Applied Intelligence (2021) 51:991–1009
https://doi.org/10.1007/s10489-020-01850-w

A novel discretization algorithm based on multi-scale and information entropy

Yaling Xun[1] · Qingxia Yin[1] · Jifu Zhang[1] · Haifeng Yang[1] · Xiaohui Cui[1]

Published online: 12 September 2020

Abstract
Discretization is one of the data preprocessing topics in the field of data mining, and is a critical issue to improve the efficiency and quality of data mining. Multi-scale can reveal the structure and hierarchical characteristics of data objects, the representation of the data in different granularities will be obtained if we make a reasonable hierarchical division for a research object. The multi-scale theory is introduced into the process of data discretization and a data discretization method based on multi-scale and information entropy called MSE is proposed. MSE first conducts scale partition on the domain attribute to obtain candidate cut point set with different granularity. Then, the information entropy is applied to the candidate cut point set, and the candidate cut point with the minimum information entropy is selected and detected in turn to determine the final cut point set using the MDLPC criterion. In such way, MSE avoids the problem that the candidate cut points are limited to only certain limited attribute values caused by considering only the statistical attribute values in the traditional discretization methods, and reduces the number of candidates by controlling the data division hierarchy to an optimal range. Finally, the extensive experiments show that MSE achieves high performance in terms of discretization efficiency and classification accuracy, especially when it is applied to support vector machines, random forest, and decision trees.

Keywords Data mining · Discretization · Information entropy · Multi-scale · MDLPC criterion

1 Introduction

Data discretization is one of the data preprocessing methods in the field of data mining and knowledge discovery, which is to transform quantitative data into qualitative data by dividing continuous domains [35]. For data mining and machine learning, the discretization of continuous attribute can effectively reduce the granularity of the information system to improve the performance and learning accuracy of data mining/ machine learning algorithms, and enhance the ability of classify, cluster and anti-noise. In addition, many machine learning and data mining algorithms can only deal with discrete attributes, for example, C4.5/ C5.0 decision trees [26], association rules [32, 33], Naive Bayes [34] and rough sets [31]. In essence, data discretization is a data reduction mechanism. Continuous data is grouped into discrete intervals, while it still ensures the correlation between each discrete value and a certain interval. Therefore, data discretization can effectively hide the defects in original data and has attracted widespread attention [11].

Actual datasets often contain a large number of attributes, which can form conceptual hierarchies with a clear partial order structure. Dividing the attribute values based on related concepts in the concept hierarchy can form attribute value with multi-scale characteristics, and can obtain different granularity representations of the attribute value set. Since all data subsets in a certain scale representation form of a dataset are divided according to the attribute value set of a concept, each data subset has a specific and clear data meaning. In traditional algorithms, such as CAIM [18], CACC [27], MDLP [10], etc, only the mean of the

✉ Yaling Xun
xunyl55@126.com

Qingxia Yin
yqx_1995@163.com

Jifu Zhang
jifuzh@sina.com

Haifeng Yang
yhftxy7537750@163.com

Xiaohui Cui
cuixh_929@126.com

[1] Taiyuan University of Science and Technology (TYUST), Taiyuan, Shanxi, 030024, China

adjacent attribute intervals is considered as the candidate cut point set, and the data division based on it is insufficient. We introduce the multi-scale theory into the discretization process, which can reasonably divide the attribute value to obtain a set of candidate cut points. The candidate set is sorted, then the information entropy is applied recursively, always selecting the cut point with the smallest entropy. And MDLPC criterion is applied to decide when to refrain from applying further binary partitioning to a given interval. Therefore, the performance of the discretization algorithms and the classification accuracy of the classifiers have been significantly improved by combining multi-scale theory.

1.1 Motivations

- The dataset usually involves the relative size of the conceptual scope and granularity, and the multi-scale characteristics can reflect the nature of the dataset from multiple perspectives and hierarchies.

  Multi-scale can reveal the nature of the natural scale of a research object in essence. The data often corresponds to an attribute set when studying data from a certain category, which can form a conceptual hierarchy with a clear partial order structure. Dividing the data according to the concept hierarchy can form a dataset with multi-scale data characteristics, which is helpful for decision makers to make decisions from different perspectives. And the complexity of handling problems can also be further reduced by using scale conversion. Recently, multi-scale theory has been attempted to apply to general datasets. Hierarchical theory, conceptual hierarchy, and inclusion theory are used as the basis for scale division to study the distribution patterns in different scale hierarchies, and then to find meaningful facts, such as multi-scale association rules [21] and multi-scale clustering [12].
- Data discretization is an important data preprocessing technique. However, most traditional discretization approaches are difficult to reach a balance between running time and classification accuracy for classifiers.

  Many data mining/machine learning algorithms can only handle discrete data. However, the original user data is often continuous. Therefore, the discretization of these continuous data is necessary to facilitate the further processing of the algorithms. Moreover, data can be more further understood and reduced, which make data analysis faster and more accurate. Most discretization algorithms are difficult to achieve a balance in running time and classification accuracy when applying them to classification algorithms, even some discretization algorithms are only applicable to specific datasets. Therefore, it is necessary to research an efficient and usual data discretization method.
- Incorporating multi-scale theory, a more reasonable candidate cut point set can be obtained through reasonable data scale partition.

  The exploration of things, phenomena or processes will vary due to the choice of different scales. As a result, the inward nature of things may be comprehensively, partially, even incorrectly reflected. Dataset also tends to involve this multi-scale nature. If we can follow the essential characteristics of a research object and divide the corresponding data reasonably based on different scale characteristics, we can obtain more valuable information. Therefore, we introduce the multi-scale theory and give specific multi-scale partition strategy to divide the data and calculate candidate cut points with different granularities, the candidate set and computational overhead are greatly reduced. In addition, the classification results are obtained through a large number of known condition attributes and decision attributes. Therefore, the larger the amount of data, the higher the prediction accuracy. However, most discretization methods only consider the attribute values that have been counted, which makes the candidate cut points only limited to the determined finite attribute values. Cut points with different hierarchies are obtained through multi-scale partition. Then we utilize these points as test data, which will make the actual classification more reasonable.

1.2 Contributions

Compared with a large number of existing discretization methods, the main contributions of MSE are summarized as follows:

- The domain attribute is hierarchically divided by introducing multi-scale theory, and a set of candidate cut points with different granularity are obtained.
- Information entropy is applied to the obtained candidate cut point set, and the cut point with the minimum entropy is recursively selected and judged by MDLPC criterion to generate the final discrete interval.
- A data discretization algorithm based on multi-scale and information entropy, called MSE, is proposed.
- We conduct extensive experiments to exhibit that MSE offers ample opportunities to boost the execution efficiency of discretization algorithms and classification accuracy for classifiers.

1.3 Organization

The rest of this paper is organized as follows. Section 2 investigates previous work related to this study. In Section 3, we describe basic concepts pertaining to data discretization as well as muti-scale. Then, a data discretization algorithm based multi-scale and information entropy (i.e., MSE) is presented in Section 4. Sections 5 and 6 detail the experimental

settings and comparison results respectively. We conclude our research work and future research directions in Section 7 .

2 Related work

In the field of data mining and machine learning, discretization of continuous attributes can not only effectively reduce the time and space overhead, but also enhance the learning accuracy and anti-noise ability of algorithms. The most attention of discretization algorithms and the multi-scale theory are summarized as follows:

- **Discretization algorithms based on class-attribute interdependence.** Kurgan et al. proposed a classic algorithm— CAIM (class-attribute interdependence maximization), which is a global, static, top-down, supervised discretization algorithm [18]. They emphasized that CAIM can generate a minimal number of discrete intervals and need not require the user to predefine the number of intervals. However, CAIM has three drawbacks. First, the importance of attributes is not fully considered during the discretization process. Second, the inconsistency rates of the decision-making table is ignored. Finally, it is unreasonable to adopt the caim value as a discrete discriminant. The above drawbacks often results in information loss, and the accuracy of machine learning is affected. To address the issues of CAIM, Cano et al. presented ur-CAIM, which extended the CAIM criterion to address interdependence, redundancy, and uncertainty of class-attributes [5]. Therefore, the algorithm is superior to CAIM, especially on unbalanced datasets, which generating fewer intervals and better discretization schemes at the lower computational overhead. The same year, Cano et al. presented LAIM (Label-Attribute Interdependence Maximization), which is inspired in the discretization heuristic of CAIM for single-label classification [4]. LAIM provides the possibility to process multi-label dataset. Tsai et al. proposed CACC (class-attribute contingency coefficient), which is a static, global, incremental, supervised and top-down discretization algorithm [27]. They developed a novel heuristic objective function that takes into account the class distribution information for all samples. CACC avoids overfitting of the algorithm to produce better discretization results, and improve the classification prediction accuracy of machine learning. However, CACC is time consuming, which reduces its appeal when applying on real-world problems. Xiaolong Liu et al. proposed an improved algorithm based on CACC, which selects the cut points using the CACC standard and increases the constraint conditions of the data inconsistency rate to reduce the amount of data loss information [20].
- **Discretization algorithms based on rough set theory.** Hong Shi et al. proposed a novel algorithm, which implements global discretization through consistency measurement, which overcomes the defect of the inconsistency rate introduced by the local discretization MDLPC criterion [25]. Cheng et al. proposed an improved continuous attribute discretization algorithm based on rough set from the perspective of decision tables and information entropy [38]. In which, the concepts of 'conditional attribute weights' and 'equivalent class projections' are defined. Unnecessary candidate cut points are quickly eliminated by judging the importance of conditional attributes to the decision table and comparing the relations between conditional attribute values and equivalent class projections, and then algorithm efficiency is significantly improved. Jiang et al. proposed a supervised multivariate discretization method (abbr. SMDNS), which uses the interdependence between class information and condition attributes to improve classification effect [15]. Cao et al. proposed a continuous attribute discretization algorithm combining binary ant colony and rough set [6]. This algorithm constructs a binary ant colony network on the multi-dimensional continuous attribute candidate breakpoint set space. According to the approximate classification accuracy of the rough set, fitness evaluation function is established to find the globally optimal discretized breakpoint set.
- **Discretization algorithm based on clustering.** Min et al. proposed a global discretization and attribute reduction algorithm based on clustering and rough set theory [22]. In which, k-means clustering algorithm is adopted by comparing different discretization methods. In order to overcome the deficiency of k-means clustering algorithm, F-analysis of variance statistics and the support strength of conditional attributes are introduced to control the effectiveness of discretization. In order to meet the premise of rough set theory, a reasonable number of clusters can be obtained based on the correlation index. Thereafter, attributes are reduced by using rough set theory and decision rules are derived. Jifu Zhang et al. first selected candidate initial fuzzy clustering center by using the density values of the samples to effectively overcome the shortcomings of sensitivity to noise data [35]. Then, the algorithm parameters are dynamically adjusted to achieve the best discretization of spectral characteristic lines based on the compatibility of the decision table.
- **Discretization algorithm based on entropy**. Recently, discretization methods based on information entropy have been widely researched. Fayyad et al. proposed a discretization algorithm based on the entropy and the Minimum Description Length Principle (MDLP) [10].

