A Distribution-Index-Based Discretizer for Decision-Making with Symbolic AI Approaches

QingXiang Wu, *Member*, *IEEE*, David A. Bell, *Member*, *IEEE*, Girijesh Prasad, *Member*, *IEEE*, and Thomas Martin McGinnity, *Member*, *IEEE*

Abstract—When symbolic AI approaches are applied to handle continuous valued attributes, there is a requirement to transform the continuous attribute values to symbolic data. In this paper, a novel distribution-index-based discretizer is proposed for such a transformation. Based on definitions of dichotomic entropy and a compound distributional index, a simple criterion is applied to discretize continuous attributes adaptively. The dichotomic entropy indicates the homogeneity degree of the decision value distribution, and is applied to determine the best splitting point. The compound distributional index combines both the homogeneity degrees of attribute value distributions and the decision value distribution, and is applied to determine which interval should be split further; thus, a potentially improved solution of the discretization problem can be found efficiently. Based on multiple reducts in rough set theory, a multiknowledge approach can attain high decision accuracy for information systems with a large number of attributes and missing values. In this paper, our discretizer is combined with the multiknowledge approach to further improve decision accuracy for information systems with continuous attributes. Experimental results on benchmark data sets show that the new discretizer can improve not only the multiknowledge approach, but also the naïve Bayes classifier and the C5.0 tree.

Index Terms—Data mining, machine learning, information theory, decision support.

1 Introduction

In general, machine learning or data mining can be regarded as a search for a good mapping from an input space (or condition space) to an output space (or decision space). The input and output spaces can be represented by variables with continuous numeric values or sets with symbolic data or nominal data. Particular machine learning approaches often prefer particular data types. For example, decision tree learning, rough set theory, evidence theory, and the Bayesian classifier are good at dealing with symbolic data. Neural networks, fuzzy set theory, support vector machines, and the kNN (k-Nearest-Neighbor) approaches are more suited to numeric data. In the real world, mixed data types are found. For machine learning or knowledge discovery in multiple databases [1], mixed data types cannot be avoided. If a machine learning approach with a numeric data preference is applied, the data must be transformed to a numeric type. If a machine learning approach with a symbolic data preference is applied, the data should be transformed to symbolic data or discretized values. These transformations may seriously affect the decision accuracy and the quality of the extracted knowledge. In addition, most data preprocessing approaches also have their own preferences for data types. Data type transformation is therefore one of the most critical aspects in data preparation for machine learning. This paper focuses only on the transformation from continuous numeric data to symbolic data by means of a continuous attribute discretizer. Since finding the best solution for discretization of continuous attributes is an NP-hard problem, many approaches have been proposed to obtain a good solution for application to real-world data sets. The approaches are always classified into two classes i.e., unsupervised and supervised discretizers. If a discretizer uses only condition attributes without the decision attribute, it is called an unsupervised discretizer. Examples are the equal interval discretizer and the equal frequency discretizer. If a discretizer uses both condition attributes and the decision attribute, it is called a supervised discretizer. In order to avoid searching all permutations and combinations of possible partitions, different heuristic approaches have been proposed, for example, approaches based on information entropy [2], [3], [4], the statistical χ^2 test [5], [6], and probability [7]. Two alternative strategies (bottom-up and top-down) are available to split a continuous attribute into several intervals. The Chimerge method [6] is a typical example of using a bottom-up strategy. This algorithm places the real values to its own intervals and then merges adjacent intervals by a measure of the expected independence based on the χ^2 test. The approach presented in [7] is a typical example of using a top-down strategy. In this algorithm, a continuous attribute is split into two intervals using Bayesian probability, and then the two intervals can be split again by analogy until a specific criterion is satisfied. The criterion or evaluation of such partitioning is a critical issue to obtain a good solution for discretization of continuous attributes. Traditionally, information entropy, the χ^2 test or probability are used to construct the criterion. For example, in [4], the entropy gain criterion was used as heuristic information, and the Minimal Description Length Principle (MDLP) was applied to determine a stopping criterion for the recursive

- *Q.X. Wu and D.A. Bell are with the School of Computer Science, Queen's University Belfast, Room 2013, SARC Building, Belfast BT7 1NN, North Ireland, UK. E-mail: {q.wu, da.bell}@qub.ac.uk.*
- *G. Prasad and T.M. McGinnity are with the School of Computing and Intelligent Systems, University of Ulster, Magee Campus, Londonberry, BT48 7JL, UK. E-mail: {g.prasad, tm.mcginnity}@ulster.ac.uk.*

Manuscript received 1 Nov. 2005; revised 14 Apr. 2006; accepted 24 Aug. 2006; published online 20 Nov. 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0496-1105.

discretization strategy. Dougherty et al. [2] have presented a very good survey and references for most approaches proposed before 1995. These approaches have been widely applied in the symbolic AI and data mining domains. However, many problems have not yet been satisfactorily solved. In order to avoid very thin partitioning in Chimerge, the FUSINTER method [5] used a bottom-up strategy to find the optimal partition based on a measure sensitive to the sample size. The results of χ^2-based approaches have ignored conflicting cases existing in the data set, and a significant value for the threshold was not initially available for different data sets. The threshold was always given after some tests were performed on the training data set. A modified Chimerge algorithm [8] was proposed as a completely automated discretization method. However, compared with the C4.5 and MDLP-based discretization algorithm [4], the modified Chimerge algorithm has no significant performance difference in terms of predictive accuracy. Furthermore, for a large data set, it generates a large tree compared with those generated with C4.5. Recently, the CAIM algorithm [3], which is based on class attribute interdependence, was proposed to obtain a continuous attribute discretization with the smallest number of intervals. This is one aspect of evaluating a set of intervals. It is worthwhile to further consider how a good evaluation can be made. For example, in [9], the stability of the continuous value discretization was proposed to evaluate a set of discretized intervals and extracted rules. Measures of the stability are based on a probability density function for each interval. These measures of stability have been applied to evaluate the results of current discretizers and rules extracted using rough set theory, but they have not yet been applied to develop a discretizer.

Current approaches are thus based on a single property (information entropy, χ^2 test or probability) of an instance information system. The improvements obtained in decision-making accuracy with these approaches are therefore limited. Additionally, complicated evaluation criteria may result in high computational complexities. Therefore, we propose a Distribution-Index-Based Discretizer (DIBD) to solve the discretization problem. In the DIBD, we take account of the natural distribution of data values. Based on a combination of entropy and homogeneity degrees of the value distribution and the decision value distribution, a dichotomic entropy is defined and applied to the DIBD to determine the best splitting point within an interval. A *value distribution index* and a *decision distribution index* are defined to create a *compound distributional index*. The *compound distributional index* for an interval always decreases when a large interval is split into two small intervals. Based on a *compound decrement* of the *compound distributional index*, our Top-Down Optimal Strategy (TDOS) is proposed to find a set of optimal intervals. The *compound decrement* is regarded as good for evaluating a *splitting operation* instead of evaluating *intervals* as in traditional approaches. This makes TDOS very different from a traditional top-down strategy based on binary entropy [4], [13]. The DIBD can adaptively discretize any continuous attribute according to simple adaptive rules based on *minimal dichotomic entropy* and *maximal compound decrement*. Based on this approach, a value area with high occurrence and high homogeneity degree is split into small intervals; a value area with low occurrence and low homogeneity degree is split into large intervals. The DIBD is also combined with the multiknowledge approach [10] so that a higher decision making accuracy can be reached. The experiments show that the DIBD works very efficiently, and can share statistical information with the multiknowledge approach and the well-known naïve Bayes classifier. Additionally, the discretizer can also be applied for use in other symbolic AI approaches to discretize continuous attributes.

The remainder of the paper is organized as follows: A representation of value distribution is introduced in Section 2. In Section 3, *dichotomic entropy*, *compound distributional index*, and *adaptive rules* are defined, and an example is presented to illustrate the algorithm in the DIBD. Experimental results and analysis, and details of the integration of the DIBD with the multiknowledge approach are given in Section 4. Section 5 concludes the paper.

2 Representation of Value Distribution

Following notations in [10], [11], [12], let $H =< U, A >$ represent *an information system*, where

$$U = \{o_1, o_2, \ldots, o_i, \ldots, o_n\}$$

is a finite nonempty set, called an object space or universe, and o_i is called an object. Each object has a finite nonempty set of attributes $A = \{a_1, a_2, \ldots a_i \ldots, a_m\}$, where m is the number of attributes. An *instance information system* is defined to distinguish an information system with decision attributes from a general information system. An *instance* is defined to distinguish an object with decision attributes from general objects. Let $I =< U, A \cup D >$ represent an *instance information system*, where $U = \{u_1, u_2, \ldots, u_i, \ldots, u_n\}$ is a finite nonempty set, called an instance space or universe, where u_i is called an *instance* in U, and n is the number of instances. Each instance has a set of attributes A and decision attributes D. D is a nonempty set of decision attributes or class attributes, and $A \cap D = \oslash$.

