MCAIM: Modified CAIM Discretization Algorithm for Classification

Shivani V. Vora
M.Tech (Research) Scholar
Department of Computer Engineering, SVNIT, Surat
India

R.G.Mehta
Ph.D Scholar
Department of Computer Engineering, SVNIT, Surat
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, by associating discrete numerical value with each of the generated intervals. Discretization is usually performed prior to the learning process and has played an important role in data mining and knowledge discovery. The results of CAIM are not satisfactory in some cases, led us to modify the algorithm. The Modified CAIM (MCAIM) results are compared with other discretization techniques for classification accuracy and generated the outperforming results. The intervals generated by MCAIM discretization are more in numbers, so to reduce them, the CAIR criterion is used to merge the intervals in MCAIM discretization. It gives better classification accuracy and the reduced number of intervals.

Keywords

Discretization, Class-attribute interdependency maximization, CAIM, MCAIM, and CAIR.

1. INTRODUCTION

In the era of Information Technology, electronic devices are widely used to store huge data. High speed accurate processing is expected on large sized stored data in many applications like prediction of the demand of customers in advance, designing a marketing strategy for the new product to be launched etc. Data mining (DM) provides solution to such problems. DM is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns (model) from data [1]. Collection and storage of large sized data and high speed accurate processing is expected from DM.

Classification is crucially important among the several functions of DM and has been applied successfully to numerous areas such as automatic text summarization and categorization, web usage mining [2], image classification, virus detection of new malicious emails and fraud detection in different fields like credit card transactions, telecommunication industries, e-commerce, insurance, diamond industries [3]. Data collected for classification task involve continues attributes. Some classification algorithms can only handle categorical attributes while others can handle continuous attributes but would perform better on categorical attributes [4]. In order to speed up classification algorithms, improve the predictive accuracy, and generate simple decision rules, lots of discretization algorithms have been proposed to pre-process learning data. Discretization is a process to partition continuous attributes into a finite set of adjacent intervals in order to generate attributes with a small number of distinct values [3]. Discretization is usually performed prior to the learning process. Many discretization processes are embedded with the classification task which is online discretization [3]. This approach increases the classification accuracy but it will increase the process complexity. The objective is to design a classification model in which the proposed algorithm performed discretization as preprocessing task of the classification which is called off line or static discretization. From research, it if found that CAIM is the most popular top-down, static discretization algorithm [5]. The results of CAIM are not satisfactory in some cases. This motivated us to modify the algorithm. Our proposed modified CAIM (MCAIM) algorithm gives improved classification accuracy [6]. Intervals generated by MCAIM algorithm is more in numbers. To remove this restriction some intervals are needed to be merged without loss of discretization information and classification accuracy and for that the discretized attribute intervals are to be merged with the help of CAIR (Class Attribute Interdependence Redundancy) criterion [7]. The results of the proposed algorithm are tested through tree based classification like C5.0 [8] using different real datasets of the UCI repository [9].

The rest of the paper is organized as follows. Section 2 covers the review of some related works. Section 3 presents modified CAIM discretization algorithm and the experimental comparisons of six discretization algorithms on seven real datasets. CAIR based merging in MCAIM discretization algorithm and result analysis is presented in Section 4. Finally, the conclusions are presented in Section 5.

2. RELATED WORKS

The section presents the review of the related works. Since we evaluated the performance of several discretization algorithms in Section 3 by using the famous classification algorithm C5.0, we first gave a brief introduction of classification in Section 2.1. Section 2.2 covers types of discretization algorithms and CAIM discretization algorithm in brief.

2.1 Classification

Classification is a data mining (DM) technique used to predict group membership for data instances. Many classification algorithms are developed such as decision tree [10], classification and regression tree [11], bayesian classification [12], neural networks [13] and K nearest neighbor classification [14]. Among them, the decision tree has become more popular algorithm as it has several advantages like [15], [16]:

- Compared to neural networks or a bayesian based approach; it is more easily interpreted by humans.

- It is more efficient for large training data than neural networks which would require a lot of time on thousands of iterations.
- A decision tree algorithm does not require a domain knowledge or prior knowledge.
- It displays good classification accuracy as compared to other techniques.

A decision tree like C5.0 [8] is a flow-chart-like tree structure, which is constructed by a recursive divide-and conquer algorithm that generates a partition of the data. In a decision tree, each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node is associated with a target class (or class). The topmost node in a tree is called the root, and each path forming from the root to a leaf node represents a rule. Classifying an unknown example begins with the root node, and successive internal nodes are visited until this example has reached a leaf node. Then the class of this leaf node is the predicted class of the example.