The algorithm selects the breakpoints that can form a boundary between classes, and uses MDLPC criterion to determine the appropriate number of discrete intervals. However, it belongs to local discretization methods and it is easy to introduce inconsistency rates. Addressing to this problem, lots of research work has been conducted. For example, a comprehensive analysis of local and global information based on information entropy is carried out by Wen et al. [28]. In the local discretization phase, k strong cut points are selected for each attribute to minimize the conditional entropy.

- **Discretization algorithms based on Chi2.** The Chi2-based algorithms are a typical supervised, global, bottom-up discretization algorithm of statistical independence. Kerber proposed the pioneering ChiMerge method in this series of methods [17]. First, continuous attribute values are sorted in ascending order, and then the set of each value of continuous attributes is used as an interval, and tests all adjacent intervals. The pair of chi-square statistics are used to determine whether the current adjacent interval is merged, that is, the minimum chi-square adjacent interval is merged iteratively. At the same time, a chi-square parameter threshold (significant level α) is artificially set, and the iterative process is terminated until the values of all adjacent interval pairs are greater than the given threshold. However, the calculation of the inconsistency rate leads to a reduction in the credibility of the original data and some classification errors. To cope with this problem, a series of studies have been proposed. Changlei Zhao et al. proposed a new data reduction method, namely RS-D (Rough Sets-Discretization), which performs attribute reduction and rule reduction on the discrete data using the Rectified Chi2 algorithm combined with rough set theory [23]. Yu et al. considered that the theoretical basis for determining the importance of a node using the value of the difference between the critical value D deviding $2V$ was insufficient, and Accuracy cannot be guaranteed [24]. So a novel discretization Method was proposed (a.k.a., Rectified Chi). Rectified Chi uses $(2k-v)/2k$ as an important part of the value of E_{ij}, and finally achieves the desired discretization result, which improves the learning accuracy of the classifier.
- **Discretization algorithms based on genetic algorithm.** Jing Zhang et al. proposed a multi-attribute discretization algorithm based on genetic algorithms and variable-precision rough sets [36]. It establishes the fitness evaluation function of genetic algorithm by approximating classification accuracy of variable precision rough set, and uses genetic algorithm to find the optimal breakpoint subset on multidimensional continuous attribute candidate breakpoint set. The algorithm achieves better data classification fault tolerance and anti-noise ability.
- **Scale theory and data mining.** Multi-scale theory has been paid close attention in data mining field. However, the research on multi-scale data mining is still in its infancy, lacking universal theory and methods. With the deepening of the application of big data, its research becomes more urgent. Mengmeng Liu et al. conducted a study of universal multi-scale data mining on theoretical and methodological aspect [21]. The point-domain Kriging method and area-domain Kerry were introduced to accomplish scale-down and scale-up mining respectively. Chao Li et al. proposed a multi-scale association rule scale-up algorithm MSARSUA, which introduced a similarity calculation method based on inclusion degree and a Gaussian pyramid scale-up theory [19]. The introduction of multi-scale theory can not only effectively reduce the scale of the problem to improve the processing efficiency but also help the decision maker to make decisions from different perspectives. In 2019, Ye Zhang et al. proposed a data scale partition method for multi-scale data mining, which is based on a discretization method using probability density estimation [37]. This method expands the scale data types and effectively reduces the scale effect caused by scale deduction in multi-scale data mining.

In summary, most of the supervised discretization algorithms ignore the more valuable information that may exist in the dataset when initially selecting the candidate cut point set, resulting in insufficient final discretization results. Therefore, candidate cut points with different granularities are obtained by incorporating the multi-scale theory to the division of the initial data, and using these points as test dataset will make the actual classification more reasonable.

3 Background

To facilitate the presentation of MSE, we summarize the notation used throughout this paper in Table 1.

3.1 Decision table

Discretization technique aims to divide the domain of the problem based on the known condition attributes and decision attributes of the decision table to ensure that the decision table has a high classification ability. The decision table is a table-like graphical tool, which is suitable for the situations that contain multiple, intercombined conditions and multiple decision-making schemes. A way to express complex logic accurately and concisely is to associate multiple conditions with actions to be performed after these conditions are met. Decision tables can clearly associate multiple independent conditions and corresponding action actions.

Table 1 Symbol and annotation

Symbol	Annotation
DT	A decision table
U	A nonempty finite set of objects called the universe
A	Conditional attributes
D	Decision attributes
V_a	V_a is the set of values for each $a \in A \cup D$
I_a	$U \rightarrow V_a$ is an information function for each $a \in A \cup D$
T	Cut point
C_i^a	$C_i^a (a \in A, 1 \leq i \leq \lvert V_a \rvert)$ is the i-th candidate cut point of the attribute A
d_0	The minimum value of attribute A
d_n	The maximum value of attribute A
O^j	The granularity of hierarchy
Ent	Information entropy

Definition 1 (**Decision table**). A decision table DT is defined as the 5-tuple [28]:

$$\mathrm{DT} = (\mathrm{U}, \mathrm{A}, \mathrm{D}, \{\mathrm{V_a} | \mathrm{a} \in \mathrm{A} \cup \mathrm{D}\}, \{\mathrm{I_a} | \mathrm{a} \in \mathrm{A} \cup \mathrm{D}\}) \quad (1)$$

where

1. U is a nonempty finite set of objects called the universe;
2. A is a nonempty finite set of conditional attributes;
3. D is a nonempty finite set of decision attributes;
4. V_a is the set of values for each $a \in A \cup D$; and
5. $I_a : U \rightarrow V_a$ is an information function for each $a \in A \cup D$.

Example 1 Table 2 illustrates a decision table DT, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $A = \{a_1, a_2\}$, and $D = \{d\}$.

3.2 Information entropy

Information entropy is a commonly used method to select cutting points in the supervised discretization algorithms. Information entropy is usually used to indicate the degree of chaos for a system. In this study, information entropy is used to indicate the purity of the divided dataset. A smaller entropy value means that the greater the data purity, that is, the higher availability of the discrete data will be obtained, and vice versa. Information entropy and its related definitions are as follows:

Table 2 Example of a decision table DT

U	a_1	a_2	d
x_1	4.0	0.4	1
x_2	4.3	1.0	1
x_3	5.1	1.0	1
x_4	4.3	2.9	0
x_5	6.2	7.5	0
x_6	7.0	7.6	1

Definition 2 (**Information entropy**). The information entropy of T is defined as [28]:

$$\mathrm{Ent(T)} = -\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{p(X_i)Log(p(X_i))} \quad (2)$$

where $p(X_i) = \frac{|X_i|}{|U|}$, and X_i is the distribution of decision attributes which have been divided according to the cut point T.

4 Data discretization based on multi-scale and information entropy

In this section, we give relevant definitions of the multi-scale theory, and elaborate on the idea of MSE.

4.1 Multi-scale partition

4.1.1 Related multi-scale definitions

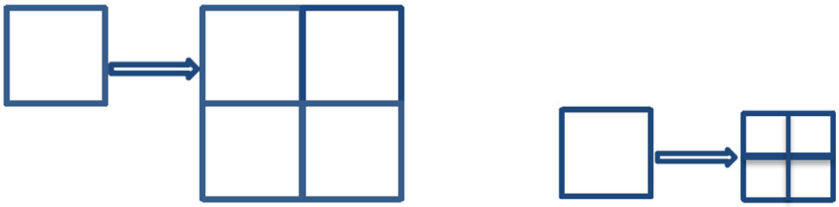
Definition 3 (**Scale**). Scale refers to the measurement unit of the research object, and is a standard for measuring research objects [37].

Broadly speaking, scale can be regarded as the unit or measurement tool of a research object. We apply scale theory to user data objects to help effectively discretize data. The 'scale measurement' includes two aspects: the range measurement (Fig. 1a) and the granularity measurement (Fig. 1b). The range scale measure studies the size of an object, and the granularity scale measure concerns the smallest measurement unit of a study object in a scale range. In this study, the granularity scale measurement method is adopted to divide the attribute values.

Definition 4 (**Concept hierarchy**). The concept hierarchy H is a partial order relation set $(H, \prec)$, where H represents a finite concept set, and $\prec$ reflects a partial order relation between two adjacent concepts contained in H [21].

The attribute set of data in certain category can form a conceptual hierarchy with a clear partial order structure: each attribute $h_i (i = 1, ..., n)$ can be regarded as a concept of the finite concept set $H = \{h_1, ..., h_i, ..., h_n\}$. Based on the domain knowledge, there is a partial order relationship among attributes, which corresponds to the partial order relationship of concepts in a finite concept set. An instantiated attribute $hi \in H$ usually corresponds to a group of specific attribute values, denoted as $V_{hi} =$

Fig. 1 Scale Measurement



(a) Range Measurement

(b) granularity measurement

$\{v_1, v_2, ..., v_{mi}\}$ (where, v_j represents a specific discrete value or a continuous interval). That is, the high abstraction of the group of attribute values in semantic forms the attribute (concept) h_i, we call the attribute value $v_j \in v_{hi}$ belongs to hi in semantic and record it as $v_j \in hi$. In practical applications, the attribute set in regional category can form a conceptual hierarchy $(H_{location}, \prec) = \{village \prec city \prec province \prec country\}$; the attribute set in the time category can also form a conceptual hierarchy $(H_{time}, \prec) = \{day \prec month \prec year\}$.

