Applied Soft Computing 124 (2022) 109025

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Weighted fuzzy rough sets-based tri-training and its application to medical diagnosis

Jinming Xing [a,b], Can Gao [a,b,c,*], Jie Zhou [a,b,c]

[a] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, PR China
[b] Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen 518060, PR China
[c] SZU Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518060, PR China

ARTICLE INFO

Article history:
Received 29 November 2021
Received in revised form 3 May 2022
Accepted 11 May 2022
Available online 19 May 2022

Keywords:
Fuzzy rough sets
Sample weighting
High-order margin
Tri-training
Partially labeled data

ABSTRACT

The theory of fuzzy rough sets is an effective soft computing paradigm for dealing with vague, uncertain, or imprecise data. However, most existing fuzzy rough sets-based methods may suffer from robustness since all samples are considered equally and also these methods are designed to cater for supervised or unsupervised learning. In this paper, we propose a weighted fuzzy rough sets-based multi-view tri-training model for partially labeled data. Specifically, considering the negative effect of noise, we first use a technique of data editing to filter potentially possible noises, and then a gradient descent algorithm is employed to optimize the weight of each sample with the objective of maximizing high-order weighted fuzzy dependency, based on which a robust weighted fuzzy rough set model is developed for labeled data. Moreover, we introduce the robust weighted fuzzy rough sets into tri-training and propose multi-view-based robust tri-training for partially labeled data by exploring data representations in the original view, the transformed view of principal component analysis, and the granular view after discretization. Extensive experiments conducted on UCI benchmark and medical diagnosis data sets show that the proposed model achieves favorable results in both supervised and semi-supervised scenarios.

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.4118969.v1.

1. Introduction

Rough sets, as a powerful mathematical tool for processing vague, imprecise, or uncertain data, have received extensive attention from the fields of soft computing, computational intelligence, and decision analysis [1–3]. In Pawlak's rough sets [1], all samples are divided into a set of equivalence classes by using an attribute or attribute subset, which is called indiscernibility relation, and the samples within the same equivalence class are indiscernible from each other. Based on the equivalence classes induced by the indiscernibility relation, two precise sets, called the lower and upper approximations, are defined to describe a vague or uncertain concept [1]. However, Pawlak's rough sets heavily rely on the equivalence relation (or indiscernibility relation), which is usually used for discrete data. To improve or extend Pawlak's rough sets, several models have been proposed, such as decision-theoretic rough sets [4,5], fuzzy rough sets [6,7], neighborhood rough sets [8,9] and others [10,11].

Fuzzy rough sets [12,13] extend from Pawlak's rough sets by replacing equivalence relation with fuzzy similarity relation, and can effectively deal with both discrete and continuous data. Dubois and Prad [6] first introduced the fuzzy equivalence relation that satisfies the properties of reflexivity, symmetry, and min–max transitivity, and used T-norms and T-conorms to construct fuzzy lower and upper approximations. Radzikowska and Kerre [14] defined a more generalized fuzzy rough sets model, in which the fuzzy similarity relationship is determined by a border implicator and T-norm. In [15], Mi and Zhang developed a new form of fuzzy rough sets based on residual implication θ and its dual. Yeung et al. [16] gave two methods for constructing fuzzy rough sets for arbitrary fuzzy relation. Hu et al. [17] combined kernel function with fuzzy rough sets and proposed kernelized fuzzy rough sets using the high-dimension mapping characteristics of the kernel function. In [18], Mieszkowicz–Rolka and Rolka introduced a new fuzzy rough sets model called variable precision

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, PR China.
E-mail address: 2005gaocan@163.com (C. Gao).

https://doi.org/10.1016/j.asoc.2022.109025
1568-4946/

fuzzy rough sets (VPFRS) to alleviate the impact of noise. Hu et al. [19] proposed another new fuzzy rough sets model based on fuzzy granulation and fuzzy inclusion. Zhao et al. [20] developed fuzzy variable precision rough sets (FVPRS) to handle the noise of misclassification and perturbation. In [21], Cornelis et al. proposed a fuzzy rough sets model based on ordered weighted average operators. Due to the generalization and applicability, fuzzy rough sets have been used in many research domains such as attribute reduction, rule extraction, and approximate reasoning [22–34].

Measuring or computing the fuzzy similarity of samples is the cornerstone of fuzzy rough sets [35]. Generally, attribute weighting and sample weighting are commonly used methods to improve the computation of fuzzy similarity of samples. In the method of attribute weighting, different attributes are given different weights according to attribute importance measure or domain prior knowledge [36]. Attribute reduction (feature selection) [37,38], as a special case of attribute weighting, aims to keep important attributes and simultaneously remove redundant and irrelevant attributes to facilitate the computation of sample similarity. Kuncheva [39] first proposed an attribute reduction algorithm based on fuzzy rough sets. Jenshen and Shen [40] constructed an attribute reduction strategy based on ant colony optimization and combined it with fuzzy rough sets for attribute reduction. Hu et al. [17] introduced an attribute reduction algorithm using kernelized fuzzy rough sets. In [41], Ganivada et al. proposed an unsupervised attribute reduction algorithm based on a three-layer fuzzy rough granular network and fuzzy rough sets. Additionally, some positive region, information entropy, and discernibility matrix-based attribute reduction methods have been proposed to deal with labeled data or unlabeled data in different scenarios [42–46].

The method of sample weighting assigns different weights to samples. Hu et al. [47] developed a sample weighting method with the objective of maximizing the margin of weighted samples, and the gradient descent algorithm was employed to optimize the weight of each sample. Further, based on the concept of fuzzy dependency in fuzzy rough sets, Hu et al. [48] proposed a gradient descent algorithm to maximize the fuzzy dependency degree by optimizing the sample weights. Du et al. [49] handled the problem of uneven distribution of class by learning sample weights based on hypothesis margin. Fan et al. [50] assigned weights to samples by calculating the similarity relationship between samples and proposed a weighted sparse representation for classification. Arazo et al. [51] proposed dynamic hard and soft bootstrapping losses by weighting each sample to improve the robustness of neural networks. Zhang et al. [52] developed a sample reweighting strategy to reduce the deviation between the source and target domains. Shu et al. [53] proposed a meta-weight-net to assign smaller weights to noisy samples by training a sample weight function with unbiased correct samples. Ghosh et al. [54] further pointed out that the meta-weight-net using a robust loss function can learn a proper sample weight function without unbiased correct samples. Kalai et al. [55] explored the boosting technique in the presence of noisy samples. Frenay et al. [56] summarized sample weighting algorithms in the case of noise. Although promising results are achieved after sample weighting, these methods do not fully consider the influence of noise, and at the same time, high-order neighborhood structure information of samples has not been explored.

The above attribute weighting and sample weighting methods are mainly used in supervised or unsupervised learning. However, most real-world applications come with few labeled data and a large number of unlabeled data (called partially labeled data hereafter). How to use both labeled and unlabeled data to learn an effective model is one of the most important problems in semi-supervised learning [57]. Parthalain and Jensen [58] used the concept of lower approximation in fuzzy rough sets to develop a self-training style model for partially labeled data. Qian et al. [59] combined local rough sets with multi-granulation decision-theoretic rough sets to deal with partially labeled data. Further, Wang et al. [60] introduced local neighborhood rough sets to handle big data. Guo et al. [61] considered the information differences of equivalence classes and proposed a local logical disjunction double-quantitative rough set model. Gao et al. [62] used prior class-distribution information to annotate unlabeled data with pseudo labels and proposed a rough sets-based semi-supervised attribute reduction algorithm. In [63], a three-way decision model with co-training was proposed for partially labeled data, where unlabeled data are divided into useful, useless, and uncertain data. Chen et al. [64] used a clustering algorithm to obtain the sample structure, so as to assign weights to samples to form a weighted graph. Ren et al. [65] constructed a model-sample dependency function, by which the samples are weighted accordingly. To the best of our knowledge, the problem of fuzzy rough sets with sample weighting has not been studied in the semi-supervised learning scenario. To address this problem, we propose a weighted fuzzy rough model and develop a tri-training framework for partially labeled data. On the whole, the main contributions of the paper are three-fold.

