ICAIM : An Improved CAIM Algorithm for Knowledge Discovery

Piriya Yaowapanee*, Ouen Pinngern**

*Research Center for Communication and Information Technology, Department of Computer Engineering,
Faculty of Engineering, King Mongkut's Institue of Technology Ladkrabang, Thailand.
(Tel : +66-2-737-3000 Ext. 3334; Email : s3061625@kmitl.ac.th)
**Research Center for Communication and Information Technology, Department of Computer Engineering,
Faculty of Engineering, King Mongkut's Institue of Technology Ladkrabang, Thailand.
(Tel : +66-2-737-3000 Ext. 3334; Email : kpouen@kmitl.ac.th)

Abstract: The quantity of data were rapidly increased recently and caused the data overwhelming. This led to be difficult in searching the required data. The method of eliminating redundant data was needed. One of the efficient methods was Knowledge Discovery in Database (KDD). Generally data can be separate into 2 cases, continuous data and discrete data.

This paper describes algorithm that transforms continuous attributes into discrete ones. We present an Improved Class Attribute Interdependence Maximization (ICAIM), which designed to work with supervised data, for discretized process. The algorithm does not require user to predefine the number of intervals. ICAIM improved CAIM by using significant test to determine which interval should be merged to one interval. Our goal is to generate a minimal number of discrete intervals and improve accuracy for classified class. We used iris plant dataset (IRIS) to test this algorithm compare with CAIM algorithm.

Keyword: ICAIM, CAIM, significant test, discretization, degree of freedom, χ^2, KDD Process.

1. INTRODUCTION

The amount of data and information in real world was rapidly growth thus we often use one of the efficient methods to process and extraction of knowledge from data. Efficient method that refer to is Knowledge Discovery in Database (KDD). KDD algorithms are used to generate classification rules from class-labeled examples that are described by a set of numerical (e.g. 1,3,5), nominal (e.g. high, medium, low) or continuous attributes[1]. In order to handle continuous data we must used preprocessing step to transform continuous data to discrete data. The preprocessing step that refer to is discretization algorithm.

Discretization can be broken into two categories [2]:

1. unsupervised algorithms that discretize attributes without taking into account respective class labels. The two representative algorithms are equal-width and equal-frequency discretizations [3].
2. supervised algorithms that discretize attributes by taking into account the interdependence between class labels and the attribute values. The representative algorithms are maximum entropy [4], Statistics-base algorithms like ChiMerge [5] and Chi2 [6], class-attribute interdependency algorithms like CADD [7], clustering-based algorithms like K-means discretization [8] .

Generally discretization have two main tasks to do. The first task is to find the number of discrete intervals. Often the user must specify the number of intervals, or provide a heuristic rule . The second task is to find the width or the boundaries for the intervals, given the range of values of a continuous attribute. An Improved Class-Attribute Interdependent Maximize (ICAIM) algorithm selects a number of discrete intervals and, at the same time, finds the width of every interval automatically.

The proposed ICAIM algorithm discretizes an attribute into the small number of intervals and makes high accuracy classification class labels for KDD process. In this paper we use IRIS dataset that consist of 3 classes, 150 examples and 4 continuous attributes for test ICAIM.

First, ICAIM algorithm automatically selects the number of discrete intervals and width of discrete intervals by using class-attribute interdependency. Second, the algorithm used significant test to combine interval that not significant. After this process we will get small number of intervals and high accuracy discretization for KDD process.

2. ICAIM Discretization Algorithm

To better facilitate supervised learning in continuous domains, a method that uses the class-attribute dependency information as the criterion for optimal discretization is used. The discretization process is viewed as the partitioning of a continuous-valued attribute into an ordered discrete attribute with a number of discrete intervals.

2.1 Definitions of CAIM

Class-Attribute Interdependet Maximize (CAIM) [9] algorithm is a supervised classification task requires a training dataset consisting of M examples, where each example belongs to only one of S classes. F indicates any of the continuous attributes from the mixed-mode data. There exist discretization scheme D on F, which discretizes the continuous domain of attribute F into n discrete intervals bounded by the pair of number

$$D : \{[d_0, d_1],(d_1,d_2],,(d_{n-1},d_n]\}, \quad (1)$$

Where:

d_0 is the minimal value,

d_n is the maximal value of attribute F.