According to the definition of concept hierarchy, we can conduct multi-scale partition on the attribute values. Accordingly, those points used to divide the attribute values are called as the candidate cut point set. Below we give the definition of the candidate cut point set.

Definition 5 (**Candidate cut point set**). Given a decision table DT, the candidate cut point set of attribute A is:

$$C_i^a = d_0 + \frac{(d_n - d_0)i}{O^j} \tag{3}$$

Where

1. $C_i^a (a \in A, 1 \leq i \leq |V_a|)$ is the i-th candidate cut point of the attribute A;
2. d_0 is the minimum value of attribute A;
3. d_n is the maximum value of attribute A;
4. O^j is the granularity of hierarchy;
5. O is the order of the tree, which is used to determine the base of the granularity in the scale. This parameter defaults to 4, which is a discretizers parameter recommended by earlier studies. The best default value has also been verified by subsequent experiments in this study;
6. J is the layer number of the tree, which is used to determine the index of granularity division. The size of J depends on the logarithm between the number of distinct attribute values $Count_A$ and order O, that is $J = Log_O(Count_A/2)$. The value of J can ensure that the divided data can be controlled within a certain range to prevent overfitting.

Example 2 Suppose an attribute value ranges from 0 to 100 and contains a total of 90 different values. According to formula (3), the value d_0 is 1, d_n is 100, O is 4, and j is 2. The division process of this attribute value is shown in Fig. 2.

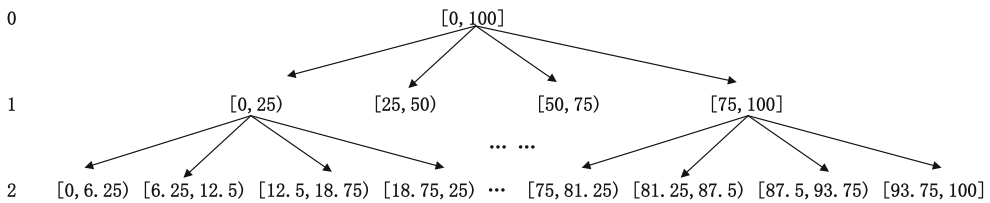
The candidate cut point set is: [0, 6.25, 12.5, 18.75, 25, 31.25, 37.5, 43.75, 50, 56.25, 62.5, 68.75, 75, 81.25, 87.5, 93.75, 100].

4.1.2 Multi-scale data partitioning

We apply multi-scale theory to study the optimal scale selection for continuous data. Due to the different value ranges and number of values of the multiple attributes that make up the dataset, we need to determine the best scale partition for each attribute. Different scale selection will affect the final conclusion, and even an irrational scale partition may lead to wrong conclusions. Therefore, we hierarchically partition the data from coarse to fine according to the concept hierarchy, then to determine the best partition scale depending on the adjustment of the scale granularity.

When scale partition is performed, the more hierarchies are divided, the finer the partition, however, the amount of

Fig. 2 An example for generating candidate cut point set



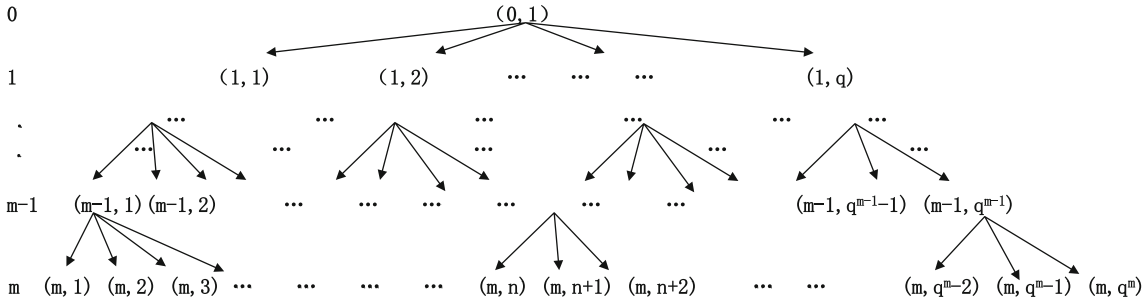


Fig. 3 Schematic representation of multi-scale interval of q-order tree

data will also increase accordingly. In the scale partition process, we can find that better candidate cutting points often correspond to a certain scale hierarchy. Therefore, the discretization algorithm can obtain the best trade-off between time and classification accuracy. Then, the corresponding hierarchy is the best scale we are looking for. Granularity scale measurement is the smallest measurement unit of the research object under the scale range, which is our main concern. Our proposed method MSE is similar to equal width discretization method. The biggest difference between them is that MSE carries out partition based on different granularities instead of the same granularity used in equal width discretization method. The specific partition idea of MSE is described as follows:

Based on the tree structure division, each interval partition maps the range of continuous values to a node of the tree structure. And the root of the tree is the value range of continuous attributes. As we gradually divide down the hierarchy, increasingly fine intervals are formed, The number of intervals determines the degree of accuracy. Figure 3 shows a general multi-scale interval representation based on a q-order tree, and Fig. 4 gives an instance of a four-order tree.

In the tree, a discrete interval is represented by two tuples (m, n), which are corresponding scale and node number respectively. That is, node (m, n) indicates that the interval is on the $n-th$ node of the $m-th$ hierarchy (scale m).

The coarsest scale locates the node $m = 0$ in which only one interval is expressed as (0, 1) interval (attribute domain). The number of intervals (a.k.a., the number of nodes) of the Q-order tree on the scale m is q^m. At each layer, we adopt equal width method to complete partition. Let $[a, b]$ be the domain of the continuous variable attribute x, then the discretization interval represented by the node (m, n) is $[a+(b-a)(n-1)/q^m, a+(b-a)n/q^m]$. The larger the values of q and m, the higher the classification accuracy, but it also means the calculation amount is also larger. From formula (4), the data partition hierarchy is determined by the number of different attribute values, that is, the larger the number of attribute values, the greater the number of hierarchies. Based on the interval partition, the obtained interval values are used as the candidate cut set to be further optimized in the next step.

4.2 Cut set detection based on information entropy and MDLPC criterion

In order to judge which points are the best after the candidate cut set is generated, information entropy is introduced. From the definition of information entropy introduced in Section 3.2, we can know that if the data in the dataset has good consistency, the corresponding information entropy value will be small. Therefore, our goal is to find a cut point with a small information entropy. The specific idea of cut set detection based on information entropy is as follows.

First, each candidate cut point is used to divide attribute value set V_a of an attribute A into two parts, and the class information entropy corresponding to each cut point is calculated according to definition 6 according to definition. Then, the cut point with the minimum entropy value will be selected as the candidate best cut point. However, whether the cut point can be determined as the final discrete interval needs to be judged by the MDLPC criterion (the reasons and definitions are given below). If a cut point is selected as the final discrete interval, the attribute value set will be divided into two subsets by this cut point. And the above process will be performed recursively on the divided attribute value subsets until all candidate cut points do not meet the MDLPC criterion.

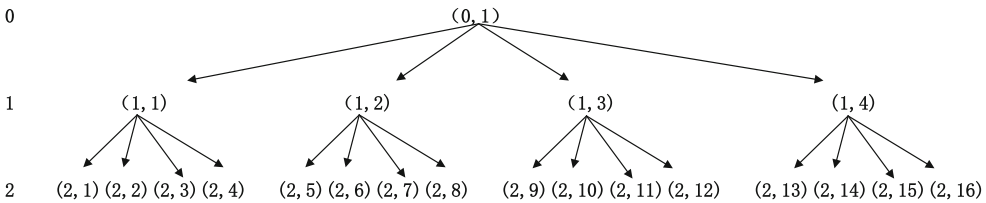


Fig. 4 An instance of four-order tree

Definition 6 (**Class information entropy**). For an example set V_a, an attribute A, and a cut value T: Let $V_{a1} \in V_a$ be the subset of examples in V_a with $A - values \leq T$ and $V_{a2} = V_a - V_{a1}$. The class information entropy of the partition induced by T, $Ent(A, T; V_a)$, is defined as [10]:

$$\mathrm{Ent(A, T; V_a)} = \frac{|\mathrm{V_{a1}}|}{|\mathrm{V_a}|}\mathrm{Ent(V_{a1})} + \frac{|\mathrm{V_{a2}}|}{|\mathrm{V_a}|}\mathrm{Ent(V_{a2})} \quad (4)$$

The discretized cut point of the attribute A is determined by selecting the cut point T_A with the minimal $Ent(A, T; V_a)$.

In most supervised discretization algorithms, the number of class labels is set to the maximum interval value of continuous data to determine the final discrete interval, such as CAIM, CACC, and so on. However, the division schema is not flexible enough and is not suitable for the data with various modes. The ideal discrete algorithm not only needs to consider enough interval values to prevent the loss of data information, but also to avoid overfitting due to too many interval values. Therefore, we use the evaluation standard proposed by Fayyad and Irani, that is MDLPC criterion [10]. MDLPC criterion compares the minimum description length without division and the minimum description length after division according to the best cut point to determine whether the cut point can be selected as the final discrete interval to divide the data. The data should be divided when the former value is greater than the latter, otherwise the division should be discarded. The MDLPC criterion is defined in Definition 7 below.