2.2 Discretization

Discretization is the process to transforms a continuous attribute values into a finite number of intervals in order to generate attribute with a small number of distinct values. Discretization methods have been developed along different approaches due to different needs: **supervised** versus **unsupervised**, **static** versus **dynamic**, **global** versus **local**, **top-down (splitting)** versus **bottom-up (merging)**, and **direct** versus **incremental** [17]. A lot of discretization algorithms have been proposed [18], [19], [20].

Many researchers have developing the dynamic discretization algorithms for some particular learning algorithms [3]. The C5.0 tree based algorithm uses dynamic discretization also known as online discretization [4]. It discretized continuous attributes when a classifier is being built. But as it discretized continuous attributes when learning starts, the computational complexity is increase in dynamic discretization. And discretized dataset is only used for that particular classification model. Whereas static discretization, also known as off-line discretization in which discretization is completed prior to the learning task [3]. The advantage of static discretization as opposed to dynamic discretization is the independence from the learning algorithms [17]. A dataset discretized by a static discretization algorithm can be used in any classification algorithm that deals with discrete attributes.

2.2.1 CAIM discretization algorithm

From literature survey, CAIM discretization algorithm found to be superior static discretization algorithms. It discretizes an attribute into the smallest number of intervals and maximizes the class attribute interdependency and, thus makes the classification subsequently performed much easier. The algorithm automatically selects the number of discrete intervals without any user supervision. Experiments in [5] showed that CAIM discretization algorithm is superior to other top-down discretization algorithms. Even though CAIM gives better result than other top down methods it still has drawbacks. (i) CAIM usually generate a simple discretization scheme in which the number of intervals is very close to the number of target classes and (ii) For each discretized interval, CAIM considers only the class with the most samples and ignores all the other target classes. Such a consideration is unreasonable in some cases and would decrease the quality. Looking to these limitations, we aimed to modify the original CAIM algorithm. The modification to CAIM algorithm is suggested in the next section.

3. PROPOSED MODIFIED CAIM (MCAIM) DISCRETIZATION

The procedure of proposed MCAIM algorithm [6] is discussed in this section. It uses the CAIM criterion as CAIM discretization.

3.1 CAIM criterion

The Class Attribute Interdependency Maximization (CAIM) criterion measures the dependency between the class variable C and the discretization variable D for attribute A, for a given quanta matrix [5] and is defined as:

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

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 q_{ir} 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 $[d_{r-1}, d_r]$.

3.2 Proposed modified CAIM algorithm

In CAIM discretization process, it only accepts the CAIM value which is highest among all corresponding CAIM values. It will ignore all possible small intervals which have less value of $\max_r$, compared to the $\max_r$ of the interval where the highest CAIM is found. The proposed algorithm changes the comparison scheme. Instead of finding the interval that produces highest CAIM, the search will be performed for the interval which produces local highest CAIM. The process of finding the Local CAIM is suggested in the Modified CAIM algorithm [6], [21]. As the initial process is same as CAIM we have not changed the initial algorithm steps. The modified steps are highlighted with bold text.