(1) To improve the robustness of fuzzy rough sets, we introduce a weighted fuzzy rough set model. The proposed model uses a data editing technique to remove potentially reliable noises and employs a high-order margin-based gradient descent optimization algorithm to weight samples. Compared with classic fuzzy rough sets, the weighted fuzzy rough sets consider the importance of samples more objectively and can capture the intrinsic information of data.

(2) To extend fuzzy rough sets to semi-supervised tasks, we develop a three-views-based tri-training model. Instead of manipulating samples with the technique of resampling in classic tri-training, we explore semi-supervised data from the original view, the transformed view after principal component analysis (PCA), and the quantized view after discretization. These views make the base classifiers more diverse, and consequently, the fuzzy rough sets-based tri-training model can effectively leverage unlabeled data to improve performance.

(3) To verify the effectiveness of the proposed model, we conduct extensive experiments on the proposed weighted fuzzy rough set model under different noise rates and very promising results are achieved. Also, we apply the weighted fuzzy rough sets-based tri-training to partially labeled benchmark and medical diagnosis data sets. Experimental results show that the proposed tri-training model achieves superior results in comparison with other state-of-the-art ones.

The rest of the paper is organized as follows. Section 2 reviews the basic concepts in fuzzy rough sets and semi-supervised learning. In Section 3, we elaborate on the proposed model. Experiment results and analysis are reported in Section 4. Finally, Section 5 concludes the paper and indicates the intended directions of future research.

2. Preliminaries

This section will review some concepts in fuzzy rough sets and semi-supervised learning. Further details of these theories can be found in [6,66,67].

2.1. Fuzzy rough sets

Formally, a fuzzy information system [6] is defined as $FIS = (U, A, V, f)$, where U is a non-empty and finite set of samples, called the universe of discourse; A is a non-empty and finite

Table 1
Typical T–norm and S–norm fuzzy operators.

	T–norm	S–norm
Min-max	$T_M(a,b) = min(a,b)$	$S_M(a,b) = max(a,b)$
Algebra	$T_p(a,b) = a \times b$	$S_p(a,b) = a+b-ab$
Lukasiewicz	$T_L(a,b) = max(a+b-1,0)$	$S_L(a,b) = min(a+b,1)$
Cosine	$T_{cos}(a,b) = max\left(ab - \sqrt{1-a^2}\sqrt{1-b^2}, 0\right)$	$S_{cos}(a,b) = min\left(a+b-ab+\sqrt{2a-a^2}\sqrt{2b-b^2}, 1\right)$

set of attributes to describe the samples; V is the union of the domains of all attributes, i.e., $V = \bigcup V_a$, where V_a denotes the domain of an attribute $a \in A$, and f is an information function, i.e., $f : U \times A \to V$. If the attribute set A can be further divided into condition attribute set C and decision attribute set D, the fuzzy information system is also called the fuzzy decision information system or simply the fuzzy decision table.

For any attribute subset $B \subseteq A$, it determines a fuzzy equivalence relation R, which forms a covering of the universe U. The fuzzy equivalence class containing x is denoted as $[x]_R$, and the membership degree of a sample $y \in U$ to the fuzzy equivalence class $[x]_R$ is denoted as $[x]_R(y) = R(x,y)$, where $R(x,y)$ represents the similarity between the samples x and y in relation R. For any sample $x \in U$ and any subset $X \subseteq U$, the fuzzy lower and upper approximations of x to X are defined as

$$\begin{aligned} \underline{R_S}X(x) &= inf_{y\in U} S(N(R(x,y)), X(y)), \\ \overline{R_T}X(x) &= sup_{y\in U} T(R(x,y), X(y)), \end{aligned} \tag{1}$$

where S and T denote the fuzzy triangular norm (T-norm) and fuzzy triangular conorm (S-norm), N is a negator and $N(x) = 1 - x$. Some commonly used fuzzy operators are shown in Table 1. Unless stated differently, fuzzy operators in this paper are assumed to be the standard min and max, i.e., $S = max$ and $T = min$.

Let $FDS = (U, A = C \cup D, V, f)$ be a fuzzy decision system and d_i be the set of samples with the decision i. For any fuzzy equivalence relation induced by $B \subseteq$C, the fuzzy lower and upper approximations of d_i with $min - max$ norms are defined as

$$\begin{aligned} \underline{R_B}d_i(x) &= inf_{y\in U} max(1 - R(x,y), d_i(y)), \\ \overline{R_B}d_i(x) &= sup_{y\in U} min(R(x,y), d_i(y)), \end{aligned} \tag{2}$$

where $d_i(y)$ indicates whether y belongs to d_i.

More generally, let $FDS = (U, A = C \cup D, V, f)$ be a fuzzy decision system and $U/D = \{d_1, d_2, \ldots, d_{|U/D|}\}$ be a set of equivalence classes induced by the decision attribute D. For any fuzzy equivalence relation induced by $B \subseteq C$, the positive region of D with respect to B is defined as

$$POS_B(D) = \bigcup_{d_i \in U/D} \underline{R_B}d_i. \tag{3}$$

The positive region of D given B reflects the degree of certainty of samples. Thus, it can be used to measure the importance of attributes or attribute subsets. The fuzzy dependency degree of decision attribute D on attribute subset B is defined as

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|}, \tag{4}$$

where $|\cdot|$ denotes the fuzzy cardinality of a fuzzy set.

The fuzzy relation reflecting the similarity of samples plays an important role in fuzzy rough sets. Fuzzy T-equivalence relation is a commonly used one in fuzzy rough sets, which satisfies reflexivity: $R(x,x) = 1$, symmetry $R(x,y) = R(y,x)$, and T-transitivity $T(R(x,y), R(y,z)) \leq R(x,z)$. Since some of the kernel functions hold the same properties as the fuzzy T-equivalence relation, Hu [17] introduced kernel functions to compute fuzzy T-equivalence relation and proposed kernelized fuzzy rough sets. For any subset $X \subseteq U$, the fuzzy lower and upper approximations of X with respect to B are redefined as

$$\begin{aligned} \underline{k_B}X(x) &= inf_{y\in U} max(1 - k(x,y), X(y)), \\ \overline{k_B}X(x) &= sup_{y\in U} min(k(x,y), X(y)), \end{aligned} \tag{5}$$

where k represents the kernel function. Some commonly used kernel functions are Gaussian kernel: $k_G(x,y) = exp(-\frac{\|x-y\|^2}{\delta})$, exponential kernel: $k_E(x,y) = exp(-\frac{\|x-y\|}{\delta})$, and rational quadratic kernel: $k_R(x,y) = 1 - \frac{\|x-y\|^2}{\|x-y\|^2+\delta}$.

2.2. Semi-supervised learning

In semi-supervised learning, training data is divided into labeled data L and unlabeled data N, where the number of samples in L is much smaller than that of N. Formally, a partially labeled data in semi-supervised learning is denoted as $PS = (U = L \cup N, A = C \cup D, V, f)$. Semi-supervised learning mainly includes semi-supervised clustering, semi-supervised regression/classification, and semi-supervised attribute reduction. In semi-supervised clustering, the supervised information of labeled data L can be used to assist clustering algorithm to obtain a better clustering result [57]. In semi-supervised regression/classification, the structure information of unlabeled data N can be captured to enhance the performance of a supervised model trained only on labeled data L. Whereas semi-supervised attribute reduction uses both the supervised label information and unsupervised structure information to evaluate the significance of an attribute or attribute subset. Refer to [57,68,69] for more details. In this study, we mainly focus on semi-supervised classification.

