



An improved ID3 algorithm based on variable precision neighborhood rough sets

Caihui Liu¹ · Jianying Lai¹ · Bowen Lin¹ · Duoqian Miao²

Accepted: 2 June 2023 / Published online: 13 July 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

The classical ID3 decision tree algorithm cannot directly handle continuous data and has a poor classification effect. Moreover, most of the existing approaches use a single mechanism for node measurement, which is unfavorable for the construction of decision trees. In order to solve the above problems, we propose an improved ID3 algorithm (called DIGGI) based on variable precision neighborhood rough sets. First, we introduce the notions of variable precision neighborhood (VPN) equivalence relation and VPN equivalence granule, and probe some basic properties of these notions. Second, we give the model of VPN rough sets and propose two extended measures: the VPN information gain and the VPN Gini index. Finally, we construct a hybrid measure by using the VPN dependence to combine the above two extended measures, and adopt the VPN equivalence granule as the splitting rule of DIGGI. Experimental results show that DIGGI is effective and its accuracy is greatly improved compared with three traditional decision tree algorithms, the neighborhood decision tree (NDT) and variable precision neighborhood decision tree (VPNDT) proposed in the latest literature.

Keywords Variable precision neighborhood rough sets · Decision tree · Attribute dependence · Uncertainty measure

1 Introduction

As an significant data analysis method, classification [1] plays significant roles in the task of data mining. In contrast to other classification algorithms [2, 3], the decision tree algorithms possess the advantages of fast classification, high accuracy, and good interpretability. Decision tree is one of the algorithms widely applied in inductive reasoning at present. It is built with the top-down recursive method, choosing the best feature as the current node, and branches down from the node according to the different values of the feature.

Leaf nodes are classes to be divided. By now, scholars have proposed numbers of improvement decision tree algorithms. For instance, Breiman et al. [4] proposed the CART algorithm, which uses the Gini index to select splitting attributes. Moreover, Quinlan [5] proposed the well-known ID3 algorithm. ID3 is one of the classical decision tree algorithms at present and it adopts the information gain as the measure of nodes. However, this measure tends to select attributes with more values, which may generate a wide decision tree. To solve this problem, Quinlan [6] presented an improved algorithm (called C4.5) to induce decision trees, which uses the information gain ratio to select the optimal attribute division [7, 8].

Besides the above-mentioned shortcomings of ID3, there are also some other serious problems, e.g., ID3 can only deal with discrete data. When using this algorithm to deal with continuous data, we should first discretize the continuous data. Discretization will contribute to the loss of information, because the membership degree of numerical discretized values is not considered. In addition, when we face large-scale data sets, the log function will be called many times because we need to calculate the information entropy, and the computation amount will greatly increase. To solve the above problems, researchers have proposed many methods

✉ Caihui Liu
liucaihui@gnnu.edu.cn

Jianying Lai
1421631021@qq.com

Bowen Lin
641017193@qq.com

Duoqian Miao
dqmiao@tongji.edu.cn

¹ Department of Mathematics and Computer Science, Gannan Normal University, Ganzhou 341000, Jiangxi, China

² Department of Computer Science and Technology, Tongji University, Shanghai 201804, China

to improve decision tree. For instance, to overcome the problem of multivalued attribute bias, Liang et al. [9] presented a new attribute weighting method by using the simulation conditional probability to calculate the correlation between conditional and decision attributes. They linked information gain and attribute weights as new splitting criterion. Wang et al. [10] proposed a new decision tree optimization algorithm. Firstly, Gini index is used to improve the definition of boundary points and the calculation formula of information entropy in Fayyad theorem, so as to reduce the calculation times of information entropy in C4.5 algorithm. Then, CFS algorithm is introduced to optimize the calculation of information gain ratio, and the effectiveness of the algorithm is verified on Spark platform. Panhalkar et al. [11] proposed an African buffalo optimization algorithm suitable for solving complex problems to improve the decision tree and create a more efficient optimal decision tree.

The classical rough set model suggested by Pawlak [12] considered the whole research object as a universe. Information granules are obtained by dividing the universe of discourse into several proprietary equivalence classes using equivalence relations, which are used to describe arbitrary concepts in the universe. By now, the rough set theory has been widely used in classification, gene selection, attribute reduction and feature selection, etc. Some studies have shown that the decision method based on rough sets can flexibly deal with a variety of decision tables and is suitable for big data, but the output result is relatively complex. While decision tree generally deals with decision tables with multiple condition attributes and single decision attribute, the output result is relatively simple. Hence, by combining decision tree with rough sets, we can avoid the computational complexity caused by the logarithm operation and the bias problem of multivalued attributes. Wei et al. [13] discussed the combination of rough sets and decision tree. Wang et al. [14] proposed a concept of consistency based on rough sets and used it as a method of attribute selection. Moreover, they used this concept to solve the problems of multi-value bias of attribute values and huge amount of calculation in decision tree.

With the development of rough set theory, we gradually find that the classical rough set theory is no longer suitable for the application of the real world, and its limitations are more obvious. In order to solve this dilemma, scholars began to explore the generalization model of rough sets, and then the improved decision tree based on rough sets also turned to the use of more adaptable generalization models. Neighborhood rough set [15–17] is a generalization model of classical rough set. It replaces the strict equivalence relation with neighborhood relation, which greatly broadens its application scenario. At the same time, the problem that Pawlak rough set cannot deal with continuous data is also solved on this extended model, which reduces the problem of information loss caused by data discretization. Xie et al. [18]

combined neighborhood rough sets with decision trees and designed an algorithm to solve the shortcoming of ID3 (i.e., it can only deal with discrete data) based on a neighborhood equivalence relation. In order to reduce the influence of noise on the construction of decision trees, the combination of variable precision rough sets [19–21] and decision trees were explored. Xie et al. considering the strict equivalence relation on the basis of the previous part, and then proposed an improved decision tree algorithm based on similarity relation, using similarity relation instead of equivalence relation, induced a variable precision decision tree algorithm, improved the classification accuracy. Liu et al. [23] found that the author of the previous literature ignored the geometric structure of the neighborhood system when considering the algebraic structure of the neighborhood system, resulting in unsatisfactory transitivity requirements, so they proposed solutions from a geometric perspective and combined with decision trees.

We observe that attribute dependence, information gain and Gini index are the most widely used splitting criteria in all literatures. Attribute dependency can best show the importance of conditional attributes to decision attributes. The information gain and gini index are two popular measures of nodes in decision trees, where the former has better classification ability for handling the chaotic data, while the latter is more effective in handling the pure data. Most of the existing decision tree algorithms use one of these metrics as a splitting criterion. If these metrics can be combined in a linear way to construct new metrics, can the classification accuracy of the decision tree be improved? There are few studies on the mixed metric, In [24], Xie et al. used the adaptive weighting of information gain and gini index to construct a hybrid measure and in [25], Jain et al. used constant weight to carry out linear combination integration. With the development of science and technology, uncertainty measurement is an important criterion in big data analysis. The emergence of hybrid measurement may provide people with new ideas. This paper attempts to combine these three metrics in a linear way to construct a new uncertainty measurement criterion to improve the classification performance of decision trees. On the basis of this idea, in this paper, we will propose two extended measures: the VPN Gini index, and the VPN information gain. Moreover, we will construct a hybrid measure by using the VPN dependence to combine the above two extended measures. This hybrid measure will be used to select nodes during the construction of decision trees. Experiments were confirmed on multiple UCI data sets, where the accuracy and the number of leaf nodes were used as the evaluation metrics. We verified the effectiveness of our algorithm by comparing it with the traditional CART algorithm, ID3 algorithm, C4.5 algorithm and the algorithms proposed in the latest literature [22, 23].

The main contributions of this work include: (1) The construction of neighborhood similarity by using Dice similarity can relax the strict equivalence relation between any two neighborhood granules. (2) Based on the neighborhood similarity and variable precision threshold, the equivalence relation of VPN is induced and a new VPN rough set model is proposed. The Gini index and information gain are extended to the variable precision neighborhood Gini index and variable precision neighborhood information gain by means of the VPN equivalence relation. (3) A new splitting criterion is defined by combining the variable precision dependence with the variable precision neighborhood Gini index and variable precision neighborhood information gain.