Algorithm for Modified CAIM (MCAIM)

```
Step 1:  1.1  Find maximum ($d_n$) and minimum ($d_o$) 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 :{[$d_0$..$d_n$]}
Step 2:  2.1  Initialize k =1.
         2.2  Tentatively add an inner boundary, which is not already in D, from B, and get CAIM value until the lower CAIM is not found.
         2.3  Set the LocalCAIM with the most recent highest CAIM
         2.4  If(k < S) then update D with the accepted in Step 2.3 boundary and set the LocalCAIM=CAIM
         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 (with replacement) in 2.2, it chooses the division boundary that gives the local highest value of the CAIM criterion.

Implementing CAIM discretization algorithm and modified CAIM discretization algorithm, we get discretized attribute for both algorithms. Now to evaluate the effect of generated discretized scheme on the performance of the classification algorithm we use C 5.0 tree based classification algorithm [8].

3.3 MCAIM discretization based classification

Fig. 1 depicts the process flow of MCAIM discretization based classification. Original data is given to Clementine 8.5 to get training data and test data. Now the training data is applied to our MCAIM discretization algorithm. As a result we get discretized attributes values for continuous attributes and also get rules for intervals. Test data is applied to rules for interval generation and we get discretized test data. Discretized training data is given to classification algorithm C5.0 of Clementine 8.5 which defines rules and prepares the model. Now the discretized test data is given to prepared model and as a result we get classification accuracy of given dataset.

To evaluate our proposed MCAIM discretization algorithm, datasets are obtained from UC Irvin ML repository [9].

Implementation and testing methodology is described in next section.

3.4 Implementation and testing methodology

In proposed model, the classification is followed by MCAIM discretization. The tree based classification based on information gain is proposed by Quinlan [8]. We have used C5.0 classification algorithm for measuring efficiency of our proposed MCAIM discretization algorithm.

The accuracy of the classification model is compared for classification with online discretization, CAIM discretization, proposed modified CAIM discretization along with the traditional static techniques like Equal width, Equal count, and Standard Deviation.

The real datasets used throughout the paper to test CAIM and modified CAIM algorithms are:

1. Iris Plants dataset (iris) [9]
2. Pima Indians Diabetes dataset (pid) [9]
3. Wine dataset (wine) [9]
4. Glass Identification dataset (gid) [9]
5. Ecoli data set (ecoli) [9]
6. Pen based recognition of handwritten digits data set (pendigit) [9]
7. Statlog Project Heart Disease data set (sphd) [9]
8. Yeast dataset (yeast) [9]
9. Mammographic Mass dataset (mmd) [9]
10. Blood Transfusion Service Center dataset (btsc) [9]
11. Contraceptive Method Choice dataset (cmc) [9]
12. Hepatitis dataset (hea) [9]