Semi-supervised classification aims to enhance the performance of a classification model trained only on labeled data L by capturing the structure information of unlabeled data N. Semi-supervised classification can be roughly divided into transductive and inductive methods [57]. The transductive methods do not generate classification models but directly provide prediction, and graph-based methods are commonly used. These methods are composed of three parts: graph construction, graph weighting, and graph reasoning. The inductive methods use labeled data and unlabeled data to train a classifier to make the prediction. Roughly speaking, the inductive methods can be divided into intrinsically semi-supervised methods and wrapper methods. The intrinsically semi-supervised methods include maximum margin, perturbation-based, manifold, and generative models. Whereas the wrapper methods mainly include self-training, co-training, and tri-training. Self-training [68] uses labeled data L to train a base classifier and employs the learned base classifier to predict unlabeled data N, from which some reliable samples are selected and be added into the training set to retrain the base classifier. The model is repeated until all unlabeled data N are used. However, self-training is very sensitive to noise, and it is easy to accumulate errors during the learning process. To improve the robustness of self-training, co-training [68] has been proposed. It requires two sufficient and redundant views to train two base classifiers. Each classifier trained on labeled data selects some highly confident samples with pseudo-labels to retrain its counterpart. The two base classifiers iteratively learn from each other on unlabeled data until the stopping conditions are met.

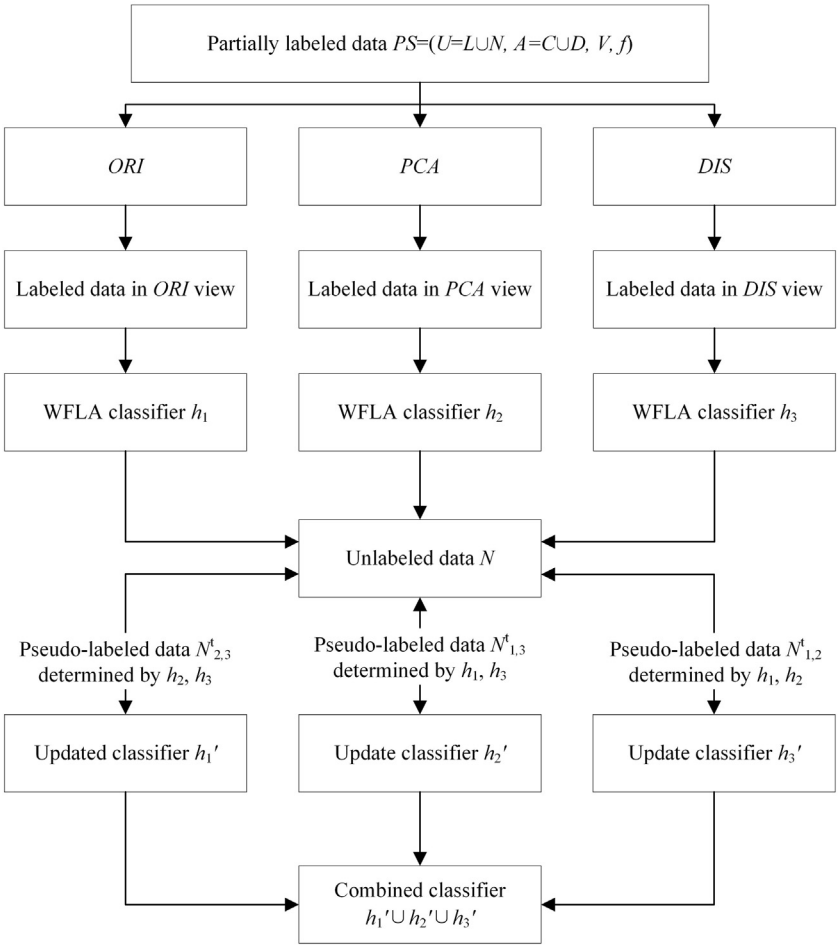


Fig. 1. Framework of weighted fuzzy rough sets-based tri-training model.

However, the assumption of two sufficient and redundant views is difficult to satisfy in real-world data. Tri-training [69] initializes three base classifiers by using the technique of resampling on labeled data *L*. In each iteration of the learning process, two classifiers vote to select the most credible samples and add them to the training set of the other classifier. The three base classifiers iteratively capitalize on unlabeled data to improve performance. Tri-training uses the technique of resampling to generate base classifiers, but it operates on the same labeled data such that the resulting classifiers have high redundancy and low diversity. In addition, practical data are always entangled with noise. How to deal with noise and reduce the redundancy between base classifiers is a problem worthy of further investigation.

3. Weighted fuzzy rough sets-based semi-supervised learning

In this section, we first present the overall framework of the proposed weighted fuzzy rough sets-based tri-training model. Then, we elaborate on the strategy of filtering noises and weighting samples for fuzzy rough sets. Finally, we develop a three-view-based tri-training model for partially labeled data.

3.1. Overall framework of the proposed model

Traditional fuzzy rough sets consider each sample equally such that their robustness to noise is limited. Moreover, fuzzy rough sets-based methods are mainly introduced to deal with labeled or unlabeled data, and they are thus ineffective to handle partially labeled data. Tri-training is one of the most popular multi-view-based semi-supervised models. It generates three base classifiers through resampling initially labeled data and makes these classifiers learn from each other on unlabeled data. Constructing base classifiers in tri-training is a key factor in determining learning performance. It is highly desired that the base classifiers can not only effectively handle data with uncertainty and also provide the tolerance ability to noise. Meanwhile, due to the scarcity of labeled data, the learned base classifiers may have poor diversity and high redundancy, thus resulting in mediocre performance. Factually, practical data can be described and investigated by different views, whereas multi-view of data is essentially beneficial to learning models. Bear this in mind, we propose a weighted fuzzy rough sets-based tri-training model for partially labeled data (see Fig. 1).

Specifically, we first generate two views of the original data through principal component analysis and discretization, which together with the original data form three different modal data, called the original data view *ORI*, principal component analysis view *PCA*, and discretized view *DIS*, respectively. In each view, a gradient descent algorithm is used to optimize the weights of initially labeled data by maximizing the sample margin, and then a weighted fuzzy lower approximation (WFLA) classifier is trained on the weighted labeled data as the base classifier for tri-training. In each iteration, one classifier is retrained on high-quality unlabeled data determined by the other two classifiers. The three classifiers learn from each other until the stopping condition is met. After refined on unlabeled data, the three classifiers are finally combined into a multi-classifier system. The WFLA

classifiers are learned from weighted data with different views and thus have great diversity and robustness to noise, whereas the mechanism of tri-training makes the proposed multi-view model to effectively capitalize on unlabeled data to improve the performance.

3.2. Sample weighting by maximizing high-order margin

In the statistic learning theory, it has been shown that the performance of a learned model is highly related to its hypothesis margin [47]. Formally, the margin of a sample x_i can be defined as [47]

$$MG(x_i) = \frac{1}{2}(\|x_i - NM(x_i)\| - \|x_i - NH(x_i)\|), \tag{6}$$

where $NM(x_i)$ means the nearest neighbor from the heterogeneous classes, called the nearest miss, $NH(x_i)$ represents the nearest neighbor from the homogeneous class, called the nearest hit, and the symbol "$\|*\|$" denotes a distance function.

Intuitively, it is easy to find a decision boundary to correctly classify all samples if each sample has a large margin. In the computation of sample margin, distance function, attribute weight, or sample weight can be considered a factor to optimize the margin of a sample. Let $\mathbf{w} = \{w_1, w_2, \ldots, w_n\}$ be a vector of sample weights. For any sample x_i, its weighted sample margin is defined as [47]

$$WMG(x_i) = \frac{1}{2}(w(NM(x_i))\|x_i - NM(x_i)\| - w(NH(x_i))\|x_i - NH(x_i)\|), \tag{7}$$

where $w(x)$ denotes the weight of a sample x. Given the appropriate sample weights, one can increase the sample margin, and improve the performance of the classification model [47].

