Class-Dependent Discretization for Inductive Learning from Continuous and Mixed-Mode Data

John Y. Ching, Andrew K. C. Wong, *Member, IEEE*, and Keith C. C. Chan

Abstract—**Inductive learning systems can be effectively used to acquire classification knowledge from examples. Many existing symbolic learning algorithms can be applied in domains with continuous attributes when integrated with a discretization algorithm to transform the continuous attributes into ordered discrete ones. In this paper, a new information theoretic discretization method optimized for supervised learning is proposed and described. This approach seeks to maximize the mutual dependence as measured by the interdependence redundancy between the discrete intervals and the class labels, and can automatically determine the most preferred number of intervals for an inductive learning application. The method has been tested in a number of inductive learning examples to show that the class-dependent discretizer can significantly improve the classification performance of many existing learning algorithms in domains containing numeric attributes.**

Index Terms—**Inductive learning, classification, discretization, continuous attributes, mixed-mode attributes, maximum entropy, mutual information, uncertainty.**

I. Introduction

In machine learning research, *inductive learning* (*IL*) has gained prominence due to promising experimental results and successful commercial applications to knowledge acquisition in expert systems (e.g., [1], [5], [6], [21]). IL systems have also been applied successfully to some of the traditional pattern recognition problems [7], [14]. A central task of IL is the construction of classification rules from examples. Given a set of pre-classified examples described in terms of some attributes, the goal of an IL system is to derive a set of rules that can be used to assign new events to the appropriate classes. This type of learning is also referred to as *supervised learning*. The most successful supervised learning systems include ID3 and related decision tree based systems [19] and the AQ family of inductive learning algorithms [15].

In a typical IL task, the training events are described by a set of characteristics or attributes. Some of the attributes characterizing an event instance may be *symbolic* or *discrete*. Other attributes may be *real* or *continuous*. While many existing inductive learning systems have been designed specifically for handling discrete and symbolic attribute values in an attempt to address the shortcomings of traditional pattern recognition methods that could only deal with continuous data [11], there is yet no fully integrated approach that can deal with *mixed-mode* continuous and discrete data [25]. In fact, the topic of handling continuous data by IL algorithms has been mostly ignored in the literature until recently. Since a continuous variable can be *discretized* into a finite number of discrete intervals, the current consensus for addressing the mixed-mode classification problem is to partition the continuous attributes into ordered discrete attributes prior to the learning process [4], [5], [12], [23].

Unfortunately, the number of ways to discretize a continuous attribute is infinite. To partition a continuous variable, two important decisions must be made. First, the number of discrete intervals must be selected. The selection of the optimal number of intervals is seldom addressed by existing discretization methods, and in most cases the human user selects (sometimes arbitrarily) the appropriate number of intervals [5], [23], [25]. Secondly, the width of the intervals must be determined. In other words, the boundaries of the discretized intervals need to be defined. Some of the traditional pattern recognition and data analysis methods have been tried with IL applications with limited success. The simplest discretization procedure is to divide the range of a continuous variable into *equal-width* intervals [25]. Given a sample of observed values of a continuous variable, the equal-width method involves the determination of the range of values from the minimum and maximum observed attribute values. The range is then divided equally by a user-defined number of intervals. The obvious weakness of this procedure is that in cases where the outcome observations are not distributed evenly, a large amount of important information can be lost after the discretization process. To reduce the amount of information loss due to discretization, a method based on the concept of *maximum marginal entropy* has been developed [25]. The method partitions a continuous attribute using a criterion to maximize Shannon's entropy and thus minimizes the loss of information. The number of intervals is determined using a rule of thumb based on the fact that more intervals generally mean less information loss. However, since the method relies on reliable probability estimation which is affected by the sample size, the upper bound of the number of intervals must be constrained by the second-order statistics required for the probability estimation. Since the problem of finding global maximum entropy is highly combinatorical, a heuristic approximation using marginal entropy has been used to discretize continuous variables in object recognition and clustering applications [25] as well as in inductive learning tasks [5].

Manuscript received July 26, 1993; revised March 28, 1995.

J.Y. Ching and A.K.C. Wong are with the PAMI Laboratory, Department o Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

K.C.C. Chan is with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

IEEECS Log Number P95095.

A variation on the entropy scheme determines the interval boundaries by making the total gain of information from the observed occurrences in each interval equal. Called the *even information intervals quantization* method [23], this procedure also relies on the human user to provide the appropriate number of intervals. Once the number of interval is selected, this method determines the optimal interval boundaries by equalizing the total information gain in each interval. This method has been used with an IL system to decompose and classify continuous myoelectric signals [23].

The common advantage of the above methods is that they can be easily incorporated into any existing IL algorithms. However, they are not ideal for supervised learning applications because their criteria of discretization fail to take into consideration the relationship between pre-assigned classes and the interval boundaries. Secondly, the selection of the number of intervals is not adequately addressed. As pointed out in [23], the number of intervals has a profound effect on learning performance and classification accuracy. Wong and Chiu suggest the best number of intervals is the largest possible one given the particular sample size and the available resources to minimize information loss [25]. For an inductive learning application, however, large numbers of intervals are not always preferred because the performance of many inductive learners deteriorates dramatically with large numbers of discrete intervals. After all, the reason for discretization is to significantly reduce the number of possible outcomes of an attribute. Therefore, for the purpose of supervised learning, the optimal number of intervals can be regarded as the *minimum* possible that does not significantly weaken the interdependency between the attribute values and the classes.

Within the machine learning community, a number of algorithm-specific discretization schemes for the decision-tree family of algorithms have also been proposed. These methods are generally based on the attribute partitioning algorithm inherent in ID3 [19]. The algorithm of Fayyad and Irani [12] formalizes the attribute binarization scheme used by some ID3 descendants such as ACLS and ASSISTANT [1] to partition all continuous attributes into binary variables using class entropy minimization heuristic. Catlett [4] proposed a similar method called D-2 which can discretize a continuous variable into multiple intervals using a training set partitioning and thresholding approach specific to decision tree generation. In both cases, the motivation for discretization was to improve the learning speed of the ID3 algorithm when continuous attributes are encountered. Unfortunately, as pointed out in [4], it is unclear how algorithms that do not partition the training set, such as AQ and others, can benefit from these techniques.

More recently, it has been suggested that standard clustering algorithms can be incorporated in classification systems to handle continuous or mixed mode features. Caelli and Pennington's PCIT classifier [3] is an evidence-based classification system integrating standard clustering techniques such as Leader and K-means methods [13] and a multi layered perceptron. An entropy minimization heuristic is also used here to obtain the least evenly distributed clusters according to class labels. Although PCIT does not explicitly address the problem of continuous attributes, it may be possible to interpret the cluster bounds as discretization intervals given a continuous feature space. Despite the similarity between discretization and some clustering problems, there has been no work done to show whether the connection can be made stronger to further contribute to the handling continuous attributes by existing symbolic learning algorithms.

Our motivation for this work is to find a discretization technique suitable for APACS [7]. We are also interested in general discretization methods that can be universally applied to all types of existing inductive learning algorithms so that we can fairly evaluate the performance of these algorithms in different continuous and mixed-mode domains. In this paper, we show that the proposed discretization method is in fact effective for *any* type of inductive learning systems. Two families of very different and well-known inductive learners, namely ID3 and AQ, in addition to APACS, have been tested with the proposed discretization algorithm using continuous and mixed-mode data from various domains.

