Discretization Algorithm that Uses Class-Attribute Interdependence Maximization

Lukasz Kurgan
Department of Computer Science and Engineering
University of Colorado at Denver,
Department of Computer Science
University of Colorado at Boulder

Krzysztof J. Cios
Department of Computer Science and Engineering
University of Colorado at Denver,
University of Colorado Health Sciences Center,
Department of Computer Science
University of Colorado at Boulder,
4cData, LLC

Abstract - *Most of the existing machine learning algorithms are able to extract knowledge from databases that store discrete attributes (features). If the attributes are continuous, the algorithms can be integrated with a discretization algorithm that transforms them into discrete attributes. The paper describes an algorithm, called CAIM (class-attribute interdependence maximization), for discretization of continuous attributes that is designed to work with supervised learning algorithms. The algorithm maximizes the class-attribute interdependence and, at the same time, generates possibly minimal number of discrete intervals. Its big advantage is that it does not require the user to pre-define the number of intervals, in contrast to many existing discretization algorithms. The CAIM algorithm and five other state-of-the-art discretization algorithms were tested on well-known machine learning datasets consisting of continuous and mixed-mode attributes. The tests show that the proposed algorithm generates discrete attributes with, almost always, the highest class-attribute interdependency when compared with other algorithms, and at the same time it always generates the lowest number of intervals. The discretized datasets were used in conjunction with the CLIP4 machine learning algorithm. The accuracy of the rules generated by the CLIP4 shows that the proposed algorithm significantly improves classification performance; it also performs best in comparison with other five discretization algorithms. The CAIM algorithm's speed is comparable to the simplest unsupervised algorithms and outperforms other supervised discretization algorithms.*

Keywords: Discretization, class-attribute inter-dependency maximization, CAIM algorithm, machine learning, classification, CLIP4 algorithm

1. Introduction

The process of automation of processing and extraction of knowledge from data becomes an important task that is often performed by machine learning (ML) algorithms. One of the most common tasks performed by ML algorithms is generation of classification rules from class-labeled examples. The examples are described by a set of numerical, nominal, or continuous attributes. Many existing inductive ML algorithms are designed expressly for handling numerical or nominal data, while some algorithms perform better with discrete-valued attributes despite the fact that they can also handle continuous attributes [1],[9]. This drawback can be overcome by using a discretization algorithm as a front-end for the learning algorithm.

Discretization is a process of transforming a continuous attribute values into a finite number of intervals and associating with each interval a discrete, numerical value. The usual approach for learning tasks that use mixed-mode (continuous and discrete) data is to perform discretization prior to the learning process [1],[7],[8],[12].

The discretization process first finds the number of discrete intervals, and then the width, or the boundaries for the intervals, given the range of values of a continuous attribute. Most often the user must specify the number of intervals, or provide some heuristic rule to be used [2]. The proposed CAIM algorithm performs both tasks by automatically selecting the number of discrete intervals and finding width of every interval based on interdependency between class and attribute values.

Discretization algorithms can be divided into two categories:

- unsupervised (class-blind) algorithms that discretize attributes without taking into account respective class labels. The representative algorithms are equal-width and equal-frequency discretizations [3].
- supervised algorithms discretize attributes by taking into account the class-attribute interdependence. The representative algorithms are: maximum entropy [16], Patterson-Niblett algorithm [11], which is built-in as a front end into a decision trees algorithm [13], and other information-gain or entropy-based algorithms [7],[18], statistics-based algorithms like ChiMerge [9] or Chi2 [10], class-attribute interdependency algorithms like CADD algorithm [2] and clustering-based algorithms like K-means discretization algorithm [15].