In fuzzy rough sets, the lower approximation of a sample x is essentially determined by the similarity $R(x, y)$ of the sample x to its neighbor sample y and the membership $d_i(y)$ of the sample y to decision class (see (2)). Factually, $d_i(y)$ represents whether y belongs to the decision d_i. Therefore, there are only two possible values 0 or 1. Formally, the lower approximation of a sample x to the decision d_i can be simplified as [48]

$$\underline{R}d_i(x) = inf_{y \notin d_i}(1 - R(x, y)). \tag{8}$$

That is, the lower approximation of the sample x to the decision class d_i is determined by the nearest sample that does not belong to the decision class d_i. Thus, the fuzzy dependency degree can be defined as

$$\gamma_B(D) = \frac{\left|\bigcup_i^k \underline{R}d_i\right|}{|U|} = \frac{\sum_{j=1}^n \underline{R}d(x_j)}{|U|} = \frac{\sum_{j=1}^n (1 - R(x_j, y))}{|U|} (x_j \in d), \tag{9}$$

where sample y is the nearest heterogeneous sample of sample x_j, and the formula $1 - R(x_i, y)$ represents the degree of dissimilarity between the samples x_i and y.

Formally, the sample margin is determined by the distances from the sample to its nearest heterogeneous and heterogeneous samples, whereas the distance between the sample and its nearest homogeneous sample is generally stable. Thus, the sample margin is mainly reflected by the nearest heterogeneous sample, i.e., the nearest miss. Intuitively, the fuzzy lower approximate also reflects the information of the nearest heterogeneous sample, and thus can be considered as a special form of margin. That is to say, the problem of optimizing the fuzzy dependency degree is equivalent to that of maximizing the margin. Therefore, we maximize fuzzy lower approximation to maximize margin. When Gaussian kernel function is used as the fuzzy similarity relation, the fuzzy dependency degree is redefined as

$$\gamma_B(D) = \frac{1}{n}\sum_{i=1}^n \left(1 - exp\left(-\frac{\|x_i - NM(x_i)\|^2}{\sigma}\right)\right), \tag{10}$$

where σ is a kernel parameter.

Similar to the weighted margin, we can define the weighted fuzzy dependency degree when sample weights are considered. Let $\mathbf{w} = \{w_1, w_2, \ldots, w_n\}$ be a vector of sample weights. The weighted fuzzy dependency degree is defined as

$$\gamma_B^w(D) = \frac{1}{n}\sum_{i=1}^n \left(1 - exp\left(-\frac{w(NM(x_i))^2\|x_i - NM(x_i)\|^2}{\sigma}\right)\right), \tag{11}$$

where $w(NM(x_i))$ is the sample weight of $NM(x_i)$.

Based on the concept of the weighted fuzzy dependency degree, we can use the gradient descent algorithm to optimize the margin by assigning proper weight to each sample. However, the weighted fuzzy dependency degree is very susceptible to noise. As shown in Fig. 2.(a), x_1 is the nearest heterogeneous sample to the blue cycle class C_1, and x_2 is the nearest heterogeneous sample to the red rectangle class C_2. Then the fuzzy dependency degree is computed as

$$\gamma_B(D) = \frac{1}{n}\left(\sum_{x_i \in C_1}(1 - R(x_1, x_i)) + \sum_{x_j \in C_2}(1 - R(x_2, x_j))\right). \tag{12}$$

However, in the case of noisy data as shown in Fig. 2.(b), x_2' is the nearest heterogeneous sample to the blue cycle class C_1, x_1' is the nearest heterogeneous sample to the red rectangle class C_2, and the fuzzy dependency becomes

$$\gamma_B(D) = \frac{1}{n}\Big(\sum_{x_i \in C_1, x_i \neq x_2'}(1 - R(x_2', x_i)) + \sum_{x_j \in C_2, x_j \neq x_1'}(1 - R(x_1', x_j)) + 1 - R(NM(x_2'), x_2') + 1 - R(NM(x_1'), x_1')\Big). \tag{13}$$

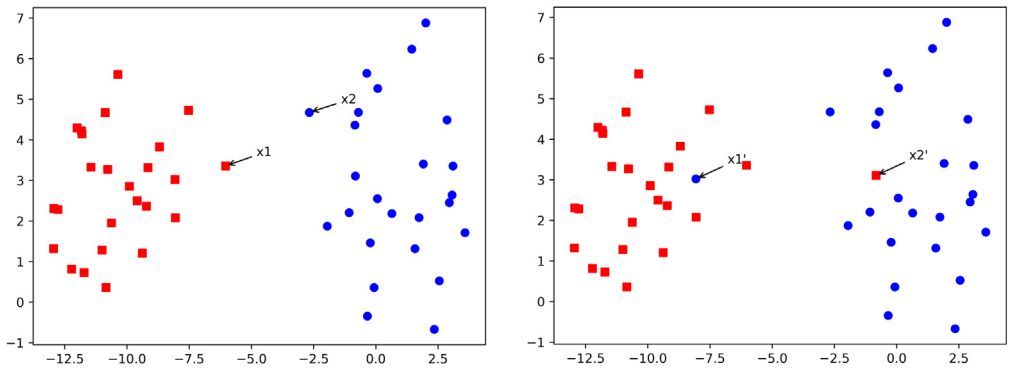
It can be seen that due to the existence of two noisy samples, the calculation of lower approximation is severely affected, and the fuzzy dependency degree is decreased significantly. In other words, the fuzzy dependency degree cannot correctly reflect the separability and discriminability of data in the presence of noise. To alleviate the influence of noise, we propose the concept of *bad-points* to preprocess noisy data.

For any sample $x_i \in U$, the kth nearest neighbor and the first k nearest neighbors of x_i are denoted as $NE^k(x_i)$ and $NS_k(x_i) = \{NE^1(x_i), NE^2(x_i), \ldots, NE^k(x_i)\}$, respectively. The decision of x_i and the decision of the kth neighbor of x_i are denotes as $ND(x_i)$, and $ND^k(x_i)$, respectively. Then, the sample x_i is considered to be the bad-points by the following 0–1 function:

$$BD(x_i) = \begin{cases} 1, & |ND_k(x_i)| < k(1-\eta) \\ 0, & otherwise \end{cases} \tag{14}$$

where $ND_k(x_i) = \bigcup\{ND^j(x_i) | ND^j(x_i) = ND(x_i), 1 \le j \le k\}$, denoting the set of neighbor samples that have the same decision as x_i.

On basis of the concept of bad-points, we can remove potential noise samples from data. For the data in Fig. 2. (b), the points x_1' and x_2' are considered to be *bad-points* when the parameters are set to $k = 9$ and $\eta = 0.25$, and then the fuzzy dependency is



(a) data without noise (b) data with noise

Fig. 2. A synthetic binary classification data set.

computed as

$$\gamma_B(D) = \frac{1}{n-2}\left(\sum_{x_i \in C_1,\ x_i \neq x_2'} (1 - R(NM(x_i), x_i)) + \sum_{x_j \in C_2, x_j \neq x_1'} (1 - R(NM(x_j), x_j))\right), \quad (15)$$

where $NM(x_i) \neq x_2'$, and $NM(x_j) \neq x_1'$.

Obviously, by removing noise from data, the fuzzy dependency can reflect the data more accurately and reasonably. However, in the aforementioned weighted fuzzy dependency, only one heterogeneous sample $NM(x_i)$ of sample x_i is considered, which may not fully capture the neighborhood structure information of samples. To improve the weighted fuzzy dependency, we further explore the high-order margin of samples.

Let $NM_j(x_i)$ be the jth heterogeneous samples closest to x_i. Then the r-order weighted fuzzy dependency is defined as:

$$\gamma_B^{w,r}(D) = \frac{1}{n * r}\sum_{i=1}^{n}\sum_{j=1}^{r} \times (1 - exp(-\frac{w(NM_j(x_i))^2 \|x_i - NM_j(x_i)\|^2}{\sigma})). \quad (16)$$

The r-order weighted fuzzy dependency of a sample x_i reflect the average margins between the r nearest misses and hits. The larger the r, the more neighborhood samples are considered, and the more neighborhood structure information is reflected. With the concepts of *bad-points* and high-order margin, we can define a robust weighted fuzzy dependency as

$$\gamma_B^{w,r}(D) = \frac{1}{(n - |BD(U)|) * r}\sum_{i=1}^{n-|BD(U)|}\sum_{j=1}^{r} \times (1 - exp(-\frac{w(NM_j(x_i))^2 \|x_i - NM_j(x_i)\|^2}{\sigma})), \quad (17)$$

where $BD(U)$ is the set of *bad-points*.