Data sets are obtained from the UC Irvin ML repository [9]. A detailed description of the data sets is shown in table 1. For measuring classification accuracy we have used C5.0 classification of Clementine 8.5 and test results are depicted in the table 2.

3.5 Result Analysis

Table 2 depicts the results of six discretization methods for seven different data sets. Among six discretization methods, Equal-width, equal-count and standard deviation are unsupervised discretization algorithms [22]. A typical problem of unsupervised method is that it is difficult to determine how many intervals are the best for given attribute. It requires users to have sound statistical information of data to determine the number of intervals for each continuous attribute. Whereas supervised methods such as online discretization, CAIM and MCAIM determine the best number of intervals for each continuous attribute. So we will consider the results of three supervised discretization methods. Online discretization method is the dynamic discretization method and remaining are static discretization methods. CAIM and MCAIM are static discretization methods and give outperforming results as compared to online discretization method. So we compare our MCAIM discretization results with the results of CAIM discretization method. Our MCAIM discretization algorithm gives superior results among all discretization methods.

For evaluation of discretization algorithms we use 20% of examples as training data set and 80% of examples as test data set. The classification goodness was measured using accuracy. For measuring classification accuracy we have used C5.0 classification algorithm of Clementine8.5. The test results are depicted in table 2.

The MCAIM algorithm achieved highest classification accuracy for six data sets out of seven data sets and for sphd data set, it achieved second highest classification accuracy.

The intervals generated by MCAIM discretization method are higher in number. So to reduce them, CAIR criterion is used for merging in MCAIM discretization

4. CAIR BASED MERGING IN MCAIM DISCRETIZATION

The procedure of CAIR based merging in MCAIM is discussed in this section.

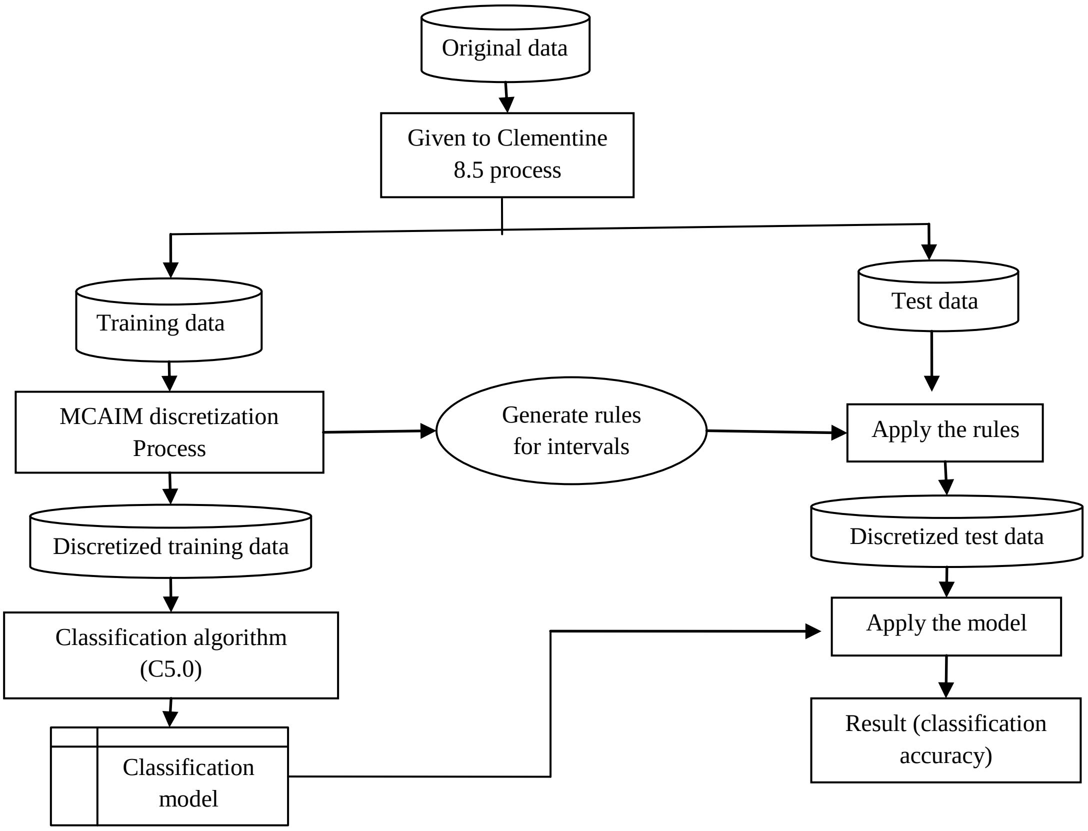
Original data
Given to Clementine 8.5 process
Training data
Test data
MCAIM discretization Process
Generate rules for intervals
Apply the rules
Discretized training data
Discretized test data
Classification algorithm (C5.0)
Apply the model
Classification model
Result (classification accuracy)

Fig. 1: Process of MCAIM discretization based classification

Table 1: Properties of datasets considered in the testing

Properties	**Data sets**											
	iris	**pid**	**wine**	**gid**	**ecoli**	**pendigit**	**sphd**	**yeast**	**mmd**	**btsc**	**cmc**	**hea**
No. Of classes	3	2	3	7	8	10	2	10	2	2	3	2
No. Of examples	150	768	178	214	336	5191	270	1484	961	748	1473	155
No. Of attributes	4	8	13	10	8	16	13	8	6	5	9	20
No. Of continuous attributes	4	7	13	9	7	16	5	4	1	4	2	6

4.1 Classification of CAIR based merging in MCAIM discretization