The remainder of this paper is structured as follows. Section 2 reviews the classical ID3 and CART algorithms, and introduces the model of neighborhood rough sets (NRSs). In Section 3, we propose the VPN equivalence relation and the VPN rough sets. Based on the VPN equivalence relation, the VPN information gain and the VPN Gini index are proposed. Moreover, a hybrid measure is constructed to improve the performance of the classical decision tree algorithms. In Section 4, we verify the effectiveness of our algorithm. Finally, we make a summary and some discussion in Section 5.

2 Preliminaries

This section reviews some basic definitions about ID3 algorithm, CART algorithm and neighborhood rough set model. Throughout the paper, U is called universe to represent a non-empty finite set.

2.1 Decision tree

We describe decision tree as a tree classifier, which is constructed from the top down and consists of nodes and directed edges. Nodes in a decision tree can be divided into internal nodes and leaf nodes, where each internal node represents a feature or attribute, and each leaf node represents a class. ID3, C4.5 and CART are three well-known decision tree algorithms, where C4.5 and CART are the improvements of ID3. The above three algorithms have been widely used in different fields because of their advantages such as good interpretability. This subsection mainly introduces ID3 and CART.

Definition 1 [5] Let $Dt = (U, C \cup D)$ be a decision table, where C is represented the set of conditional attributes and D is called the set of decision attributes. Let $U/D = \{D_1, D_2, \dots, D_n\}$ be the decision classification, the informa-

tion entropy $H(D)$ of U/D is expressed as follows:

$$H(D) = - \sum_{i=1}^n \frac{|D_i|}{|U|} \log \frac{|D_i|}{|U|}. \tag{1}$$

where for each $1 \leq i \leq n$, $\frac{|D_i|}{|U|}$ represents the probability of a sample falls into the decision class D_i .

For any $A \subseteq C$, let $U/A = \{A_1, A_2, \dots, A_m\}$, the conditional entropy $H(D|A)$ of decision attribute D with respect to conditional attribute A is defined as follows:

$$H(D|A) = \sum_{j=1}^m \frac{|A_j|}{|U|} H(A_j). \tag{2}$$

where for each $1 \leq j \leq m$, $H(A_j) = - \sum_{i=1}^n \frac{|A_j \cap D_i|}{|A_j|} \log \frac{|A_j \cap D_i|}{|A_j|}$. Therefore, the information gain of attribute subset A is

$$Gain(A) = H(D) - H(D|A). \tag{3}$$

ID3 algorithm needs to calculate the information entropy of each attribute, and selects the attribute with the largest information gain as the splitting attribute. The decision tree generated by ID3 is easy to understand for us and it shows the potential information behind the data clearly. Although the idea of ID3 is simple, it tends to prefer attributes with more values. There are many improved algorithms for ID3, and CART is one of them.

Definition 2 [4] Let $Dt = (U, C \cup D)$ be a decision table, and let $U/D = \{D_1, D_2, \dots, D_n\}$ be the decision classification, the gini index $Gini(D)$ of U/D is defined as follows:

$$\begin{aligned} Gini(D) &= \sum_{i=1}^n \frac{|D_i|}{|U|} \left(1 - \frac{|D_i|}{|U|} \right) \\ &= 1 - \sum_{i=1}^n \left(\frac{|D_i|}{|U|} \right)^2. \end{aligned} \tag{4}$$

For any $A \subseteq C$, let $U/A = \{A_1, A_2, \dots, A_m\}$, the gini index $Gini(D, A)$ of D on A is defined as follows:

$$Gini(D, A) = \sum_{j=1}^m \frac{|A_j|}{|U|} Gini(A_j). \tag{5}$$

where for each $1 \leq j \leq m$, $Gini(A_j) = 1 - \sum_{i=1}^n \left(\frac{|A_j \cap D_i|}{|A_j|} \right)^2$.

CART selects the attribute with the smallest Gini index as the splitting attribute, and builds the corresponding regression tree or classification tree according to whether the value of category attribute is continuous or discrete.

2.2 Neighborhood rough set model

As we all know that the classical rough set model needs to use the step of discretization to preprocess the continuous data, which is easy to generate to information loss. To avoid this problem, the neighborhood rough set model is proposed and attracts much attention. In the following, we review some basic notions in neighborhood rough sets.

Definition 3 [16] Let $Dt = (U, C \cup D)$ be a decision table, where U is a non-empty finite set (called universe) with samples $\{ay_1, ay_2, \dots, ay_n\}$, $C = \{c_1, c_2, \dots, c_m\}$ is the set of condition attributes, and D represents the set of decision attributes. For any $ay_i \in U$ and $A \subseteq C$, the neighborhood $\delta_A(ay_i)$ of ay_i on the attribute subspace A is defined as:

$$\delta_A(ay_i) = \{ay_j \in U \mid \Delta^A(ay_i, ay_j) \leq \delta\}. \tag{6}$$

where Δ is a distance function, and for any $ay_1, ay_2, ay_3 \in U$, Δ satisfies that:

- (1) $\Delta(ay_1, ay_2) \geq 0$, $\Delta(ay_1, ay_2) = 0$ only when $ay_1 = ay_2$;
- (2) $\Delta(ay_1, ay_2) = \Delta(ay_2, ay_1)$;
- (3) $\Delta(ay_1, ay_3) \leq \Delta(ay_1, ay_2) + \Delta(ay_2, ay_3)$.

Assume that $A = \{c_1, c_2, \dots, c_n\}$ is an attribute subset of C , for any two samples $ay_1, ay_2 \in U$, the Minkowsky distance $d_A(ay_1, ay_2)$ between ay_1 and ay_2 under A is defined as:

$$d_A(ay_1, ay_2) = \left(\sum_{i=1}^m |f(ay_1, c_i) - f(ay_2, c_i)|^p\right)^{1/p}. \tag{7}$$

where $f(ay_1, c_i)$ is the value of sample ay_1 on attribute c_i , $1 \leq i \leq m$.

When $p = 1$, the Minkowsky distance is called the Manhattan distance, it is called the Euclidean distance, and the Chebyshev distance when $p = 2$ and $p = \infty$. The NRSs model usually uses the Manhattan distance. It can be seen that when $\delta = 0$, the NRSs model will degenerate into the classical rough set model.

Definition 4 [16] Given a decision table $Dt = (U, C \cup D)$ and attribute subset $B \subseteq C$. For any $AY \subseteq U$, the upper approximation and the lower approximation of set AY with respect to B are defined as follows:

$$\begin{aligned} \overline{N_B}(AY) &= \{ay_i \in U \mid \delta(ay_i) \cap AY \neq \emptyset\}, \\ \underline{N_B}(AY) &= \{ay_i \in U \mid \delta(ay_i) \subseteq AY\}. \end{aligned} \tag{8}$$

Moreover, $B_N(AY) = \overline{N_B}(AY) - \underline{N_B}(AY)$ is called the boundary region of set AY with respect to B .

Definition 5 [16] Given a decision table $Dt = (U, C \cup D)$, let $U/D = \{D_1, D_2, \dots, D_n\}$ be the equivalence classes divided by decision attribute D on U . For any $B \subseteq C$, the upper approximation set $\overline{N_B}(D)$ and the lower approximation set $\underline{N_B}(D)$ of D with respect to A are defined as follows:

$$\begin{aligned} \overline{N_B}(D) &= \bigcup_{i=1}^n \overline{N_B}(D_i), \\ \underline{N_B}(D) &= \bigcup_{i=1}^n \underline{N_B}(D_i). \end{aligned} \tag{9}$$

Then we can get the positive region as follows,

$$POS_B(D) = \underline{N_B}(D). \tag{10}$$

Definition 6 [26] The Dice similarity between set Y_1 and set Y_2 is defined as:

$$D(Y_1, Y_2) = \frac{2 * |Y_1 \cap Y_2|}{|Y_1| + |Y_2|}. \tag{11}$$

Dice coefficient is a measure function used to describe the similarity of sets, which is mainly used to calculate the similarity of two given sets.

3 Variable precision neighborhood rough sets and measurement function

In this section, we first introduce the notions of VPN equivalence relation and VPN equivalence granule. Then we discuss the basic properties of them. At the same time, we propose a VPN rough set model and explore some of its digital characteristics. Based on the variable precision neighborhood equivalence relation, the Gini index is extended to the VPN Gini index, and the information gain is extended to the VPN information gain. Finally, a hybrid measure is established by using the VPN attribute dependence to combine the VPN Gini index and the VPN information gain.