In the definition of the robust weighted fuzzy dependency, the potential noises are not considered, and the high-order neighborhood information is employed to reflect the weighted sample margin. To improve the margin, we can maximize the proposed robust weighted fuzzy dependency by optimizing the weight of samples. In machine learning, a loss function is often defined to optimize the objective function [47]. The margin loss of all samples in U can be defined as

$$L(U) = \frac{1}{n}\sum_{i=1}^{n} l(\theta(x_i)), \quad (18)$$

where $\theta(x_i)$ represents the margin of sample x_i, and $l(\theta(x_i))$ denotes the margin loss of the sample x_i.

Different loss functions are used in different situations. For example, SVM uses hinge loss, and AdaBoost uses exponential loss [47]. Other commonly used loss functions are the linear loss function: $l(\theta) = 1 - \theta$, the logistic loss function $l(\theta) = log(1 + exp^{(-\theta)})$, the exponential loss function $l(\theta) = exp^{\frac{-\theta^2}{\mu}}$, and the surrogate loss function $l(\theta) = exp^{(-\theta)} - \theta$. It has been shown that the performance of the above four loss functions is approximately the same [47]. In this paper, we select the simplest linear loss function. Therefore, the overall loss is defined as

$$Loss(U) = 1 - \gamma_B^{w,r}(D). \quad (19)$$

The loss function not only depends on the sample weight w but also on $NM(x)$. $NM(x)$ may change during the optimization of sample weights, and the optimization process of loss function will be cumbersome [52]. Therefore, we assume that in the subtle change of sample weights, the neighbors of a sample remain the same. Under the above assumption, we know that the loss function $Loss(U)$ is smooth, and the gradient descent algorithm can be used to optimize sample weights to minimize the loss. In the iterative process of the gradient descent algorithm, the sample weight $w(NM_k(x_i))$ is updated by

$$w(NM_k(x_i)) = w(NM_k(x_i)) - lr * \frac{\partial Loss}{\partial NM_k(x_i)}, \quad (20)$$

where lr represents the learning rate and can be set a fixed value for all samples.

Given a data set, the sample weighting algorithm with the objective of minimizing overall loss can be described by Algorithm 1.

In Algorithm 1, the weights of all samples are first initialized to 1, and the technique of *bad-points* is used to remove possible noises from the original data. On the noise-free data, high-order neighborhood information is exploited to update the weight of each sample according to its gradient. The iterative optimization process is terminated if the loss increment is less than ε. Finally, a weight vector of samples is obtained.

The time complexity of finding bad-points and constructing RWLA classifier are $O(n^2)$ and $O(rn^2T)$, where r is the order of neighborhood information, T is the number of iterations, and n is the number of samples.

Based on the sample weights, a Robust Weighted Fuzzy Lower Approximation (RWFLA) classifier can be defined. For any sample $x_i \in U$, the prediction of x_i is defined as $RWFLA(x_i) = argmax_{d_j \in U/D}(\underline{R}d_j(x_i))$, where $\underline{R}d_j(x_i)$ is computed as the averaged lower approximation of sample x_i to the equivalence class d_j using the sample weights. To further improve the generalization and

robustness of the proposed classifier, the weighted average lower approximation is defined as

$$\underline{R}d_j(x_i) = \frac{\sum_{k=1}^{K} w(NM_k(x_i))\,(1 - R(NM_k(x_i), x_i)))}{\sum_{k=1}^{K} w(NM_k(x_i))}, \tag{21}$$

where $NM_k(x_i)$ represents the weight of the kth closest heterogeneous sample to sample x_i, and, similar to KNN, the parameter K can be set to 3.

Algorithm 1 Sample weighting by minimizing loss of high-order margin

Input:
A labeled data $FDS = (U, A = C \cup D, V, f)$, the number of nearest samples in *bad-points* k, the threshold parameter η, the number of high-order information r;
Output:
A vector **w** of sample weights;
Initialize sample weights **w**$=<1, 1, \ldots, 1>$, $loss = 0$, learning rate $lr = 0.1$, loss increment threshold $\varepsilon = 0.001$;
Remove noises from U by using the technique of *bad-points*, and form a noise-free data U';
For $\forall x \in U'$, compute r nearest miss and hit samples of x, $NS'(x) = \{NM_1(x), NH_1(x), \ldots, NM_r(x), NH_r(x)\}$, and calculate the kernel parameter σ according to (7);
Compute the loss with U' and sample weights **w** using (19) and (20), $loss = Loss(U')$;
while *True* **do**
  for $i = 1, 2, \ldots, n$ **do**
    for $j = 1, 2, \ldots, r$ **do**
      $w(NM_j(x_i)) = w(NM_j(x_i)) - lr * \frac{\partial loss}{\partial NM_j(x_i)}$;
    end for
  end for
  $loss' = Loss(U')$;
  if $|loss - loss'| < \varepsilon$ **then**
    break;
  else
    $loss = loss'$;
  end if
end while
return the sample weights **w**

3.3. Robust tri-training with fuzzy rough subspaces

The proposed weighted fuzzy rough set model could effectively handle vague, uncertain, or imprecise data with noise. However, it cannot be directly applied to the problem of semi-supervised learning. Therefore, we combine the proposed weighted fuzzy rough set model with tri-training, which is an effective algorithm in semi-supervised learning, and propose a robust fuzzy rough sets-based tri-training model.

Due to the resampling technique, the initial base classifiers in tri-training tend to have high redundancy and low diversity, thus affecting the quality of unlabeled data learned by the model. For this reason, we explore multi-views of the data to train the base classifiers. Data transformation, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), is a technique of converting data into different attribute spaces. Different from the original attribute space, the transformed attribute space is more informative and discriminative, and its description ability for data is also retained. Additionally, discretization is another form of data transformation. It abstracts and generalizes data into different levels of granulation, which is conducive to dealing with complicated data. Therefore, we use the technique of PCA transformation and data discretization to generate two different views of data. The two views along with the original view of data are employed to train the base classifiers. Since these views describe the data in different spaces or levels of granulation, the trained base classifiers with high diversity and better performance could be achieved after exploiting unlabeled data. For the construction of the base classifiers, tri-training can select the distance-based and discriminative classifiers such as k-Nearest Neighbors, Decision Tree, or Naive Bayes. However, practical data may be contaminated by noise and also be entangled with uncertainty. To tackle these problems, we use the proposed RWFLA classifier to train base classifiers in tri-training. As a result, the robust and diverse base classifiers could effectively capitalize on unlabeled data to improve performance. The fuzzy rough sets-based tri-training model can be described by Algorithm 2.