II. Class-Dependent 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 (a continuous random variable with some probability distribution function) into an ordered discrete attribute with a number of discrete intervals. In practice, since only a sample of observed outcomes of a continuous attribute is often available, discretization is equivalent to the process of reducing the number of states of an ordered discrete random variable by combining some of its states together.

A. Basic Data Representation and Definitions

Given a classification problem, suppose there is a set of M training instances which may be events, objects, observations, processes, etc. Each of these instances has been preclassified into one of S classes, c_s, $s = 1, \cdots, S$, and is described by n distinct attributes, $A_1, \cdots, A_j, \cdots, A_n$. For any attribute A_j, there is a domain of plausible values defined as domain $(A_j) = \{v_{jk} | k=1, \cdots, K\}$, where v_{jk} can be numeric, symbolic, or both.

DEFINITION 1: *Let the interval $[a, b]$ be the range space of the continuous-valued attribute A_j, where $a \le v_{jk} \le b$. A partition T_j on A_j as a set of L_j intervals is defined as*:

$$T_j: \{[e_0, e_1], [e_1, e_2], \cdots, [e_{Lj}\text{-}1, e_{Lj}]\}$$

where $e_0 = a$ represents the lower boundary of the observed value range, $e_{Lj} = b$ represents the upper value boundary of the attribute, and $e_{i-1} < e_i$ for $i = 1, 2, \cdots, L_j$.

DEFINITION 2: *Associated with the partition, there is a boundary set B_j which is defined to be the set of ordered endpoints $e_0, e_1, \cdots, e_{Lj}$ which delimits the L_j intervals. Suppose C represents the random variable whose values c_s are the class labels among S possible classes. Let Q_j denote a set of 2D frequency quanta such that*:

$$Q_j: \{q_{sr} \mid s=1, 2, \cdots, S, r = 1, 2, \cdots, L_j\}$$

where $q_{sr} = \sum_{v_{jk}=e_{r-1}}^{e_r} o_{sk}$, *and* o_{sk} *is the observed number of instances of the sample set having class label* c_s *and attribute value* v_{jk}.

DEFINITION 3: *Let a finite marginal probability scheme P be defined as the set of probability values* $\{p_1, p_2, \cdots, p_{Lj}\}$ *such that*:

$$p_i = \int_{e_{i-1}}^{e_i} f(A_j)\, dA = F(e_i) - F(e_{i-1})$$

where f and F are the probability density function and the cumulative density function of A_j in the range $[a, b]$, respectively, and e_i and e_{i-1} mark the lower and upper boundaries, respectively, of the sub-interval i.

DEFINITION 4: A joint class-attribute probability scheme P' is the set of joint probability values $\{p'_{si}\}$ so that:

$$p'_{si} = \int_{c_1}^{c_S} \int_{e_{i-1}}^{e_i} f'(C, A_j)\, dC dA = F(c_s, e_i) - F(c_s, e_{i-1})$$

where f and F are the joint probability density function and the joint cumulative density function of class variable C and attribute A_j in the range $[a, b]$, respectively.

In general, *discretization* is a process that transforms the range of the continuous attribute A_j into a discrete partition T_j consisting of L_j intervals. Associated with each T_j, there is a boundary set B_j and a quanta set Q_j. Given f and B, a finite marginal probability scheme P can be determined. From f and B, we can obtain the joint class-attribute probability scheme P.

B. The Discretization Criterion

In this section, a new discretization criterion based on the concept of Class-Attribute dependence is introduced. The new discretization method seeks to maximize the dependency relationship between the class variable and a continuous-valued attribute. Since IL problems are usually given a sample of observed outcomes of a continuous attribute, discretization can be considered as the process of reducing the number of states of an ordered discrete random variable by combining some of its states together. For any number of intervals L_j, and an intermediate resultant boundary set B_j, one can form a 2D quanta matrix. This quanta matrix representation is depicted in Table I, where each element q_{sr} denotes the total number of observed instances belonging to class c_s, and whose value of A_j falls within the boundary set $[e_{r-1}, e_r]$. The set of interval boundary pairs represents the *new* and *reduced* set of possible attribute values, where the original number of K possible continuous values has been reduced to L_j possible ordered discrete intervals. For convenience, we use $A_j \in e_r$ to denote the fact that the actual value v_{jk} of A_j is within the interval bounded by e_{r-1} and e_r.

Since the partitioned attribute is treated as an ordered discrete random variable, we can easily calculate the estimated joint probability of the event that an object belongs to c_s while its attribute value of A_j falls in between the boundary pair $[e_{r-1}, e_r]$:

TABLE I
A 2D DISCRETIZATION QUANTA MATRIX

		Boundary						Total
		$[e_0, e_1]$	$[e_1, e_2]$	•••	$[e_{r-1}, e_r]$	•••	$[e_{L_j-1}, e_{L_j}]$	
	c_1	q_{11}	q_{12}	•••	q_{1r}	•••	q_{1L_j}	q_{1+}
	•	•	•		•		•	•
Class	c_s	q_{s1}	q_{s2}	•••	q_{sr}	•••	q_{sL_j}	q_{s+}
	•	•	•		•		•	•
	c_S	q_{S1}	q_{S2}	•••	q_{Sr}	•••	q_{SL_j}	q_{S+}
Total		q_{+1}	q_{+2}	•••	q_{+r}	•••	q_{+L_j}	M'

$$p(C = c_s, A_j \in e_r) = p_{sr} = \frac{q_{sr}}{M'} \tag{1}$$

Similarly, one can calculate the estimated marginal probabilities of $C = c_s$ and $A_j \in e_r$ as follows:

$$p(C = c_s) = p_{s+} = \frac{q_{s+}}{M'} \tag{2}$$

$$p(A_j \in e_r) = p_{+r} = \frac{q_{+r}}{M'} \tag{3}$$

The *CA* (*Class-Attribute*) *mutual information* between the class variable and the attribute interval boundaries of A_j with its associated quanta set Q_j is calculated as follows:

$$I(C: A_j) = \sum_{C} \sum_{A_j} p_{sr} \log \frac{p_{sr}}{p_{s+} \cdot p_{+r}} \tag{4}$$

Here, p_{sr} represents the joint probability that the object belongs to class c_s while its attribute A_j has a value v_{jk}, where $e_{r-1} \le v_{jk} < e_r$, and p_{+r} denotes the marginal probability that $A_j \in e_r$. Therefore, our definition of $I(C : A_j)$ represents the mutual information calculated for some *discretized* state of a continuous-valued attribute using probabilities based on the interval quanta set Q_j resulting from the discretization process. This CA mutual information thus refers to the mutual information between the class labels and the discrete attribute intervals.

To maximize the classification utility of attribute A_j after discretization, we wish to maximize the class-attribute dependence relationship during the discretization process. The CA mutual information initially appears to be a good candidate for such a discretization criterion. It is bounded by zero, if C and A_j are completely independent, and the larger of $H_{\max}(A_j) = \log(L_j)$ and $H_{\max}(C) = \log(S)$, if there is perfect correlation between C and A_j. Unfortunately, the CA mutual information is very much affected by the number of intervals chosen to partition the original real-valued attribute. In fact, the expected CA mutual information is at maximum *before* any discretization, and it decreases as the number of intervals is reduced. As a result $I(C : A_j)$ is not suitable for selecting the optimal number of intervals.