3.1 Variable precision neighborhood rough sets

Definition 7 Let $NDS = (U, C \cup D, V, f, d, \delta)$ be a neighborhood decision system, and $\delta \in [0, 1]$ be a neighborhood parameter. For any $ay_1, ay_2 \in U$ and $A \subseteq C$, the similarity of neighborhood granules $\eta_A^\delta(ay_1)$ and $\eta_A^\delta(ay_2)$ is expressed as follows:

$$NV_A^\delta(ay_1, ay_2) = \frac{2 * |\eta_A^\delta(ay_1) \cap \eta_A^\delta(ay_2)|}{|\eta_A^\delta(ay_1)| + |\eta_A^\delta(ay_2)|}. \tag{12}$$

The above definition indicates a similarity relation between two neighborhood granules. Obviously, $NV_A^\delta(ay_1, ay_2) \in [0, 1]$.

Example 1 Let $U = \{bx_1, bx_2, bx_3, bx_4, bx_5, bx_6\}$ be a universe, and let $\eta_A^\delta(bx_1) = \{bx_1, bx_2, bx_5, bx_6\}$ and $\eta_A^\delta(bx_2) = \{bx_1, bx_2, bx_4, bx_6\}$ be two neighborhood granules. The granule similarity between $\eta_A^\delta(bx_1)$ and $\eta_A^\delta(bx_2)$ is $NV_A^\delta(bx_1, bx_2) = \frac{2 * |\eta_A^\delta(bx_1) \cap \eta_A^\delta(bx_2)|}{|\eta_A^\delta(bx_1)| + |\eta_A^\delta(bx_2)|} = \frac{2 * |\{bx_1, bx_2, bx_6\}|}{|\{bx_1, bx_2, bx_5, bx_6\}| + |\{bx_1, bx_2, bx_4, bx_6\}|} = 0.75$.

Let $NDS = (U, C \cup D, V, f, d, \delta)$ be a neighborhood decision system, for any $ay \in U$ and $A \subseteq C$, let $\eta_A^\delta(ay)$ be the condition neighborhood granule of ay on A , and $\eta_D^\delta(ay)$ be the decision neighborhood granule of ay on D . The granule similarity between the condition neighborhood granule and the decision neighborhood granule is

$$NV(\eta_A^\delta(ay), \eta_D^\delta(ay)) = \frac{2 * |\eta_A^\delta(ay) \cap \eta_D^\delta(ay)|}{|\eta_A^\delta(ay)| + |\eta_D^\delta(ay)|} \tag{13}$$

Definition 8 Given a decision table $Dt = (U, C \cup D)$, for any $A \subseteq C$, the neighborhood equivalence relation on A is expressed as follows:

$$NER_A^\delta = \{(ay_1, ay_2) \in U \times U | \eta_A^\delta(ay_1) = \eta_A^\delta(ay_2)\} \tag{14}$$

where $\delta \in [0, 1]$ is a given neighborhood radius, and $\eta_A^\delta(*)$ represents the δ -neighborhood of sample $*$ on A . The neighborhood equivalence partition induced by the neighborhood equivalence relation ENR_A^δ is defined as follows:

$$U / NER_A^\delta = \{[ay]_{NER_A^\delta} | ay \in U\} = \{Y_1^A, Y_2^A, \dots, Y_n^A\} \tag{15}$$

where $Y_r^A = \{ay_i \in U | \eta_A^\delta(ay_i) = \eta_A^\delta(ay_j)\}$, $ay_j \in Y_r^A$ ($r = 1, 2, \dots, n$). $[ay]_{NER_A^\delta}$ represents equivalent granule and $[ay]_{NER_A^\delta} = \{ay_i \in U | (ay, ay_i) \in NER_A^\delta\}$.

Example 2 Given a dataset listed in Table 1, according to the definition of neighborhood equivalence relation, we can divide the samples in the dataset into a set of equivalence classes (ECs) by a given attribute subset. For instance, the ECs induced by the attribute subset $\{a_1\}$ are $\{bx_1, bx_4\}, \{bx_2, bx_3\}, \{bx_5\}$. The ECs induced by the attribute subset $\{a_2\}$ are $\{bx_1, bx_2\}, \{bx_3\}, \{bx_4\}, \{bx_5\}$. The ECs induced by the attribute subset $\{a_3\}$ are $\{bx_1\}, \{bx_2, bx_3\}, \{bx_4\}, \{bx_5\}$.

From the definition of neighborhood equivalence relation, we can easily see that only when the neighborhood of two samples is exactly equal, the two samples will be judged to be equivalent. The above definition is undoubtedly strict, hence we propose the variable precision neighborhood equivalence relation.

Table 1 A given dataset

U	a_1	a_2	a_3	D
bx_1	0.6	0.8	0.9	1
bx_2	0.7	1.0	0.7	2
bx_3	0.8	0.7	0.4	1
bx_4	0.5	0.4	0.6	1
bx_5	0.3	0.2	0.5	2

Definition 9 In $NDS = (U, C \cup D, V, f, d, \delta)$, for any $(ay_1, ay_2) \in U \times U$ and $A \subseteq C$, if ay_1 and ay_2 satisfy the following equation

$$VNER_A^{\delta, \alpha} = \{(ay_1, ay_2) \in U \times U | NV_A^\delta(ay_1, ay_2) \geq \alpha\} \tag{16}$$

then we consider that samples ay_1 and ay_2 are equivalent, where $\alpha \in [0.5, 1)$ is the variable precision threshold. Here, we call it VPN equivalence relation.

In real-world applications, it is difficult for us to find exactly equal samples of two neighborhoods. If the above strict equivalence relationship is adopted, it will be greatly limited in the actual application scenario and produce unsatisfactory results. Especially when we apply this equivalence relation to the decision tree, it may lead to the generation of a large decision tree, which is an unfavorable classification effect, so we must improve this equivalence relation (Table 2).

Definition 10 Let $\delta \in [0, 1]$ be the neighborhood radius, and α be the variable precision threshold, we obtain the VPN equivalence class based on the VPN equivalence relation as below.

$$U / VNER_A^{\delta, \alpha} = \{[ay]_{VNER_A^{\delta, \alpha}} | ay \in U\} = \{VX_1^A, VX_2^A, \dots, VX_n^A\} \tag{17}$$

where $VX_r^A = \{ay \in U | NV_A^\delta(ay_i, ay) \geq \alpha\}$; $ay_i \in VX_r^A$; $1 \leq r \leq n$.

Example 3 Here we use neighborhood similarity to calculate the similarity of two tuples above the attribute subset $\{a_1\}$ with the variable precision threshold $\alpha = 0.7$, the calculation results are shown in Tables 3, and 4 shows the results of equivalence classes.

Proposition 1 In $NDS = (U, C \cup D, V, f, d, \delta)$, α is the variable precision threshold,

1. If $\delta_1 \leq \delta_2$, then $U / VNER_A^{\delta_1, \alpha} \subseteq U / VNER_A^{\delta_2, \alpha}$;

Table 2 The sample neighborhoods of example decision table ($\delta = 0.3$)

attribute subset	bx_1	bx_2	bx_3	bx_4	bx_5
$\{a_1\}$	U	$\{bx_1, bx_2, bx_3, bx_4\}$	$\{bx_1, bx_2, bx_3, bx_4\}$	U	$\{bx_1, bx_4, bx_5\}$
$\{a_2\}$	$\{bx_1, bx_2, bx_3\}$	$\{bx_1, bx_2, bx_3\}$	$\{bx_1, bx_2, bx_3, bx_4\}$	$\{bx_3, bx_4, bx_5\}$	$\{bx_4, bx_5\}$
$\{a_3\}$	$\{bx_1, bx_2, bx_4\}$	$\{bx_1, bx_2, bx_3, bx_4\}$	$\{bx_1, bx_2, bx_3, bx_4\}$	U	$\{bx_1, bx_4, bx_5\}$

2. If $\alpha_1 \leq \alpha_2$, then $U/VNER_A^{(\delta, \alpha_2)} \subseteq U/VNER_A^{(\delta, \alpha_1)}$.

Proof 1 1. If $\delta_1 \leq \delta_2$, for any $ay_1, ay_2 \in U$, obviously, we can obtain $\eta_A^{\delta_1}(ay_1) \subseteq \eta_A^{\delta_2}(ay_1)$. Then we can get $NV_A^{\delta_2}(ay_1, ay_2) \geq NV_A^{\delta_1}(ay_1, ay_2) \geq \alpha$, therefore, $U/VNER_A^{(\delta_1, \alpha)} \subseteq U/VNER_A^{(\delta_2, \alpha)}$.