Algorithm 2 Fuzzy rough sets-based tri-training for partially labeled data

Input:
A partially labeled data $PS = (U = L \cup N, A = C \cup D, V, f)$;
Output:
A combined classifier h;
Preprocess the original data PS and obtain three view data: *ORI*, *PCA*, and *DIS*;
Use labeled data L in the three views to train three base classifiers h_1, h_2, h_3, and initialize the updating state, the error rate, the set of unlabeled data, and the number of useful unlabeled data of the three base classifiers $update_1 = update_2 = update_3 = True$, $e_1 = e_2 = e_3 = 0.5$, $N_1 = N_2 = N_3 = N$, and $l'_1 = l'_2 = l'_3 = 0$;
while *True* **do**
  for $i = 1, 2, 3$ **do**
    $L'_i = \emptyset$, $update_i = False$;
    Compute the error e'_i of h_j and h_k on $L(j, k \neq i)$;
    if $e'_i < e_i$ **then**
      Select samples L'_i from N_i according to the classifiers h_j and h_k;
      if l'_i *is* 0 **then**
        $l'_i = \left\lfloor \frac{e'_i}{e_i - e'_i} + 1 \right\rfloor$;
      end if
      if $l'_i < |L'_i|$ **then**
        if $e'_i * |L'_i| < e_i * l'_i$ **then**
          $update_i = True$;
        else if $l'_i > \frac{e'_i}{(e_i - e'_i)}$ **then**
          Remove $\left\lceil \frac{e_i * l'_i}{e'_i} - 1 \right\rceil$ samples from L'_i randomly;
          $update_i = True$;
        end if
      end if
    end if
  end for
  if $update_1 = False$ and $update_2 = False$ and $update_3 = False$ **then**
    break;
  end if
  for $i = 1, 2, 3$ **do**
    if $update_i$ *is True* **then**
      Update h_i with $L_i \cup L'_i$, $e_i = e'_i$, $l'_i = |L'_i|$;
    end if
  end for
end while
Combine h_1, h_2, and h_3 into a new classifier h;
return h

In the algorithm, two views of the original data are first generated through principal component analysis and discretization. Then, these two views along with the original view are used to train three robust weighted lower approximation classifiers on initial labeled data. On unlabeled data, the three base classifiers iteratively learn from each other to improve the performance. Specifically, for unlabeled data of one classifier, the counterpart two classifiers try to predict the class labels of each sample, and the unlabeled samples with the same predicted result are

considered as the augmented data for the classifier. In each iteration of tri-training, the error rate of one classifier is estimated on initial labeled data. If the performance of the classifier is improved after learning from unlabeled data determined by the other two classifiers, then this classifier can be further learned from unlabeled data; otherwise, it will stop updating. In the early iterations of tri-training, one classifier may have a large number of unlabeled data selected by its counterpart classifiers. According to the noise learning theory, a weak classifier can be boosted to a strong classifier after learning from a certain number of samples with noise. Although there may have a large number of useful unlabeled samples, one classifier is restricted to only learn from a certain number of noisy samples that do not degrade its performance. Therefore, some unlabeled data are randomly discarded in the algorithm, and the classifier is only updated on the remaining certain amount of unlabeled data. The process of tri-training is terminated if all classifiers are not updated, and the refined base classifiers on unlabeled data are combined into a final classifier.

Assume a partially labeled data has $n = |U|$ samples, where $l = |L|$ samples are labeled. According to Algorithm 1, the time complexity of constructing a RWFLA classifier is $O(rn^2T)$, where r is the order of neighborhood information, T is the number of iterations. Therefore, the time complexity of training a base classifier on labeled data is $O(rl^2T)$. In the worst case, each classifier can only learn one unlabeled sample in each iteration. Thus, the time complexity for learning from unlabeled data is at most $O(rl^2(n-l)T)$. The overall time complexity of Algorithm 2 is $O\left(rl^2T\right) + O(rl^2(n-l)T)$, which is approximate to $O(rl^2(n-l)T)$, and the space complexity is $O(n)$.

4. Experimental analysis

In this section, we first show the effect of sample weights on model performance through two artificial data sets and a real UCI data set. Then, we conduct extensive experiments to verify the effectiveness of the proposed fuzzy rough set model on real UCI data sets, particularly, in the case of noise. Finally, we perform validation experiments of our proposed model for partially labeled data, especially in the field of medical diagnosis.

4.1. The impact of sample weight on model performance

To show the impact of sample weights on the performance of distance-based learning models, we synthesized a two-dimensional Gaussian data set as shown in Fig. 3, where there are two classes centered at $(-5, -5)$ and $(5, 5)$, respectively, each of which has 500 samples with a standard deviation of 5.

In Fig. 3, the samples from two classes are intertwined, and even some samples seem to be noises due to high class variances. Generally speaking, we can divide samples into three groups, and they are boundary, external, and internal one in terms of the distance between the sample to its nearest miss. To carry out an in-depth analysis of the effect of sample weights, we intentionally manipulate the weights of different types of samples and observe the performance of learning models on the weighted data. More specifically, for each class, we first compute the distance from each sample to its nearest miss and obtain the maximum miss distance of the class. Then, we set two quantile points of 0.25 and 0.75 to divide the samples. When the miss distance of a sample is less than 0.25 times the maximum miss distance of its class, the sample is considered as the boundary one; and when the miss distance of a sample is greater than 0.75 times the maximum miss distance of its class, the sample is regarded as the external one; Whereas the remaining samples in the class are classified as the internal ones. Meanwhile, we set the range of sample weights to [0-100] and use the quantile points of 25 and 75 to divide the range of sample weights into 3 intervals, namely [0,25], (25,75), and [75,100]. We enumerate the combination of sample types and weight intensities and obtain 6 valid combinations of weight intensities for the boundary, internal, and external samples of each class, i.e., 1:([0,25], (25,75), [75,100]), 2:([0,25], [75,100), (25,75)), 3:((25,75), [0,25], [75,100]), 4:((25,75), [75,100], [0,25]), 5:([75,100], (0,25), [25,75]), and 6:([75,100], (25,75), [0,25]). Finally, we randomize each type of sample with the weights from the corresponding range and use the KNN classifier ($k = 3$) and Fuzzy Lower Approximate (FLA) classifier to evaluate the performance of the weighted data set. The technique of 10-fold cross-validation is employed to ensure the effectiveness of the experiments, and experimental results are shown in Tables 2 and 3.

In Tables 2 and 3, columns 1 to 6 indicate the error rates of classifiers under different sample weights. Their mean and variance are listed in the 7th and 8th columns, respectively.

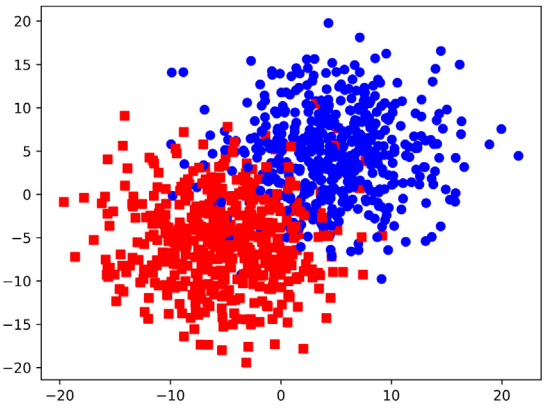


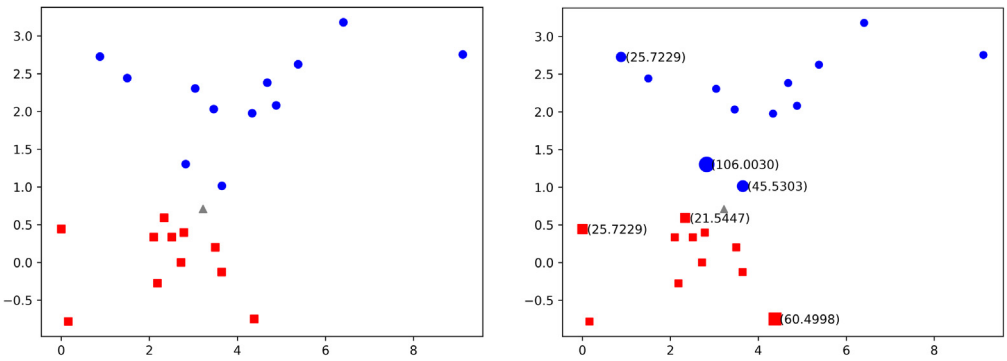
Fig. 3. A synthetic data set.

Table 2
The error rate of the KNN classifier on the synthetic data set with different weights (%).