Definition 7 (**MDLPC criterion**). The MDLPC criterion is an evaluation standard for attribute selection metrics, that is, to determine whether a cutting point is a final cut point. For a set V_a of N examples, if a cut point T can be accepted as the final discrete cut point iff [10]):

$$\mathrm{Gain(A, T; V_a)} > \frac{\mathrm{Log_2(N-1)}}{\mathrm{N}}\mathrm{Ent(V_{a1})} + \frac{\Delta\mathrm{(A, T; V_a)}}{\mathrm{N}}\mathrm{Ent(V_{a2})} \quad (5)$$

Where, $\Delta(A, T; V_a) = Log_2(3^k - 2) - [kEnt(V_a) - k_1 Ent(V_{a1}) - k_2 Ent(V_{a2})]$, $Gain(A, T; V_a) = Ent(V_a) - Ent(A, T; V_a)$. Accept when the conditions are met, otherwise reject.

According to MDLPC criterion, the final number of interval values can be obtained objectively without causing overfitting and information loss.

4.3 Discretization algorithm based on multi-scale and information entropy

4.3.1 Algorithm description

Step 1: Select candidate cut points. The continuous attributes in the decision table are sorted in ascending order according to their values. we label the minimum value as d_{min} and the maximum value as d_{max}. And we initialize the discretization scheme $D[d_{min}, d_{max}]$ and candidate cut point set $C[d_{min}, d_{max}]$. The candidate cut point set $C[d_{min}, d_{max}]$ are then mapped to the initial root nodes of the O-order tree (for the initial value of O, see definition 5). The continuous attribute value range represented by $C[d_{min}, d_{max}]$ is divided layer by layer according to Definition 5. The boundary values corresponding to each node in the tree are calculated by the formula $C_i^a = d_0 + \frac{(d_n - d_0)i}{O^j}$ (a and i represent attribute and the number of cut points, respectively) until the number of division layers J is reached, then a complete O-order tree will be obtained. During this process candidate cut point set C will be updated by adding all newly generated boundary values C_i^a to C (see lines 1 to 9 in Algorithm 1).

Step 2: Select a optimal cut point. The class information entropy value corresponding to each candidate cut point is calculated based on definition 6, and the cutting point with the smallest entropy value is selected as the best cut point (see lines 12 to 13 of the following algorithm 1).

Step 3: Determine the final discrete interval set. The MDLPC criterion is adopted to judge whether the best cut point in step 2 can be selected as the final discrete interval. If it meets, add it to the discretization scheme $D[d_{min}, d_{max}]$, that is, it is determined as the final cut point to divide the data. Then, return to step 2 and start repeatedly on each divided data block. Otherwise, the cut point is discarded and the next cut point is selected to return to step 2. The algorithm terminates until all candidate cut points are judged (see lines 14 to 19 in Algorithm 1)

Algorithm 1 MSE algorithm.

Input: Dataset of M instances, S classes, and A attributes;
Output: Discretization scheme D for all attributes;
1: //A_i is the $i-th$ attribute in dataset
2: **for** each A_i in $dataset$ **do**
3: Sort all distinct values of A_i in ascending order;
4: Find the minimum d_{min} and maximum d_{max} values of A_i;
5: Set discretization scheme $D = [d_{min}, d_{max}]$;
6: Initialize candidate cut point set C with d_{min}, d_{max};
7: // See Definition 5
8: Calculate all possible candidate cut point values, $C_i^a = d_0 + \frac{(d_n - d_0)i}{O^j}$;
9: Add C_i^a into C;
10: **for** each item in inner candidate cut point C which is not already in scheme D **do**
11: // Calculation of information entropy see definition 6
12: Calculating the cut point corresponding information entropy(IE);
13: Select current minimal information entropy value in C and marked as IE_{min};
14: **if** $MDLP_Criterion(IE_{min}) == True$ **then**
15: Add IE_{min} into D;
16: **else**
17: Go to step 10;
18: **end if**
19: **end for**
20: **end for**

4.3.2 Time complexity analysis

In this section, the time complexity of the MSE algorithm for a single attribute is analyzed. The time complexity of the MSE algorithm is mainly determined by the calculation of all possible candidate cut points and the information entropy of each cut point. The specific steps involved are as follows:

All attribute values are searched to find the maximum and minimum values of the attribute values in Line 3. Suppose an attribute contains N objects, so the time complexity corresponding to this step is $O(N)$.

Line 8 is to calculate all candidate cut point using the given formula (5). In the worst case, the time complexity of this step is $O(m)$, where m is the number of unique values of the attribute.

Lines 10 to 19 in the algorithm are used to determine the final discrete interval. First, lines 12 to 13 calculate the entropy value of each candidate cut point. It can be known from the above algorithm description that the complexity of entropy calculation is $O(N)$, so the time complexity corresponding to these steps is $O(mN)$. Then the MDLPC criterion is used to detect these cut point. From the definition 7, it can be seen that the time complexity of the detection process i is $O(N)$, so the overall time complexity from line 10 to 19 is $O(mN) + O(N)$.

Therefore, for a dataset containing K attributes, the time complexity of the algorithm is $O(k) \times O(N + m + mN + N) = O(2NK + mK + mNK) = O(mNK)$.

5 Experimental setup

In this section, some existing classic algorithms (see Section 5.2 in details) and classifiers(see Section 5.3 in details) are selected to evaluate the performance of our proposed algorithm MSE. The performance differences among them are examined and analysis using the ten UCI datasets (see Section 5.4 in details).

5.1 Experimental setup

We implement the MSE and its comparison algorithms using Python 3.6.2 on a computing node equipped with Windows 10 operating system, InterCore i5-7500G Hz CPU and SamsungDDR44GB memory.

5.2 Discretization algorithms for comparison

The following typical discretization algorithms are chosen to effectively evaluate our algorithm MSE.

1. Multi-scale Data and Information Entropy
2. EW [29]: Equal Width
3. EF [29]: Equal Frequency
4. KMeans [8]: Clustering-based
5. MDLP [10]: Minimum Description Length Principle
6. CAIM [18]: Class-Attribute Interdependence Maximization
7. CACC [27]: Class-Attribute Contingency Coefficient
8. UrCAIM [5]: Improved CAIM Discretization
9. TSD [28]: A Two-stage Discretization

5.3 Classifiers for comparison

In order to avoid the bias of particular classifiers to data, 5 different classifiers belonging to different families are used to evaluate the classification performance, which increases the strength of the experimental study. The classifiers are:

1. CART [2]: This is a typical binary decision tree, considered one of the top 10 DM algorithms [30].
2. Naive Bayes [16]: This is another of the top 10 DM algorithms [30]. Its aim is to construct a rule which will allow us to assign future objects to a class, assuming independence of attributes when probabilities are established.
3. RandomForest [3]: This is an algorithm that integrates multiple trees through the idea of ensemble learning. It is unexcelled in accuracy among current algorithms.
4. SVM [7]: It is a supervised learning method, which is widely used in statistical classification and regression analysis. It is also considered one of the top 10 DM algorithms [30].
5. OneR [14]: This is a very simple classification method, which can quickly build a model for classification prediction. The basic idea of OneR is to use the most important features found in all the features of the dataset for classification.

5.4 Experimental dataset

We choose 10 datasets from the University of California Irvine Machine Learning UCI Database (http://archive.ics.uci.edu/ml) [1] to evaluate our algorithm. These datasets are typical

Table 3 The summary of 10 UCI experimental datasets

No.	Name	Examples	Attributes	Classes
1	abalone	4177	8	28
2	glass	214	10	6
3	ionosphere	351	33	2
4	iris	150	4	3
5	optdigit	5620	64	10
6	pendigits	10992	16	10
7	satellite	6435	36	7
8	shuttle	58000	9	10
9	waveform	5000	21	3
10	winequality	4894	11	7

due to their differences in complexity, number of classes, number of attributes, number of instances, etc., and they are often used by other algorithms to evaluate discretization performance [11].

1. Abalone Data (abalone)
2. Glass Identification Database (glass)
3. Johne Hopkins University ionosphere Database (ionosphere)
4. Iris Plants dataset Iris (iris)
5. Optical Recognition of Handwritten Digits (optdigit)
6. Pen-Based Recognition of Handwritten Digits Dataset (pendigits)
7. Statlog (Landsat Satellite) Dataset (satellite)
8. Statlog (shuttle) Dataset (shuttle)
9. Waveform Database Generator (Version 1) dataset (waveform)
10. Wine Quality (winequality)

The main characteristics of these datasets are summarized in Table 3. And the corresponding probability density functions are shown in Fig. 5 to express the probability of the attribute values of the datasets. In addition, the covariance matrix (Fig. 6) is adopted to demonstrate the actual relation among the attributes of each dataset. However, too large conditional attributes will make it impossible to display all the attribute information at the same time in the visualization process. Therefore, for cases where there are too many conditional attributes in individual datasets (such as ionosphere, optdigits, satellite, shuttle, waveform, winewuality), we select the conditional attributes with greater correlations to plot by calculating the correlation between conditional attributes and decision attributes using the chi-square.

6 Experimental analysis

In the experimental evaluation process, ten UCI datasets were tested to evaluate the performance of MSE. Note that the best results are marked in bold in all tables exhibiting experimental results below.