Basic procedure is same as MCAIM discretization algorithm. First the attribute is discretized with MCAIM discretization algorithm. Now the discretized attributes are then again discretized with the help of CAIR criterion. And the remaining procedure for classification is same as MCAIM discretization shown in fig. 2. In fig. 2 the procedure indicated in different color (green and blue) is the CAIR based merging in MCAIM discretization. The CAIR criterion [7] and the CAIR based merging process described in detail in the next section.

4.2 CAIR criterion

The CAIR criterion can effectively represent the interdependency between the target class and discretized attributes and thus it is widely used to measure the quality of discretization scheme. The larger its value the better correlated is class labels and the discrete intervals. It is also independent of the number of class labels and the number of unique values of the continuous attribute.

The proposed model uses CAIR criterion to merge the intervals. CAIR criterion is used to measure the interdependence between classes and the discretized attribute and expected to be as large as possible [23]. CAIR is defined as:

$$\text{cair} = \sum_{i=1}^{s} \sum_{i=1}^{n} p_{ir} \log_2 \frac{p_{ir}}{p_{i+} p_{+r}} / \sum_{i=1}^{s} \sum_{i=1}^{n} p_{ir} \log_2 \frac{1}{p_{ir}}$$

where $p_{ir} = q_{ir}/M$, $p_{i+} = M_{i+}/M$ and $p_{+r} = M_{+r}/M$ from quanta matrix.

Table 2 Comparisons of six discretization schemes using seven different datasets

Criterion	Discretization methods	Data sets					
		iris	pid	wine	gid	ecoli	sphd
Classification accuracy (%)	Equal- width	84.17%	65.04%	65.73%	60.47%	69.14%	74.07%
	Equal-count	86.67%	66.99%	83.92%	56.14%	72.86%	**75.46%**
	Std. deviation	90%	68.94%	74.13%	54.65%	72.86%	70.37%
	Online discretization	88.33%	69.43%	83.22%	58.14%	80.67	74.07%
	CAIM	72.5%	66.99%	85.31%	63.37%	80.67%	70.37%
	MCAIM	**94.17%**	**70.89%**	**87.41%**	**64.53%**	**81.04%**	74.07%
Total no. of Intervals	Equal- width	5	5	5	5	5	5
	Equal-count	5	4	4	4	4	4
	Std. deviation	-	-	-	-	-	-
	Online discretization	-	-	-	-	-	-
	CAIM	14	22	54	23	10	31
	MCAIM	20	374	383	220	32	151
Tree depth C5.0	Equal- width	1	4	1	5	3	3
	Equal-count	2	2	2	4	3	3
	Std. deviation	2	3	4	5	4	2
	Online discretization	2	5	2	5	5	3
	CAIM	2	9	2	5	3	2
	MCAIM	2	5	2	5	3	3

*Std. stands for standard

4.3 CAIR based merging in MCAIM discretization algorithm

The procedure is to be applied to the Quanta matrix (Q) [5],[24] prepared for the discretized attribute and the class attribute. The sum of column and Rows are to be calculated. Then the CAIR calculation is applied to the Q matrix which is called Global CAIR (GCAIR). Now all consecutive columns of Q are to be merged. We can start either from first column or the last. We started the process from the last column. Temporary matrix Q1 is created which contains one pair of merged columns and remaining as it is. Find the CAIR of Q1, we call it local CAIR (LCAIR) and compare it with GCAIR. If LCAIR found greater then GCAIR, the merge is favorable and copy Q1 to Q. Repeat the process till you find favorable merge. The process is depicts in fig. 3.