	1	2	3	4	5	6	Avg.	Variance
1	10.00	10.00	9.00	8.00	9.00	9.00	9.17	0.01
2	10.00	5.00	10.00	8.00	10.00	9.00	8.67	0.04
3	11.00	11.00	12.00	9.00	10.00	8.00	10.17	0.02
4	7.00	6.00	4.00	6.00	5.00	6.00	5.67	0.01
5	7.00	7.00	9.00	6.00	8.00	5.00	7.00	0.02
6	8.00	7.00	7.00	6.00	7.00	5.00	6.67	0.01
7	9.00	9.00	6.00	6.00	6.00	8.00	7.33	0.02
8	5.00	5.00	8.00	5.00	8.00	5.00	6.00	0.02
9	12.00	15.00	11.00	12.00	10.00	8.00	11.33	0.05
10	9.00	7.00	6.00	7.00	6.00	6.00	6.83	0.01
Avg.	8.80	8.20	8.20	7.30	7.90	6.90	7.88	0.00

Table 3
The error rates of the FLA classifier on the synthetic data set with different weights (%).

	1	2	3	4	5	6	Mean	Variance
1	6.00	8.00	8.00	7.00	7.00	7.00	7.17	0.01
2	10.00	9.00	10.00	9.00	7.00	9.00	9.00	0.01
3	11.00	11.00	13.00	11.00	12.00	11.00	11.50	0.01
4	5.00	6.00	6.00	6.00	6.00	4.00	5.50	0.01
5	6.00	6.00	6.00	4.00	5.00	6.00	5.50	0.01
6	5.00	6.00	6.00	8.00	7.00	8.00	6.67	0.01
7	9.00	9.00	10.00	8.00	8.00	8.00	8.67	0.01
8	6.00	5.00	6.00	6.00	7.00	6.00	6.00	0.00
9	13.00	12.00	12.00	10.00	10.00	9.00	11.00	0.02
10	5.00	5.00	4.00	6.00	4.00	5.00	4.83	0.01
Avg.	7.60	7.70	8.10	7.50	7.30	7.30	7.58	0.00



(a) Samples without weights (b) Samples with learned weights

Fig. 4. A simple binary classification data set.

Each row in the table represents the results of one-fold cross-validation, and the last row "Avg." shows the average error rates of the classifier across 10-fold cross-validation.

It can be seen from Tables 2 and 3 that different sample weights have different effects on the performance of the classifiers. The KNN classifier has relatively worse performance when the external samples have large weights (see columns 1, 2, 3, and 5 in Table 2), and has good performance when both the internal samples and the boundary samples are attached large weights (see columns 4 and 5 in Table 2). Particularly, The KNN classifier achieves the best performance when the boundary samples have large weights, and the external samples have small weights. Intuitively, the boundary samples are those samples that lie on the decision boundary and are difficult to classify. Thus, the classifier should pay more attention to these samples. The internal samples are representative ones that are near the center regions of different classes. They reflect the structure information of data so that the learning model should take into consideration these samples. Whereas the external samples are far away from the decision boundary and take less effect on the learning model such that smaller weights can be assigned. For the FLA classifier, a similar tendency has been shown on the given data set. But the average error rate and the variance of the FLA classifier are generally smaller than that of the KNN classifier. A possible explanation is that the computation of lower approximation in FLA is more robust to noise than the computation of neighbor in KNN. To sum up, sample weights have a substantial effect on the performance of the learning model, and better results can be achieved when larger weights are imposed on the non-external samples, especially the boundary samples. In order to further illustrate the effect of boundary samples with weights, we generated a simple binary classification data set as shown in Fig. 4, where there are only 24 samples described by two attributes.

As shown in Fig. 4, the gray triangle point is a sample to be classified or predicted, which is located in the boundary region of two classes and is difficult for a discriminative model to decide the class. For the fuzzy rough sets-based classifier, the lower approximations of the gray triangle sample to the red rectangle class and blue circle class are 0.6041 and 0.6155, respectively. However, from the perspective of geometric structure, the red rectangle samples are more compact, and the blue circle samples are relatively high scattered. After weighting all samples (see Fig. 4.(b), where we only show the sample weights that are greater than the initial value 1), the boundary samples are assigned large weights. Specifically, the weight of the blue cycle sample located in the upper right corner of the gray triangle sample is increased from 1 to 45.5303. Accordingly, the lower approximation of the gray triangle sample to the red square class is $45.5303 * 0.6041 = 27.5049$. The weight of the nearest red square sample at the bottom left of the triangle sample is still 1. Thus, the lower approximation of the gray triangle sample to the blue circle class is $1 * 0.6155 = 0.6155$. The lower approximation of the gray triangle sample to the red square class becomes larger, so it will be correctly classified.

Table 4
The error rates on the data set wine (%).

	FLA	WFLA	KNN	WKNN
1	11.11	0.00	0.00	0.00
2	11.11	5.56	11.11	11.11
3	5.56	5.56	5.56	5.56
4	11.11	11.11	11.11	11.11
5	0.00	0.00	0.00	0.00
6	5.56	5.56	11.11	5.56
7	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00
9	17.65	5.88	5.88	5.88
10	5.88	5.88	5.88	5.88
Avg.	6.80	3.95	5.07	4.51

Additionally, we conducted a comparative experiment on a real data set "wine". The data set contains a total of 178 samples described by 13 attributes. These samples are from 3 classes and the number of samples in each class is 59, 71, and 48, respectively. In the experiment, the sample weighting algorithm is first used to assign weights for each sample, and then different classifiers are trained on the data set with weighted samples. To ensure the reliability of the results, we performed 10-fold cross-validation experiments, and the results are shown in Table 4.

In Table 4, each row represents one-fold in the 10-fold cross-validation. The columns FLA, WFLA, KNN, and WKNN represent the lower approximate classifier, weighted lower approximate classifier, KNN classifier, and weighted KNN classifier, respectively. The average performance of the classifiers over the 10-fold cross-validation is shown in the last row "Avg."

It is observed that the performance of the weighted approximation classifier and the weighted KNN classifier are better than that of their corresponding none-weighted classifiers. On the data set without sample weights, the performance of the lower approximate classifier FLA is worse than that of the KNN classifier. But after sample weighting by the proposed optimized algorithm, the classification accuracy of the FLA is improved by 41.91% and is, in turn, better than that of the KNN classifier. This may be because the sample weighting algorithm is to optimize the lower approximation classifier, and the obtained sample weights are not necessarily optimal for the KNN classifier. In other words, different classifiers may need slightly different weights to achieve the best performance. These results further show that sample weighting can indeed improve the performance of the classifier.

Table 5
Investigated data sets.

Data sets	$\|U\|$	$\|C\|$	$\|U/D\|$	Missing
autos(autos)	205	26	6	Y
car(car)	1728	6	4	N
glass(glass)	214	10	7	N
iris(iris)	150	4	3	N
sonar(sonar)	208	60	2	N
tic-tac-toe(ttt)	958	9	2	N
breast-cancer-wisconsin-prognostic(wpbc)	198	34	2	Y
ecoli(ecoli)	336	8	8	N
liver-disorders(liver)	345	6	2	N
new-breast-tumor(tumor)	286	8	2	N
parkinsons(parkinson)	195	22	2	N
thoracic-surgery(thoracic)	470	16	2	N

Table 6
The error rates (%) of the selected methods on UCI data sets ($\beta = 0$).

	KNN	CART	LSVM	FLA	WFLA	RWFLA
autos	34.63	**17.56**	33.66	23.90	24.88	28.78
car	10.47	**1.56**	27.95	19.21	26.56	9.95
glass	28.50	32.71	42.52	28.04	**23.36**	27.57
iris	4.67	4.00	**3.33**	4.67	5.33	4.67
sonar	16.35	30.29	23.08	**13.94**	14.90	17.79
ttt	20.77	**12.11**	34.66	56.89	22.55	18.27
ecoli	15.48	19.94	16.07	18.75	17.86	**13.99**
liver	38.84	**33.91**	42.03	40.00	39.71	40.58
tumor	32.17	30.07	**27.62**	34.62	32.52	31.47
parkinson	12.31	11.79	11.28	9.74	9.23	**9.23**
thoracic	18.09	20.85	**15.11**	23.19	20.00	17.23
Wpbc	26.77	32.83	**23.23**	29.80	27.78	26.77
Avg.	21.59	20.64	25.04	25.23	22.06	**20.53**

4.2. The effectiveness of the weighted fuzzy rough sets

To test the effectiveness of the proposed weighted fuzzy rough sets, we selected 12 UCI data sets for experiments, of which 6 data sets are medical diagnosis-related data sets. The details are summarized in Table 5.