Given that C and A_j are both considered as random variables, and the joint entropy between the class variable and the attribute variable $H(C, A_j)$, the CA mutual information $I(C : A_j)$ can be normalized as follows [26]:

$$R_{CA_j} = \frac{I(C : A_j)}{H(C, A_j)} \tag{5}$$

where R_{CA_j} is called the Class-Attribute (CA) *Interdependence Redundancy*. Clearly $R_{CA_j} \geq 0$ since $I(C : A_j) \geq 0$ and $H(C, A_j) \geq 0$. In fact, it is known that $0 \leq R_{CA_j} \leq 1$, and $R_{CA_j} = 1$ if C and A_j are totally dependent, and $R_{CA_j} = 0$ if they are totally independent.

Since R_{CAj} is normalized, it has the property that it is independent of the composition of the attribute and class variables. That is, unlike the absolute mutual information measure, the value of R_{CAj} is not dependent on the number of class labels or the number of unique attribute values in a particular classification problem. That means, while the process of discretization always reduces the absolute amount of mutual information between the class and the attribute, the interdependence between the class and attribute, as measured by the interdependence redundancy, does not necessarily decrease. Discretization for inductive learning can be viewed as a process of reducing the number of unique attribute values by combining some of them into optimal intervals. Intuitively, the concept of redundancy means that the unique number of attribute value representations can be sometimes reduced without destroying the interdependence relationship between the class and attribute variables, because the original continuous scheme is *redundant*.

The properties of the CA interdependence redundancy measure clearly make it an ideal candidate as a class-dependent discretization criterion. For the purpose of inductive learning, it is desirable for the discretized training events to minimize the loss of correlation between the class labels and the attribute values. We therefore propose a new class-dependent discretization method that uses the interdependence redundancy as an optimality criterion. For any partially discretized state of a continuous-valued attribute A_j, and its associated quanta set Q_j, the interdependence redundancy R_{CAj} of the class variable and the attribute variable can be calculated using (5).

Formally, let Ψ represent the set of all possible finite probability schemes that can be derived by all of the discretization processes and the resulting quanta matrices for a given class-attribute variable pair. The problem of maximizing class-attribute interdependence redundancy is to find a ψ_{max}, such that:

$$R_{CA_j}(\psi_{max}) \geq R_{CA_j}(\psi) \qquad \forall \psi \in \Psi$$

C. Determine the Optimal Number of Intervals

Besides the discretization criterion, the selection of the number of intervals to partition a continuous-valued attribute into is the second most important decision in the discretization process. Since the theory of maximum mutual information dictates that the absolute mutual information is greatest when the number of intervals is the largest possible, a rule of thumb is to select the maximum allowable number of intervals without violating the statistical assumptions for estimating the second order probabilities needed for class-dependent discretization. So if the number of classes is S, the rule states that the number of intervals L_j for attribute A_j should not be greater than $M'/(N \times S)$, where M' is again the total number of training samples. The parameter N is usually suggested to be 3 for liberal estimation according to [25]. This rule of thumb, however, is not ideal, since the goal of discretization is to reduce the number of unique values so inductive learning algorithms can be applied on continuous-valued data. Smaller numbers of intervals are always preferred in inductive learning applications because a large number of intervals means large number of possible attribute values, and that contributes to slow and inefficient learning process [4]. Therefore, in addition to the maximization of interdependence between class labels and attribute values, an ideal discretization method should have a secondary goal to minimize the number of intervals without significant loss of class-attribute mutual dependence.

The CA mutual information measure can be used as a statistical test for interdependence between the class variable C and the attribute variable A_j. For any intermediate discretization state, we can measure the statistical significance of the class-attribute interdependence as follows:

$$I(C : A_j) > \frac{1}{2M'} \chi^2_{(S-1)(L_j-1)} \tag{6}$$

By normalizing this test on both sides with respect to $H(C, A_j)$, we get:

$$R_{CA_j} \geq \frac{\chi^2_{(S-1)(L_j-1)}}{2M' \cdot H(C, A_j)} \tag{7}$$

If (7) is true, we say that C and the discretized attribute A_j are statistically interdependent [25].

This statistical test allows us to eliminate any "redundant" intervals so we can minimize the number of intervals. Given an intermediate partition result with a boundary set B_j and its associated quanta matrix as depicted in Table I, all neighboring pairs of intervals are analyzed one pair at a time. We can then calculate the partial CA mutual information and the partial joint entropy according to the distributed frequencies and subtotal instances of each pair of neighboring columns. The statistical test of (7) can then be used to determine if the frequency distribution among the two neighboring intervals and the class labels are significantly interdependent. If the test is significant at some confidence interval, the analysis for the next pair of neighboring intervals is performed. If the test fails, it concludes that the two intervals will not likely contribute to IL classification and can therefore be combined.

D. Heuristic Implementation of Class-Dependent Discretization

The problem of class-dependent discretization to maximize interdependence redundancy is highly combinatorical. Global maximization of the interdependence redundancy measure between the class variable and a discretized attribute is impractically expensive in terms of computational requirements. In this section, we describe a heuristic-based "local optimization" implementation that is both effective and efficient. This implementation consists of three main processes: interval initialization, interval improvement, and interval reduction.

The first step of the discretization process requires the sorting of the *unique* values of a real-valued attribute observed from a training set in increasing order. An initial default number of intervals is selected, either as a user input, or as calculated based on the maximum allowed for reliable second order probability estimation. The goal is to partition the initial intervals so that the sample is distributed as evenly as possible to minimize information loss (the maximum entropy criterion can be used). Once the initial interval boundaries are set, a quanta matrix similar to that in Table I is constructed from the boundary set and the training samples. The initial interdependence redundancy measure between the class labels and the initial attribute intervals is then calculated and recorded.

The boundary improvement procedure attempts to improve upon the initial interdependence redundancy measure by altering the initial quanta matrix through local perturbation of the interval boundaries. Interval adjustments can be made to either the lower boundary or the upper boundary of a given interval. Boundary adjustments are made in increments of the next ordered observed unique attribute values. For any given interval, and to adjust a lower boundary down to include the next lower observed attribute value, the algorithm also adjusts downward the upper boundary of the interval just before it to exclude the same attribute value.

To ensure a good estimation of global optimal interdependence, the algorithm perturbs each boundary up and then down a boundary value in turn starting from the first ordered interval. The procedure records the interdependence change as measured by the interdependence redundancy criterion. After all possible adjustments and their associated interdependency measure have been tried and recorded in the current pass, it determines which adjustment causes the maximum gain of interdependence and modifies the boundary set and the associated quanta matrix according to that adjustment. The entire process is repeated until no improvement of the interdependence criterion is found.

The third major part of the proposed discretization algorithm is the combination of statistically insignificant intervals. Due to redundancy, frequency quanta of some of the adjacent intervals may be very similarly distributed in respect to the class labels. In these cases, the similar intervals may be combined without significant loss of degree of interdependence. The algorithm extracts pairs of adjacent intervals and performs a statistical test of interdependence described by (7). If two neighboring intervals do not contribute to class-attribute dependency, they are combined into a single interval. The interval reduction algorithm is performed for all pairs of adjacent intervals until all of them pass the test of statistical interdependence.

Even though the proposed algorithm is heuristic in nature, it is believed to be a good compromise between providing good results and requiring acceptable computational resources. The heuristics used are simple, reasonable and effective. The algorithm has been implemented in a computer-based system for effective class-dependent discretization of continuous attributes in inductive learning applications.