Discretization should significantly reduce the number of possible values of the continuous attribute since large number of possible attribute values contributes to slow and ineffective process of inductive machine learning [1]. Thus, a supervised discretization algorithm should seek possibly minimum number of discrete intervals, and at the same time it should not weaken the interdependency between the attribute values and the class label. The proposed CAIM discretization algorithm not only discretizes an attribute into the small number of intervals but also makes it easier for the subsequent machine learning task by maximizing the class-attribute interdependency. The CAIM algorithm does not require user interaction since it automatically picks proper number of discrete intervals. The CAIM algorithm is compared with five well-known discretization algorithms two of which are unsupervised (equal-width and equal frequency algorithms) and the remaining three are supervised (Patterson-Niblett, Maximum Entropy, and CADD) giving always the smallest number of discrete intervals and almost always the highest class-attribute interdependency. The CAIM algorithm and the five algorithms were used with the CLIP4 machine learning algorithm [4],[5] to generate rules. The accuracy of the rules generated from discretized data shows that the CAIM algorithm significantly improves the classification performance and performs best among all considered discretization algorithms.

1.1. Basic definitions of the class-attribute interdependent discretization

The CAIM algorithm uses the class-attribute dependency information as the criterion for the optimal discretization, which has the minimum number of discrete intervals and minimum loss of the class-attribute interdependency. After Ching, Wong & Chan [2] we introduce several basic definitions.

For a certain classification task, let us assume that we have a training data set consisting of *M* examples, and that each example belongs to only one of the *S* classes. *F* will indicate any of the continuous attributes from the mixed-mode data. Then there exists a discretization scheme *D* on *F*, which discretizes the continuous domain of attribute *F* into *n* discrete intervals bounded by the pairs of numbers:

$$\mathrm{D}:\{[\mathrm{d}_0,\mathrm{d}_1],(\mathrm{d}_1,\mathrm{d}_2],\ldots,(\mathrm{d}_{n-1},\mathrm{d}_n]\}$$

where d_0 is the minimal value and d_n is the maximal value of attribute *F*, and the values are arranged in the ascending order. These values constitute the boundary set $\{d_0, d_1, d_2, \ldots, d_{n-1}, d_n\}$ for discretization *D*.

In *D* each value belonging to attribute *F* can be classified into only one of the *n* intervals. With the change of discretization *D*, the membership of each value in a certain interval for attribute *F* may also change. The class variable and the discretization variable of attribute *F* can be treated as two random variables, thus a two-dimensional frequency matrix (called quanta matrix) can be set up as shown in Table 1.

In Table 1, 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, and 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*.

Table 1. 2-D frequency matrix for attribute F and discretization scheme *D*

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

Based on the quanta matrix the Class-Attribute Interdependence Redundancy (CAIR) criterion [17] has been proposed. The CAIR has been used as a discretization criterion in the class-attribute dependent discretizer (CADD) algorithm [2]. In the nutshell, the CAIR criterion reflects the interdependence between classes and the discretized attribute, being at the same time independent of the number of class labels and the number of unique values of the continuous attribute. The larger the value of the CAIR the better correlated are the class labels and the discrete intervals. For details on the CAIR criterion the reader is referred to [6]. The CADD algorithm has several problems. It uses user-specified number of intervals and the maximum entropy discretization method to initialize the intervals, which may cause the algorithm to remain in the worst starting point in terms of the CAIR criterion. Finally, experience is required for selection of a confidence interval for the significance test used in the algorithm. The CAIM algorithm has no disadvantages associated with the CADD algorithm.

2. The CAIM Algorithm

The CAIM's algorithm goal is to maximize the dependency relationship between the class labels and the continuous-valued attribute, and at the same time to minimize the number of discrete intervals. Additional goal is to design an algorithm that performs the discretization at reasonable computational cost so that it can be applied to continuous attributes that have large number of unique values.

2.1. The Discretization Criterion

Given the quanta matrix defined in Table 1, the Class-Attribute Interdependency Maximization (CAIM) criterion that measures the dependency between the class variable *C* and the discretization variable *D* for attribute *F* is defined as:

$$CAIM(C,D \mid F) = \frac{\sum_{i=1}^{n} \frac{\max_i^2}{M_{ir}}}{n}$$

where: n is the number of intervals

i iterates through all intervals, i.e. i=1,2,...,n

max_i is the maximum value among all q_{ir} values (maximum value within the i^{th} column of the quanta matrix), r=1,2,...,*S*

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

The CAIM criterion is used as a discretization criterion in the proposed class-attribute interdependency maximization algorithm, also called CAIM. The CAIM criterion is a heuristic measure that quantifies the interdependence between classes and the discretized attribute. The criterion is independent of the number of classes and the number of unique values of the continuous attribute.