To measure the classification accuracy we use real data sets and which are obtained from UCI repository. Main properties of datasets are described earlier in table 1 (in section 3.4 section 3). For evaluation of MCAIM with merging discretization algorithm we use 20% of examples as training data set and 80% of examples as test data set which is 2 fold methods. For measuring classification accuracy we have used C5.0 classification algorithm of Clementine8.5. The test results are depicted in table 3. The datasets used to test CAIM and modified CAIM algorithms are:

1. Yeast dataset (yeast) [9]
2. Pima Indians Diabetes dataset (pid) [9]
3. Wine dataset (wine) [9]
4. Mammographic Mass dataset (mmd) [9]
5. Ecoli data set (ecoli) [9]

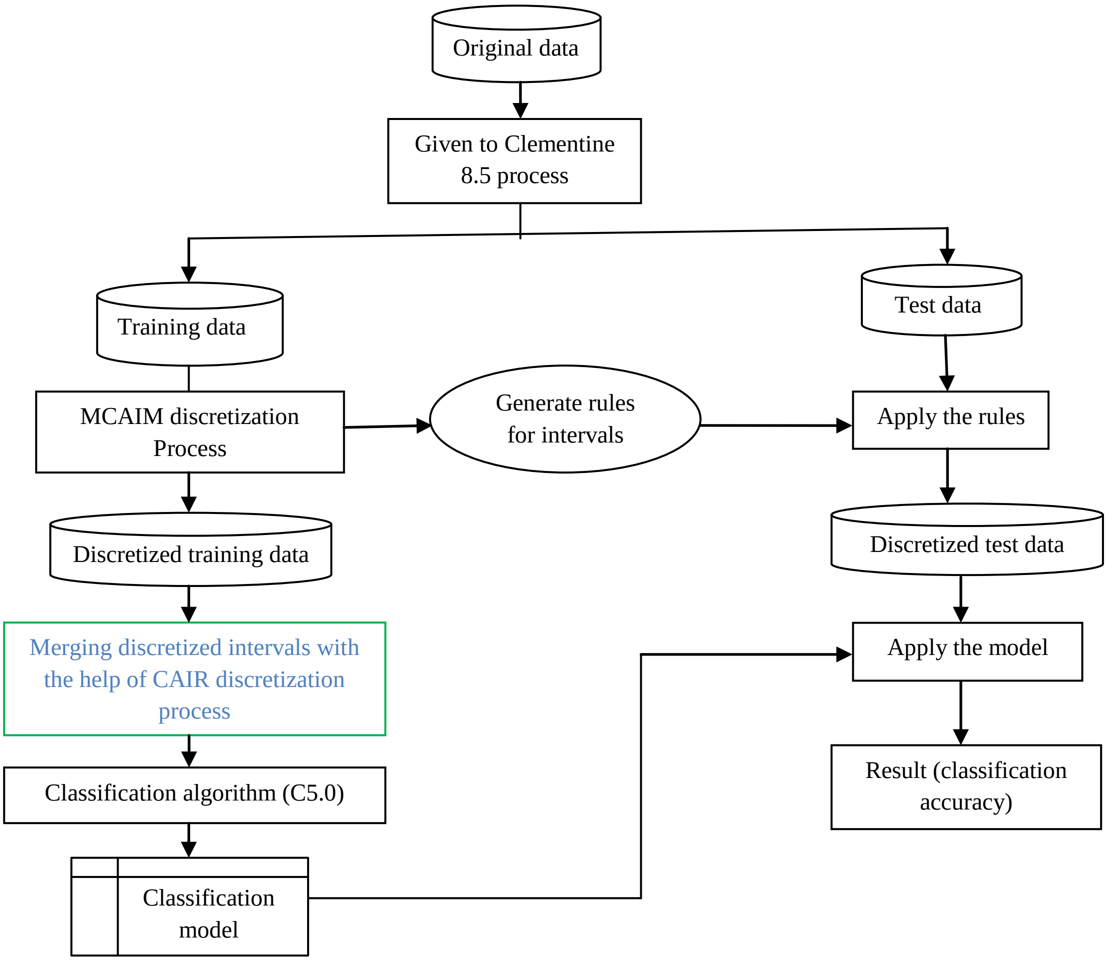
Original data
Given to Clementine 8.5 process
Training data
Test data
MCAIM discretization Process
Generate rules for intervals
Apply the rules
Discretized training data
Discretized test data
Merging discretized intervals with the help of CAIR discretization process
Apply the model
Classification algorithm (C5.0)
Result (classification accuracy)
Classification model

Fig. 2: The classification process of CAIR based merging in MCAIM discretization

6. Statlog Project Heart Disease data set (sphd) [9]
7. Blood Transfusion Service Center data set (btsc) [9]

Table 3 depicts the result of MCAIM with merging along with online discretization, CAIM and MCAIM without merging discretization method for seven different real data sets. We discussed earlier that the (section 3.5 section 3) online discretization is dynamic discretization method whereas the CAIM, MCAIM without merging, and MCAIM with merging are static discretization methods. MCAIM with merging method reduce the number of intervals generated by MCAIM discretization method. From the resultant table 3 we can say that MCAIM with merging discretization algorithm gives better result than the online and CAIM discretization algorithms.

For data sets yeast, pimadiabetes (pid), wine and mammography (mmd), the classification accuracy of MCAIM with merging is better and the intervals generated are also reduced. For ecoli test (ecoli) data set, the classification accuracy is same for MCAIM with merging and MCAIM without merging discretization algorithms and better than online and CAIM discretization. The intervals generated by MCAIM with merging are fewer in numbers. For sphd (statlog heart disease) data set, classification accuracy for MCAIM with merging, MCAIM without merging and online discretization methods remain same but classification accuracy for CAIM is less than the other discretization algorithms. And for btsc (blood transfusion service center) data set, classification accuracy for all discretization algorithms remains same. In sphd, btsc data sets, all attributes other than class attribute do not give more information than class field so there is no effect of CAIM, MCAIM without merging, and MCAIM with merging discretization on classification accuracy.

The above results show that the MCAIM with merging discretization methods gives better result and the reduced number of intervals generated by MCAIM discretization method.

5. CONCLUSION