E. An Example

To demonstrate how the proposed discretization algorithm works in a real-world situation, we use a set of mixed-mode data from the micro computer fault diagnosis domain [6]. A set of 110 problem situations have been preclassified into one of six common PC power-up faults. Each problem record is characterized by 13 attributes—some of them continuous—to represent the symptoms. The objective is to identify which subsystem (functional group of hardware components) is at fault, causing the appearance of the problem symptoms.

One of the continuous-valued attributes is the measured voltage between pin number 2 and pin number 4 from the power supply to the major peripherals. This attribute provides important information since certain ranges of voltage measures can indicate power-related problems as opposed to other types of problems. In [6], a knowledge-based approach was used to discretize the voltage attribute since the engineers knew the proper voltage range at pin 2 and pin 4 on a typical PC compatible computer is between 4.8 and 5.2 volts. A reasonable "common sense" partition scheme would be to divide the attribute into three ranges, too low (<4.8), normal (4.8 - 5.2), and too high (> 5.2). In many application domains, knowledge about the attributes is not always easily available, and our proposed method is capable of utilizing the inherent class versus attribute range information to define the set of partition boundaries which emphasizes the relations between the class assignment and the attribute ranges.

Suppose we wish to acquire diagnostic rules for classifying PC hardware faults using an IL algorithm. In this simple example, 109 samples are used for training, and a single sample is selected for testing. The test cases are not generally available prior to classification training, and they will not be considered in the discretization analysis. Instead, test cases will be discretized prior to the class prediction procedure according to the discretization intervals derived from the training samples. There are only 85 training sample values available for this attribute due to missing values. Since there are six possible classes, it is determined that the maximum default number of intervals is $85 \div (3 \times 6)$ which is equal to 4. Therefore, the proposed algorithm begins with the default number of intervals 4, and proceeds to partition the attribute into 4 as evenly distributed intervals as possible in order to maximize $H(A_j)$. This frequency matrix associated with the initial partition is presented in Table II.

The initial mutual information between the class labels and the attribute intervals is 0.161820, and the calculated interdependence redundancy is 0.055825. After a few local boundary perturbations, the class-attribute mutual information is increased to 0.239157, while the related interdependence redundancy measure is improved to 0.095790. The corresponding boundary set and the frequency distribution matrix are shown in Table III. Note that the marginal frequency distribution for the classes does not change.

Upon further examination of the frequency matrix in Table III, it is evident that the frequencies of the observed attribute values in intervals 2 and 3 (4.8 - 4.8 and 4.9 - 5.3) are

TABLE II
INITIAL INTERVALS AND ASSOCIATED FREQUENCY MATRIX

Class	Attribute Value Intervals				Total
	0.0-4.7	4.8-4.9	5.0-5.0	5.1-7.1	
1	7	2	1	5	15
2	2	6	6	5	19
3	0	7	4	6	17
4	0	5	3	7	15
5	0	4	3	4	11
6	0	3	2	3	8
Total	9	27	19	30	85

TABLE III
IMPROVED INTERVALS AND ASSOCIATED FREQUENCY MATRIX

Class	Attribute Value Intervals				Total
	0.0-4.7	4.8-4.8	4.9-5.3	5.4-7.1	
1	7	1	4	3	15
2	2	2	12	3	19
3	0	3	14	0	17
4	0	2	13	0	15
5	0	3	8	0	11
6	0	2	6	0	8
Total	9	13	57	6	85

TABLE IV
FINAL OPTIMAL INTERVALS AND ASSOCIATED FREQUENCY MATRIX.

Class	Attribute Value Intervals			Total
	0.0-4.7	4.8-5.3	5.4-7.1	
1	7	5	3	15
2	2	14	3	19
3	0	17	0	17
4	0	15	0	15
5	0	11	0	11
6	0	8	0	8
Total	9	70	6	85

fairly similarly distributed among the class labels. In fact, by using the statistical test of interdependence (7) on these two adjacent intervals, the algorithm discovers that they may be combined without reducing the degree of class-attribute interdependence significantly. Therefore, the original default number of intervals is now reduced to three and after an additional local boundary perturbation pass, the algorithm produces the final optimized boundary set and the corresponding frequency distribution matrix in Table IV. Recall that human experts knew that voltages between 4.8 volts and 5.2 volts are considered acceptable, and the proposed algorithm produced "normal" range of 4.8 to 5.3 volts is indeed very similar.

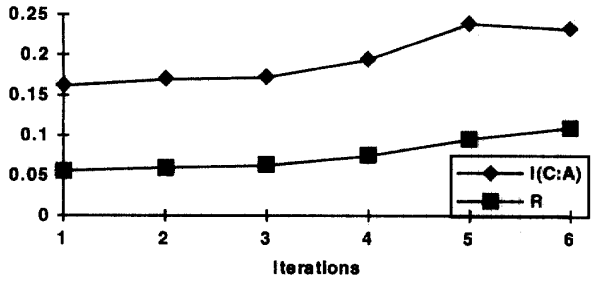

Fig. 1. Mutual information and interdependence redundancy by iterations.

It is interesting that the final partition resulted in an optimal interdependence redundancy measure of 0.110098. As expected, the absolute mutual information actually decreased slightly to 0.232157, as a result of reducing the number of intervals from 4 to 3. This result illustrates the importance of using normalized interdependence redundancy rather than the absolute mutual information as a discretization criterion. The values of the absolute mutual information and the interdependence redundancy measure for each of the partition iterations are plotted in Fig. 1, where the legends $I(C : A)$ and R represents the absolute CA mutual information and the CA interdependence redundancy, respectively.

F. Evaluation of the Automatic Number of Intervals Selection Method

The proposed discretization approach has the ability to efficiently maximize the CA interdependence redundancy while at the same time minimize the number of discrete intervals required. As the above discretization example indicates, the proposed approach using statistical interdependence tests to reduce the maximum default number of intervals is both efficient and effective. To further demonstrate this point, we present the following example. A less efficient way to determine the optimal number of intervals to use for the discretization of a given continuous attribute is to actually try all of the possible numbers greater than 2 but less than the maximum allowed under the second order probability estimation rule described earlier. In the example of the pin 2 - 4 voltage attribute, the maximum allowed by the size of the sample set and the number of possible classes is 4. So one of 2, 3 or 4 must be the best possible number of intervals according to the proposed discretization criterion. Since the method automatically settled on 3 discrete intervals, it is reasonable to predict that 3 is the best number of intervals.

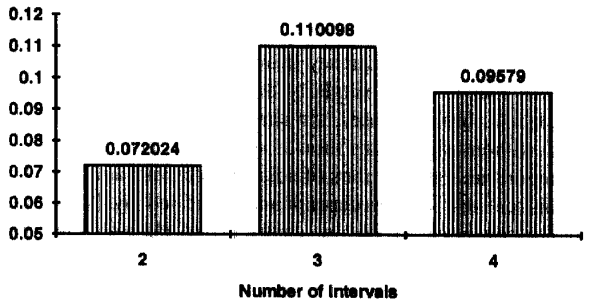

Fig. 2. The effect of the number of intervals on dependency.