Let $a \in A$ and V_a represent a domain of attribute a. There is a mapping $a(u) : U \rightarrow V_a$ from U into the domain V_a. The mapping $a(u)$ represents the value of attribute a of instance u. For a given instance space U, domain V_a of attribute a is represented by the following expression:

$$V_a = \{d(u) : u \in U\} \; for \; a \in A. \tag{1}$$

The domain of a decision attribute is denoted by

$$V_a = \{d(u) : u \in U\} \; for \; d \in D. \tag{2}$$

The *condition vector space*, which is generated from attribute domain V_a, is denoted by

$$\begin{aligned} V_{\times A} &= \underset{a \in A}{\times} V_a = V_{a1} \times V_{a2} \times \ldots \times V_{a|A|}, \\ |V_{\times A}| &= \prod_{i=1}^{|A|} |V_{a_i}|, \end{aligned} \tag{3}$$

where $|V_{\times A}|$ is the size of the *condition vector space*.

The *decision vector space*, which is generated from the decision domain V_d, is denoted by

$$\begin{aligned} V_{\times D} &= \underset{d \in D}{\times} V_d = V_{d1} \times V_{d2} \times \ldots \times V_{d|D|}, \\ |V_{\times D}| &= \prod_{i=1}^{|D|} |V_{d_i}|, \end{aligned} \tag{4}$$

TABLE 1
Example Instance Information System

U	a1	a2	a3	a4	d
u1	1	1	1	4	+
u2	1	2	3	3	-
u3	2	3	1	4	+
u4	2	4	2	1	-
u5	3	4	2	2	-
u6	4	4	2	3	+
u7	4	3	3	3	-
u8	5	2	2	4	+
u9	6	1	1	4	+
u10	7	4	2	3	+
u11	7	2	3	1	-
u12	7	3	3	2	-

where $|V_{\times D}|$ is the size of the *decision vector space*. A conjunction of the values of condition attributes for an instance corresponding to a *condition vector* in the *condition vector space* is denoted by

$$\vec{A}(u) = (a_1(u), a_2(u), \ldots, a_{|A|}(u)). \quad (5)$$

Let AU represent a set of *condition vectors* which exist in the instance information system.

$$AU = \{\vec{A}(u) : u \in U\}, \quad (6)$$

where AU is a set of condition vectors without duplicated vectors. If $|AU| = |V_{\times A}|$, the system is called a *complete instance system*.

In the real world, training sets for decision making or in a classification task are rarely complete instance systems. In order to illustrate algorithms in this paper, Table 1 is taken as an example of an instance information system. In this instance information system, there are four attributes $A = \{a_1, a_2, a_3, a_4\}$, 12 instances $U = \{u_1, u_2, \ldots, u_{12}\}$, and one decision attribute with two values $V_d = \{+, -\}$. The value domains for each attribute are as follows:

$$V_{a_1} = \{1,2,3,4,5,6,7\}, |V_{a_1}| = 7.$$
$$V_{a_2} = \{1,2,3,4\}, |V_{a_2}| = 4.$$
$$V_{a_3} = \{1,2,3\}, |V_{a_3}| = 3.$$
$$V_{a_4} = \{1,2,3,4\}, |V_{a_4}| = 4.$$

The size of the condition vector space:

$$|V_{\times A}| = 7 \times 4 \times 3 \times 4 = 336.$$

The number of condition vectors appearing in the table: $|AU| = 12$. So, 324 possible condition vectors (or conjunctions of attribute values) did not appear in Table 1. Thus, Table 1 is not a complete instance information system, as there are 324 unseen instances. Machine learning or data mining approaches can be applied to extract knowledge from such an incomplete training set and make decisions for all instances, including the 324 unseen instances. The multiknowledge approach [10] was proposed to make decisions with high accuracy for unseen instances.

As the multiknowledge approach is good at dealing with symbolic data, it has the potential to further improve decision-making accuracy when combined with a continuous attribute discretizer. When the multiknowledge approach or the naïve Bayes classifier are combined for decision making, a high decision accuracy for instances with missing values can be obtained [10]. A statistical distribution is already used in the multiknowledge approach. In order to avoid increasing computational cost, it is proposed that the DIBD shares this statistical distribution. The statistical distribution is represented by a value statistical distribution table. In order to obtain the statistical table, different numeric summary variables are defined as follows.

Suppose that there is an instance information system $I =< U, A \cup D >$. Let N_{d_k} represent the number of instances with decision value d_k

$$N_{d_k} = |\{u : d(u) = d_k \text{ for all } u \in U\}|. \quad (7)$$

Let N_{d_k,a_i,v_x} represent the number of instances with decision value d_k and attribute value $v_x \in V_{a_i}$.

$$N_{d_k,a_i,v_x} = |\{u : d(u) = d_k \text{ and } a_i(u) = v_x \text{ for all } u \in U\}|. \quad (8)$$

Let N_{a_i,v_x} represent the number of instances for all decisions $d_x \in V_d$ and attribute value $v_x \in V_{a_i}$.

$$N_{a_i,v_x} = |\{u : a_i(u) = v_x \text{ for all } u \in U\}|. \quad (9)$$

We call a table showing such summary variables *a value statistical distribution*. For example, the value statistical distribution for Table 1 is shown in Table 2. The number N_{d_k,a_i,v_x} is a basic value distribution number. Based on the number N_{d_k,a_i,v_x}, the numbers N_{d_k} and N_{a_i,v_x} can be calculated by the following expressions:

$$N_{d_k} = \sum_{v_x \in V_{a_i}} N_{d_k,a_i,v_x} \text{ for any } a_i. \quad (10)$$

$$N_{a_i,v_x} = \sum_{d_k \in V_d} N_{d_k,a_i,v_x}. \quad (11)$$

Each value distribution number defined here is similar in concept to a histogram and a binning transformation is applied. A histogram is usually based on bins with equal sizes of intervals, whereas the value distribution number is based on the sampled values. The value interval is regarded as a range from (the value—lower neighbor value)/2 to (the upper neighbor value—the value)/2 (see reference [7]), for example, value range of bin v_4 is $[(v_4 - v_3)/2, (v_5 - v_4)/2]$. So, different values correspond to bins with different sizes of value interval. The statistical distribution in Table 2 can be shared with the naïve Bayes classifier. Based on the numbers in the table, the modified naïve Bayes classifier [10] can be written as follows:

$$d_{mp} = \arg\max_{d_k \in V_d} \frac{N_{d_k}}{|U|} \prod_i \frac{N_{d_k,a_i,a_i(u)} + \beta \bullet |U|}{N_{d_k} + \beta \bullet |U| \bullet |V_a|}, \quad (12)$$

where β is a small constant number [10] (a typical value of $\beta = 0.02$ is chosen for our experiments).

TABLE 2
Value Statistical Distribution

Decision	Attribute		N_{d_k,a_i,v_x} for $v_x \in V_{a_i}$							N_{d_k}
	Name	Domain V_a	v_{x1}	v_{x2}	v_{x3}	v_{x4}	v_{x5}	v_{x6}	v_{x7}	
d1='+'	a_1	{1,2,...,7}	1	1	0	1	1	1	1	6
	a_2	{1,2,3,4}	3	1	1	1	---	---	---	
	a_3	{1,2,3}	3	3	0	---	---	---	---	
	a_4	{1,2,3,4}	0	0	2	4	---	---	---	
d2='-'	a_1	{1,2,...,7}	1	1	1	1	0	0	2	6
	a_2	{1,2,3,4}	0	2	2	2	---	---	---	
	a_3	{1,2,3}	0	2	4	---	---	---	---	
	a_4	{1,2,3,4}	2	2	2	0	---	---	---	

3 Distributional-Index-Based Discretizer

In order to discretize a continuous attribute, the number of intervals and the borders of intervals have to be determined. If all numbers and borders of intervals are searched to get the "best" discretization solution, the computational complexity is excessive, so a Top-Down Optimal Strategy (TDOS) is applied to split an attribute into several nonidentical intervals with different splitting points or borders. The TDOS is similar to that in [7], but *distributional index* concepts are applied here instead of Bayesian probability. A *compound decrement* is applied to evaluate a splitting operation instead of a set of intervals.