The CAIM criterion has the following properties:

- The larger the value of the CAIM the better correlated are the class labels and the discrete intervals
- The algorithm favors discretization schemes where each interval has all of its values grouped within a single class label. This observation was our motivation for using the max_i values within each of the n intervals, and summing them for all intervals.
- The squared max_i value is scaled by the M_{ir} to eliminate negative impact that the values belonging to other classes have on the class with the maximum number of values on the discretization scheme (all values other than the max value for the i^{th} column of the quanta matrix)
- The summed value is divided by the number of intervals n because then the criterion favors discretization schemes with smaller number of

intervals, which is one of the goals of the CAIM algorithm

The value of the CAIM criterion is calculated with a single pass over the quanta matrix. The CAIM criterion maximizes the class-attribute interdependency.

2.2. The CAIM Algorithm

Since the problem of finding discretization scheme with globally optimal value of the class-attribute interdependency is highly combinatorial the CAIM algorithm uses greedy approach, which finds local maximum values of the CAIM criterion. Although the CAIM does not guarantee finding the global maximum it is computationally efficient and effective, as shown in the experimental section. The pseudocode of the CAIM algorithm follows:

```
Given: Data consisting of M examples, S classes, and continuous attributes $F_i$
For every $F_i$ do:
Step1.
  1.1 find maximum ($d_n$) and minimum ($d_o$) values of $F_i$
  1.2 form a set of all 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 $D:\{[d_0,d_n]\}$, set GlobalCAIM=0
Step2.
  2.1 initialize k=1;
  2.2 tenatively add an inner boundary, which is not already in D, from B, and calculate corresponding CAIM value
  2.3 after all the tentative addition 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 CAIM algorithm works in a greedy top-down manner. It starts with a single interval and divides it iteratively, using for the division the boundary that gave the highest values of the CAIM criterion. The algorithm assumes that every discretized attribute needs at least number of intervals equal to the number of classes.

The CAIM algorithm uses trade-off between finding a discretization with the highest possible class-attribute interdependency, and a reasonable computational cost. The main advantage of CAIM algorithm is that it finds small number of discretization intervals, which gives the low computational cost, and at the same time high class-attribute interdependency.

3. Experiments

The four datasets used to test the CAIM algorithm are:

1. Statlog Project Heart Disease dataset (*hea*)
2. Pima Indians Diabetes dataset (*pid*)
3. Thyroid Disease dataset (*thy*)
4. Waveform dataset (*wav*)

The datasets were obtained from the UCI Irvine ML repository [14]. Detailed description of the datasets is shown in the Table 2.

The CAIM algorithm performance was compared with five state-of-the-art discretization algorithms. Two were unsupervised: equal-width and equal frequency algorithms, and three supervised: Patterson-Niblett, Maximum Entropy, and CADD. All the algorithms were used to discretize all four datasets. The quality of the discretization was evaluated based on the CAIR criterion value, number of generated intervals, and the time of execution. The CAIM algorithm performance was compared with the five discretization algorithms.

Later, the discretized datasets were used to generate classification rules by the CLIP4 machine algorithm [4],[5] and the accuracy of generated rules was compared among the six discretization algorithms over all datasets.

Table 2. Major properties of datasets considered in the experimentation

Dataset	# of classes	# of examples	# of training / testing examples	# of attributes	# of continuous attributes
hea	2	270	10 x cross-validation	13	6
pid	2	768	10 x cross-validation	8	8
thy	3	7200	3772 / 3428	21	6
wav	3	3600	600 / 3000	21	21

Table 3. Comparison of the six discretization schemes using four continuous and mixed-mode datasets (bolded values show the best results)

Criterion	**Discretization Method**	**Dataset**			