To test this prediction, the three possible numbers of intervals were used, one at a time, as fixed user-defined number of intervals with the interval reduction function of the algorithm disabled. The final maximum interdependence redundancy measures for the three discretization runs are plotted in Fig. 2. In this case, it is evident from the graph that discretization us-

ing the interval number 3 produced the best results. In general, however, the optimal number of interval selected by the algorithm may not always provide the largest possible IR measure, but the number selected will always be the *smallest possible* without losing *significant* mutual dependency information in a statistical sense. This strategy is important because it is also desirable to minimize the number of discrete intervals to maximize inductive learning systems efficiency.

III. Empirical Analysis of Discretization Methods for Inductive Learning

The proposed class-dependent discretization method has been implemented in a system called class-attribute dependent discretizer (CADD). To evaluate the performance of CADD, and to test the effects of different discretizers on inductive learning accuracy and speed, we conducted several experiments with different inductive learners in continuous domains. The goal of this comparative study is to determine if CADD can significantly improve the performance of common inductive learning methods, particularly APACS, relative to other general discretization methods: maximum entropy (ME), equal information gain (EIG), and equal interval width (EIW).

Since the ME, EIG and EIW discretization methods require user-provided number of intervals, the selection rule of thumb discussed similar to the one recommended in [25] was used to determine the default number of intervals used in our experiments. For our experiments, a maximum number of intervals ranging from 8 to 12 were imposed depending on the size of the data and the amount of the computer memory required. The current implementation of CADD uses a *initial* number of intervals that maximizes the entropy and minimizes information loss. Several versions of AQ and ID3 algorithms, in addition to our own APACS system, are used as the classification engine in our experiments. In each of the experiments, we compare the classification results of five different versions of existing inductive learning systems on a set of discretized real-life data by CADD and three other discretization methods. The IL systems used in the experiments are briefly described below.

ID3—The ID3 algorithm [19] is the best known IL algorithm in the machine learning community. It constructs a decision tree using a top-down divide and conquer approach. The goal of the ID3 algorithm is to derive an optimal decision tree from a set of preclassified sample set. Each interior node of the tree denotes a single attribute, and arcs leaving that node represents the possible attribute values. Each leaf node is a conjunction of attribute values. Associated with a leaf node, there is a class label which represents the class assignment of all training events belonging to that class and satisfying that conjunctive attribute test. ID3 is designed to handle training data with discrete and symbolic attribute values. In order to deal with continuous-valued attributes, ID3 must treat them as discrete attribute with many possible values. Since the size of a tree is directly dependent on the number of attribute value tests represented by the arcs, continuous-valued attributes can significantly increase the size of the decision trees.

ID3 with pre-pruning—Overfitting due to noise can cause the decision tree to be unmanageably large. The strategy to solve this problem is to find a way to reduce the size of the tree without significantly affecting the classification accuracy of the resulting tree. Almost all of the tree-pruning techniques can be classified as pre-pruning or post-pruning depending on where the pruning process occurs during the tree-construction procedure. Quinlan's pre-pruning method [20], which is used in our experiments, halts the tree-growing process when it determines that no attribute is going to significantly increase the information gain in the classification task. It uses the chi-square statistical test as a pruning criterion.

ID3 with post-pruning—The pruning of decision trees "on the fly" sometimes misses important information that cannot be detected locally [1]. A solution to this problem is to introduce pruning of already completed trees. Such post-pruning strategies are usually based on criteria that compare a tree's complexity to its observed classification accuracy [18]. A number of post-pruning methods are describe in [20]. In our test, the ID3 with post-pruning implementation constructs a set of rules equivalent to a decision tree and then simplifies the conditional side of the rule by comparing the predicted class with the actual class and test for statistical dependence using Fisher's Exact Test [20].

AQ—Michalski's AQ is another well-known IL algorithm [15]. Training examples are given in the form of events, which are described in terms of a feature set. Training events from a given class are considered positive examples of that class, while all other events are considered as negative examples. A cover, which is a disjunction of features, is generated for each class. An ideal cover of a class describes all the positive examples while excluding all the negative ones. The goal of AQ is to induce a set of decision rules, one for each class, in the form of if *<cover>* then predict *<class>*, where *class* is the most common class described by cover. AQ uses the MAX-STAR parameter to specify how many disjunctive feature sets are retained at each search step in order to limit the search. To determine which complexes are kept, a user defined criterion function is used. Although there are many possible alternative measures, the most common one is to "maximize the positive events covered", and "minimize the negative events covered" [14], [15]. To handle continuous values, AQ must treat each possible value as a unique discrete value. Because AQ attempts to find a cover that satisfies a single attribute value while excluding all others, a large number of possible attribute values in the form of a real-valued attribute can create many specific and long disjuncts rather than a few simple and general ones. In addition, continuous attribute values add significant overhead to AQ's search process. The basic AQ algorithm is implemented as one of the IL tools for our comparative study.

AQ15—The original AQ algorithm does not handle uncertainty very well. The best known AQ-based system for learning from noisy data is AQ15. AQ15 utilizes a built-in rule truncation and flexible matching procedure called TRUNC during the rule interpretation process [16]. In the presence of

uncertainty, some training events may be misclassified, and some of the rules generated using such noisy data may be incorrect. AQ15, as implemented here, uses a rule truncation technique to remove (truncate) portions of the rules that may be due to noise. With the rules truncated, a flexible match routine, as opposed to a strict match, is used.

M-APACS—The original APACS used the maximum entropy discretization method with some success [5]. We soon discovered that for many continuous and mixed-mode application domains, the performance of the maximum entropy method was inadequate for supervised learning tasks. The M-APACS implementation is an *integrated* APACS/CADD system (*M* stands for mixed-mode). This system can automatically discretize any real-valued attributes in a continuous or mixed-mode environment prior to performing inductive learning using its APACS engine.

The standard APACS algorithm consists of three steps and has been described in detail elsewhere [5], [6], [7]. The first step is to detect the underlying regularities in the training data set. The major goal of this step is to determine which attribute values contribute to the given class membership. The significant attribute values of a class are called the *relevant features* of the class. The irrelevant attribute values or features of the training data, according to a standard statistical test based on *adjusted residual* [7], are discarded from further analysis to minimize the negative effects due to overfitting.

Once the relevant features of a class are known, the *weight of evidence* of an object belonging to a particular class given a relevant feature can be calculated. Decision rules, in the form of IF <condition> THEN <conclusion> WITH WEIGHT OF EVIDENCE *W* are generated so they may be used later to classify unknown objects. The condition side specifies the attribute and attribute values an object must possess in order to be classified into the object class indicated on the conclusion side. The weight of evidence *W* is a measure of uncertainty in a noisy environment based on mutual information [7]:

$$W(Class = c_p / Class \neq c_p | Attr_j = v_{jk}) = \log \frac{\Pr(Attr_j = v_{jk} | Class = c_p)}{\Pr(Attr_j = v_{jk} | Class \neq c_p)} \quad (8)$$

where c_p is any particular class and v_{jk} is the kth possible attribute value for the jth attribute in the attribute set.

The final step of the APACS method is the determination of class membership of an unknown object using the rules generated from the training set. The rule base is searched for all rules whose condition side matches any of the relevant features of the given object. If a match is found, it is said that there is positive or negative evidence supporting the classification of the new event into the class specified by the conclusion side of the rule, depending on the sign of the rule's weight of evidence [7]. In most cases, there are multiple rules that match the event's attributes. Let val_1, ..., val_j, ..., val_n be the n attribute values associated with the event e to be classified, then the total weight of evidence by all n attributes of e in favour of it being assigned to c_p as opposed to being assigned to any other class is simply the sum of the weight of evidence provided by each relevant attribute value of e [7]:

$$W(C_e = c_p / C_e \neq c_p | val_{[1]}, \ldots, val_{[m]}) = \sum_{j=1}^{m} W(C_e = c_p / C_e \neq c_p | val_{[j]}) \quad (9)$$

where m attributes were found to match one or more classification rules, and $m \leq n$.