3.1 Definition of Dichotomic Entropy

Our approach for DIBD suggests that an interval is split by a border value (or splitting point) and then the border is adjusted to reach a minimum of the dichotomic entropy. Let $v_x \in V_{a_i}$ be a value of continuous attribute a_i and let N_{d_k,a_i,v_x} represent the number of instances with decision value $d_k \in V_d$ and value v_x for attribute a_i. Suppose that a_i is split by border value v_{bd} (i.e., a splitting point). The number of instances with decision d_k and value $a_i(u) \leq v_{bd}$ is represented by $N_{d_k,v_{bd},left}$.

$$N_{d_k,v_{bd},left} = \sum_{v_x \leq v_{bd}} N_{d_k,a_i,v_x}. \tag{13}$$

The number of instances with value $a_i(u) \leq v_{bd}$ for all decisions $d_k \in V_d$ is represented by $N_{a_i,v_{bd},left}$.

$$N_{a_i,v_{bd},left} = \sum_{d_k \in V_d} N_{d_k,v_{bd},left}. \tag{14}$$

The number of instances with decision d_k and value $a_i(u) > v_{bd}$ is represented by $N_{d_k,v_{bd},right}$.

$$N_{d_k,v_{bd},right} = \sum_{v_x > v_{bd}} N_{d_k,a_i,v_x}. \tag{15}$$

The number of instances with value $a_i(u) > v_{bd}$ for all decision $d_k \in V_d$ is represented by $N_{a_i,v_{bd},right}$.

$$N_{a_i,v_{bd},right} = \sum_{d_k \in V_d} N_{d_k,v_{bd},right}. \tag{16}$$

In order to indicate instance number and homogeneity degree over a decision space within an attribute value interval, a *decision distributional index* is defined as follows:

$$E_d(v_{start} \rightarrow v_{end}) = \sum_{d_k \in V_d} -N_{d_k,v_{start} \rightarrow v_{end}} \log_2 \left(\frac{N_{d_k,v_{start} \rightarrow v_{end}}}{N_{a_i,v_{start} \rightarrow v_{end}}} \right). \tag{17}$$

Here, $N_{d_k,v_{start} \rightarrow v_{end}}$ represents the number of instances with decision value d_k and attribute value between v_{start} and v_{end}, and $N_{a_i,v_{start} \rightarrow v_{end}}$ represents the number of instances with attribute value from v_{start} to v_{end} for all decision values.

Now, clearly $E_d(v_{start} \rightarrow v_{end})/N_{a_i,v_{start} \rightarrow v_{end}}$ is Shannon's entropy in the interval for the decision space. Actually, $N_{d_k,v_{start} \rightarrow v_{end}}/N_{a_i,v_{start} \rightarrow v_{end}}$ can be regarded as a probability. The *decision distributional index* is equal to the product of $N_{a_i,v_{start} \rightarrow v_{end}}$ and the Shannon's entropy. The *decision distributional index* thus depends not only on the entropy but also on the number of instances falling in the interval. If there are two intervals with the same distribution and a different number of instances, entropy and probability cannot identify them. In addition, applying the numbers $N_{a_i,v_{start} \rightarrow v_{end}}$ and $N_{d_k,v_{start} \rightarrow v_{end}}$ instead of probabilities as variables for the equation, $E_d(v_{start} \rightarrow v_{end})$ can be implemented easily in online learning paradigms for dynamic environments, i.e., the numbers can be incremented according to a single value of a new instance. Suppose that the new instance $u_{13} = (a_1 = 7, a_2 = 3, a_3 = 3, a_4 = 1, d = '-')$ is added to the instance information system in Table 1. Only five numbers in Table 2 need to be updated as follows:

$$\begin{aligned}
N_{d_-,a_1,v_{x7}} &= N_{d_-,a_1,v_{x7}} + 1 = 2 + 1 = 3.\\
N_{d_-,a_2,v_{x3}} &= N_{d_-,a_2,v_{x3}} + 1 = 2 + 1 = 3.\\
N_{d_-,a_3,v_{x3}} &= N_{d_-,a_3,v_{x3}} + 1 = 4 + 1 = 5.\\
N_{d_-,a_4,v_{x1}} &= N_{d_-,a_4,v_{x1}} + 1 = 2 + 1 = 3.\\
N_{d_-} &= N_{d_-} + 1 = 6 + 1 = 7.
\end{aligned}$$

If the probabilities instead of the numbers are stored, an update of the probabilities requires all existing instances and the new instance. We have called $E_d(v_{start} \rightarrow v_{end})$ the *decision distributional index* in order to indicate the difference from the Shannon entropy. Note that

$$N_{a_i,v_{start} \rightarrow v_{end}} = \sum_{d_k \in V_d} N_{d_k,v_{start} \rightarrow v_{end}}.$$

Therefore, the larger the number $N_{a_i,v_{start}\rightarrow v_{end}}$ of instances within the interval, the larger the index. For example,

1. Consider a homogeneous distribution. $N_{a_i,v_{start}\rightarrow v_{end}} = 2$; $N_{d_-,v_{start}\rightarrow v_{end}} = 1$; $N_{d_+,v_{start}\rightarrow v_{end}} = 1$. We have
$$E_d(v_{start} \rightarrow v_{end}) = -1^* \log_2(1/2) - 1^* \log_2(1/2) = 2.$$
2. Consider the same homogeneous distribution with a larger number of occurrence instances
$$N_{a_i,v_{start}\rightarrow v_{end}} = 4, N_{d_-,v_{start}\rightarrow v_{end}} = 2,$$
and $N_{d_+,v_{start}\rightarrow v_{end}} = 2$. We have
$$E_d(v_{start} \rightarrow v_{end}) = -2^* \log_2(2/4) - 2^* \log_2(2/4) = 4.$$
Note that cases 1 and 2 are the same homogeneous distribution, but the *decision distributional index* for the two cases is different. Case 2 has a high *decision distributional index* because there are more instances in the interval. Shannon's entropy or probability cannot identify two different cases.
3. Consider a low homogeneous distribution with the same number of occurrence as that in case 2, i.e., $N_{a_i,v_{start}\rightarrow v_{end}} = 4$. Suppose that the distribution is represented by $N_{d_-,v_{start}\rightarrow v_{end}} = 1$ and $N_{d_+,v_{start}\rightarrow v_{end}} = 3$.
$$\begin{aligned} E_d(v_{start} \rightarrow v_{end}) &= -1^* \log_2(1/4) - 3^* \log_2(3/4) \\ &= 3.25. \end{aligned}$$
Cases 2 and 3 have the same occurrence number within their interval, but the index values are different. The index for case 3 is less than that in case 2 because case 3 has low homogeneous distribution. The more homogeneous the instance distribution over decision space, the larger the index.
4. Consider the same low homogeneous distribution with a larger occurrence number than that in case 3, i.e.,
$$N_{a_i,v_{start}\rightarrow v_{end}} = 8, N_{d_-,v_{start}\rightarrow v_{end}} = 2; N_{d_+,v_{start}\rightarrow v_{end}} = 6;$$
$$\begin{aligned} E_d(v_{start} \rightarrow v_{end}) &= -2^* \log_2(2/8) - 6^* \log_2(6/8) \\ &= 6.49. \end{aligned}$$

Cases 3 and 4 have the same distribution, but the $E_d(v_{start} \rightarrow v_{end})$ for case 4 is larger than that for case 3. If there are two intervals corresponding to cases 3 and 4, respectively, it is reasonable to select the interval corresponding to case 4 to further splitting. Shannon's entropy and probability cannot differentiate between two cases.

Based on the definition of *decision distributional index*, two *decision distributional indexes* can be obtained when an interval is split. A *left* decision distributional index can be represented by the following expression.

$$E_{left}(v_x \le v_{bd}) = \sum_{d_k \in V_d} -N_{d_k,v_{bd},left} \log_2 \left(\frac{N_{d_k,v_{bd},left}}{N_{a_i,v_{bd},left}} \right). \quad (18)$$

A *right* decision distributional index can be represented by the following expression:

$$E_{right}(v_x > v_{bd}) = \sum_{d_k \in V_d} -N_{d_k,v_{bd},right} \log_2 \left(\frac{N_{d_k,v_{bd},right}}{N_{a_i,v_{bd},right}} \right). \quad (19)$$

A *dichotomic entropy* for splitting point v_{bd} is defined as

$$E(v_{bd}) = \frac{E_{left}(v_x \le v_{bd})}{N_{a_i,v_{start}\rightarrow v_{end}}} + \frac{E_{right}(v_x > v_{bd})}{N_{a_i,v_{start}\rightarrow v_{end}}}, \quad (20)$$

where $N_{a_i,v_{start}\rightarrow v_{end}}$ is the total number of instances in the interval. According to the machine learning theory [2], [3], [4], [13], [14], the smaller the entropy, the better the attribute discretization. Applying (20), a border value v_{border} can be obtained by minimizing dichotomic entropy.

$$\begin{aligned} v_{border} &= \arg\min_{v_{bd} \in V_{a_i}} E(v_{bd}) \\ &= \arg\min_{v_{bd} \in V_{a_i}} \left(\frac{E_{left}(v_x \le v_{bd})}{N_{a_i,v_{start}\rightarrow v_{end}}} + \frac{E_{right}(v_x > v_{bd})}{N_{a_i,v_{start}\rightarrow v_{end}}} \right). \end{aligned} \quad (21)$$

In other words, the minimal entropy can be obtained if the value v_{border} is applied to split the attribute into two intervals.