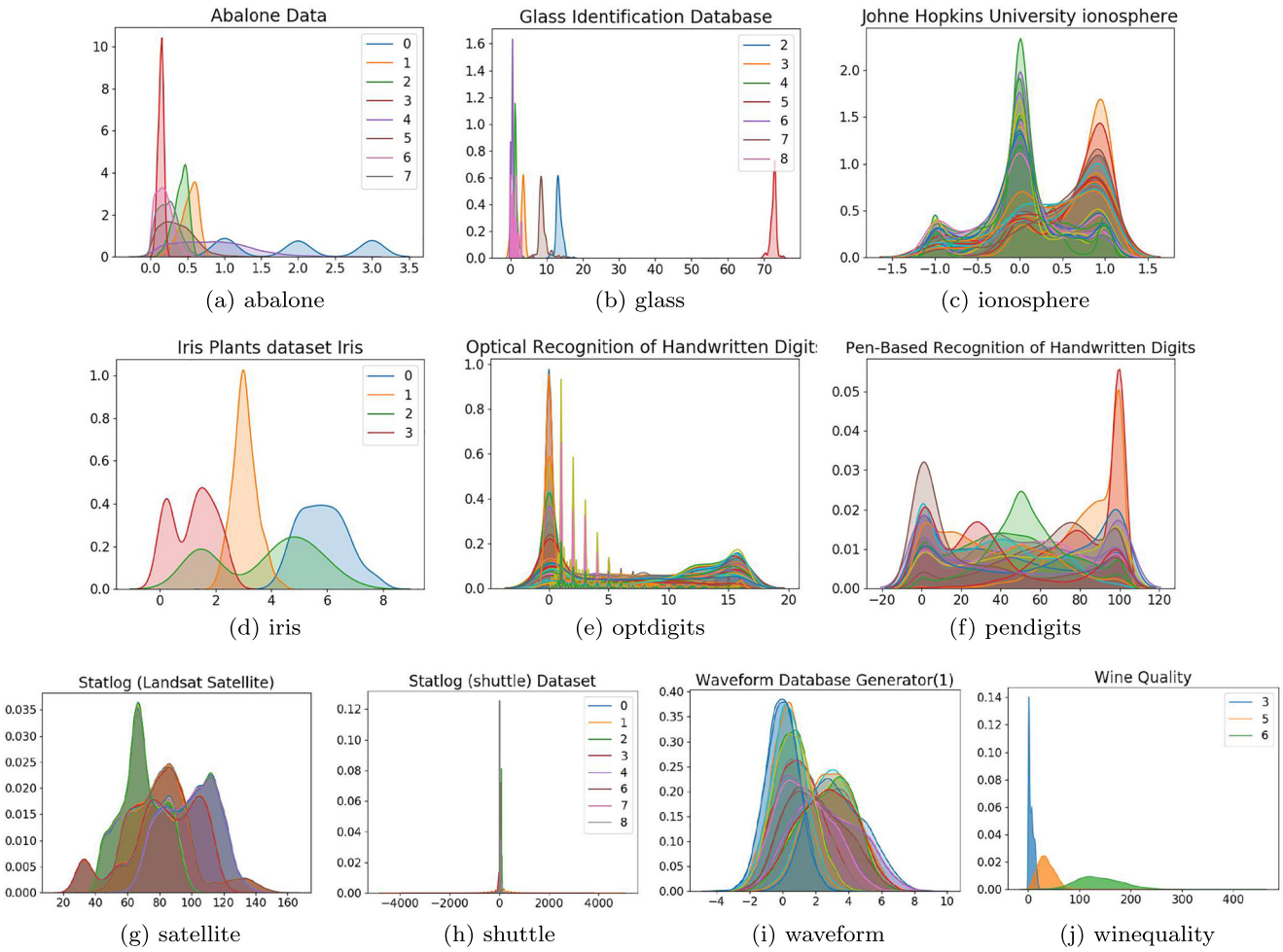


Fig. 5 The probability density functions of datasets

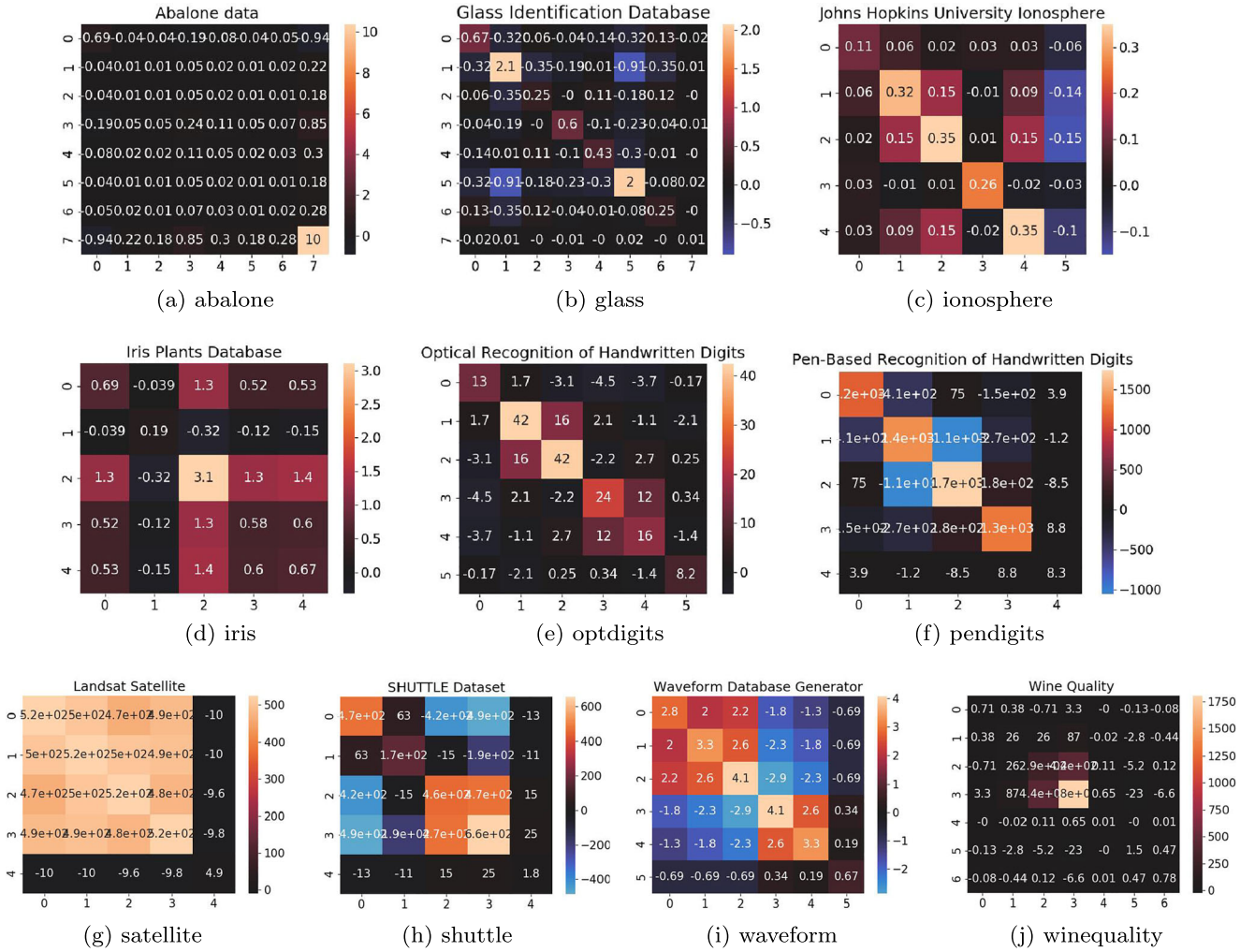


Fig. 6 The covariance matrix of datasets

In the subsequent experimental results, we added the last row or column (i.e., Rank) to compare the grades of the nine algorithms. Each Rank value is the grade mean of the corresponding discretization algorithm on 10 datasets. That is, for each dataset, we assign different grade to each algorithm according to performance. The algorithm grade with the best performance is assigned to 1, and so on. To further evaluate the significant differences in algorithm performance, we used the Friedman test and Holm post-hoc test [13] [9] to verify statistically. When the Friedman statistic value is greater than a certain threshold, it indicates that there is a statistical difference among the Rank of the discretization methods. The Friedman statistical distribution corresponds to the F-distribution with degrees of freedom $k-1$ and $(k-1)(N-1)$, in which k is the number of algorithms, N is the number of data sets. When the Rank difference between the two comparison algorithms is greater than the critical difference obtained by Holm post-hoc test, it indicates that the algorithm with the larger Rank value has a significant performance advantage.

Table 4 Impacts of different partition orders on the classification accuracy of CART

Orders	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality
3	24.9	86.21	–	**95.33**	89.02	96	83.48	99.04	**76.8**	**46.45**
4	**25.43**	90.05	**92.3**	93.33	89.02	96.45	**83.75**	**99.96**	76.1	44.49
5	25.07	**90.95**	90.6	94	**90.05**	96.23	83.1	99.07	75.8	45.53
6	24.47	–	91.45	93.33	90.05	**96.64**	83.26	98.93	76.48	45.53

Table 5 Impacts of different partition orders on the classification accuracy of RandomForest

Orders	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality
3	24.44	84.31	–	94.67	95.99	**98.71**	88.41	99.04	81.6	47.77
4	**25.47**	**86.19**	93.74	**94.7**	95.37	98.61	**88.42**	**99.96**	82.18	47.33
5	24.68	86.19	**94.31**	94	**96.03**	98.54	87.99	99.08	**82.88**	47.59
6	24.04	–	93.75	92.67	95.44	98.67	88.02	98.94	82.22	**47.94**

Table 6 Impacts of different partition orders on the runtime of MSE

Orders	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality
3	5.58	0.1825	–	0.015	0.7725	3.0575	1.7625	3.195	4.6675	1.3025
4	4.3425	0.1375	1.525	0.02	1.2025	2.5475	2.3725	2.9525	3.78	1.12
5	4.5475	0.2025	1.1025	0.0175	0.5025	1.07	1.18	3.1725	8.7275	1.315
6	5.235	–	1.8375	0.0175	0.615	1.62	1.73	3.325	3.3725	0.7775

Table 7 Impacts of different partition orders on the interval number of MSE

Orders	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality
3	46	28	–	11	241	162	438	68	105	36
4	45	28	130	13	253	161	420	73	109	40
5	45	28	131	12	206	161	424	73	109	38
6	45	–	132	11	224	164	442	68	106	38

Table 8 Impacts of different partition hierarchies on the classification accuracy of CART