2. If $\alpha_1 \leq \alpha_2$, we can get $VNER_A^{(\delta, \alpha_2)} \subseteq VNER_A^{(\delta, \alpha_1)}$, therefore, we have $U/VNER_A^{(\delta, \alpha_2)} \subseteq U/VNER_A^{(\delta, \alpha_1)}$.

Proposition 1 mainly reflects the sizes of neighborhood granule and variable precision equivalence granule. According to Proposition 1, we can see that the size of a neighborhood granule increases with the increase of the neighborhood radius, which means that the larger the neighborhood radius is, the more samples are contained in the neighborhood granule. Variable precision threshold α can control the thickness of the variable precision neighborhood equivalence granule. A lower threshold makes it easier to assume that two samples are similar, whereas the higher the degree of similarity between samples can be judged they are equal. A larger variable precision threshold means that a single VPN equivalence granule is rougher, while a smaller one is more accurate.

Definition 11 In $NDS = (U, C \cup D, V, f, d, \delta)$, α is the variable precision threshold, for any $P \subseteq C$ and $ay \in U$, the VPN lower approximation and the VPN upper approximation are defined as follows:

$$\begin{aligned} \underline{R}_P(D)_\delta^\alpha &= \{ay \in U \mid NV(\eta_P^\delta(ay), \eta_D^\delta(ay)) \geq \alpha\}, 0.5 \leq \alpha < 1, \\ \overline{R}_P(D)_\delta^\alpha &= \{ay \in U \mid NV(\eta_P^\delta(ay), \eta_D^\delta(ay)) > 1 - \alpha\}. \end{aligned} \tag{18}$$

Table 3 The results of neighborhood similarity between samples

The two tuple	Neighborhood similarity	The two tuple	Neighborhood similarity
(1,2)	1.0	(2,4)	(0.8889)
(1,3)	0.8889	(2,5)	0.5714
(1,4)	1.0	(3,4)	0.8889
(1,5)	0.75	(3,5)	0.5714
(2,3)	1.0	(4,5)	0.75

The VPN lower approximation is also called the VPN positive region, which is denoted as:

$$POS_P(D)_\delta^\alpha = \underline{R}_P(D)_\delta^\alpha. \tag{19}$$

The VPN boundary region is defined as:

$$BN_P(D)_\delta^\alpha = \overline{R}_P(D)_\delta^\alpha - \underline{R}_P(D)_\delta^\alpha. \tag{20}$$

Obviously, for any α , $\underline{R}_P(D)_\delta^\alpha \subseteq \overline{R}_P(D)_\delta^\alpha$.

Proposition 2 In $NDS = (U, C \cup D, V, f, d, \delta)$, α is the variable precision threshold, if $A_1 \subseteq A_2 \subseteq C$, then $POS_{A_1}(D)_\delta^\alpha \subseteq POS_{A_2}(D)_\delta^\alpha$.

Proof 2 Given $A_1 \subseteq A_2$, ay and metric Δ , $\forall ay_i \in U$, $ay_i \in \delta_{A_1}(ay)$ if $ay_i \in \delta_{A_2}(ay)$ because $\Delta_{A_1}(ay_i, ay) \leq \Delta_{A_2}(ay_i, ay)$. Therefore, we have $\delta_{A_1}(ay) \supseteq \delta_{A_2}(ay)$ if $A_1 \subseteq A_2$. Assume $\delta_{A_1}(ay) \subseteq \underline{R}_{A_1}(D)_\delta^\alpha$, where D represents one of the decision classes, then we have $\delta_{A_2}(ay) \subseteq \underline{R}_{A_2}(D)_\delta^\alpha$. At the same time, it may exist ay_i , $\delta_{A_1}(ay_i) \not\subseteq \underline{R}_{A_1}(D)_\delta^\alpha$ and $\delta_{A_2}(ay_i) \subseteq \underline{R}_{A_2}(D)_\delta^\alpha$. Therefore, $\underline{R}_{A_1}(D)_\delta^\alpha \subseteq \underline{R}_{A_2}(D)_\delta^\alpha$. Assuming $D = \{D_1, D_2, \dots, D_m\}$, we have $\underline{R}_{A_1}(D_1)_\delta^\alpha \subseteq \underline{R}_{A_2}(D_1)_\delta^\alpha, \dots, \underline{R}_{A_1}(D_m)_\delta^\alpha \subseteq \underline{R}_{A_2}(D_m)_\delta^\alpha$. $POS_{A_1}(D)_\delta^\alpha = \bigcup_{i=1}^m \underline{R}_{A_1}(D_i)_\delta^\alpha$; $POS_{A_2}(D)_\delta^\alpha = \bigcup_{i=1}^m \underline{R}_{A_2}(D_i)_\delta^\alpha$, so we can obtain $POS_{A_1}(D)_\delta^\alpha \subseteq POS_{A_2}(D)_\delta^\alpha$.

Definition 12 In $NDS = (U, C \cup D, V, f, d, \delta)$, α is the variable precision threshold, for any $P \subseteq C$, the degree of dependence of D on P is defined as:

$$\gamma_P(D)_\delta^\alpha = \frac{|POS_P(D)_\delta^\alpha|}{|U|}. \tag{21}$$

The degree of knowledge dependence indicates the degree of dependence of knowledge D on knowledge P .

Table 4 The results of ECs

Samples	Equivalent calsses
bx_1	$\{bx_2, bx_3, bx_4, bx_5\}$
bx_2	$\{bx_1, bx_3, bx_4\}$
bx_3	$\{bx_1, bx_2, bx_4, bx_5\}$
bx_4	$\{bx_1, bx_2, bx_3\}$
bx_5	$\{bx_1, bx_4\}$

Proposition 3 In $NDS = (U, C \cup D, V, f, d, \delta)$, α is the variable precision threshold, for any $A \subseteq C, U/VENR_A^{(\delta, \alpha)}$ is the VPN equivalence granule,

1. If $\alpha = 1$, then $\gamma_{U/VNER_A^{(\delta, \alpha)}}(D) = \gamma_{U/NER_A^\delta}(D)$.
2. If $\alpha = 1$ and $\delta = 0$, then $\gamma_{U/VENR_A^{(\delta, \alpha)}}(D) = \gamma_{U/RA}(D)$.

Proof 3 1. If $\alpha = 1$, the VPN equivalence granule becomes the strict neighborhood equivalence granule and the VPN rough sets degenerates into the neighborhood rough sets. Therefore, the VPN dependence becomes the neighborhood attribute dependence.

2. If $\alpha = 1$ and $\delta = 0$, the VPN equivalence granule degenerates into the ordinary Pawlak’s equivalence granules and the VPN rough sets degenerates into the Pawlak’s rough sets. Therefore, the VPN dependence naturally becomes the classical Pawlak’s dependence.

Proposition 3 shows that the VPN rough sets are associated with classical rough sets and NRSs. When we set the neighborhood radius δ to 0 and the threshold α to 1, the VPN equivalence relation degenerates into the ordinary Pawlak’s equivalence relation, that is, the variable precision neighborhood rough set model becomes the classical rough set model. Therefore, the VPN attribute dependence becomes the Pawlak’s attribute dependence. When the threshold α is set to 1 and the neighborhood radius δ is retained, it degenerates into the NRSs model. So the VPN attribute dependence becomes the neighborhood attribute dependence.

Proposition 4 In $NDS = (U, C \cup D, V, f, d, \delta)$, α is the variable precision threshold, if $A_1 \subseteq A_2 \subseteq C$, then we have $\gamma_{A_1}(D)_\delta^\alpha \leq \gamma_{A_2}(D)_\delta^\alpha$.

Proof 4 According to Proposition 2, we can obtain that $POS_{A_1}(D)_\delta^\alpha \subseteq POS_{A_2}(D)_\delta^\alpha$, so we have that $\gamma_{A_1}(D)_\delta^\alpha \leq \gamma_{A_2}(D)_\delta^\alpha$.