3.2 Algorithm Development of Distributional-Index-Based Discretizier (DIBD)

Applying (21), a continuous attribute can be split into two intervals. The next step is to determine which interval should be split further. Although the decision distributional index can give some information for selecting an interval to split further, it is not enough to construct a good index for the top-down optimal strategy. Therefore, a *value distributional index* is defined as follows:

$$E_v(v_{start} \rightarrow v_{end}) = \sum_{v_{start} \le v_x < v_{end}} \sum_{d_k \in V_d} N_{d_k,a_i,v_x} \log_2 \left(\frac{N_{d_k,a_i,v_x}}{N_{a_i,v_x}} \right). \quad (22)$$

Clearly, $E_v(v_{start} \rightarrow v_{end})$ is small if the distribution varies very frequently with value v_x. $E_v(v_{start} \rightarrow v_{end})$ is large if the distribution varies very slowly with value v_x; in other words, the distribution over value v_x is homogeneous. In principle, an interval with a high homogeneous *decision distribution*, a low homogeneous *value distribution*, and a large number of instances should be split. Based on the difference of $E_d(v_{start} \rightarrow v_{end}) - E_v(v_{start} \rightarrow v_{end})$, a *compound distributional index* is defined as follows:

$$E_{com}(v_{start} \rightarrow v_{end}) = \frac{E_d(v_{start} \rightarrow v_{end}) - E_v(v_{start} \rightarrow v_{end})}{|U|}, \quad (23)$$

where $|U|$ is the total number of instances in the instance information system. As $|U|$ is a constant for an instance information system, dividing by $|U|$ ensures that the value of E_{com} is within [0,1] and makes it possible to get a threshold that is not highly sensitive to different data sets. The E_d and E_v for each interval always become small when an interval is split into two smaller intervals. Therefore, E_{com} for each interval always becomes small when an interval is split. This makes it possible to calculate new E_{com} only for split intervals and thus makes TDOS run very efficiently. This is very different from a traditional top-down strategy. The *compound distributional index* is related to the difference

between the decision distribution and the value distribution. Mathematically, it is easy to prove that the maximal value of $E_v(v_{start} \rightarrow v_{end})$ is equal to the *decision distributional index* $E_d(v_{start} \rightarrow v_{end})$ for the same interval. According to the concept of entropy, the maximal value of $E_v(v_{start} \rightarrow v_{end})$ corresponds to the most homogeneous distribution, i.e.,

$$\frac{N_{d_k,a_i,v_x}}{N_{a_i,v_x}} = \frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}} \text{ for } v_x \text{ within } v_{start} \rightarrow v_{end}.$$

From (22), the maximal value can be rewritten as follows:

$$E_{v,\max}(v_{start} \rightarrow v_{end}) = \sum_{d_k \in V_d} -\log_2\left(\frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}}\right) \sum_{v_{start} \le v_x < v_{end}} N_{d_k,a_i,v_x}$$

$$\Rightarrow E_{v,\max}(v_{start} \rightarrow v_{end}) = \sum_{d_k \in V_d} -N_{d_k,v_{start}\rightarrow v_{end}} \log_2\left(\frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}}\right).$$

Therefore, the minimal compound distributional index is 0.

$$E_{com}(v_{start} \rightarrow v_{end})_{\min} = 0. \tag{24}$$

The value distributional index in (22) is the entropy of the instance distribution over the value interval and decision space. The minimum of the value distributional index is zero (i.e., $E_v(v_{start} \rightarrow v_{end}) = 0$), corresponding to the state with the most inhomogeneous value number distribution over the interval. Hence, we have the maximum of the compound distributional index as follows:

$$E_{com\max}(v_{start} \rightarrow v_{end}) = \frac{1}{|U|} E_d(v_{start} \rightarrow v_{end}) = \frac{1}{|U|} \sum_{d_k \in V_d} -N_{d_k,v_{start}\rightarrow v_{end}} \log_2\left(\frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}}\right), \tag{25}$$

where $N_{a_i,v_{start}\rightarrow v_{end}} = \sum_{d_k \in V_d} N_{d_k,a_i,v_{start}\rightarrow v_{end}}$.

The equation can be rewritten as follows:

$$E_{com\max} = \frac{N_{a_i,v_{start}\rightarrow v_{end}}}{|U|} \sum_{d_k \in V_d} -\frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}} \log_2\left(\frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}}\right), \tag{26}$$

where $\sum_{d_k \in V_d} -\frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}} \log_2\left(\frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}}\right)$ is Shannon's entropy. If $|V_d| = 2$, the maximal value is 1. If the interval covers the whole value domain of the attribute a_i, $N_{a_i,v_{start}\rightarrow v_{end}} = |U|$. $E_{com\max} = 1$. If the attribute is split into several intervals, the E_{com} for the split interval is less than 1. The more intervals are split, the smaller is the value of E_{com} for each interval. The value of E_{com} for an interval is a real number within [0,1]. It is clear that E_{com} for an interval is dependent on the instance number within the interval. If an interval is small, i.e., $N_{a_i,v_{start}\rightarrow v_{end}}$ is small, the compound distributional index E_{com} is small. From (23), we have

$$E_{com}(v_{start} \rightarrow v_{end}) = \frac{1}{|U|}\left(\sum_{d_k \in V_d} -N_{d_k,v_{start}\rightarrow v_{end}} \log_2\left(\frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}}\right) - \sum_{v_{start} \le v_x < v_{end}} \sum_{d_k \in V_d} -N_{d_k,a_i,v_x} \log_2\left(\frac{N_{d_k,a_i,v_x}}{N_{a_i,v_x}}\right)\right).$$

Let $C_{d_k} = -\log_2\left(\frac{N_{d_k,v_{start}\rightarrow v_{end}}}{N_{a_i,v_{start}\rightarrow v_{end}}}\right)$ and $C_{v_x} = -\log_2\left(\frac{N_{d_k,a_i,v_x}}{N_{a_i,v_x}}\right)$. We have

$$E_{com}(v_{start} \rightarrow v_{end}) = \frac{1}{|U|}\left(\sum_{d_k \in V_d} N_{d_k,v_{start}\rightarrow v_{end}} C_{d_k} - \sum_{v_{start} \le v_x < v_{end}} \sum_{d_k \in V_d} N_{d_k,a_i,v_x} C_{v_x}\right).$$

As C_{d_k} depends only on the decision distribution and C_{v_x} depends only on the value distribution, and $N_{d_k,v_{start}\rightarrow v_{end}} = \sum_{v_{start} v_x < v_{end}} N_{d_k,a_i,v_x}$, we have

$$E_{com}(v_{start} \rightarrow v_{end}) = \frac{1}{|U|} \sum_{d_k \in V_d} \sum_{v_{start} \le v_x < v_{end}} N_{d_k,a_i,v_x}(C_{d_k} - C_{v_x}).$$

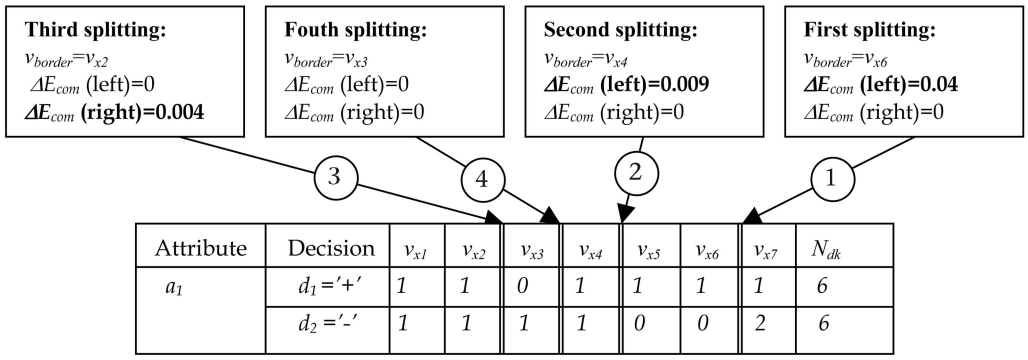
Putting the interval instance number $N_{a_i,v_{start}\rightarrow v_{end}} = \sum_{d_k \in V_d} \sum_{v_{start} v_x < v_{end}} N_{d_k,a_i,v_x}$ into the expression, we have