Using the sum of the weight of evidence as a measure, the APACS algorithm considers the class c_p with the greatest total weight of evidence is the class to be assigned. That is:

$$W(C_e = c_p / C_e \neq c_p | val_{[1]}, \ldots, val_{[m]}) > W(C_e = c_h / C_e \neq c_h | val_{[1]}, \ldots, val_{[m]}) \quad (10)$$

where $h = 1,2,\ldots,P'$, and $h \neq p$. $P' \leq P$ is the number of classes matched by the attribute values according to the rules [7].

All user-selected parameters such as MAXSTAR for AQ algorithms were selected to maximize classification accuracy while providing reasonable learning efficiency. Statistical test needed in APACS/CADD and ID3 with pruning implementations used a mid-range confidence level of 95%. For each experiment, a set of preclassified data was randomly divided into two sets, one for training and the other for testing. The training and test data were discretized according to discretization intervals determined from the training data, before they were used as input to the five inductive learning systems in separate test runs. To compare the effects of the four discretization methods on classifier performance, the discretization portion of each experiment was performed using each of CADD, ME, EIG and EIW methods, one at a time. For M-APACS, the discretization process using CADD was done automatically before learning in an integrated step. For M-APACS using other discretizers and for all ID3 and AQ runs, discretization of real-valued attributes was conducted as a separate data preprocessing step. Each learning experiment was repeated 10 times using 10 *different* sets of randomly selected training and test data. The average results in terms of classification accuracy and learning time of the 10 experiments for the five classifiers and four discretizers are reported in this paper.

The experiments were conducted on continuous data sets from four different domains. These domains are described below. Data sets 2-4 were obtained from [17].

1) **Chemical and Overt Diabetes**: This set of clinical data was previously used for the study of the relationship between chemical subclinical and overt nonketotic diabetes in adult non-obese subjects (Table V) [22]. The data for a total of 145 subjects was available. Each subject is described by five continuous-valued variables. The subjects have been classified into three classes, normal (76 cases), chemical diabetic (36 cases), and overt diabetic (33 cases), according to standard medical criteria. A set of 30 subjects for testing was randomly selected, and the remaining 80% of the available data was used for training.
2) **Glasses:** This is a set of 214 instances of data describing six types of different types of glasses for the purpose of forensic science (Table VI). Each instance is characterized

TABLE V
STATISTICAL PROPERTIES OF DIABETES DATA

Attribute	*Min*	*Max*	*Mean*	*SD*	*Class Correlation*
Relative weight	0.71	1.2	0.98	0.13	-0.2146
Fasting plasma glucose	70	353	121.99	63.93	-0.7306
Glucose resistance	269	1568	543.61	316.95	-0.8367
Insulin resistance	10	748	186.12	120.94	0.1138
Steady state plasma glucose	29	480	184.21	106.03	-0.7835

by 9 attributes, all of them continuous. The attributes represent the unit measurement in percentage of the different type of oxides. Class distribution is as follows: 70 building windows, 17 float processed vehicle windows, 76 non-float processed vehicle windows, 13 containers, 9 tableware, and 29 headlamps. For the 10 randomly selected training and test sets, 65 cases were reserved for testing and the remaining 70% were used for training.

TABLE VI
STATISTICAL PROPERTIES OF GLASSES DATA

Attribute	*Min*	*Max*	*Mean*	*SD*	*Class Correlation*
Refractive Index	1.5112	1.5339	1.5184	0.0030	-0.1642
Na	10.73	17.38	13.4079	0.8166	0.5030
Mg	0	4.49	2.6845	1.4424	-0.7447
Al	0.29	3.5	1.4449	0.4993	0.5988
Si	69.81	75.41	72.6509	0.7745	0.1515
K	0	6.21	0.4971	0.6522	-0.0100
Ca	5.43	16.19	8.9570	1.4232	0.0007
Ba	0	3.15	0.1750	0.4972	0.5751
Fe	0	0.51	0.0570	0.0974	-0.1879

3) **Liver disorders:** This set of medical data consists of 345 adult male patients with two different types of liver disorder (145 type A and 200 type B). Each patient has been pre-classified into one of two possible liver disease types, and was described in terms of six continuous-valued attributes (Table VII). The first five variables are all blood tests which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption, and the sixth attribute indicates the number of half-pint equivalents of alcoholic beverages consumed per day. For each of the 10 runs, the test set consisted of 70 randomly selected patient records, while the remaining data was used for training.

TABLE VII
STATISTICAL PROPERTIES OF LIVER DISORDERS DATA

Attribute	*Min*	*Max*	*Mean*	*SD*	*Class Correlation*
Mean copuscular volume	65	103	90.16	4.45	-0.0911
Alkaline phosphotase	23	138	69.87	18.35	-0.0981
Alamine aminotransferase	4	155	30.41	19.51	-0.0350
Aspartate aminotransferase	5	82	24.64	10.06	0.1574
Gamma-glutamyl transpeptidase	5	297	38.28	39.25	0.1464
Number of drinks	0	20	3.46	3.34	-0.0221

4) **Iris:** This is a well-known set of data in the pattern recognition field (e.g., [10]) containing three classes of iris plants with 50 instances belonging to each class (Table VIII). One of the iris types was linearly separable from the other two, although the two were not linearly separable from each other. Each record is described by four numeric attributes. Out of the 150 instances available, a set of 70 instances (50%) was selected randomly for testing and the remaining half were used for training. The results for experiments using the first data set are summarized in Table IX. In terms of classification accuracy, ID3 and AQ based systems using data discretized by CADD had a consistent overall improvement of about 10%. The improvement in M-APACS over the other discretizers was less pronounced. By automatically reducing the number of intervals without sacrificing useful classification information, the proposed discretizer also managed to shave the average learning time of the IL systems significantly. Careful review of the test results for each inductive learning system confirmed the initial observation. All six inductive learning methods produced the best classification results as well as the shortest processing time using CADD as the discretizer as opposed to using the other three methods. The most significant computational efficiency improvement as a result of the proposed discretization method occurred in AQ systems with their relatively expensive search-based problem solving strategy. By retaining only the useful intervals, CADD allowed AQ to run faster while producing better accuracy. ID3 with post-pruning is also known for being slow normally [18]. Its learning time was also effectively reduced by using CADD to discretize the original continuous data. For already fast methods like M-APACS and ID3 without pruning, the performance improvement was less visible. Nevertheless, using CADD resulted in the shortest processing time in every case.