		thy	wav	hea	pid
CAIR (mean value through all intervals)	Equal Width	0.07	0.07	0.09	0.06
	Equal Frequency	0.04	0.06	0.08	0.05
	Paterson-Niblett	0.14	**0.14**	0.09	0.05
	Maximum Entropy	0.03	0.06	0.08	0.05
	CADD	0.03	0.07	0.09	0.06
	CAIM	**0.17**	0.13	**0.14**	**0.08**
total # of intervals	Equal Width	126	630	57	106
	Equal Frequency	126	630	57	106
	Paterson-Niblett	45	252	47	59
	Maximum Entropy	126	630	57	97
	CADD	80	627	56	96
	CAIM	**18**	**63**	**12**	**16**
time [s]	Equal Width	6.78	9.77	0.09	0.31
	Equal Frequency	6.52	9.48	0.08	0.30
	Paterson-Niblett	218.88	4808.66	1.79	21.87
	Maximum Entropy	51.39	299.27	0.34	2.47
	CADD	628.64	8287.80	2.03	27.86
	CAIM	103.47	1143.91	0.31	4.95

3.1. Analysis of the results

First, the four datasets were discretized using the six discretization methods mentioned above, and the quality of the discretization was evaluated based on the CAIR criterion value, the number of generated intervals, and time of execution. The CAIR criterion was used to evaluate different discretization algorithms since the goal of discretization is to maximize the class-attribute interdependence redundancy. After [2] this can be done by finding a discretization scheme, D_{MAX}, out of all possible discretization schemes, D, such that:

$$CAIR(D_{MAX}) \geq CAIR(D_i)\ \forall(D_i \in D)$$

Although the CAIM criterion has the same goal, it is a new heuristic measure and thus the CAIR criterion was used to evaluate the discretization schemes.

Table 3 shows the results of discretizing the datasets using all considered discretization schemes. The CAIM algorithm achieved the highest class-attribute interdependency for 3 out of 4 datasets, and for *wav* datasets was the second highest. That verifies that the greedy approach and the CAIM criterion work in practice, and results in higher interdependence between class and attribute variables than the interdependence achieved by other algorithms.

For all datasets the CAIM algorithm generated discretization scheme with significantly smaller number of intervals than schemes generated by other discretization algorithms. It is a very significant advantage that helps to better understand the meaning of the discretized attributes, and reduces the size of data.

Table 4. Comparison of the accuracies achieved by the CLIP4 algorithm for the four datasets using the six discretization schemes (bolded are the best results, hea and pid results are averaged over 10CV)

Accuracy	Discretization Method	Dataset				RANK
		thy	wav	hea	pid	
CLIP4 accuracy [%]	Equal Width	86.0	50.7	65.5	64.5	4
	Equal Frequency	**98.2**	42.9	63.3	72.6	3.3
	Paterson-Niblett	96.7	62.8	72.7	68.5	3.0
	Maximum Entropy	96.8	42.4	63.4	62.6	4.8
	CADD	76.9	42.1	65.5	72.2	4.5
	CAIM	98.1	**74.1**	**72.9**	**79.3**	**1.3**

The shortest execution time obviously was achieved for unsupervised discretization algorithms. Within the group of supervised algorithms the CAIM and Maximum Entropy algorithms achieved comparable execution time, outperforming the CADD and Paterson-Niblett algorithms, in particular for the *wav* dataset.

After we discretized the datasets they were used as input to the CLIP4 algorithm to generate classification rules. The purpose of this experiment was to show the impact of the selection of a good discretization algorithm on the accuracy of the subsequently used machine learning algorithm. Thus, again the accuracy was compared for the six discretization algorithms, over all discretized datasets. The results can be easily compared by looking at the RANK column that defines each algorithm's rank for a particular dataset among the six algorithms, averaged over all four datasets. Table 4 shows the accuracy results.

The best accuracy was achieved for the data that was discretized using the CAIM algorithm. The difference between the rank of the CAIM algorithm (1.3) and the next best algorithm (Paterson-Niblett with rank 3.0) is substantial. The accuracy results show that the CAIM algorithm generates the data that performs better then the data generated by the other discretization algorithms when subsequently used for supervised learning.