Discretization methods have played an important role in classification, as it produces concise summarization of continuous attributes to make them easily understandable and make learning more accurate and faster. From the research, CAIM discretization is proven to be the very efficient discretization technique for classification algorithms [5]. But in some cases CAIM algorithm’s results are not satisfactory and to improve results the modified CAIM (MCAIM) discretization algorithm is proposed.

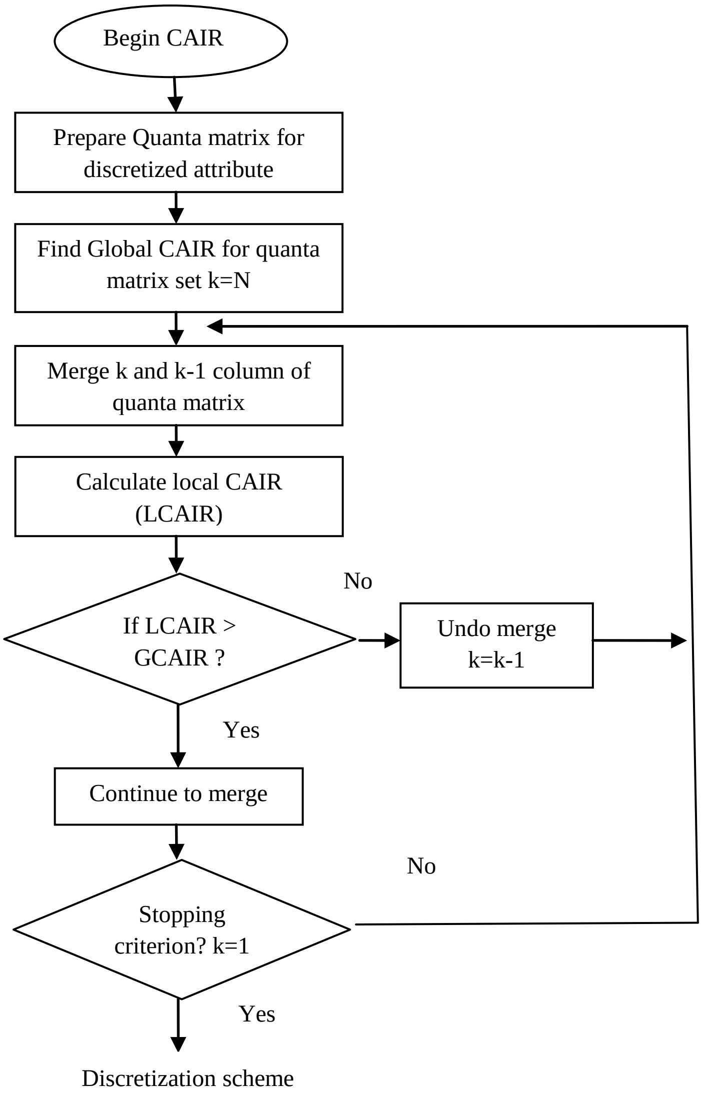
Begin CAIR
Prepare Quanta matrix for discretized attribute
Find Global CAIR for quanta matrix set k=N
Merge k and k-1 column of quanta matrix
Calculate local CAIR (LCAIR)
No
If LCAIR > GCAIR ?
Undo merge k=k-1
Yes
Continue to merge
No
Stopping criterion? k=1
Yes
Discretization scheme

Fig. 3: The Process of CAIR based merging in MCAIM discretization

From the implementation and testing, it is found that our proposed MCAIM discretization algorithm outperforms the other static and dynamic discretization algorithms. Intervals generated by MCAIM discretization are more in numbers so some intervals are needed to be merged without loss of discretization information. For merging we use CAIR criterion. From the performance, it is found that MCAIM with merging discretization gives improved results than Online, CAIM and MCAIM without merging discretization methods.

6. REFERENCES

[1] Jiawei Han and Micheline Kamber, Data Mining –Concept and Techniques, Elsevier: Second Edition

[2] D.P. Rana, R.G Mehta, M.A Zaveri, 2008. Hash based Pattern Discovery Algorithm for Web Usage Mining, ADIT Journal of Engineering, ISSN: 0973 3663, vol. 5, No. 1, (December 2008), pp No. 3-10

[3] Cheng-Jung Tsai, Chien-I. Lee, Wei-Pang Yang, 2007. A discretization algorithm based on Class-Attribute Contingency Coefficient; Elsevier; sciencedirect; Received 27 September 2006; received in revised form 24 August 2007, accepted 2 September 2007

[4] Q. Wu, D.A. Bell, T.M. McGinnity, G. Prasad, G. Qi, X. Huang, 2006. Improvement of decision accuracy using discretization of continuous attributes, in: Proceedings of the Third International Conference on Fuzzy Systems and Knowledge Discovery, Lecture Notes in Computer Science 4223, pp. 674–683

[5] Lukasz A. Kurgan, Member, IEEE, and Krzysztof J. Cios, Senior Member, IEEE, 2004. CAIM Discretization Algorithm; IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 2

[6] R.G Mehta, 2009. A Novel Fuzzy Based Classification algorithm for Data Mining using Fuzzy Discretization" World congress on Computer Science and Information Engineering (CSIE-2009), Sponsored by IEEE, Los Angeles, USA

[7] K.J. Cios, W. Pedrycz and R. Swiniarski, 1998. Data Mining Methods for Knowledge Discovery, Kluwer, http://www.wkap.nl/ book.htm/0-7923-8252-8

[8] J. Ross Quinlan, 1993. C4.5: programs for machine learning, Morgan Kaufmann Publishers Inc

[9] http://archive.ics.uci.edu/ml/datasets.html

[10] S. Cohen, L. Rokach, O. Maimon, 2007. Decision-tree instance-space decomposition with grouped gain-ratio, Information Sciences, pp. 3592–3612

[11] Breiman L., Friedman, J. H., Olshen R. A., and Stone C. J., 1984. Classification and Regression Trees, Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