TABLE VIII
STATISTICAL PROPERTIES OF IRIS DATA

	Min	*Max*	*Mean*	*SD*	*Class Correlation*
Sepal Length	4.3	7.9	5.84	0.83	0.7826
Sepal Width	2.0	4.4	3.05	0.43	-0.4194
Petal Length	1.0	6.9	3.76	1.76	0.9490
Petal Width	0.1	2.5	1.20	0.76	0.9565

TABLE IX
COMPARATIVE TEST RESULTS OF DIABETES DATA

IL Systems	Equal Width		Equal Information		Maximum Entropy		CADD	
	Correct	Time (s)	Correct	Time (s)	Correct	Time (s)	Correct	Time (s)
M-APACS	89.3%	0.11	89.7%	0.11	90.3%	0.11	92.3%	0.10
AQ	77.0%	1.71	75.7%	2.26	78.7%	2.06	87.7%	0.61
AQ15	83.3%	2.00	82.7%	2.08	80.7%	2.14	90.0%	0.64
ID3	77.0%	0.07	72.0%	0.07	74.0%	0.06	90.0%	0.05
ID3 pre-pruning	71.0%	0.13	72.3%	0.13	76.7%	0.08	90.7%	0.06
ID3 post-pruning	78.7%	5.65	74.0%	1.53	79.3%	3.64	91.3%	0.47

Among the different inductive learning implementations, M-APACS produced the best classification results in every case. Algorithms with better uncertainty tolerance such as M-APACS, AQ15, and ID3 with pruning produced better classification accuracy in general. Overall, ID3 algorithms also performed better than the AQ versions for this medical domain.

Table X presents the results from the second experiment. Again, the test results confirmed that the proposed discretization method implemented in CADD consistently contributed to better classification performance of all of the five IL systems. In general, the degree of classification accuracy improvement by using CADD as opposed to the other methods was fairly consistent among the different inductive learners. ID3 based methods appeared to take the most advantage of class-dependent discretization because CADD's information theoretic approach actually helped the decision-tree based systems to deal with this noisy data.

The slower IL techniques again benefited more in terms of reduced learning time. Due to the complexity and the size of

TABLE X
COMPARATIVE TEST RESULTS OF THE GLASSES DATA

IL Systems	Equal Width		Equal Information		Maximum Entropy		CADD	
	Correct	Time (s)	Correct	Time (s)	Correct	Time (s)	Correct	Time (s)
M-APACS	56.9%	0.16	55.8%	0.17	55.6%	0.17	65.7%	0.16
AQ	53.7%	19.17	60.5%	14.06	58.5%	16.88	63.8%	8.80
AQ15	55.1%	17.05	63.8%	15.88	64.2%	15.39	68.6%	14.50
ID3	57.2%	0.44	59.8%	0.58	61.4%	0.46	64.9%	0.28
ID3 pre-pruning	50.0%	0.11	58.5%	0.11	57.5%	0.11	65.9%	0.11
ID3 post-pruning	54.9%	27.39	58.6%	42.33	55.6%	39.90	63.4%	22.22

the data, the overall learning time was relatively slower compared to the other sets of data tested. In any case, the improvement due to CADD's ability to minimize the number of intervals needed for effective classification was still quite evident in AQ and ID3 with post-pruning. CADD actually helped to reduce overfitting and eliminated some of the pruning otherwise needed. ID3 with pre-pruning was the fastest method overall in this experiment most likely because its chi-square test for terminating the branching process was quite effective due to the large size of this data set.

In this experiment, AQ15 produced the best classification accuracy followed closely by M-APACS and ID3 with pre-pruning. Given the significant CPU time penalty of AQ15, ID3 and M-APACS may still be considered as more suitable methods in this application. This set of data also showed that CADD can greatly improve the performance of our APACS system in continuous domains.

The results in Table XI for the third test domain show a similar trend. In every case, the classifiers' accuracy increased as a result of using CADD to discretize the data set. For this set of data, the degree of classification accuracy improvement of all of the tested inductive learners appeared quite consistent at around 5%. The results for training time also confirmed the earlier findings. There was a clear trend indicating that IL systems using CADD generally have a shorter learning time. Again, M-APACS performed well in this experiment both in terms of classification accuracy and efficiency.

TABLE XI
COMPARATIVE TEST RESULTS OF THE LIVER DISEASE DATA

IL Systems	Equal Width		Equal Information		Maximum Entropy		CADD	
	Correct	Time (s)	Correct	Time (s)	Correct	Time (s)	Correct	Time (s)
M-APACS	59.6%	0.17	58.1%	0.16	60.6%	0.17	65.7%	0.15
AQ	58.9%	18.29	59.3%	19.88	59.6%	18.78	62.9%	11.01
AQ15	58.4%	23.2	60.3%	15.4	59.1%	20.1	65.6%	7.8
ID3	54.7%	0.77	55.1%	0.72	58.0%	0.99	62.6%	0.6
ID3 pre-pruning	59.0%	0.26	55.4%	0.24	56.0%	0.23	64.0%	0.21
ID3 post-pruning	61.7%	59.2	58.3%	51.7	59.9%	47.62	63.3%	30.52

TABLE XII. COMPARATIVE TEST RESULTS OF THE IRIS PLANTS DATA

IL Systems	Equal Width		Equal Information		Maximum Entropy		CADD	
	Correct	Time (s)	Correct	Time (s)	Correct	Time (s)	Correct	Time (s)
M-APACS	94.8%	0.17	96.1%	0.19	94.1%	0.21	96.3%	0.16
AQ	69.3%	2.10	65.3%	2.44	62.5%	2.35	90.8%	1.10
AQ15	78.9%	1.96	78.4%	2.46	85.1%	2.30	91.2%	1.12
ID3	91.5%	0.15	91.1%	0.11	90.8%	0.11	95.1%	0.05
ID3 pre-pruning	82.7%	0.05	82.8%	0.06	84.5%	0.06	95.2%	0.05
ID3 post-pruning	92.0%	0.39	83.9%	0.39	88.3%	0.48	94.7%	0.28

The fourth set of results in Table XII showed that the improvements in classification accuracy by using CADD as the discretizer can be very dramatic for some inductive learning systems in certain domains. The classification accuracy of AQ improved to over 90% from consistently under 70% as a result of using the proposed class-dependent discretization method. Impressive improvements were also recorded for AQ15, and ID3 with pre-pruning. Although CADD clearly had a positive effect on the classification results of ID3 without pruning, this IL system performed well even with the class-ignorant discretizers. Due to the small size of the training data in this domain, the learning speed improvement from the use of our discretizer was less apparent. In any case, inductive learning using CADD discretized Iris data still resulted in at least a 35% reduction overall in learning time. By simply using a superior discretization method in an inductive learning application involving continuous attributes, significantly better classification accuracy as well as shorter training time can be achieved no matter which inductive learner is used. The proposed class-dependent discretization approach is clearly highly suitable for any supervised learning tasks in continuous or mixed mode domains. Both M-APACS and ID3 performed well overall in this experiment with the aid of the proposed class-dependent discretization algorithm.

In summary, our integrated inductive learning method for mixed-mode data, M-APACS, performed well in all of the experiments using its new class-dependent discretization module. In terms of classification efficiency, M-APACS and ID3-based systems generally were several times faster than AQ-based systems.

IV. CONCLUSIONS AND DISCUSSIONS

In this paper, we are concerned with the handling of continuous and mix-mode attributes in inductive learning applications. Particularly, we are interested in general discretization algorithms that can be used to enhance the performance of existing symbolic learning algorithms. We found that current discretization methods are either limited to a particular learning algorithm or they tend to ignore the important associative information between the continuous attributes and class assignment. We proposed a general class-dependent discretization method based on the concept of maximum class-attribute interdependence redundancy. The proposed discretization algorithm (CADD) takes into consideration the class assignment information in the training data of an inductive learning application, and seeks to maximize the mutual dependence of the class labels and the attribute intervals. As part of an integrated inductive learning system called M-APACS, or as a stand-alone general discretization system, CADD has been tested with the major inductive learning systems including ID3 and AQ. The test results showed our method is universally applicable and effective for inductive learning applications involving continuous attributes. Because of its ability to automatically select the minimum number of intervals without significantly reducing useful mutual information, CADD can speed up the learning time of most inductive learners with little classification accuracy loss. The information theoretic discretization criterion and statistical interdependence tests in CADD contribute to its ability to handle noise and uncertainties in the training data.