Proposition 4 illustrates the monotonicity of variable precision neighborhood dependence. From Proposition 2, we can obtain that if we continue to add attributes to the condition attribute subset, then the positive domain will become larger. When U remains the same, the VPN dependence increases with the increase of the positive domain, showing a monotonically increasing trend.

3.2 Variable precision neighborhood Gini index and variable precision neighborhood information gain

In the classical CART algorithm, the gini index is used for measuring the nodes of the decision tree, and the attribute with the smallest Gini index is selected as the splitting attribute. In this subsection, the Gini index is extended to the variable precision neighborhood Gini index.

Definition 13 In $NDS = (U, C \cup D, V, f, d, \delta)$, α is the variable precision threshold. Let $\{VX_1^A, VX_2^A, \dots, VX_n^A\}$ be the VPN equivalence partition induced by $U/VENR_A^{(\delta, \alpha)}$, for any $A \subseteq C$, the VPN Gini index is defined as follows.

$$VGini(U/VENR_A^{(\delta, \alpha)}) = \sum_{i=1}^n \frac{|VX_i^A|}{|U|} \left(1 - \frac{|VX_i^A|}{|U|}\right) = \sum_j^m \sum_i^n \frac{|VX_i^A \cap D_j|}{|U|} \left(1 - \frac{|VX_i^A|}{|U|}\right). \tag{22}$$

where $U/D = \{D_1, D_2, \dots, D_m\}$, and $VX_k^A \in U/VENR_A^{(\delta, \alpha)}$ represents a granule in the equivalence granule of variable precision neighborhood. We can see that the above equation is the sum of information carried by each granule, so the neighborhood Gini index of a single granule can be constructed as follows:

$$VGini(VX_k^A) = \frac{|VX_k^A|}{|U|} \left(1 - \frac{|VX_k^A|}{|U|}\right). \tag{23}$$

Therefore, the neighborhood Gini index of D with respect to $VX_k^A \in U/VENR_A^{(\delta, \alpha)}$ is expressed as follows:

$$VGini(D, VX_k^A) = \sum_{j=1}^m \frac{|VX_k^A \cap D_j|}{|VX_k^A|} \left(1 - \frac{|VX_k^A \cap D_j|}{|U|}\right). \tag{24}$$

The Gini index of variable precision neighborhood of D with respect to $A \subseteq C$ is defined as follows:

$$VGini(D, U/VENR_A^{(\delta, \alpha)}) = \sum_{i=1}^n \frac{|VX_i^A|}{|U|} VGini(D, VX_i^A). \tag{25}$$

where $\frac{|VX_i^A|}{|U|}$ represents the weight coefficient of D on the neighborhood Gini index of a single granule.

Proposition 5 Let $U/VENR_A^{(\delta, \alpha)}$ be the VPN equivalence granule, when $\delta = 0$ and $\alpha = 1$,

1. $VGini(U/VENR_A^{(\delta, \alpha)}) = Gini(A)$.
2. $VGini(D, U/VENR_A^{(\delta, \alpha)}) = Gini(D, A)$.

Proof 5 When $\delta = 0$ and $\alpha = 1$, the VPN equivalence relation will degenerate into a general equivalence relation, and the VPN equivalence granule will degenerate into a general

equivalence granule structure, therefore, the VPN Gini index will also become the general *Gini* index.

The VPN Gini index is a generalization of the Gini index, and their properties are basically the same. When the neighborhood radius δ and the variable precision threshold α are set to 0 and 1, the VPN Gini index degenerates into the familiar Gini index.

Definition 14 In $NDS = (U, C \cup D, V, f, d, \delta)$, α is the variable precision threshold. Assume that the VPN equivalence partition is $U/VENR_A^{(\delta, \alpha)} = \{VX_1^A, VX_2^A, \dots, VX_n^A\}$, for any $A \subseteq C$, the VPN information entropy of A and the VPN information entropy of D are defined as follows:

$$\begin{aligned} VH_\delta^\alpha(A) &= - \sum_{i=1}^n \frac{|VX_i^A|}{|U|} \log_2 \frac{|VX_i^A|}{|U|} \\ &= - \sum_j^m \sum_i^n \frac{|VX_i^A \cap D_j|}{|U|} \log_2 \frac{|VX_i^A|}{|U|}. \end{aligned} \quad (26)$$

$$VH_\delta^\alpha(D)_{U/VENR_A^{(\delta, \alpha)}} = - \sum_j^m \sum_i^n \frac{|VX_i^A \cap D_j|}{|U|} \log_2 \frac{|D_j|}{|U|}. \quad (27)$$

Hence, we can obtain the VPN conditional entropy as follows:

$$VH_\delta^\alpha(D|A) = - \sum_j^m \sum_i^n \frac{|VX_i^A \cap D_j|}{|U|} \log_2 \frac{|VX_i^A \cap D_j|}{|VX_i^A|}. \quad (28)$$

Therefore, the VPN information gain is defined as follows:

$$VGAIN_\delta^\alpha(D, A) = VH_\delta^\alpha(D)_{U/VENR_A^{(\delta, \alpha)}} - VH_\delta^\alpha(D|A). \quad (29)$$

Proposition 6 When $\delta = 0$ and $\alpha = 1$, we have that $U/VENR_A^{(\delta, \alpha)} = U/IND(A)$ and $VH_\delta^\alpha(A) = H(A)$, then $VH_\delta^\alpha(D|A) = H(D|A)$ and $VGAIN_\delta^\alpha(D, A) = gain(D, A)$.

Proof 6 When $\delta = 0$ and $\alpha = 1$, the VPN equivalence relation will degenerate into the classical equivalence relation and the VPN information gain will also degenerate into the classical information gain.

3.3 The hybrid measure for selecting splitting attributes

As we all know, both information gain and gini index are two common measures used to select splitting attributes of decision trees. Information gain is derived from information entropy, which focuses on the representation of information. It is used to characterize the uncertainty in the data set and has a better ability to characterize the chaotic data. Normally, the attribute with the largest information gain will be selected as the current splitting attribute. Moreover, we select the attribute with the smallest gini index as the splitting attribute. Gini index mainly focuses on the algebraic representation and has better classification ability for the pure data. We can see that the above two measures are complementary, hence we can combine the advantages of the two measures and construct a hybrid measure for selecting splitting attributes.

In general, let $C = \{c_1, c_2, \dots, c_{|C|}\}$ be the set of condition attributes, for any $A \subseteq C$, the attribute increment chain $A_1 = \{c_1\} \subseteq A_2 = \{c_1, c_2\} \subseteq \dots \subseteq A_{|C|} = \{c_1, c_2, \dots, c_{|C|}\}$ is used to explore the rules of knowledge granularity, including the granulation monotonicity of uncertainty measure [27]. The information gain is characterized by granular monoadditivity, while the Gini index is characterized by granular non-monoadditivity [24], which are expressed as follows:

$$A_1 \subseteq A_2 \subseteq C \Rightarrow gain(A_1) \leq gain(A_2),$$

$$A_1 \subseteq A_2 \subseteq C \not\Rightarrow gini(A_1) \leq gini(A_2),$$

$$A_1 \subseteq A_2 \subseteq C \not\Rightarrow gini(A_1) \geq gini(A_2).$$

According to the above monotonicity analysis, we can see that the extended variable precision neighborhood information gain and the variable precision neighborhood Gini index have the same granulation monotonicity, which are expressed as follows:

$$A_1 \subseteq A_2 \subseteq C \Rightarrow VGain(A_1) \leq VGain(A_2),$$

$$A_1 \subseteq A_2 \subseteq C \not\Rightarrow VGini(A_1) \leq VGini(A_2),$$

$$A_1 \subseteq A_2 \subseteq C \not\Rightarrow VGini(A_1) \geq VGini(A_2).$$

Definition 15 By combining the variable precision neighborhood information gain and the variable precision neighborhood Gini index, we define a hybrid measure for selecting splitting attributes, which is given as follow:

$$\begin{aligned} VS(D, A) &= \varphi(A) * VGain(D, A) + (1 - \varphi(A)) \\ &\quad * VGini(D, A). \end{aligned} \quad (30)$$

where $\varphi(A) = \gamma_A(D)_\delta^\alpha$. We elect an attribute with the largest $VS(D, A)$ as the splitting attribute.