[12] David HeckerMann, A Tutorial On Learning With Bayesian Networks, March 1995 (Revised November 1996)

[13] Raul Rojas, 1996. Neural Networks - A Systematic Introduction, Springer-Verlag

[14] Cover, T., Hart, P., 1967. Nearest neighbor pattern classification, IEEE Trans. on Information Theory, vol.13, no.1,pp. 21–7

[15] Shivani V. Vora and Rupa G. Mehta, “Classification techniques for environmental data: A survey”, International Congress of Environment Research (ICER-11), SVNIT, Surat, Dec 15-17, 2011.

[16] R. Rastogi, K. Shim, A decision tree classifier that integrates building and pruning, Proc. of the twenty forth Int’l Conf. on Very Large Databases, (1998) , pp. 404–415

[17] H. Liu, F. Hussain, C.L. Tan, M. Dash, 2002. Discretization: an enabling technique, Journal of Data Mining and Knowledge Discovery 6(4) 393–423

[18] M. Boulle, Khiops, A statistical discretization method of continuous attributes, Machine Learning 55 (1) (2004) 53–69

[19] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, in: Proceeding of Twelfth International Conference on Machine Learning, 1995, pp. 194–202

[20] U.M. Fayyad, K.B. Irani, On the handling of continuous-valued attributes in decision tree generation, Machine Learning 8 (1992) 87– 102

[21] Shivani Vora and Rupa G. Mehta, 2011. MCAIM: modified CAIM discretization, International Journal of computer Science and Engineering, Vol.8, Issue 1, pp.16-20 ISSN (online): 2043-9091

[22] Catlett, J. 1991. On changing continuous attributes into ordered discrete attributes. In proc. of fifth European working session on learning. Berlin: Springer-Verlag, pp. 164–177

[23] Michalski, R. S., Chilausky, R. L., 1980. Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing and expert system for soybean disease diagnosis, Policy Analysis and Information Systems

[24] Y. Linde, A. Buzo, R. M. Gray, 1980. An Algorithm for Vector Quantizer Design, IEEE Trans. Comm., vol. 28, no. 1, pp. 84-95

Table 3 comparisons of four discretization schemes using seven different datasets with 2 fold method

Criterion	Discretization methods	Data sets						
		Yeast	**pid**	**wine**	**mmd**	**ecoli**	**sphd**	**btsc**
Classification accuracy (%)	**Online discretization**	52.10%	69.93%	83.22%	76.72%	80.67%	74.07%	75.63%
	CAIM	52.27%	66.99%	85.31%	79.97%	80.67%	70.37%	75.63%
	MCAIM without merging stage	51.35%	70.89%	87.41%	76.72%	81.04%	74.04%	75.63%
	MCAIM with merging	**56.48%**	**72.52%**	**88.81%**	**81.92%**	**81.04%**	**74.04%**	**75.63%**
Total no. of Intervals	**Online discretization**	-	-	-	-	-	-	-
	CAIM	3,6,6,7,6,5	4,5,7,6	4,4,3,3,4,10,5,4,3,2,3,4,5	5	6,4	6,5,9,5,6	6,5,6,5
	MCAIM without merging stage	28,30,21,26,24,22	40,62,248,20	30,48,32,20,14,42,30,15,45,24,16,37,30	14	12,20	15,65,33,23,15	14,13,13,34
	MCAIM with merging	**3,3,2,4,2,3**	**2,2,2,2**	**2,3,2,2,2,2,2,2,2,2,2,2,2**	**2**	**3,3**	**2,2,2,2,2**	**2,2,2,3**

Tree height C5.0	**Online discretization**	17	5	2	3	5	3	-
	CAIM	6	9	2	3	3	2	-
	MCAIM without merging stage	13	5	2	3	3	3	-
	MCAIM without merging stage	**5**	8	**2**	**3**	**3**	**3**	-