Hierarchies	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Suttle	Waveform	Winequality
$J = log_O(Count_A/2) - 2$	**25.62**	58.48	88.05	-	9.27	95.88	68.72	87.98	75.48	**47.43**
$J = log_O(Count_A/2) - 1$	25.26	87.21	91.18	**94.67**	**90.12**	96.55	82.86	97.97	74.98	45.39
$J = log_O(Count_A/2)$	25.43	**90.05**	**92.3**	93.3	89.02	**96.45**	**83.75**	**99.96**	76.1	44.49
$J = log_O(Count_A/2) + 1$	–	–	–	94	89.88	96.38	83.09	99.96	**76.44**	45.27
$J = log_O(Count_A/2) + 2$	–	–	–	–	89.88	96.38	83.45	–	–	–

Table 9 Impacts of different partition hierarchies on the classification accuracy of RandomForest

Hierarchies	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Suttle	Waveform	Winequality
$J = log_O(Count_A/2) - 2$	**25.83**	57.58	90.87	–	9.32	97.42	72.91	87.98	81.62	46.78
$J = log_O(Count_A/2) - 1$	24.61	83.89	92.32	94.67	95.68	98.54	87.99	97.97	82.04	**47.65**
$J = log_O(Count_A/2)$	25.47	**86.19**	**93.74**	94.67	95.37	98.61	88.42	**99.96**	82.18	47.33
$J = log_O(Count_A/2) + 1$	–	–	–	**95.33**	**95.93**	**98.77**	88.28	99.96	**82.4**	47.47
$J = log_O(Count_A/2) + 2$	–	–	–	–	95.93	98.77	**88.78**	–	–	–

Table 10 Impacts of different partition hierarchies on the runtime of MSE

Hierarchies	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Suttle	Waveform	Winequality
$J = log_O(Count_A/2) - 2$	**0.40**	**0.01**	**0.17**	–	**0.17**	**0.28**	**0.18**	**0.32**	**0.39**	**0.14**
$J = log_O(Count_A/2) - 1$	1.39	0.065	0.67	**0.005**	0.44	0.97	0.92	1.20	1.31	0.27
$J = log_O(Count_A/2)$	6.02	0.2	1.93	0.03	1.45	3.77	3	3.95	5.38	1.30
$J = log_O(Count_A/2) + 1$	–	–	–	0.06	5.61	12	11.29	18.32	19.08	4.97
$J = log_O(Count_A/2) + 2$	–	0	–	–	-	49.93	41.42	70.55	–	–

Table 11 Impacts of different partition hierarchies on the interval number of MSE

Hierarchies	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Suttle	Waveform	Winequality
$J = log_O(Count_A/2) - 2$	**35**	**18**	**80**	–	**74**	**64**	**117**	**19**	109	**24**
$J = log_O(Count_A/2) - 1$	43	24	121	**12**	186	150	322	36	**105**	36
$J = log_O(Count_A/2)$	45	28	130	13	253	161	420	73	109	40
$J = log_O(Count_A/2) + 1$	–	–	–	**12**	278	165	448	134	106	37
$J = log_O(Count_A/2) + 2$	–	–	–	–	–	165	448	190	–	–

6.1 Impact of parameters on MSE

The parameters that affect the performance of the MSE algorithm mainly include the order and number of layers of the data partition. In this group of experiment, we verified the impact of these two parameters on the performance of the MSE algorithm in terms of running time, number of intervals, and classification accuracy.

In order to make the experimental results more convincing, experiments were performed on CART and RandomForest. CART has the characteristics of strong fault tolerance for outliers and high robustness, so it is not affected by some abnormal parameters in our experiments. For RandomForest, compared with other classification algorithms, it has a strong ability to resist overfitting, and can also balance errors on unbalanced datasets.

The effects of different partition orders on the performance of CART and RandomForest are shown Tables 4 and 5. It can be seen that when the order is 4, the classification accuracy performs best. From Tables 6 and 7, we can also see when the other parameters of the algorithm are fixed, the running time and interval numbers on most datasets are not much different for different orders. This trend is consistent with the quantile theory in statistics. The quartile, as a form of quantile, has very important meaning and function in statistics, which can effectively help us identify the characteristics of the data: (1) intuitively identify outliers in the dataset, (2) judge the degree of data dispersion and bias of the dataset.

Tables 8 and 9 show the effect of different partition hierarchies on the performance of CART and RandomForest. The experimental results clearly show that the classification accuracy can reach the optimal value when the number of hierarchies $J = log_O(Count_A/2)$. The experimental results in Table 10 clearly show that the execution time will increase as the number of partition hierarchies increases. And, the experimental results illustrated in Table 11 show that the number of interval values also increases with the number of partition hierarchies until a specific value is reached. The sign ' - ' in the table indicates that the running time is too long due to too many candidate cut points generated by the attribute value division. The reason for this trend is that more candidate cut points provide a better opportunity to choose the best cut point, so that the classification accuracy is improved. When the partition hierarchies reaches a certain value, the classification accuracy is no longer significantly improved because the number of selected cut points has met the selection criteria.

6.2 Discretization efficiency

In the comparative experiment, the parameter intervals of the unsupervised discretization algorithm are all set to 4 referring to other literatures [28], and the analysis

Table 12 Discretization efficiency of the nine algorithms

Algorithms	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality	Rank
MSE	6.17	**0.13**	1.98	**0.016**	1.84	**4.28**	**3.19**	**5.49**	6.81	**1.5**	4.15
EW	0.023	0.024	0.072	0.01	0.12	0.04	0.07	0.045	0.06	0.025	1.1
EF	0.034	0.051	0.065	0.026	0.13	0.05	0.215	0.067	0.115	0.123	2.2
KMeans	0.478	0.226	1.349	0.07	4.31	1.421	1.58	2.976	1.844	0.558	4.1
MDLP	13.48	0.189	2.01	0.025	2.15	6.667	4.06	9.378	16.75	3.102	5.8
CAIM	969.6	0.74	0.45	0.016	5.23	24.68	9.1	21.48	9.14	9.04	6.65
CACC	3535	7.2	100.7	0.156	6.78	112.9	88.4	653.7	4290.5	341.8	8.9
UrCAIM	32.97	0.5825	**0.275**	0.0325	**1.3175**	4.795	3.275	6.6475	**5.8825**	1.635	4.9
TSD	**1.598**	5.46	2.866	0.142	27.944	18.31	13.548	197.20	8.58	4.562	7.2

Table 13 Number of discretized interval generated by the nine algorithms

Algorithms	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality	Rank
MSE	45	**28**	130	13	253	161	420	73	109	40	6.2
EW	32	40	132	16	256	64	144	36	84	44	4.6
EF	**28**	32	130	16	207	64	144	**34**	84	44	4.2
KMeans	32	40	132	16	256	64	144	36	84	44	6.0
MDLP	50	29	130	12	278	165	448	199	106	37	6.4
CAIM	198	70	64	12	571	160	216	63	63	77	6.4
CACC	37	148	135	**11**	**196**	55	120	37	469	93	5.5
UrCAIM	32	45	110	11	201	64	133	**29**	53	29	3.1
TSD	37	27	**24**	12	18	**40**	**58**	35	**38**	**22**	2.3

results are shown in Table 12. For the supervised discretization algorithms, their parameter settings defer to the recommended configuration.

Table 12 shows The two classic unsupervised discretization algorithms, equal width and equal frequency, run faster than other algorithms because unsupervised algorithms do not involve heuristic knowledge and avoid extra judgement time. It can also be seen from Table 12 that our MSE algorithm outperforms most other supervised discretization algorithms in execution time. That is, there are significant differences between MSE and other algorithms, and it performs significantly better than the time-consuming CACC algorithm, which is also verified by Friedman test (the Friedman statistical value 6.43 is greater than the threshold of 3.1) and Bonferroni-Dunn test (the critical difference is 3.34). Because MSE effectively controls the amount of candidate set of cut points through scale division, thereby the calculation time of the optimal cut point selection is reduced. It is worth noting that the CACC and CAIM discretization algorithms obviously behave poorly on abalone dataset. This is mainly because the dataset is unevenly distributed and the large number of class labels affect the algorithm runtime. However, we also found that the UrCAIM algorithm performs better on certain datasets, because UrCAIM improves the efficiency of the CAIM algorithm by improving the discretization standard. However, TSD consumes a lot of time due to its two-stage execution strategy.

For different datasets, in the case that the amount of attribute data is less than 1000, such as glass, ionosphere, and iris, the algorithms run faster because the amount of data that needs to be processed is relatively small. However, from the time comparison of various discretization algorithms running on these three datasets, it is obvious that the ionosphere dataset consumes longer than the other two datasets. This is because the ionosphere dataset requires 33 discrete attributes, which is much larger than the other two datasets. As a result, the number of cycles of the algorithm will increase, and the running time will also increase, especially for CACC discretization algorithm. In the case where the data volume of an attribute is around 5000, such as abalone, optdigit, satellite, waveform, and winequality, most algorithms take a relatively long time on the two datasets abalone and waveform. This is because these two datasets contain more unique attribute values. And, for the two datasets pendigits and shuttle, although the data volume is large (the data volume of an attribute exceeds 10,000), the values are all integer values and the data distribution is relatively concentrated. Therefore, the running time on the two datasets is relatively stable.