In a nutshell the CAIM algorithm discretized the datasets in a way that resulted in the smallest number of intervals, and the highest class-attribute interdependency when compared with other state-of-the-art discretization algorithms. The CAIM algorithm has execution time that assures its applicability for real-life problems. In addition, the use of the CAIM algorithm significantly improves the accuracy of results achieved by a subsequently used machine learning algorithm. The results show high applicability of the CAIM algorithm since it outperformed the other five discretization algorithms. The future work will include more extensive experimental work.

4. Summary and Conclusions

We proposed a new algorithm, called CAIM, for discretization of continuous attributes. The CAIM algorithm can be used with any class-labeled data. The CAIM maximizes mutual interdependence of the class labels and the attribute intervals, and at the same time generates possibly the smallest number of intervals for a given continuous attribute. The tests performed using the CAIM algorithm show that it generates discretization schemes with almost always the highest dependence between the class labels and the discrete intervals, and always with significantly lower number of intervals, when compared with other state-of-the-art discretization algorithms. The use of the CAIM algorithm as a preprocessing step for a machine learning algorithm significantly improves the results in terms of the accuracy, which are better than by using other discretization algorithms.

An important feature of the CAIM algorithm is that it automatically selects the number of intervals, in contrast to many existing discretization algorithms. The CAIM algorithm's execution time is comparable to the time of the simplest unsupervised discretization algorithms, and outperforms some supervised algorithms. The above advantages make the CAIM algorithm suitable for discretization of data representing a variety of real life problems.

References

[1] Catlett, J.: On Changing Continuous Attributes into ordered discrete Attributes, Proc. European Working Session on Learning, pp.164-178, 1991

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

[3] Chiu D., Wong A. & Cheung B.: Information Discovery through Hierarchical Maximum Entropy Discretization and Synthesis, In: Piatesky-Shapiro G., Frowley W.J. (Eds.) Knowledge Discovery in Databases,MIT Press, 1991

[4] Cios, K.J., & Kurgan, L., Hybrid Inductive Machine Learning Algorithm that Generates Inequality Rules, *Information Sciences*, Special Issue on *Soft Computing Data Mining*, in review, 2001

[5] Cios, K. J., & Kurgan, L., Hybrid Inductive Machine Learning: An Overview of CLIP Algorithms, In: Jain L.C., and Kacprzyk J. (Eds.) *New Learning Paradigms in Soft Computing*, pp.276-322, Physica-Verlag (Springer), 2001

[6] Cios, K. J., Pedrycz, W. & Swiniarski, R.: Data Mining Methods for Knowledge Discovery. Kluwer, 1998

[7] Dougherty J., Kohavi R. & Sahami M.: Supervised and Unsupervised Discretization of Continuous Features, Proc. of the 12th International Conference on Machine Learning, pp.194-202, 1995

[8] Fayyad U.M. & Irani K.B.: On the Handling of Continuous-Valued Attributes in Decision Tree Generation, Machine Learning, v.8, pp.87-102, 1992

[9] Kerber R.: ChiMerge: Discretization of Numeric Attributes, Proc. AAAI-91, 9th International Conference on Artificial Intelligence, pp.123-128, 1992

[10] Liu H. & Setiono R.: Feature Selection via Discretization, IEEE Transactions on Knowledge and Data Engineering, v.9, no.4, pp.642-645, 1997

[11] Paterson, A. & Niblett, T.B.: ACLS Manual, Edinburgh: Intelligent Terminals, Ltd, 1987

[12] Pfahringer B.: Compression-Based Discretization of Continuous Attributes, Proc. of the 12th International Conference on Machine Learning, pp.456-463, 1995

[13] Quinlan, J.R.: C4.5 Programs for Machine learning, Morgan-Kaufmann, 1993

[14] The University of California, UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html

[15] Tou J.T. & Gonzalez R.C.: Pattern Recognition Principles, Addison-Wesley, 1874

[16] Wong A.K.C. & Chiu D.K.Y.: Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data, IEEE Transactions on Pattern Analysis and Machine Intelligence, v.9, pp. 796-805, 1987

[17] Wong A.K.C. & Liu T.S.: Typicality, diversity and feature pattern of an ensemble, IEEE Trans. Computers, v.24, pp.158-181, 1975

[18] Wu X.: A Bayesian Discretizer for Real-Valued Attributes, The Computer Journal, v.39, 1996