$$E_{com}(v_{start} \rightarrow v_{end}) = \frac{N_{a_i,v_{start}\rightarrow v_{end}}}{|U|} \sum_{d_k \in V_d} \sum_{v_{start} \le v_x < v_{end}} \frac{N_{d_k,a_i,v_x}}{N_{a_i,v_{start}\rightarrow v_{end}}}(C_{d_k} - C_{v_x}), \tag{27}$$

where

$$\sum_{d_k \in V_d} \sum_{v_{start} v_x < v_{end}} \frac{N_{d_k,a_i,v_x}}{N_{a_i,v_{start}\rightarrow v_{end}}}(C_{d_k} - C_{v_x})$$

is the average difference between the decision distributional index and value distributional index. From this equation, it is clear that the compound index is dependent on the instance number $N_{a_i,v_{start}\rightarrow v_{end}}$ and the average difference of the two distributions (*decision distributional index* and *value distributional index*). If the average difference between the decision distribution and the value distribution is large, $E_{com}(v_{start} \rightarrow v_{end})$ is large, and the interval should be split further. If there are a many of instances (or sampled values) within the interval (i.e., $N_{a_i,v_{start}\rightarrow v_{end}}$ is large), the value of $E_{com}(v_{start} \rightarrow v_{end})$ is large, and then the interval should be split further. Value $E_{com}(v_{start} \rightarrow v_{end})$ is thus a compound index for identifying an interval that should be split. Therefore, this *compound distributional index* is applied as a criterion to determine whether an interval is to be split further. *Compound distributional indexes* are calculated for all split intervals. When an interval is split into two new intervals, the maximal compound distributional index in two intervals will be reduced because the value occurrence number in the new interval becomes smaller. Let $E_{com}(v_{start} \rightarrow v_{end})$ represent the compound distributional index for the interval from v_{start} to v_{end}, $E_{com}(v_{start} \rightarrow v_{split})$ and $E_{com}(v_{split} \rightarrow v_{end})$ represent two indexes for two new intervals. A *compound decrement* due to this splitting is defined as



Attribute	Decision	v_{x1}	v_{x2}	v_{x3}	v_{x4}	v_{x5}	v_{x6}	v_{x7}	N_{dk}
a_1	d_1 ='+'	1	1	0	1	1	1	1	6
	d_2 ='-'	1	1	1	1	0	0	2	6

Fig. 1. Procedure for discretization.

$$\Delta E_{com}(v_{start} \rightarrow v_{end}) = E_{com}(v_{start} \rightarrow v_{end})\{E_{com}(v_{start} \rightarrow v_{end}) - \max[E_{com}(v_{start} \rightarrow v_{split}), E_{com}(v_{split} \rightarrow v_{end})]\}. \quad (28)$$

$\Delta E_{com}(v_{start} \rightarrow v_{end})$ is a product of the index $E_{com}(v_{start} \rightarrow v_{end})$ and its decrement due to splitting this interval. All the split intervals can be indicated with compound decrement ΔE_{com}. The interval with the largest $\Delta E_{com}(v_{start} \rightarrow v_{end})$ is selected to split further. In our experiments, the DIBD stops splitting when $\Delta E_{com}(v_{start} \rightarrow v_{end})$ is less than a threshold (0.001), or intervals reaches a desired number. It was found in practice that the threshold is not sensitive to different data sets and a maximum of five intervals are sufficient for reaching high decision accuracy in most cases. So, the maximal number of intervals is set to 5 in our experiments. Generally, the DIBD results in only two, three, or four intervals, as it is adaptive to data sets under the control of the threshold. It is thus very different from current approaches. For example, the discretization of attribute a_1 in Table 1 is illustrated in Fig. 1. Note that attribute a_1 is regarded as a continuous attribute. For simplicity, only seven values are sampled. In order to avoid calculating log(0), all distribution numbers are increased by a small number 0.0001, which makes programming easier.

In this case, it can be seen that the compound decrement $\Delta E_{com}(v_{start} \rightarrow v_{end})$ decreases from 0.04 to 0 as the number of occurrences within an interval becomes smaller in the splitting procedure. The largest compound decrement indicates the interval that should be split. The borders at the end of the procedure of discretization are very reasonable. v_{x5} is the same as v_{x6} fully supporting the decision $d1 = '+'$. They have been integrated in a single interval. Value v_{x1} does not support a specific decision, and the same is true for value v_{x2}. They have been integrated in another single interval. As this discretizer is based on the distributional indexes, it is called the DIBD (Distributional-Index-Based Discretizer). The formal algorithm for the DIBD is summarized as follows.

3.3 The DIBD Algorithm

These are three main steps:

1. Calculate the distribution numbers according to (7)-(11) and get the distribution numbers over the sampled values and over the decision space (an example is shown in Table 2).
2. Calculate dichotomic entropy and determine the splitting point
   2.1. Set initial values.
   Interval control number $n = 0$;
   $\mathrm{v_{start}} = \mathrm{v_{min}}$; $\mathrm{v_{end}} = \mathrm{v_{max}}$;
   Splitting point sequence list $\mathrm{S\_list} = [v_{min}, v_{max}]$.
   2.2. Determine the splitting point.
$$v_{bd\_n} = \arg\min_{v_{bd} \in V_{a_i}} E(v_{bd}) = \arg\min_{v_{bd} \in V_{a_i}} \left(\frac{E_{left}(v_x \le v_{bd})}{N_{a_i, v_{start} \rightarrow v_{end}}} + \frac{E_{right}(v_x > v_{bd})}{N_{a_i, v_{start} \rightarrow v_{end}}} \right).$$
   2.3. Add the splitting point into the splitting point sequence list.
   $\mathrm{S\_list} = [\mathrm{v_{min}}, \mathrm{v_{border\_n}}, \mathrm{v_{max}}]$.
3. Select an interval for splitting further.
   3.1 Calculate compound distributional index and compound decrement.
$$E_{com}(v_{start} \rightarrow v_{end}) = \frac{E_d(v_{start} \rightarrow v_{end}) - E_v(v_{start} \rightarrow v_{end})}{|U|}.$$
$$\Delta E_{com}(v_{start} \rightarrow v_{end}) = E_{com}(v_{start} \rightarrow v_{end}) \{E_{com}(v_{start} \rightarrow v_{end}) - \max[E_{com}(v_{start} \rightarrow v_{split}), E_{com}(v_{split} \rightarrow v_{end})]\}.$$
   3.2 Record compound decrement for each interval.
$$\mathrm{Dec\_list} = [\Delta E_{com}(\mathrm{v_{min}} \rightarrow v_{bd\_n}), \Delta E_{com}(v_{bd\_n} \rightarrow \mathrm{v_{max}})].$$
   3.3 Adaptive rule control.
   $n = n + 1$.
   $\Delta E_{com\_max} = \max \Delta E_{com}$ in Dec_list.
   If $\Delta E_{com\_max} < 0.001$ then end.
   If $n >= N_{max}$ (the maximal number of intervals) then end.
   3.4 Select the interval corresponding to the maximal ΔE_{com_max} for next splitting, i.e.,
   $\mathrm{v_{start}} = \mathrm{v_{start-with-max}}$; $\mathrm{v_{end}} = \mathrm{v_{end-with-max}}$.
   Goto 2.2.

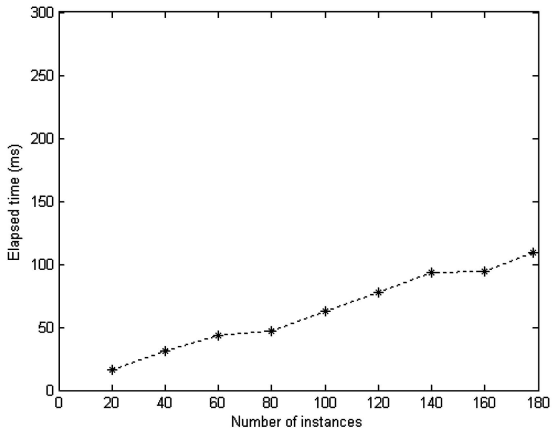


Fig. 2. Elapsed time versus number of instances.

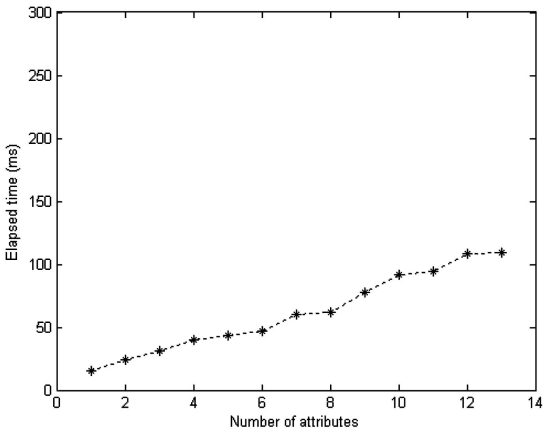


Fig. 3. Elapsed time versus number of attributes.