The value in D are arranged in ascending order. The class variable and the discretization variable of attribute F are treated as two random variables defining a two-dimensional frequency matrix called quanta matrix as shown in Table 1.

Table 1 Quanta Matrix

Class	Interval					Class Total
	$[d_0, d_1]$	...	$(d_{r-1}, d_r]$	...	$(d_{n-1}, d_n]$	
C_1	q_{11}	...	q_{1r}	...	q_{1n}	M_{1+}
:	:	...	:	...	:	:
C_i	q_{i1}	...	q_{ir}	...	q_{in}	M_{i+}
:	:	...	:	...	:	:
C_S	q_{S1}	...	q_{Sr}	...	q_{Sn}	M_{S+}
Interval Total	M_{+1}	...	M_{+r}	...	M_{+n}	M

Q_{ir} is the total number of continuous values belonging to the i^{th} class that are within interval $(d_{r-1},d_r]$,

M_{i+} is the total number of objects belonging to the i^{th} class,

M_{+r} is the total number of continuous values of attribute F that are within the interval $(d_{r-1},d_r]$, for i=1,2,..,S and r=1,2,..,n,

M is the total number of continuous values or objects in dataset.

The CAIM algorithm works in a top-down manner, dividing one of the existing intervals into two new intervals using criterion that results in achieving the optimal class-attribute interdependency after the split, and starts with a single, $[d_0, d_n]$, interval.

2.2 Discretization Criterion

First, ICAIM criterion measures the dependency between the class variable C and the discretization variable D for attribute F.

For a given quanta matrix as shown in table 1 criterion that measures the dependency is defined as:

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

Where:

n is the number of intervals,

r iterates through all intervals, r=1,2,..,n,

max_r is the maximum value among all q_{ir} values,

i=1,2,..,S ,

M_{+r} is the total number of continuous values of attribute F that are within the interval $(d_{r-1},d_r]$.

Second measures statistically significant [10] between two adjacent intervals. Proceeds by using χ^2 test to determine when adjacent intervals should be merged. The χ^2 test is a statistical measure used to test hypothesis that two discrete attributes are statistically independent. In discretization process χ^2 tests the hypothesis that the class attribute is independent of the two adjacent intervals an example belong to. If the conclusion of the χ^2 test is that the class is independent of the intervals or not statistically significant, then the intervals should be merged. On the other hand, if the χ^2 test concludes that they are not independent, it indicates that the difference in relative class frequencies is statistically significant and, therefore, the intervals should remain separate. The criterion that measures statistically significant is:

$$\chi^2_{(S-1)(n-1)}=\sum_{1}^{S-1}\sum_{1}^{n}\frac{(A_{sn}-E_{sn})^2}{E_{sn}}, \quad (3)$$

Where :

(S-1)(n-1) is degree of freedom of quanta matrix,

S is number of classes,

n is number of intervals,

A_{sn} is number of examples in n^{th} interval, s^{th} class,

E_{sn} is expected frequency of A_{sn} , defined as

$$E_{sn}=(M_{+r}*M_{i+})/M, \quad (4)$$

2.3 The ICAIM Algorithm

The ICAIM algorithm is searching over the space of all possible discretization schemes to find the highest value of the CAIM criterion. After that they bring the adjacent intervals to test statistically significant and merge adjacent intervals that not significant. The pseudocode of ICAIM algorithm is follow:

Given: Data consisting of M examples, S classes, and continuous attributes F_i

For every F_i do:

Step 1.

1.1 find maximum (d_n) and minimum (d_0) values of F_i

1.2 form a set of distinct values of F_i 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:{$[d_0,d_n]$}

1.4 tentatively add an inner boundary, which is not already in D, from B, and calculate corresponding CAIM value with eq. (2).

1.5 after all the tentative additions have been tried accept the one with the highest value of CAIM.

Step 2.