Proposition 7 The hybrid measure $VS(D, A)$ has the property of granulation non-monoadditivity, which is described as follows:

$$A_1 \subseteq A_2 \subseteq C \not\Rightarrow VS(A_1) \geq VS(A_2),$$

$$A_1 \subseteq A_2 \subseteq C \not\Rightarrow VS(A_1) \leq VS(A_2).$$

Attribute dependence can be used to measure the uncertainty of knowledge. From Definition 14, we can see that when $\varphi(A)$ increases, $VGain(D, A)$ becomes larger and $VGini(D, A)$ becomes smaller, hence $VS(D, A)$ will be larger. When $\varphi(A)$ decreases, $VGain(D, A)$ becomes smaller and $VGini(D, A)$ becomes larger, hence $VS(D, A)$ will be larger. According to the definition and the granulation monotonicity of the above two measures, from Proposition 6, we can obtain that the hybrid measure has the property of granulation non-monotonicity. Attribute dependence is used to weight the two measures, which can regulate the two measures and make them complement each other. The information gain tends to select attributes with more values, so calculating the dependence degree of the attribute and multiplying it can avoid the above problem to a certain extent. At the same time, the Gini index with more emphasis on algebraic representation is more reliable for data classification.

There are three states of attribute dependence: partial dependence, complete dependence and complete independence. When the attribute dependence equals 0, the hybrid measure depends on the Gini index, that is, $VS(D, A) = VGini(D, A)$. When the attribute dependence equals 1, the hybrid measure depends on the information gain and the attribute with the largest information gain is selected as the splitting attribute. This also solves the problem that the attribute dependence equals 0 when using a single measure, which will not occur when using the hybrid measure (Tab 5).

3.4 Algorithm design

The algorithm structure of DIGGI is similar to those of ID3 and CART, but there are still some differences between them. The differences are as follows: 1) DIGGI adopts a hybrid measure which combines information gain and Gini index in the way of attribute dependence weighting, while ID3 and CART adopt a single measure. The equivalence class is induced by the variable precision neighborhood equivalence relation. At the same time, the thickness of node branches can be adjusted by setting the variable precision threshold α . When $\alpha = 1$, it changes into the original strict neighborhood equivalence granule. To some extent, the hybrid measure overcomes the shortcomings of single measures. 2) ID3 needs to use the step of discretization to preprocess the continuous data, which may generate to the loss of information, and DIGGI could handle the continuous data directly.

The time complexity of the algorithm mainly comes from the calculation of uncertainty measure $VS(D, a)$ in step 1. The generation of neighborhood granules involves the granulation operation of all samples in the domain, and its time complexity is $O(M * N^2)$. M represents the number of features, and N represents the number of samples in the domain. The time complexity of calculating the variable precision neighborhood information entropy and the variable precision neighborhood Gini index are $O(N * \log N)$, and the granulation operation has been completed before the calculation of these two metrics. The calculation of each attribute dependency is relatively simple, and the time complexity is $O(M)$. Then it is an equivalent partition of the decision tree, which needs to find the equivalence class involving all samples under each attribute, and its time complexity is $O(M * N^2)$. Therefore, the time complexity of the DIGGI algorithm is $O(2 * M * N^2 * \log N)$.

As for the ID3 algorithm, the time complexity of calculating information entropy is $O(N * \log N)$, because each

Table 5 The DIGGI algorithm

Algorithm: Decision tree algorithm DIGGI

Input: A neighborhood decision system $NDS = (U, C \cup D, V, f, d, \delta)$ and variable precision threshold α .

Output: A DIGGI decision tree

- 1: Traverse all the single attributes in A (where $A \subseteq C$), and calculate $VS(D, a)$;
- 2: Determine the set $\arg \max_{a \in A \subseteq C} VS(D, a)$. Randomly select an optimal attribute b from $\arg \max_{a \in A \subseteq C} VS(D, a)$, and b is used as the splitting attribute;
- 3: According to the splitting attribute b in Step (2), calculate the variable precision neighborhood equivalence partition $U/VENR_A^{(\delta, \alpha)} = \{VX_1^A, VX_2^A, \dots, VX_n^A\}$, where each equivalence granule (i.e., $VX_i^A, 1 \leq i \leq n$) represents a branch of node splitting;
- 4: For each sample, determine the equivalence granule to which it belongs. Then divide the current neighborhood decision system into N sub-decision systems $NDS_1, NDS_2, \dots, NDS_n$;
- 5: Recursively execute Steps (1)-(4) for each of the N sub-decision systems until all conditional granules belong to the same decision class, or all condition attributes are detected;
- 6: Return a DIGGI decision tree.

Table 6 The description of 14 datasets

No.	Dataset	Number of samples	Number of condition attributes	Number of classes
1	Crayo	90	6	2
2	Wpbc	194	33	2
3	Plrx	182	18	1
4	Wine	178	13	3
5	Iris	150	4	3
6	Seg	210	19	7
7	Glass	214	10	6
8	Heart	270	13	2
9	Ecoil	336	7	7
10	ILPD	583	10	2
11	END2012	768	9	5
12	MF	1152	4	2
13	Segment	2310	19	7
14	SCADI	70	205	7

attribute value needs to be sorted. For the selection of the best partition attribute, the information gain corresponding to each attribute needs to be calculated, and the time complexity is $O(M * N * \log N)$. For each non-leaf node, it is necessary to recursively select the best partition attribute and establish the number of words, and the time complexity is $O(M * N * \log N)$. So the time complexity of the ID3 algorithm is $O(M * N * \log N)$. The C4.5 algorithm is improved compared with the ID3 algorithm, and the continuous attributes are optimized. The time complexity is $O(M * N * \log N)$, so the time complexity of the C4.5 algorithm is $O(M * N * \log N)$. The pruning of CART algorithm

is improved compared with ID3 algorithm, and the time complexity is $O(M * N * \log N)$, so the time complexity of CART algorithm is $O(M * N * \log N)$. Both the NDT algorithm and the VPNDT algorithm involve the granulation operation of all samples in the universe of domain. The time complexity of the operation is $O(M * N^2)$, and the equivalent partitioning operation time is consistent with this article, that is $O(M * N^2)$. NDT algorithm uses variable precision Gini index as the splitting node, and the time required to calculate the metric is $O(N * \log N)$. The time complexity of NDT algorithm is $O(M * N^2)$. The VPNDT algorithm uses the attribute dependency as the splitting node, and the time required to calculate the metric is $O(M)$. Therefore, the time complexity of the VPNDT algorithm is $O(M * N^2)$.

4 Experimental analysis

This section describes the comparative results of DIGGI and the existing algorithms. We used 14 public UCI datasets (as shown in Table 6) to verify the effectiveness of DIGGI and most attributes in these datasets are continuous attributes, and MF is the abbreviation of messidor featureis dataset. DIGGI was compared with three traditional algorithms, the NDT algorithm and VPNDT algorithm in the latest literature. Considering that the traditional algorithms cannot deal with continuous data, we discretize the data by equi-distance discretization method. In order to evaluate the quality of decision tree, we usually consider the accuracy of the algorithm and the number of leaves. In this experiment, these two indexes are also selected as the evaluation criteria of DIGGI and the ten-fold cross-validation method is used to verify the data set. The experimental results are shown in Tables 7, 8 and 9.