The DIBD algorithm contains three parts. The first part is the calculation of the distribution numbers defined in (7)-(11). Its time complexity is $O(|A|^*|U|)$. Here, $|U|$ is the total number of instances in an instance information systems and $|A|$ is the number of attributes. Its distribution numbers can be used with the multiknowledge approach or the naive Bayes classifier. The second part is the calculation of the dichotomic entropy according to (20). Its time complexity is $O(|A|^*|V_a|)$. The third part is the calculation of the compound distributional index and compound decrement according to (23) and (28). Its time complexity is $O(|A|^*|V_a|)$. Here, $|V_a|$ is the number of different sampled values for attribute a. As $|V_a| \ll |U|$, the computational cost increases very little in applying the DIBD to the multiknowledge approach or the naive Bayes classifier.

The efficiency of the algorithm mainly depends on two key factors—the number of instances and number of attributes. In order to show the scalability of the algorithm, a group of data sets with instance numbers [20, 40, ..., 140, 160, 178] was created by randomly drawing instances from the Wine data set in the UCI Machine Learning Repository. These data sets are applied to test the algorithm implemented using Jbuilder 4.5 on a PC with Pentium 4 (1.6GHz CPU). Discretization times for the data sets are shown in Fig. 2. It can been seen that the elapsed time of the algorithm increases linearly with the increase of instance number. Another group of data sets with different attribute numbers [1, 2, 3, ..., 12, 13] is formed by dropping different attribute numbers from the Wine data set. By using these data sets to test the algorithm, the curve for the elapsed time versus attribute number shown in Fig. 3 is obtained. It can been seen that the elapsed time of the algorithm again increases linearly with the increase of attribute number.

4 Experimental Results

4.1 Splitting Points and Statistical Distribution

Discretization results for the well-known Iris data set are shown in Fig. 4. We can very clearly see the relationship between the statistical distribution and the splitting intervals. The value statistical distribution is visualized using a histogram. As there are three decision values in the Iris data, three decision distributions for each attribute are shown in three histograms, indicated by N_{d_1,a_i,v_x} ,N_{d_2,a_i,v_x}, and N_{d_3,a_i,v_x}, respectively. Splitting points are marked with asterisks. The numbers beneath the asterisks indicate the splitting order when the DIBD applies the TDOS (top-down optimal strategy).

In Fig. 4a, the asterisk with number "1" is at $x = 9$. This point is determined by the minimal dichotomic entropy (21) for first splitting. Then, the compound distributional indexes and compound decrement for two intervals are calculated according to (23) and (28).

Compound distributional index: $E_{com}(v_1 \rightarrow v_9) = 0$; $E_{com}(v_{10} \rightarrow v_{45}) = 0.53$.

Compound decrement:

$$\Delta E_{com}(v_1 \rightarrow v_9) = 0; \Delta E_{com}(v_{10} \rightarrow v_{45}) = 0.25.$$

So, the interval $v_{10} \rightarrow v_{45}$ with the largest ΔE_{com} is selected to split further and the splitting point is found at $x = 24$. Two intervals $v_{10} \rightarrow v_{24}$ and $v_{25} \rightarrow v_{45}$ replace the interval $v_{10} \rightarrow v_{45}$, and three intervals are obtained. The interval with the largest ΔE_{com} is selected to be split further. The operation is repeated until the threshold is met or the maximal number of intervals is reached. All splitting points obtained are shown in Fig. 4a. It can be seen that values in the interval $v_1 \rightarrow v_9$ fully support decision d_1. Values in the interval $v_{28} \rightarrow v_{45}$ fully support decision d_3. Values in the interval $v_{10} \rightarrow v_{24}$ almost fully support decision d_2. Values in the interval $v_{25} \rightarrow v_{27}$ weakly support decision d_2 and strongly support decision d_3. A similar situation occurs in the discretization of A4 shown in Fig. 4b. The value statistical distributions for A1 and A2 shown in Fig. 4c and Fig. 4d are more complicated. More splitting points are required for separating the complicated distributions and small intervals are obtained. In this case, although the maximal number of splitting points is set to 5, only three splitting points are obtained for A3 and A4 as the DIBD adapts to the data set below the threshold.

Note that the x-axis represents integer labels for sampled intervals (or bins) in ascending values. For example, the integer labels for A4 corresponds to the real values shown in Table 3. The bin is itself adaptive to the sampled values instead of using equal intervals specified manually by a

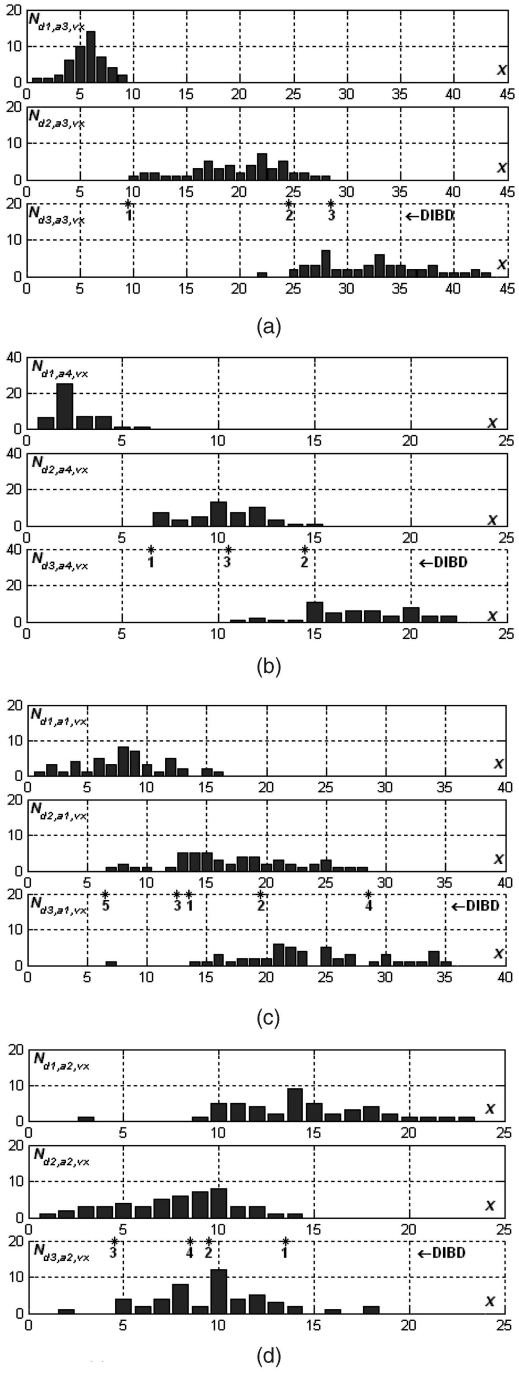


Fig. 4. Attribute discretization for the Iris data set. (a) Splitting points for A3 in the Iris data. (b) Splitting points for A4 in the Iris data. (c) Splitting points for A1 in the Iris data. (d) Splitting points for A2 in the Iris data.

user. The middle value is applied to transfer the integer labels to real values. For example, splitting point at $x = 6$ for A4 is transferred to value $(0.6 + 1.0)/2 = 0.8$. Transferred to real values, the splitting points are shown in Table 4.

TABLE 4
Splitting Points for Attributes in the Iris Data

Attributes	Splitting points				
A1	4.85	5.45	5.55	6.15	7.05
A2	2.45	2.85	2.95	3.35	
A3	2.45	4.75	5.15		
A4	0.80	1.35	1.75		

4.2 Controlling Maximal Interval Numbers

As the TDOS is applied, the maximal number of intervals is easier to control by DIBD than other current approaches. Different approaches were applied to the data set from [5]. The results are shown in Fig. 5. Splitting points marked with asterisks were found using the DIBD. Splitting points marked with squares were found using MDLPC. Splitting points marked with triangles were found using Chimerge with $\alpha = 0.05$. Splitting points marked with diamonds were found using FUSINTER.

As the DIBD finds a splitting point for each running step, the order number under an asterisk in Fig. 5 is the running step number for obtaining the splitting point. In the DIBD algorithm, the running step number can be controlled by both the maximal number of splitting points and a threshold value for compound distribution index. If the maximal number is set to 4, the DIBD obtained four splitting points that are the same as those for MDLPC and FUSINTER in this case. If there is no control for the maximal number of splitting points, the DIBD gives a very similar result to Chimerge with $\alpha = 0.05$.