2. compute statistically significant of distinct adjacent intervals of F_i with eq. (3)

2.2 If the conclusion of the χ^2 test is that the class is independent of the intervals or not statistically significant, then the intervals should be merged and if the χ^2 test concludes that they are not independent, therefore, the intervals should remain separate

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 points that are tried, it chooses the division boundary that gives the highest value of CAIM criterion. After that it define new boundary and iterative find highest CAIM of that boundary until last point select.

In step 2 we test adjacent intervals for statistically significant, if which adjacent intervals that not significant we merge it and try to test all adjacent intervals. Finally we get small discrete intervals and high accuracy for using in KDD process.

3. EXPERIMENTS

For experiment, IRIS dataset use to test the ICAIM algorithm. IRIS dataset consist of 150 examples, 3 classes, and 4 continuous attributes. We test IRIS dataset with ICAIM compare with CAIM.

First we test IRIS dataset with original CAIM. After tested IRIS dataset we got the results as follow:

Table 2 CAIM discretization

SepalLength			SepalWidth		
order w	*CAIM*	*interval*	*order w*	*CAIM*	*interval*
13	31.9126	5.55	10	23.7907	3.05
20	26.6363	6.25	13	17.3420	3.35
28	20.2728	7.00	18	13.0536	3.80

Table 2 CAIM discretization (Cont.)

PetalLength			PetalWidth		
order w	*CAIM*	*interval*	*order w*	*CAIM*	*interval*
9	37.5000	1.95	6	37.5000	0.65
24	45.5589	4.75	14	46.1616	1.75
27	34.4923	5.05	22	34.6212	2.50

From Table 2, we get SepalLength have 3 intervals, SepalWidth have 3 intervals, PetalLength have 3 intervals, and PetalWidth have 3 intervals. So number of intervals of CAIM algorithm are 12 intervals.

Then we tested IRIS dataset with ICAIM in step 1 from ICAIM algorithm. The result is shown follow:

Table 3 Step 1 in ICAIM algorithm

SepalLength			SepalWidth		
order w	*CAIM*	*interval*	*order w*	*CAIM*	*interval*
13	31.9126	5.55	10	23.7907	3.05
20	26.6363	6.25	13	16.9661	3.35
28	20.2728	7.05	23	12.7245	4.40
35	16.2183	7.90			
PetalLength			PetalWidth		
order w	*CAIM*	*interval*	*order w*	*CAIM*	*interval*
9	37.5000	1.95	6	37.5000	0.65
24	45.5589	4.75	14	46.1616	1.75
27	34.4923	5.05	15	34.6366	1.85
28	27.6141	5.15	22	20.7093	2.50
43	23.0117	6.90			

From Table 3, in step 1 of ICAIM we get SepalLength have 4 intervals, SepalWidth have 3 intervals, PetalLength have 5 intervals, and PetalWidth have 4 intervals. So number of intervals of step 1 ICAIM algorithm are 16 intervals.

We bring result from step 1 of ICAIM to test with step 2 of ICAIM. After all adjacent intervals have passed statistically significant test in step 2. We get results as table 4:

Table 4 Results after pass step 2 of ICAIM

SepalLength			SepalWidth		
order w	*CAIM*	*interval*	*order w*	*CAIM*	*interval*
13	31.9126	5.55	10	23.7907	3.05
20	26.6363	6.25	13	16.9661	3.35
28	20.2728	7.05	23	12.7245	4.40
35	16.2183	7.90			
PetalLength			PetalWidth		
order w	*CAIM*	*interval*	*order w*	*CAIM*	*interval*
9	37.5000	1.95	6	37.5000	0.65
24	45.5589	4.75	14	46.1616	1.75
28	27.6141	5.15	22	20.7093	2.50
43	23.0117	6.90			

After passed step 2 of ICAIM, all intervals of SepalLength had significant so those intervals must separate. The result is shown in fig.1.