Table 7 The accuracies of various algorithms

No.	Data	ID3	CART	C4.5	VPNDT	NDT	DIGGI
1	Crayo	0.5556	0.6667	0.5556	0.8889	0.7778	1
2	Iris	0.5333	0.8667	0.6	0.8000	0.7733	1
3	Wine	0.8889	0.8333	0.8889	0.7500	0.5056	0.9444
4	Plrx	0.7368	0.7895	0.7368	0.7838	0.5165	0.9474
5	Wpbc	0.7	0.75	0.85	0.8718	0.4639	0.95
6	Seg	0.5714	0.619	0.6667	0.7381	0.4571	0.9524
7	Glass	0.6364	0.7273	0.7727	0.8410	0.514	0.9545
8	Heart	0.8148	0.7778	0.8519	0.8372	0.5926	0.963
9	Ecoil	0.7647	0.7941	0.8235	0.8382	0.6012	0.9706
10	ILPD	0.8136	0.7797	0.7458	0.8547	0.6849	0.9492
11	END2012	0.3766	0.4416	0.4286	0.7792	0.5599	0.961
12	MF	0.7241	0.6746	0.7446	0.7446	0.6823	0.7931
13	Segment	0.9394	0.9113	0.9091	0.8776	0.5463	0.9481
14	SCADI	0.7143	0.5714	0.8571	0.8095	0.4571	0.8571
15	Arithmetic mean	0.6979	0.7288	0.7451	0.8153	0.5809	0.9422

Table 8 The number of the leaves

No.	Data	ID3	CART	C4.5	VPNDT	NDT	DIGGI
1	Crayo	32	50	32	65	69	39
2	Iris	16	49	16	112	86	66
3	Wine	52	287	67	130	207	147
4	Plrx	179	305	197	142	355	158
5	Wpbc	247	386	205	159	387	168
6	Seg	186	158	182	191	382	197
7	Glass	152	160	158	188	229	213
8	Heart	186	404	191	186	250	239
9	Ecoil	152	221	161	302	536	280
10	ILPD	529	432	344	492	583	497
11	END2012	1060	834	1187	841	732	991
12	MF	525	763	498	508	841	463
13	Segment	1603	2161	1100	2081	58050	1984
14	SCADI	23	967	27	199	1020	233
15	Arithmetic mean	353	512.62	311.79	399.71	4551.93	405.36

The experimental environment is Intel(R) Core(TM) I7-8700 CPU @ 3.20ghz 3.19ghz, RAM 16.0GB.

The selection of neighborhood radius is closely related to the size of neighborhood granules, and its selection affects the whole neighborhood model. This paper mainly uses the adaptive radius constructed by standard deviation [23], which is expressed as $\delta_{C_1} = \frac{std(C_1)}{w}$, where $std(C_1) = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$ represents standard deviation of the sample in the attribute C_1 and w represents adjustive parameter. In this paper, we set $w = 2$. In this experiment, we select 90% of the data in the data set as the training set, 10% of the data as the test set, and the number of intervals used for equal distance divergence is 6. The neighborhood radius used by

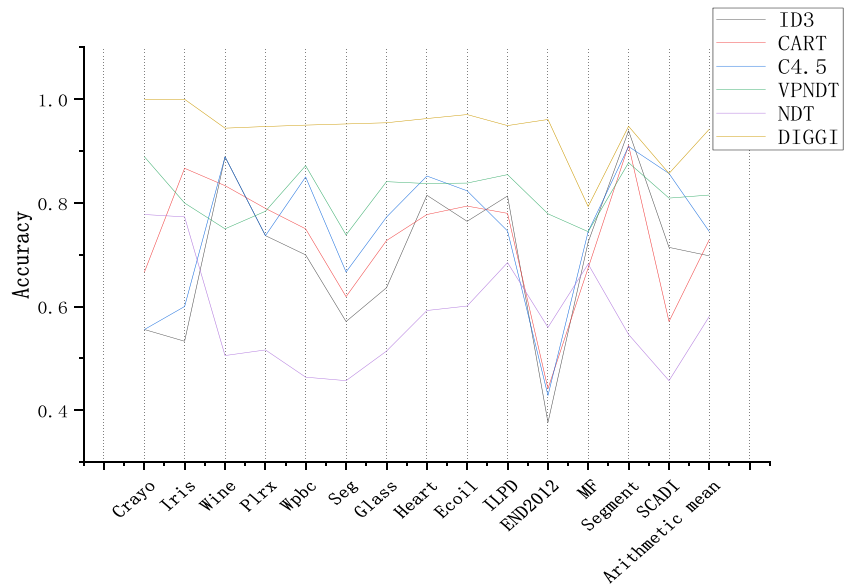
the NDT algorithm and the VPNDT algorithm is consistent with this paper. Their variable precision thresholds are set to 0.8, and the variable precision threshold α in this paper is set to 0.7.

We can see from the experimental results in Table 7 that DIGGI is effective and its accuracy has been greatly improved. Figure 1 specifically show the accuracy comparison of the five algorithms on different datasets. DIGGI not only can be directly applied to the classification and prediction of continuous data, but also has higher classification accuracy than the existing algorithms. From the statistical perspective of arithmetic average, the accuracy of DIGGI is 0.35 higher than ID3. Compared with all algorithms, it

Table 9 The running time of various algorithms

No.	Data	ID3	CART	C4.5	VPNDT	NDT	DIGGI
1	Crayo	0.0112	0.0314	0.0194	3.4637	2.04642	8.889
2	Iris	0.0056	0.0062	0.01033	5.8446	2.3669	19.6659
3	Wine	0.0421	0.4383	0.10176	29.624	11.055	94.1479
4	Plrx	0.1146	0.3518	0.224	20.5421	11.8675	71.8047
5	Wpbc	0.3867	3.073	0.66307	57.2925	39.1085	214.0631
6	Seg	0.2136	0.6133	0.4384	142.896	92.6474	418.226
7	Glass	0.0874	0.284	0.1871	60.5966	20.384	172.4266
8	Heart	0.1117	0.4796	0.233	60.8479	50.14802	219.8214
9	Ecoil	0.0713	0.2271	0.131	309.04575	53.208	448.0677
10	ILPD	0.2233	0.41587	0.2074	189.7859	144.7494	1136.7457
11	END2012	0.6517	1.7933	1.3982	1297.2029	223.3739	2847.0397
12	MF	0.1945	1.5047	0.3475	1941.161	922.1475	7807.3785
13	Segment	1.9742	8.3889	5.5608	28559.2745	18623.8633	85056.6382
14	SCADI	0.4239	50.2981	1.13037	893.7827	1474.9017	2297.0355
15	Arithmetic mean	0.3223	4.8504	0.7609	2397.9543	1457.9905	7200.8536

Fig. 1 The accuracy comparison of four algorithms with DIGGI



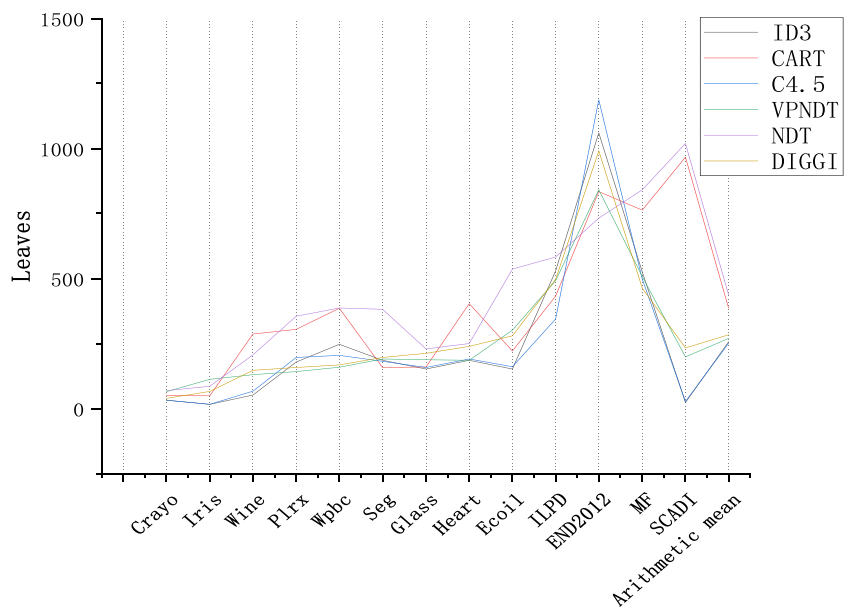
is improved by more than 10%. Among the 13 datasets, the results on Crayo, Seg, Glass and END2012 show the most obvious improvement, and the results on other datasets also show relatively large improvement, which indicate that DIGGI is effective.

We then observe the specific number of leaves of these algorithms on these 14 data sets from Table 8. Figure 2 shows them in a more intuitive form and shows the comparison of the number of leaves of the five algorithms on different datasets. We can see that DIGGI has fewer leaves than the existing algorithms on some datasets and has a higher number of leaves on the remaining datasets, but the number of leaves is still within the receivable range. In the high-dimensional

data set SCADI, it can be clearly found that our proposed algorithm is still effective, while the CART algorithm and NDT algorithm obviously show the inapplicability in high-dimensional data.