A possible improvement is the use of a more formal method, perhaps a clustering algorithm like K-means, to select the *initial* number of intervals and interval boundaries. Once the initial partitions are selected, the class-attribute interdependence redundancy heuristic can be used to refine the inter-

val boundaries and to minimize the number of intervals. In any case, it is clear that a superior discretizer such as CADD is the easiest and quite effective way to improve classifier performance in continuous and mixed-mode domains, using *existing* inductive learning algorithms.

REFERENCES

[1] I. Bratko and I. Kononenko, "Learning diagnostic rules from incomplete and noisy data," *Interactions in Artificial Intelligence and Statistical Methods*, B. Phelps, ed., Hants: Technical Press, 1987.

[2] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Belmont, Calif: Wadsworth, 1984.

[3] T. Caelli and A. Pennington, "An improved rule generation method for evidence-based classification systems," *Pattern Recognition*, vol. 26, no. 5, pp. 733-740, 1993.

[4] J. Catlett, "On changing continuous attributes into ordered discrete attributes," *Proc. European Working Session on Learning*, pp. 164-178, 1991.

[5] K.C.C. Chan, J.Y. Ching, and A.K.C. Wong, "A probabilistic inductive learning approach to the acquisition of knowledge in medical expert systems," *Proc. Fifth IEEE Computer-Based Medical Systems Symp.*, Durham, N.C., 1992.

[6] K.C.C. Chan, J.Y. Ching, and A.K.C. Wong, "Learning system fault diagnostic rules: a probabilistic inference approach," *Proc. Conference on Artificial Intelligence Applications in Engineering 92*, Waterloo, Canada, 1992.

[7] K.C.C. Chan and A.K.C. Wong, "APACS: a system for automated pattern analysis and classification," *Computational Intelligence*, vol. 6, 1990.

[8] K.C.C. Chan and A.K.C. Wong, "A statistical technique for extracting classificatory knowledge from databases," *Knowledge Discovery in Databases*, pp. 107-123, 1991.

[9] J.Y. Ching, "Class-dependent discretization of continuous attributes for inductive learning," MASc Thesis, University of Waterloo, Canada, 1992.

[10] P. Clark and T. Niblett, "Induction in noisy domains," *Progress in Machine Learning: Proc. of EWSL 87*, I. Bratko and N. Larvac, eds., Bled, Yugoslavia, 1987.

[11] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.

[12] U.M. Fayyad and K.B. Irani, "On the handling of continuous-valued attributes in decision tree generation," *Machine Learning*, vol. 8, pp. 87-102, 1992.

[13] J.A. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.

[14] R.S. Michalski, "Pattern recognition as rule-guided inductive inference," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 4, pp. 349-361, 1980.

[15] R.S. Michalski "A theory and methodology of inductive learning," *Machine Learning: An Artificial Intelligence Approach*. vol. 1, R. Michalski, J. Carbonell, and T. Michell, eds., Los Altos, Calif., 1983.

[16] R.S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, "The AQ15 inductive learning system: an overview and experiments," *UIUCDCS-R-86-1260*, Computer Science Department, University of Illinois at Urbana-Champaign, 1986.

[17] P.M. Murphy and D.W. Aha, UCI Repository of machine learning databases [Machine-readable data repository]. Irvine, Calif: University of California, Department of Information and Computer Science, 1991.

[18] T. Niblett, "Constructing decision trees in noisy domains," *Progress in Machine Learning: Proc. of EWSL 87*, I. Bratko and N. Larvac, eds., Bled, Yugoslavia, 1987.

[19] J.R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81-106, 1986.

[20] J.R. Quinlan, "Simplifying decision trees," *Int'l J. Man-Machine Studies*, vol. 27, pp. 221-234, 1987.

[21] J.R. Quinlan, P.J. Compton, K.A. Horn, and L. Lazarus, "Inductive knowledge acquisition: a case study," *Applications of Expert Systems*, J.R. Quinlan, ed., pp. 157-173, Sydney, Australia: Addition-Wesley, 1987.

[22] G.M. Reaven and R.G. Miller, "An attempt to define the nature of chemical diabetes using a multidimensional analysis," *Diabetologia* vol. 16, pp. 17-24, 1979.

[23] D.W. Stashuk and R.K. Naphan, "Probabilistic inference based classification applied to myoelectric signal decomposition," *IEEE Trans. Biomedical Engineering*, June, 1992.

[24] A.K.C. Wong and D.K.Y. Chiu, "An event-covering method for effective probabilistic inference," *Pattern Recognition*, vol. 20, no. 2, pp. 245-255, 1987.

[25] 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. 9, no. 6, pp. 796-805, 1987.

[26] A.K.C. Wong and T.S. Liu, "Typicality, diversity and feature pattern of an Ensemble," *IEEE Trans. Computers*, vol. 24, pp. 158-181, 1975.

John Y. Ching received the BASc degree in industrial engineering from the University of Toronto, Toronto, Ontario, Canada in 1987. He received the MASc degree in systems design engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 1992. He is currently a PhD candidate, and a student member of the Pattern Analysis and Machine Intelligence Laboratory of the Institute for Computer Research at the University of Waterloo.

He was a Canadian National Science and Engineering Research Council Scholar, and is a current recipient of the Ontario Graduate Scholarship. His research interests include machine learning, pattern analysis, neural networks, computer imagery and robot vision.

Andrew K.C. Wong (M 79) received his PhD from Carnegie Mellon University, Pittsburg, Penn., USA, in 1968, and taught there for several years thereafter. He is currently a professor of systems design analysis and the director of the Pattern Analysis and Machine Intelligence Laboratory at the University of Waterloo and an honorary professor at the University of Hull, UK.

Dr. Wong has authored and coauthored chapters and sections in a number of books on engineering and computer science, and has published many articles in scientific journals and conference proceedings. He is the 1991 recipient of the Federation of Chinese-Canadian Professionals Award of Merit.

Keith C.C. Chan graduated from the University of Waterloo, Ontario, Canada with a BMath degree in computer science and statistics. He obtained his MASc and PhD degrees in systems design engineering from the same university in 1985 and 1989, respectively,.

Dr. Chan was a research assistant in the Pattern Analysis and Machine Intelligence Laboratory at the Institute for Computer Research in the University of Waterloo from 1984 to 1989. Soon after graduation, he joined the IBM Canada Laboratory where he was involved in software development projects in the Image and Multimedia Systems Center and the Application Development Technology Center. He joined the Department of Electrical and Computer Engineering at Ryerson Polytechnic University, Toronto, Ontario as an associate professor in 1993. Since 1994, he has been with the Department of Computing at the Hong Kong PolytechnicUniversity, and is currently also adjunct professor in the Department of Systems Design Engineering, University of Waterloo, and the Department of Electrical Engineering, The Univesity of Western Ontario.

His research interest is in pattern recognition, artificial intelligence and neural and fuzzy systems.