Tailored Discretization Technique for Rule-Based Classification"* IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 8, August 2012.

[2] A.K.C. Wong and D.K.Y. Chiu, *"Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data,"* IEEE Trans. Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 6, pp. 796-805, Nov. 1987.

[3] A. An and N. Cercone, *"Discretization of Continuous Attributes for Learning Classification Rules,"* Proc. Third Pacific-Asia Conf. Methodologies for Knowledge Discovery and Data Mining, pp. 509-514, 1999.

[4] U.M. Fayyad and K.B. Irani, *"Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning,"* Proc. 13th Int'l Joint Conf. Artificial Intelligence, pp. 1022-1027, 1993.

[5] R. Kerber, *"ChiMerge: Discretization of Numeric Attributes,"* Proc. 10th Nat'l Conf. Artificial Intelligence, pp. 123-128, 1992.

[6] K.M. Ho and P.D. Scott, *"Zeta: A Global Method for Discretization of Continuous Variables,"* Proc. Third Int'l Conf. Knowledge Discovery and Data Mining (KDD '97), pp. 191-194, 1997.

[7] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen, *Classification and Regression Trees*. Wadsworth, 1984.

[8] P. Clark and T. Niblett, *"The CN2 Induction Algorithm,"* Machine Learning, vol. 3, pp. 261-283, 1989.

[9] J.R. Quinlan, *"Learning Efficient Classification Procedures and Their Application to Chess End Games,"* Machine Learning: An Artificial Intelligence Approach,S.R. Michalski, G.J. Carbonell, and M.T. Mitchell, eds., vol. I, pp. 463-482, Tioga Publishing Co., 1983.

[10] J.R. Quinlan, *"Induction of Decision Trees,"* Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.

[11] J.R. Quinlan, *"Learning Logical Definitions from Relations,"* Machine Learning,vol. 5, no. 3, pp. 239-266, 1990.

[12] I.H. Witten and E. Frank, *"Data Mining—Practical Machine Learning Tools and Techniques"*. Morgan Kaufmann Publishers, 2005.

[13] Lukasz A. Kurgan, and Krzysztof J. Cios, *"CAIM Discretization Algorithm"* IEEE Transactions On Knowledge And Data Engineering, Vol. 16, No. 2, February 2004.

[14] Xiao-Hang Zhang, Jun Wu, Ting-Jie Lu, Yuan Jiang *"A discretization algorithm based on gini criterion"* , Proceedings Of The Sixth International Conference On Machine Learning And Cybernetics, Hong Kong, 19-22 august 2007.

[15] Marc Boulle "*Khiops: A Statistical Discretization Method of Continuous Attributes*" Machine Learning, 55, 53–69, 2004.

[16] Qiusha Zhu, Lin Lin, Mei-Ling Shyu, Shu-Ching Chen " Effective Supervised Discretization for Classification based on Correlation Maximization" *Information Reuse And Integration (Iri)*, 2011 IEEE *International Conference On* 3-5 Aug. 2011.

[17] P. Clark and R. Boswell, *"Rule Induction with CN2: Some Recent Improvements,"* Proc. Fifth European Working Session on Learning, pp. 151-163, 1991.

[18] C.L. Blake and C.J. Merz, *"UCI Repository of Machine Learning Databases,"* Univ. of California, Dept. of Information and Computer Science, Irvine, CA, http://archive.ics.uci.edu/ml/, 1998.

[19] H. Ishibuchi and T. Yamamoto, *"Deriving Fuzzy Discretization from Interval Discretization,"* Proc. 12th IEEE Int'l Conf. Fuzzy Systems, pp. 749-754, 2003.

[20] I. Kuwajima, Y. Nojima, and H. Ishibuchi, *"Effects of Constructing Fuzzy Discretization from Crisp Discretization for Rule-Based Classifiers,"* Artificial Life and Robotics, vol. 13, no. 1, pp. 294-297, 2008.