4.3 Application of DIBD to Multiple Benchmark Data Sets

The DIBD was applied to improve the multiknowledge approach [10], [15]. The multiknowledge approach with the DIBD and without the DIBD was, respectively, applied to a set of 16 benchmark data sets from the UCI Machine Learning Repository. The decision accuracies under the ten-fold cross validation standard are given in column MK in Table 5. Subcolumn *Org* lists decision accuracies for the multiknowledge approach without the discretizer. Subcolumn *DIBD* lists decision accuracies for the multiknowledge approach with the adaptive discretizer. In order to compare these with an unsupervised discretizer, subcolumn $D5$ lists decision accuracies for the multiknowledge approach with a 5-identical-interval discretizer. Column Att shows attribute numbers in the data sets. The string "60c60" indicates that there are 60 attributes and 60 attributes are continuous attributes. Column N is for instance numbers in the data sets. The names with "♣" indicate that some attribute values are missing from the data set. The DIBD uses only

TABLE 3
Integer Labels of Attribute A4 Corresponding to Real Values

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0.1	0.2	0.3	0.4	0.5	0.6	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	2.1	2.2	2.3	2.4	2.5

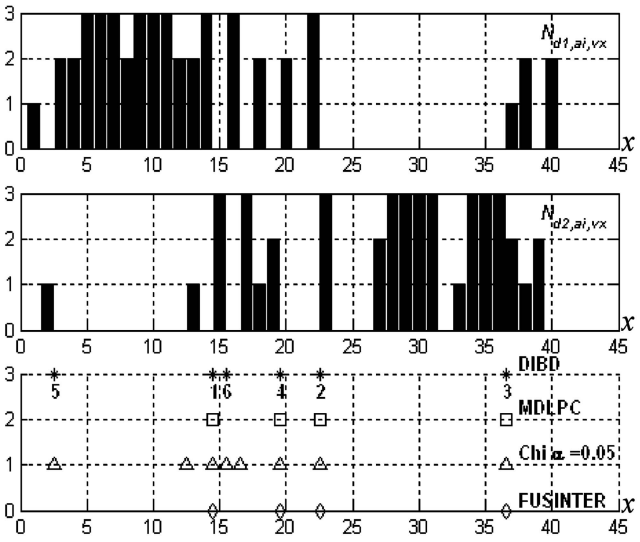


Fig. 5. Comparison of different discretizers.

the known values to find a good discretization solution. The missing values are handled by the multiknowledge approach [10], decision tree learning, or the Bayes classifier.

The results for the DIBD in Table 5 are obtained under a maximal number of five intervals. It can be seen that the decision accuracies for most data sets are improved. The multiknowledge approach with the DIBD improved decision accuracies for 14 of the data sets. The DIBD was also applied to the modified naive Bayes classifier. The results are shown in the column *Bayes*. The accuracies are improved for 15 of the data sets. Although the C5.0 tree contains a binary discretizer, the decision accuracies are still improved for 13 of the data sets when the DIBD is used for data preparation. In order to compare results with that obtained using 5-identical intervals, the maximal number of intervals is set to 5. In fact, different value distributions require different numbers of intervals

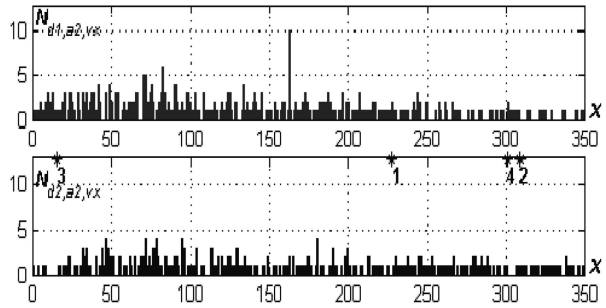


Fig. 6. Value distribution of attribute 2 in the Australian data set.

as shown in Section 4.1. This is the reason that the decision accuracy cannot be improved for some data sets with complicated distributions for statistical numbers. For example, the statistical distribution of Attribute 2 in the Australian data set is shown in Fig. 6. It can be seen that the value statistical distributions for both Decision 1 and Decision 2 are very complicated. Therefore, a large number of intervals is required to improve the decision accuracy. If the maximal number of intervals is set to 7 in the DIBD, the accuracies can reach 87.2 percent for the naive Bayes classifier and 87.9 percent for the MK.

An experiment on different interval numbers has been performed for the data set Anneal with the modified Bayes classifier. The results are shown in Table 6. It can be seen that the decision accuracy increases as the interval number increases, and reaches maximum at interval number 20. If the interval number limit is not applied to stop the splitting, and the splitting is stopped at a compound index threshold of 0.001, the decision accuracy can reach 95.9 percent. The decision accuracy for the other data sets can also be improved by cancelling the interval number limit. For example, without the interval number limit the accuracy for the Echocard data set can reach 82.0 percent. As most symbolic AI approaches will encounter a large number of

TABLE 5
Comparable Results for DIBD Discretizer

Data Name	Att	N	Bayes			MK			C5.0 tree		
			Org	DIBD	D5	Org	DIBD	D5	Org	DIBD	D5
Sonar	60c60	208	64.0	89.9	81.7	77.8	97.1	91.4	71.0	74.0	73.0
Horse-colic ♣	27c7	300	73.7	75.0	73.7	80.0	86.3	80.3	78.3	80.0	80.7
Ionosphere	34c34	351	82.6	90.3	88.6	90.6	93.7	92.6	88.3	91.5	89.5
Wine	13c13	178	98.3	98.3	96.1	98.9	99.4	97.8	93.2	97.7	95.5
Crx_data♣	15c6	690	81.0	86.5	85.0	85.1	86.5	85.0	85.1	86.4	84.1
Heart	13c6	270	76.3	84.4	84.8	83.3	86.3	85.1	78.1	76.3	79.3
Hungarian♣	13c6	294	83.3	83.7	83.7	85.4	85.4	84.0	79.2	80.9	78.2
SPECTF	44c44	80	66.3	88.8	75.0	73.8	98.8	92.5	70.0	76.2	68.8
Australian	14c6	690	85.2	85.5	86.5	85.8	85.8	86.5	83.2	85.9	84.5
Echocard♣	12c8	132	60.6	73.5	68.0	61.2	77.5	80.0	68.2	57.5	69.0
Bupa	6c6	345	64.4	70.2	67.0	65.5	70.2	67.0	67.2	65.2	66.4
Iris_data	4c4	150	94.0	96.7	82.0	96.7	96.7	93.3	96.7	96.7	96.0
Ecoli	6c6	336	71.5	75.3	75.0	71.5	75.3	75.0	71.5	83.6	82.4
Anneal♣	38c6	798	95.4	86.7	86.2	99.4	99.7	99.7	98.6	98.6	98.6
Hepatitis♣	19c6	155	69.6	70.8	69.6	75.5	75.5	76.7	62.0	68.5	67.3
Bands♣	39c20	540	64.8	66.3	68.5	77.8	79.6	76.5	68.0	68.3	69.4

TABLE 6
Decision Accuracy versus the Number of Intervals in Data Set Anneal by Using the Bayes Classifier Combined with DIBD

Maximum interval number	5	10	15	20	No interval number control
Decision accuracy	86.7	95.2	95.7	95.9	95.9

rules for an information system with a large number of intervals, there is a trade-off between the number of intervals and accuracy. The DIBD algorithm allows a user to control the maximal number of intervals easily. If one wants to obtain high accuracy, no limit is placed on the interval number. If one wants to obtain a compact information system, a small maximal number of intervals can be set. Note that the numbers of intervals for the attributes with simple normal distributions are determined by the threshold. The maximal number of intervals is only used for those attributes with informal and complicated distributions. If one does not want to control the maximal number of intervals, the following empirical formula can be used in the algorithm. For a given attribute,

$$N_{\max} = round(N_0 e^{-\lambda}), \tag{29}$$

where $N_0 = 20$, and

$$\lambda = \Delta E_{com}(v_{start} \rightarrow v_{end}) / E_{com}(v_{start} \rightarrow v_{end}).$$

Each attribute may have its own value of λ. The value is determined by the statistical distribution indexes for the whole value range of the attribute according to (23) and (28).

Applying (29) to the algorithm in Section 3.2.1, it is not necessary to specify a limit to the maximal number of intervals.

4.4 Comparative Evaluation

In order to compare the DIBD with other approaches, the DIBD results and recently published results are shown in Table 7. The results for CAIM, and ME (Maximum Entropy) and IEM (Information Entropy Maximum) were published in [3]. The results for Chi (Chimerge), MChi (Modified Chimerge), and MDLPC are published in [8]. The machine learning approaches used are C4.5, C5.0, CLIP4, naive Bayes classifier, and MK (Multi-Knowledge). The decision accuracies are represented by percentages. Based on these experimental results, it can be seen that a combination of the DIBD and MK gives the best decision accuracies for the used data sets. However, many more combinations of discretizers and machine learning approaches are possible. An exhaustive comparison is a topic for a further study. As seen in Table 7, it appears that different data sets prefer different combinations of discretizer and machine learning approaches. Applied to the data sets in Table 7, the DIBD and MK combination is superior to other approaches.