Table 9 shows the time spent by each algorithm on 14 datasets. We can find that the running time of the three classical decision tree algorithms is very short, and the running time of the other three algorithms increases with the increase of the number of samples or attributes in the data set. Therefore, we can see that in the large data set, running the algorithm requires a computer with super computing power to run, so it is not suitable for using the algorithm in the large data set.

Fig. 2 Comparison of leaf numbers of four algorithms with DIGGI



5 Conclusions

The strictness of neighborhood equivalence partition is difficult to achieve in real-world data. To overcome this problem, we propose the notion of neighborhood approximate equivalence, and thus propose a variable precision neighborhood equivalence relation and a new model of VPN rough sets. The VPN Gini index and the VPN information gain are constructed by the VPN equivalence relation. In the construction of decision tree, this paper proposes a new metric function for the measurement of splitting nodes. The experimental results show that DIGGI is effective, and its accuracy is also improved obviously. At the same time, DIGGI no longer needs to discretize the continuous data, so as to avoid the information loss.

As a summary of this work, we discuss the advantages and disadvantages of this work.

Now we first describe some of the contributions and advantages of the work done in this paper. (1) We propose a variable precision domain rough set based on similarity relation, and explore some related properties of the rough set, which enriches the rough set theory. (2) Based on the similarity relation, we extend the variable precision neighborhood information entropy and the variable precision neighborhood Gini index to omit the operation of data discretization and reduce the information loss caused by data discretization, which has certain reference significance for the improvement of decision tree. (3) The hybrid metric we constructed avoids the situation where the metric is 0 in a single metric to a certain extent, so the exploration of hybrid metrics is necessary, which also provides a new idea for the exploration and improvement of uncertain metrics. (4) From our experimental data, our improved algorithm is effective, which improves the classification accuracy of the decision tree within the acceptable range of leaf number, and is also effective in high-dimensional data (from the experimental results of the data set SCADI). Therefore, from the above summary, we can know that the work has certain significance and provides some new ideas for the improvement of decision tree.

Of course, at the same time, there are also some shortcomings in this work: (1) The improved decision tree algorithm in this paper is based on the neighborhood system. The generation of neighborhood granules involves the operation of neighborhood granulation. However, when the number of samples is large, the algorithm takes a long time and the time complexity is high, which cannot improve our work efficiency. (2) When our device configuration is low, it takes a lot of time to wait for a classification result, and we need to configure higher devices to run the algorithm. We also need to think about how to reduce the time complexity while

maintaining good classification results, which will be a focus of future work.

Funding This work is supported by National Natural Science Foundation of China under Grant Nos.: 62166001, 61976158, 62266003, Jiangxi Provincial Natural Science Foundation under Grant Nos: 20224BAB212022, and Science and Technology Project of Education Department of Jiangxi Province under Grant Nos: GJJ211435

Declarations

Competing of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

References

1. Bishop CM, Nasrabadi NM (2006) Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: Springer
2. Myles AJ, Feudale RN, Liu Y, Woody NA, Brown SD (2004) An introduction to decision tree modeling. *J Chemometr: J Chemometr Soc* 18(6):275–285
3. Safavian SR, Landgrebe D (1991) A survey of decision tree classifier methodology. *IEEE Trans Syst Man Cybern Syst* 21(3):660–674
4. Breiman L (2017) Classification and regression trees. Routledge
5. Quinlan JR (1996) Learning decision tree classifiers. *ACM Comput Surv* 28(1):71–72
6. Quinlan JR (2014) C4. 5: programs for machine learning. Elsevier
7. Hssina B, Merbouha A, Ezzikouri H, Erritali M (2014) A comparative study of decision tree ID3 and C4.5. *Int J Adv Comput Sci Appl* 4(2):13–19
8. Ruggieri S (2002) Efficient C4. 5 [classification algorithm]. *IEEE Trans Knowl Data Eng* 14(2):438–444
9. Liang X, Qu F, Yang Y, Cai H (2015) An improved ID3 decision tree algorithm based on attribute weighted. In 2nd International Conference on Civil, Materials and Environmental Sciences (pp. 613–615). Atlantis Press
10. Wang S, Jia Z, Cao N (2022) Research on optimization and application of Spark decision tree algorithm under cloud-edge collaboration. *Int J Intell Syst* 37(11):8833–8854
11. Panhalkar AR, Doye DD (2022) Optimization of decision trees using modified African buffalo algorithm. *Journal of King Saud University-Computer and Information Sciences* 34(8):4763–4772
12. Pawlak Z (1982) Rough sets. *Int J Comput Inform Sci* 11:341–356
13. Wei JM, Wang SQ, Wang MY, You JP, Liu DY (2007) Rough set based approach for inducing decision trees. *Knowl-Based Syst* 20(8):695–702
14. Wang Z, Liu Y, Liu L (2017, December) A new way to choose splitting attribute in ID3 algorithm. In 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 659–663). IEEE
15. Hu Q, Yu D, Xie Z (2008) Neighborhood classifiers. *Expert Syst Appl* 34(2):866–876
16. Hu Q, Yu D, Liu J, Wu C (2008) Neighborhood rough set based heterogeneous feature subset selection. *Inf Sci* 178(18):3577–3594
17. Sun L, Zhang X, Xu J, Zhang S (2019) An attribute reduction method using neighborhood entropy measures in neighborhood rough sets. *Entropy* 21(2):155

18. Xie X, Zhang X, Yang J (2022) Improved ID3 Decision Tree Algorithm Induced by Neighborhood Equivalence Relations. *Comput Appl Res* 39(01):102–105+112 (In Chinese with English Abstract)
19. Ziarko W (1993) Variable precision rough set model. *J Comput Syst Sci* 46(1):39–59
20. Zakowski W (1983) Approximations in the space (u, π) . *Demonstratio mathematica* 16(3):761–770
21. Chen Y, Chen Y (2021) Feature subset selection based on variable precision neighborhood rough sets. *Int J Comput Intell Syst* 14(1):572–581
22. Xie X, Zhang X, Wang X, Tang P (2022) Neighborhood Decision Tree Construction Algorithm of Variable Precision Neighborhood equivalent Kernel. *J Comput Appl* 42(02):382–388 ((In Chinese with English Abstract))
23. Liu C, Lin B, Lai J, Miao D (2022) An improved decision tree algorithm based on variable precision neighborhood similarity. *Inf Sci* 615:152–166
24. Xie X, Zhang X, Yang J (2022) Decision Tree algorithm combining information gain and Gini Index. *Comput Eng Appl* 58(10):139–144 (In Chinese with English Abstract)
25. Jain V, Phophalia A, Bhatt JS (2018) Investigation of a joint splitting criteria for decision tree classifier use of information gain and gini index. In *TENCON 2018-2018 IEEE Region 10 Conference* (pp. 2187-2192). IEEE
26. Dice LR (1945) Measures of the amount of ecologic association between species. *Ecology* 26(3):297–302
27. Zhang X, Tang X, Yang J, Lv Z (2020) Quantitative three-way class-specific attribute reducts based on region preservations. *Int J Approx Reason* 117:96–121

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

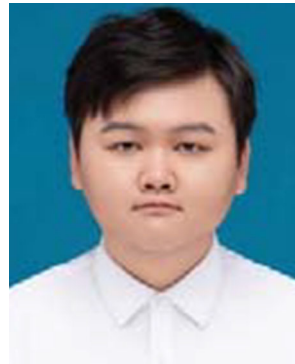
Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Caihui Liu received the Ph.D. degree from Tongji University, China, in 2013. He is currently a Professor with the Department of Mathematics and Computer Science, Gannan Normal University, China. His research interests include artificial intelligence, machine learning, and granular computing.



Jianying Lai received the bachelor's degree from Gannan Normal University in 2021. She is currently pursuing the master's degree with the School of Mathematics and Computer Science, Gannan Normal University. Her research interests include granular computing and machine learning.



Bowen Lin received the bachelor's degree from Gannan Normal University in 2021. He is currently pursuing the master's degree with the School of Mathematics and Computer Science, Gannan Normal University. His research interests include granular computing and machine learning.



Duoqian Miao received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 1997. He is currently a Professor with the Department of Computer Science and Technology, Tongji University. His research interests include artificial intelligence, machine learning, big data analysis, and granular computing.