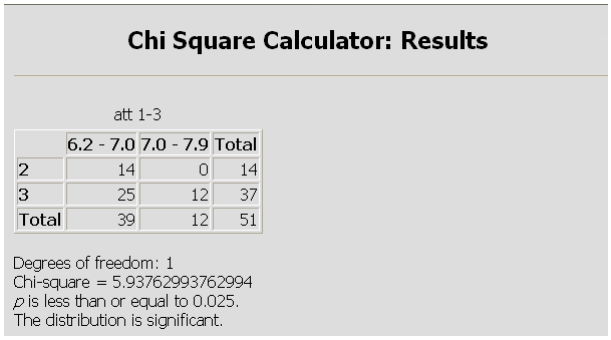


Fig. 1 significant test for SepalLength between 6.25 – 7.05 and 7.05 – 7.90

At SepalWidth, all intervals is same as CAIM and all intervals have significant is shown in fig. 2.

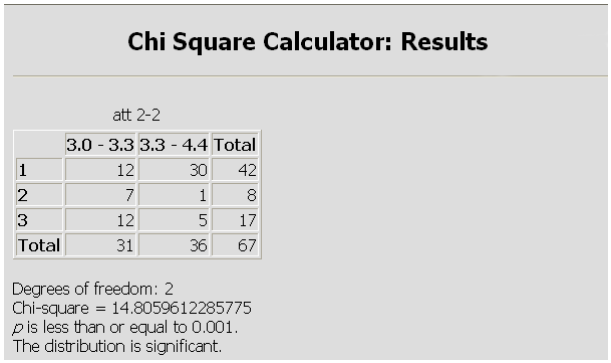


Fig. 2 significant test for SepalWidth between 3.05 – 3.35 and 3.35 – 4.40

For PetalLength, we test significant all adjacent intervals. We get intervals 4.75 – 5.05 and 5.05 – 5.15 is not significant so we merge it. The number of intervals is reduce to 4 intervals. The result is shown follow:

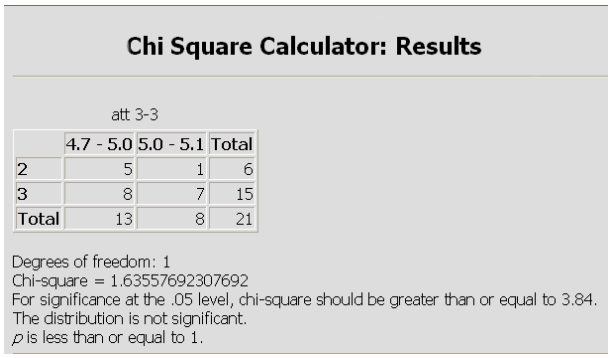
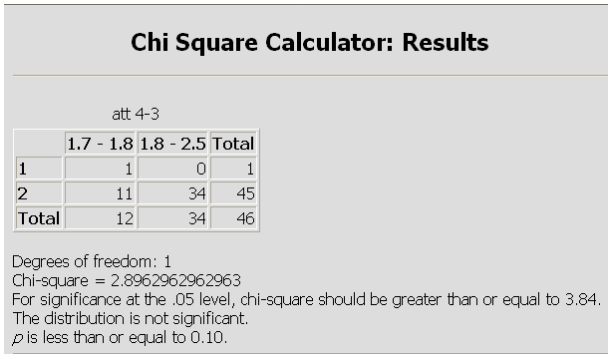


Fig. 3 significant test for PetalLength between 4.75 – 5.05 and 5.05 – 5.15

At PetalWidth adjacent intervals between 1.75 – 1.85 and 1.85 – 2.5 is not significant and the results is shown in fig. 4:



Chi Square Calculator: Results

att 4-3

	1.7 - 1.8	1.8 - 2.5	Total
1	1	0	1
2	11	34	45
Total	12	34	46

Degrees of freedom: 1
Chi-square = 2.8962962962963
For significance at the .05 level, chi-square should be greater than or equal to 3.84.
The distribution is not significant.
p is less than or equal to 0.10.

Fig. 4 significant test for PetalWidth between 1.75 – 1.85 and 1.85 – 2.5

So we merged the intervals 1.75 – 1.85 and 1.85 – 2.50 into one interval.