5 Conclusion

In this paper, the concept of dichotomic entropy is defined and it is shown that the minimal dichotomic entropy can be applied to determine a border value for splitting an interval. A compound distributional index, which is composed of a decision distributional index and a value distributional index, is defined and applied to identify the interval that should be split during the discretization. Based on these concepts, a continuous attribute can be split into two intervals at the border point with minimal dichotomic entropy, and then the compound decrement is applied to select an interval to split into further smaller intervals until the compound decrement meets a general threshold or the desired maximum number of intervals is reached. The DIBD was combined with the multiknowledge approach, the modified naive Bayes classifier, and the C5.0 tree method. Experimental results on 16 benchmark data sets show that the average accuracy has been improved at different rates. The multiknowledge approach and the modified naive Bayes classifier show greater improvement than the C5.0 tree. The DIBD can share statistical information with the multiknowledge approach and the modified naive Bayes classifier. Compared to other published results, the combination of the DIBD and MK gives the best accuracies for the same data sets. The DIBD is based on the TDOS strategy and provides a very simple way to allow a user to control the maximal number of intervals; it provides a very compact information system for symbolic AI approaches. The DIBD can also be combined with other symbolic machine learning approaches.

TABLE 7
Comparison of Different Methods

Method	Iris	Ion	Heart	Method	Iris	Wine	Bupa	Heart
CAIM-C5.0	95.3	89.0	76.3	Original-C4.5	95.33	92.7	68.4	53.8
CAIM-CLIP4	92.7	92.7	79.3	Chi-C4.5	94.0	88.8	50.0	55.1
ME-C5.0	93.3	86.5	73.3	MChi-C4.5	94.7	93.2	50.3	32.9
IEM-C5.0	95.3	92.6	73.4	MDLPC-C4.5	94.0	87.7	63.5	54.5
CAIM-Bayes	95.3	90.1	80.8		95.3	97.8	**70.2**	80.8
CAIM-MK	95.3	92.8	82.6		95.3	98.3	**70.2**	82.6
DIBD-C5.0	**96.7**	91.5	76.3		**96.7**	97.7	65.2	76.3
DIBD-Bayes	**96.7**	90.3	84.4		**96.7**	98.3	**70.2**	84.4
DIBD-MK	**96.7**	**93.7**	**86.3**		**96.7**	**99.4**	**70.2**	**86.3**

REFERENCES

[1] S. Zhang, C. Zhang, and X. Wu, *Knowledge Discovery in Multiple Databases.* Springer-Verlag, 2004.

[2] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features," *Proc. Int'l Conf. Machine Learning,* pp. 194-202, 1995.

[3] A.K. Lukasz and J.C. Krzysztof, "CAIM Discretization Algorithm," *IEEE Trans. Knowledge and Data Eng.,* vol. 16, no. 2, pp. 145-153, Feb. 2004.

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

[5] D.A. Zighed, S. Rabaseda, and S. Rakotomala, "FUSINTER: A Method for Discretisation of Continuous Attributes," *Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 6, pp. 307-326, 1998.

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

[7] X. Wu, "A Bayesian Discretizer for Real-Valued Attributes," *The Computer J.,* vol. 39, no. 8, pp. 688-691, 1996.

[8] F.E.H. Tay and L. Shen, "A Modified Chi2 Algorithm for Discretization," *IEEE Trans. Knowledge and Data Eng.,* vol. 14, no. 3, pp. 666-670, May/June 2002.

[9] M.J. Beynon, "Stability of Continuous Value Discretisation: An Application within Rough Set Theory," *Int'l J. Approximate Reasoning,* vol. 35, pp. 29-53, 2004.

[10] Q.X. Wu, D.A. Bell, and T.M. McGinnity, "Multi-Knowledge for Decision Making," *Int'l J. Knowledge and Information Systems,* vol. 7, no. 2, pp. 246-266, 2005.

[11] *Rough Set Methods and Applications, New Developments in Knowledge Discovery in Information Systems,* L. Polkowski, S. Tsumoto and T.Y. Lin, eds. Physica-Verlag, a Springer-Verlag Company, 2000.

[12] *Rough Set and Data Mining,* T.Y. Lin and N. Cercone, eds. Kluwer Academic Publishers, 1997.

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

[14] M.T. Mitchell, *Machine Learning.* McGraw Hill, copublished by the MIT Press Companies, Inc., 1997.

[15] Q.X. Wu and D.A. Bell, "Multi-Knowledge Extraction and Application," *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing,* G.Y. Wang, Q. Liu, Y.Y. Yao and A. Skowron, eds., LNAI 2639, pp. 274-279, Springer, 2003.

QingXiang Wu received the PhD degree in intelligent systems from the University of Ulster in United Kingdom in 2005. He is currently a research fellow in the School of Computer Science, Queen's University. He has been a research associate in the School of Computing and Intelligent Systems at the University of Ulster, and he has also fulfilled a role as professor in the School of Physics and OptoElectronics Technology, Fujian Normal University, in China. He is a member of the IEEE and a member of the International Rough Sets Society. He cooperated with Professor Bell to design the IFOMIND autonomous robot control system which won the Machine Intelligence Prize of 2005 competition organized by the British Computer Society. His research interests include intelligent systems, robotics, spiking neural networks, machine learning, knowledge discovery, data mining, automatic disease diagnosis, and Chinese information processing. He has more than 60 publications on these topics in international journals, Chinese journals, and proceedings of conferences.

David A. Bell graduated in 1969 in pure mathematics and has three research degrees in a variety of topics: programming language design, database performance, and AI in database systems. He has been a full professor since 1986. He has around 400 publications, including coauthoring *Distributed Databases* (Addison-Wesley) and *Evidence Theory and Its Applications* (North Holland), and has supervised more than 30 PhDs to completion. He has been a prime investigator on many national and EU-funded projects (e.g., MAP, ESPRIT, DELTA, COST, and AIM) in IT since 1981. He is a member of the editorial boards of *Information Systems* and *The Computer Journal*, and he has been guest editor of special issues of *Information and Software Technology* and *Information Systems*, on data mining and semantic Web, respectively. He has also chaired /cochaired several program committees, including those for VLDB and ICDE. His research is in data and knowledge management: the linking of reasoning under uncertainty, machine learning, and other artificial intelligence techniques with more established distributed database work exploiting the close relationship between evidence and data. He is a member of the IEEE.

Girijesh Prasad (M'98) received the BTech degree in electrical engineering from Regional Engineering College Calicut, India, in 1987 and the MTech degree in computer science and technology from the University of Roorkee, Roorkee, India, in 1992 and the PhD degree from the Queen's University of Belfast, Belfast, UK, in 1997. He has been working as a lecture at the University of Ulster since November 1999. He is a member of Computer Science Research Institute and Intelligent Systems Engineering Laboratory (ISEL). Previously, he worked as a digital systems engineer, as a power plant engineer, and finally as a research fellow on an EPSRC/industry project. His research work has been focused on devising self-organizing hybrid intelligent systems based mainly on computational intelligence techniques. This is mainly to effectively address the real-world problems associated with both industrial as well as biological systems having complex nonlinear, uncertain, and time-varying characteristics and diverse data types. Adopting a holistic approach to the system development, his work involves innovative fusion of neural computation, fuzzy neural networks, type-2 fuzzy logic, local model networks, evolutionary algorithms, adaptive predictive modeling and control, performance monitoring and optimization, and principal component analysis (PCA). His application area has been Brain Computer Interface (BCI) and Assistive Robotics, medical packaging production systems, autonomic computing systems, and thermal power plant. Dr. Prasad is a Chartered Engineer and a member of the IEE and the IEEE Computer Society.

Thomas Martin McGinnity has received the first class honors degree in physics, and a doctorate from the University of Durham. He has been a member of the University of Ulster academic staff since 1992, and holds the post of professor of intelligent systems engineering within the Faculty of Engineering. He is a fellow of the IEE, member of the IEEE, and a Chartered Engineer. He has 26 years of experience in teaching and research in electronic and computer engineering, leads the research activities of the Intelligent Systems Engineering Laboratory at the Magee campus of the University of Ulster, and is currently Acting Associate Dean of the Faculty of Engineering, with responsibility for research and development, and knowledge and technology transfer. He has authored or coauthored more than 150 refereed research papers and is the recipient of major research grant income from a wide range of sources. His current research interests relate to the creation of intelligent computational systems in general, particularly in relation to hardware/software implementations of neural networks, fuzzy systems, genetic algorithms, embedded intelligent systems utilising reconfigurable logic devices, brain-computer interfacing and, in particular, bio-inspired intelligent systems.