6.3 Number of discretized intervals

Table 13 statistics the number of intervals generated by the discrete algorithm. In this group of tests, the corresponding Friedman statistical test value 5.52 of Table 13 is greater than the threshold value of 3.1, indicating that there is a statistical difference among the Rank of these discretization methods. The subsequent Bonferroni-Dunn test (the critical difference is 3.34) shows that MSE has no significant performance advantage over other algorithms. Since the

Table 14 Parameters of the discretizers and classifiers

Method	Parameters
CART	Pruned tree, 2 example per split, 1 example per leaf
RandomForest	Pruned tree, 2 example per split, 1 example per leaf
SVM	K=3, threshold =0.01
EW	numIntervals = 4
EF	numIntervals = 4
KMeans	numIntervals = 4
MDLP	Recommended by the authors
CAIM	Recommended by the authors
CACC	Recommended by the authors
urCAIM	Recommended by the authors
TSD	Recommended by the authors

Table 15 Impacts on classification accuracy of CART

Algorithms	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality	Rank
MSE	**25.43**	90.05	**92.3**	95.33	89.02	**96.45**	**83.75**	**99.96**	**76.1**	44.49	2.1
EW	24.85	70.74	88.64	87.33	88.64	95.67	81.7	89.55	72.7	40.42	6.4
EF	24.83	67.38	87.76	88.67	88.78	95.53	81.06	89.45	74.32	40.92	6.5
KMeans	24.28	67.08	83.76	**95.33**	83.35	95.72	80.89	99.1	72.1	41.42	6.55
MDLP	24.71	**95.71**	90.90	94.0	**89.88**	96.38	83.45	**99.96**	75.98	**44.83**	2.2
CAIM	21.91	94.76	90.03	94.0	89.72	96.14	83.22	99.93	74.56	44.69	3.65
CACC	23.08	90.95	88.9	94.0	89.18	95.05	23.95	99.94	71.98	43.04	5.9
UrCAIM	25.06	90	90.04	94	89.59	95.3	83.5	99.82	73.18	44.53	4.1
TSD	24.49	57.64	84.91	94	10.48	89.77	55.79	99.81	60.98	43.26	7.6

difference in Rank between MSE and other algorithms does not exceed this critical value observed from Table 13.

The interval of the two unsupervised algorithms (i.e., equal width and equal frequency algorithms) and the KMeans algorithm are directly given by the user with strong randomness. Therefore, the interval are fixed. For the supervised discretization algorithm, the number of interval of MSE algorithm and MDLP algorithm is more than the other two supervised discretization algorithms. The main reason is that these two algorithms use the MDLPC criterion to obtain the interval instead of the fixed value given in advance. In the such way, continuous data can be divided more fully, which can reduce information loss caused by inconsistent data. Overall, the CACC and TSD algorithms perform better on the number of discretized interval. Because the number of intervals generated by the CACC algorithm is always kept within the number of class labels. And, TSD discretizes the data using two stages, which makes the result of local discretization further reduced during the global discretization process, resulting in fewer discrete intervals.

6.4 Impacts on classification accuracy

We evaluate the the impact of the MSE discretization algorithm on classification accuracy by applying our MSE to five classic classification algorithms, widely used to verify the classification accuracy for the discretization algorithms. The datasets used in the experiment are partitioned using the 10-fold cross-validation (10-fcv) procedure. The parameters we used in the discretizer and classifier experiments were recommended by their respective authors, and we assume that these parameters are optimal. The specific parameters are listed in Table 14. From the Rank value, Friedman and Bonferroni-Dunn tests show that the MSE algorithm has the best comprehensive performance, because it ranks the first in four classification algorithms. On the contrary, the TSD algorithm performs the worst among the remaining algorithms. Below, we elaborate on the impact of these discretization methods on the accuracy of each classifier.

All experimental results shows the two unsupervised discretization methods, the equal width and equal frequency

Table 16 Impacts on classification accuracy of RandomForest

Algorithms	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality	Rank
MSE	**25.47**	86.19	93.74	94.7	95.37	98.61	88.42	**99.96**	**82.18**	47.33	2.35
EW	24.71	65.67	90.06	85.33	93.73	97.38	85.52	89.55	78.94	47.96	6.3
EF	24.45	66.1	90.12	90.67	93.65	97.53	86.06	89.09	81.34	48.63	5.6
KMeans	24.23	63.18	91.46	93.33	90.37	97.49	86.28	99.1	78.16	47.35	6.2
IEM	24.06	**87.21**	94.02	**95.33**	95.93	**98.77**	**88.78**	**99.96**	82.02	46.88	2.45
CAIM	23.56	82.38	92.9	94.0	**96.05**	98.49	87.69	99.93	79.8	**49.27**	3.75
CACC	23.41	84.76	92.6	93.33	94.68	96.38	25.1	99.93	78.24	46.06	6.45
UrCAIM	24.9	83.42	**94.04**	94.67	95.78	97.48	86.71	99.82	78.46	46.45	4.1
TSD	24.22	55.37	86.33	94.66	10.44	91.38	57.71	99.81	67.36	44.85	7.8

Table 17 Impacts on classification accuracy of Naive Bayes

Algorithms	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality	Rank
MSE	**19.08**	81	92.6	95.33	**78.15**	**84.91**	79.16	51.97	81.76	7.76	3.0
EW	11.4	45.52	90.63	84.7	70.18	77.54	72.7	0.4	78.14	2.16	7.3
EF	12.12	55.23	90.34	86	70.89	77.58	77.24	16.83	81.28	39.89	5.4
KMeans	15.9	29.33	74.94	91.3	42.28	70.18	76.28	14.79	73.38	37.01	7.0
IEM	13.31	**95.24**	89.7	94.7	76.16	84.79	**79.34**	**93.64**	**82.04**	10.84	3.0
CAIM	12.28	90.02	92.3	94.0	76.98	84.41	77.58	90.85	81.26	18.15	3.95
CACC	8.24	90.47	92.3	94.0	68.08	82.75	24.77	98.38	80.08	39.28	5.15
UrCAIM	11.25	90.48	**93.15**	94	74.77	83.36	78.26	91.03	80.84	38.06	4.0
TSD	12.35	31.32	70.96	**96**	10.559	72.51	51.12	71.91	71.76	**40.38**	6.2

discretization methods, behave poor classification accuracy because of a large data inconsistency rate caused by the equal division of data without considering decision attributes. The classification effect of KMeans discretization algorithm be superior to the two unsupervised algorithms, because it clusters close data into one category when dividing dataset. However, the classification effect of KMeans is not perfect because it lacks guidance knowledge without considering the decision attributes.

The supervised discretization algorithms are applied to a binary decision tree algorithm CART to evaluate the classification accuracy performance of our proposed algorithm. Table 15 shows results of this comparison of various discretization algorithms on different datasets. It can be seen that our proposed MSE algorithm appears good classification accuracy, especially for six datasets. This is because we divide the dataset into appropriate scales to obtain candidate cut point sets with different manifestations. Therefore, these cut point sets can better reflect the essential characteristics of the research objects, thereby improving the classification accuracy.

Tables 16, 17 and 18 reveals the impacts of the ten discretization algorithmths on three commonly used classification algorithms. It can be seen from the experimental results MSE significantly improves classification accuracy, especially for naive Bayes (see Table 17) and support vector machines (see Table 18), the classification accuracy is significantly improved. Compared with other types of discretization algorithms, MSE is more suitable for datasets with a larger number of class labels, a larger amount of data, and uneven distribution. For this type of dataset, the candidate cut point set selected by MSE can get a wider range of data, thereby finding more valuable cut points. For this kind of data, the candidate cut point set selected by MSE can get a wider range of data, thereby finding more valuable cut points. In other cases, MSE can also maintain relatively stable classification accuracy compared with other discretization algorithms.

Table 19 shows the classification accuracy on the 1R classification algorithm, which is the simplest classification algorithm. 1R is to constructs rules for each feature in a dataset based on a single feature, that is 1-rules. As

Table 18 Impacts on classification accuracy of SVM

Algorithms	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality	Rank
MSE	**27.32**	85.71	93.46	**95.33**	98.13	**99.18**	88.56	99.73	**86.64**	53.58	2.35
EW	25.64	63.23	92.89	90.67	98.11	97.65	87.16	89.55	82.46	50.04	6.1
EF	25.06	64.13	92.32	91.33	98.05	97.33	87.49	89.76	85.36	53.68	5.3
KMeans	26.17	50.22	88.9	90.67	78.72	93.96	86.82	99.07	77.06	49.96	7.6
IEM	27.29	**90**	92.31	94.0	98.13	99.16	88.64	99.74	86.56	**54.49**	2.7
CAIM	26.31	75.80	**94.29**	93.33	**98.38**	99.08	**88.81**	99.85	83.64	52.27	2.8
CACC	23.49	65.91	92.32	93.33	96.74	96.78	29.14	**99.88**	35.22	45.63	6.7
UrCAIM	26.6	70.54	94.02	94	97.94	97.97	87.07	99.78	82.66	53.39	3.95
TSD	24.97	51.49	85.76	93.33	10.39	89.33	57.79	99.77	72.98	51.94	7.8

Table 19 Impacts on classification accuracy of OneR

Algorithms	Abalone	Glass	Ionosphere	Iris	Optdigit	Pendigits	Satellite	Shuttle	Waveform	Winequality	Rank
MSE	25.9	92.6	81.82	**97.4**	24.78	35.8	60.66	89.48	54.7	45.96	4.55
EW	24.3	81.48	80.68	86.8	23.77	32.64	48.47	87.5	55	49.96	6.1
EF	23.5	80.76	80.11	73.7	23.34	32.18	51.34	87.3	54.4	49.06	7.4
KMeans	24.21	81.48	82.95	100	24.48	32.5	51.46	85.25	54.2	46.53	6.1
IEM	27.17	**100**	**86.36**	97.4	25.48	**39.7**	**60.97**	**94.89**	55.5	47.76	2.15
CAIM	**30.52**	**100**	84.1	97.37	**26.19**	38.46	60.85	94.87	**55.92**	**50.45**	2.0
CACC	23.14	**100**	86.36	97.37	24.84	36.0	32.7	92.7	55.52	45.8	4.85
UrCAIM	24.4	88.89	**86.36**	97.37	25.05	35.63	58.11	92.7	54.56	45.8	4.85
TSD	24.21	62.96	63.63	97.37	9.7	26.1	50.4	89.27	55.6	47.18	7.0

can be seen from Table 19, the CAIM gains the optimal performance. Because the CAIM algorithm finds the most number of points in the set for data partitioning, which is similar to the 1R algorithm. However, the selection of cut points by the MSE algorithm is based on information entropy and MDLPC criterion, which may result in more data division points and relatively fewer datasets. Therefore, the performance of classification accuracy is slightly unsatisfactory. But we cannot deny the validity of MSE because 1R algorithm is only suitable for the case where only focuses on one attribute.