After we compute CAIM and ICAIM, we bring the results to classify classes in IRIS dataset. In CAIM we observed that if att.1 =0 – 5.55, att.2 =0 – 3.05, att.3 =1.95 – 4.75, att.4 = 0.65 – 1.75, att.1 =5.55 – 6.25, att.2 =0 – 3.05, att.3 =4.75 – 6.90, att.4 = 0.65 – 1.75 and att.1 =6.25 – 7.90, att.2 =0 – 3.05, att.3 =4.75 – 6.90, att.4 =0.65 – 1.75 class of dataset were in class 2 and 3 in same time. But for att.1 =0 – 5.55, att.2 =0 – 3.05, att.3 =1.95 – 4.75, att.4 = 0.65 – 1.75 we observed that cause a noise of dataset because it has one value that was in class 3 so we didn't ignore it.

In ICAIM, we bring the results to classify classes in same dataset. We observed that classify classes remain conflict with att.1 = 0 – 5.55, att.2 =0 – 3.05, att.3 =1.95 – 4.75, att.4 = 0.65 – 1.75 and att.1 =5.55 – 6.25, att.2 =0 – 3.05, att.3 = 4.75 – 5.15, att.4 =0.65 – 1.75 but in case att.1 =0 – 5.55, att.2 =0 – 3.05, att.3 =1.95 – 4.75, att.4 =0.65 – 1.75 is in class 2 11 objects while in class 3 1 object and case att.1 = 5.55 – 6.25, att.2 =0 – 3.05, att.3 =4.75 – 5.15, att.4 =0.65 – 1.75 is in class 2 1 object while in class 3 5 objects. So if we bring this result into KDD process these conflict will discard them that make result better.

4. CONCLUSIONS

From our experiments we conclude that ICAIM can discretize continuous attribute value to discrete interval. In CAIM, IRIS dataset could discretize into 12 intervals. In ICAIM, IRIS dataset could discretize into 14 intervals. But in CAIM IRIS dataset have conflict classes between define att.1 5.55 – 6.25, att.2 0 – 3.05, att.3 4.75 – 5.15, att.4 0.65 – 1.85 to be class 2 or class 3 and att.1 6.25 – 7.05, att.2 0 – 3.05, att.3 4.75 – 5.15, att.4 0.65 – 1.85 to be class 2 or class 3. In ICAIM, IRIS dataset could discretize into 14 intervals that nearby CAIM and all conflicts could be defined into classes even though they are conflict but could define to be noise because it had one object in class 2 but five objects in class 3 and 1 object in class 3, 11 objects in class 2. So ICAIM can improve precision in classification for KDD process.

REFERENCES

[1] Knut Magne Risvik. "Discretization of Numerical Attributes." Knowledge Systems Group, Department of Computer and Information Science, Norwegian University of Science and Technology, 1997.

[2] J. Dougherty., R. Kohavi., M. Sahami. "Supervised and Unsupervised Discretization of Continuous Features." Machine Learning: Proceeding of the 12th International Conference, 1995, Morgan Kaufmann Publishers.

[3] D. Chiu, A. Wong and B. Cheung, "Information Discovery through Hierarchical Maximum Entropy Discretization and Synthesis", Knowledge Discovery in Databases, MIT Press,1991.

[4] A.K.C. Wong and D.K.Y. Chiu, Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 9,pp. 796-805,1987.

[5] R. Kerber, "ChiMerge: Discretization of Numeric Attributes", Proceedings of the Ninth International Conference on Artificial Intelligence,pp. 123-128,1992.

[6] H. Liu and R. Setiono,"Feature Selection via Discretization:, IEEE Transaction on Knowledge and Data Engineering, Vol. 9, no. 4, pp. 642-645, 1997.

[7] J.Y. Ching, A.K.C. Wong and K.C.C. Chan, "Class-Dependent Discretization for Inductive Learning from Continuous and Mixed Mode Data", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 17, no. 7, pp. 641-651, 1995.

[8] J.T. Tou and R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, 1874.

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

[10] Huan Liu., Rudy Setiono., "Chi2 : Feature Selection and Discretization of Numeric Attributes.", Proceeding of IEEE 7th International Conference on Tools with Artificial Intelligence.