In general, CAIM, CACC and UrCAIM belong to the discretization algorithms whose class attributes are interdependent. And the latter two algorithms are the improvement of CAIM algorithm, that is, they maximize the class attribute interdependence and calculate the best interval according to their own criteria. It can be seen that CAIM and UrCAIM algorithms have similar Ranks, and UrCAIM performs best on two classification algorithms and CAIM performs best on one classification algorithm for the ionosphere dataset. The ionosphere dataset contains many conditional attributes, and the number of class labels is only two. The discretization algorithm based on the interdependence of class attributes performs better in this type of feature distribution dataset.

MSE, IEM and TSD algorithms all use information entropy in the selection of interval points. We found that IEM and MSE algorithms have similar ranks, and they perform well in most datasets. Because they can obtain appropriate discrete interval number that are more in line with the data distribution feature by adopting the MDLPC standard. In addition, we can also see that the classification accuracy on the two datasets abalone and winequality is relatively low. Because the abalone dataset contains up to 28 class labels, it is difficult to effectively distinguish the category to which the attribute belongs in the discretization process. In particular, the MSE algorithm has higher classification accuracy than other discrete algorithms. Because it divides the candidate set more widely. For the winequality dataset, the low classification accuracy is mainly due to its large attribute value difference.

7 Conclusion and future work

In this study, we developed a supervised, top-down, static discretization algorithm called MSE, which addresses to balance the running time and classification accuracy. The algorithm performs reasonable multi-scale partitioning on the dataset and can generate the smallest candidate cut for a given continuous attribute. For the evaluation of each best candidate cut point, the MDLPC criterion is used to make the selection of the cut point more objective and reasonable. We verified the performance of our proposed algorithm through extensive experiments by comparing five classic classification algorithms and nine discretization algorithms on 10 UCI datasets. The evidence shows our MSE exhibits higher execution efficiency than that of other supervised discretization algorithms and better prediction classification accuracy for the five classification algorithms.

The importance of attributes will be considered in our future research work, as well as the relationship between attributes. According to the importance of the attributes, unnecessary attributes are eliminated by setting a reasonable weight value for each attribute to further improve the running time of the algorithm. Besides, in recent years, a big evolution of information technology has brought a sudden growth in data size. Such big data are not only large in size but also complex-structured. Therefore, distributed and parallel computing based on cluster environment are widely to adopted to discretization process.

Funding This work is supported by the National Natural Science Foundation of P.R. China (No.61602335, 61876122), Natural Science Foundation of Shanxi Province, P. R. China (No.201901D211302), Taiyuan University of Science and Technology Scientific Research Initial Funding of Shanxi Province, P. R. China (No.20172017), and

Scientific and Technological Innovation Team of Shanxi Province, P. R. China (No. 201805D131007).

Compliance with Ethical Standards

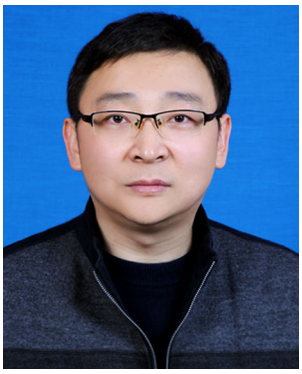
Conflict of interests The authors declare that we have no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

1. Bache K, Lichman M (1998) Uci repository of machine learning databases http://archive.ics.uci.edu/ml
2. Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. belmont, ca: Wadsworth. Int Group 432:151–166
3. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
4. Cano A, Luna JM, Gibaja EL, Ventura S (2016a) Laim discretization for multi-label data. Inf Sci 330:370–384
5. Cano A, Nguyen DT, Ventura S, Cios KJ (2016b) ur-caim: improved caim discretization for unbalanced and balanced data. Soft Comput 20(1):173–188
6. Cao F, Tang C, Zhang J (2017) Algorithm of continuous attribute discretization based on binary ant colony and rough sets. Comput Sci 44(9):222–226
7. Chang CC, Lin CJ (2011) Libsvm: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):1–27
8. Chmielewski MR, Grzymala-Busse JW (1996) Global discretization of continuous attributes as preprocessing for machine learning. Int J Approx Reason 15(4):319–331
9. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
10. Fayyad U, Irani K (1993) Multi-interval discretization of continuous-valued attributes for classification learning
11. Garcia S, Luengo J, Sáez JA, Lopez V, Herrera F (2012) A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. IEEE Trans Knowl Data Eng 25(4):734–750
12. Han Y, Zhao S, Liu M, Luo Y, Ding Y (2016) Multi-scale clustering mining algorithm. Comput Sci 43(8):244–248
13. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6(2):65–70
14. Holte RC (1993) Very simple classification rules perform well on most commonly used datasets. Mach Learn 11(1):63–90
15. Jiang F, Sui Y (2015) A novel approach for discretization of continuous attributes in rough set theory. Knowl-Based Syst 73:324–334
16. John GH, Langley P (2013) Estimating continuous distributions in bayesian classifiers. arXiv:13024964, pp 338–345
17. Kerber R (1992) Chimerge: Discretization of numeric attributes. In: Proceedings of the tenth national conference on Artificial intelligence, pp 123–128
18. Kurgan LA, Cios KJ (2004) Caim discretization algorithm. IEEE Trans Knowl Data Eng 16(2):145–153
19. Li C, Zhao S, Zhao J, Gao L, Chi Y (2017) Scaling-up algorithm of multi-scale association rules. Comput Sci 44(08):285–289
20. Liu X, Jiang H, Wu D (2013) Improved algorithm based on cacc for discretization of continuous data [j]. Computer Engineering 4
21. Liu M, Zhao S, Min C (2015) Scaling-up mining algorithm of multi-scale association rules mining. Appl Res Comput 32(10):2924–2929
22. Min H (2009) A global discretization and attribute reduction algorithm based on k-means clustering and rough sets theory. In: 2009 Second international symposium on knowledge acquisition and modeling, vol 2. IEEE, pp 92–95
23. Ramírez-Gallego S, García S et al (2016) Data discretization: taxonomy and big data challenge. Wiley Interdiscip Rev Data Min Knowl Discov 6(1):5–21
24. Sang Y, Li K, Shen Y (2010) Ebda: An effective bottom-up discretization algorithm for continuous attributes. In: 2010 10th IEEE International Conference on Computer and Information Technology. IEEE, pp 2455–2462
25. Shi H, Fu J (2005) A global discretization method based on rough sets. In: 2005 International conference on machine learning and cybernetics, vol 5. IEEE, pp 3053–3057
26. Thaiphan R, Phetkaew T (2018) Comparative analysis of discretization algorithms on decision tree. In: 2018 IEEE/ACIS 17Th international conference on computer and information science (ICIS). IEEE, pp 63–67
27. Tsai CJ, Lee CI, Yang WP (2008) A discretization algorithm based on class-attribute contingency coefficient. Inf Sci 178(3):714–731
28. Wen LY, Min F, Wang SY (2017) A two-stage discretization algorithm based on information entropy. Appl Intell 47(4):1169–1185
29. Wong AK, Chiu DK (1987) Synthesizing statistical knowledge from incomplete mixed-mode data. IEEE Transactions on Pattern Analysis and Machine Intelligence (6):796–805
30. Wu X, Kumar V (2009) The top ten algorithms in data mining. CRC Press, Boca Raton
31. Xie H, Cheng H, Niu D (2005) Discretization of continuous attributes in rough set theory based on information entropy. Chin J Comput 28(9):1570–1574
32. Xun Y, Zhang J, Qin X (2015) Fidoop: Parallel mining of frequent itemsets using mapreduce. IEEE Trans Syst Man Cybern Syst 46(3):313–325
33. Xun Y, Zhang J, Qin X, Zhao X (2016) Fidoop-dp: Data partitioning in frequent itemset mining on hadoop clusters. IEEE Trans Parallel Distrib Syst 28(1):101–114
34. Yang Y, Webb GI (2009) Discretization for naive-bayes learning: managing discretization bias and variance. Mach Learn 74(1):39–74
35. Zhang J, Li X et al (2012) A soft discretization method of celestial spectrum characteristic line based on fuzzy c-means clustering. Spectrosc Spectr Anal 32(5):1435–1438
36. Zhang J, Feng C, Tang C (2018) Discretization algorithm based on genetic algorithm and variable precision rough set. J Central China Normal Univ 52(03):322–328
37. Zhang F, Zhao S, Wu Y (2019) Data scaling method for multi-scale data mining. Computer Science
38. Zhao J, Zhou YH (2009) New heuristic method for data discretization based on rough set theory. Journal of China Universities of Posts and Teleconnunications (6):113–120

Dr. Yaling Xun received the BS in Computer Science and Technology from Harbin University of Science and Technology(HUST), and the MS and Ph.D. degrees from Taiyuan University of Science and Technology. She is currently a associate professor in the School of Computer Science and Technology at TYUST. She is a member of China Computer Federation (CCF). Her research interests include data mining and parallel computing.



Dr. Haifeng Yang is a professor Computer Application Technology, Taiyuan University of Science and Technology, China. He is the long-term member of the Institute for Intelligent Information and Data Mining. He is a member of China Computer Federation (CCF) and Chinese Astronomical Society (CAS). His research concerns the Data Mining and Machine Learning methods in the specific backgrounds especially for the astronomical Big Data.

MS. Qingxia Yin is a graduate student at Taiyuan University of Science and Technology. Her main research interests are data mining and parallel computing.

MS. Xiaohui Cui is a graduate student at Taiyuan University of Science and Technology. Her main research interests are data mining and parallel computing.

Dr. Jifu Zhang received the BS and MS in Computer Science and Technology from Hefei University of Tchnology, China, and the Ph.D. degree in Pattern Recognition and Intelligence Systems from Beijing Institute of Technology, in 1983, 1989, 2005, respectively. He is currently a Professor in the School of Computer Science and Technology at TYUST . His research interests include data mining, parallel and distributed computing, and artificial intelligence.