In Table 5, the second to fourth columns report the number of samples, condition attributes, and classes in each data set, and the last column indicates whether there are missing values. To facilitate the experiment, the missing values in each data set are filled with the mean or mode according to the attribute type. Note that the last 6 data sets are from the task of medical diagnosis.

The proposed weighted fuzzy lower approximation (WFLA) classifier and robust weighted fuzzy lower approximate (RWFLA) classifier are compared to the original fuzzy lower approximation (FLA) classifier without sample weights, KNN, CART, and LSVM. In RWFLA, we set the threshold $\eta = 0.25$ to remove noise in the 9 nearest neighbors, the high-order information $r = 5$ to optimize the weight of each sample, and the parameter $k = 3$ for prediction. For fair comparison, the parameters in KNN are also set to $k = 3$, and the parameters in other classifiers are initialized by default.

To verify the robustness of the selected methods to noise, different levels of class noise are added to each data set. Specifically, the class noise rate β varies from 0 to 0.2 with a step size of 0.05. Under a given class noise rate, a certain number of samples are selected, and their class labels are randomly flipped to other class labels. In the experiments, the technique of 10-fold cross-validation is employed, and the experimental results are shown in Tables 6–10, where the best results among all selected methods are boldfaced.

From Tables 6–10, we can see that the average performance of RWFLA on all data sets is better than other methods under different levels of class noise. Specifically, when the noise rate β is 0, namely the original data without noise, the classifier of CART achieves the best results on 4 of 12 data sets. Although RWFLA only achieves the best results on 2 data sets, its average performance on all data sets is higher than all comparison methods.

Table 7
The error rates (%) of the selected methods on UCI data sets ($\beta = 0.05$).

	KNN	CART	LSVM	FLA	WFLA	RWFLA
autos	61.46	58.05	**55.61**	59.51	60.49	58.05
car	38.72	44.39	41.61	53.01	46.01	**37.27**
glass	28.97	35.98	42.52	30.84	**25.70**	28.97
iris	26.67	33.33	**20.00**	29.33	28.67	22.00
sonar	32.69	37.50	32.21	35.1	34.62	**32.21**
ttt	19.94	**11.69**	34.66	57.41	23.38	19.00
ecoli	14.29	19.94	16.37	17.86	18.15	**13.99**
liver	39.71	38.26	42.61	40.0	**36.52**	40.00
tumor	30.42	30.42	**27.62**	33.92	30.07	30.42
parkinson	28.21	31.79	26.15	29.23	**25.64**	26.67
thoracic	28.72	35.32	27.66	31.49	31.06	**27.66**
wpbc	36.36	41.41	**34.34**	38.38	35.86	37.88
Avg.	32.18	34.84	33.45	38.01	33.01	**31.18**

Table 8
The error rates (%) of the selected methods on UCI data sets ($\beta = 0.1$).

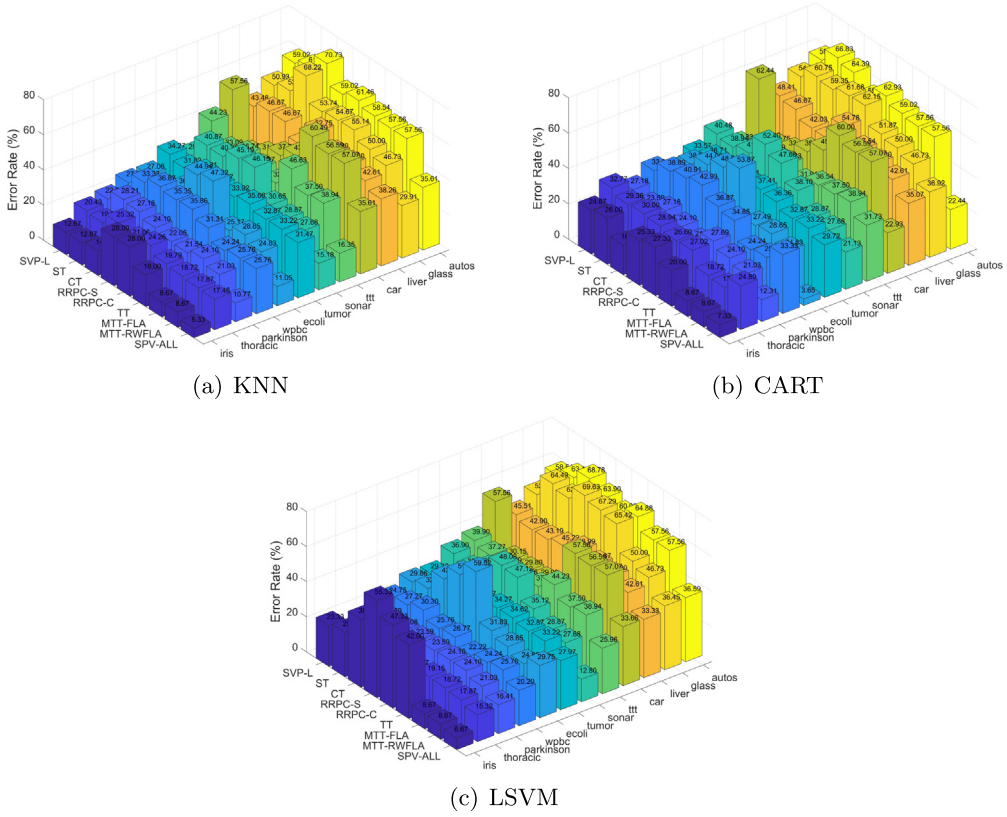
	KNN	CART	LSVM	FLA	WFLA	RWFLA
autos	67.32	**62.93**	67.32	63.41	65.85	64.39
car	57.64	61.63	**53.12**	67.94	62.56	56.13
glass	50.93	57.48	56.54	57.48	51.87	**49.07**
iris	30.67	29.33	22.67	32.67	30.67	**26.67**
sonar	44.71	44.71	**41.35**	44.71	45.67	44.23
ttt	43.84	**43.32**	48.33	55.64	46.14	43.74
ecoli	39.58	49.40	38.10	45.83	42.26	**36.9**
liver	46.09	**42.61**	46.96	44.44	46.48	44.07
tumor	45.10	47.90	41.96	44.41	44.41	**41.96**
parkinson	35.38	32.31	36.92	36.41	33.33	**32.31**
thoracic	36.38	39.36	36.17	36.17	38.30	**34.26**
wpbc	45.45	45.45	46.97	45.45	**40.40**	43.94
Avg.	45.26	46.37	44.70	47.88	45.66	**43.14**

Table 9
The error rates (%) of the selected methods on UCI data sets ($\beta = 0.15$).

	KNN	CART	LSVM	FLA	WFLA	RWFLA
autos	72.68	74.15	**71.22**	75.12	76.59	73.17
car	64.53	67.48	**63.83**	70.54	65.39	64.70
glass	70.56	76.17	74.77	73.83	71.03	**69.63**
iris	39.33	36.67	42.00	35.33	**32.00**	36.00
sonar	**35.58**	46.63	49.04	39.90	41.83	37.50
ttt	46.45	**44.68**	46.24	55.64	47.91	50.31
ecoli	53.57	62.20	**48.81**	64.29	61.31	57.14
liver	**46.11**	48.52	48.89	46.30	47.78	47.22
tumor	48.95	47.20	**44.06**	52.80	53.50	46.15
parkinson	40.51	47.18	38.97	41.03	41.03	**37.95**
thoracic	40.85	41.70	38.51	41.91	40.64	**38.51**
wpbc	43.94	48.99	42.93	**39.39**	46.97	42.42
Avg.	50.26	53.46	50.77	53.01	52.16	**50.06**

Table 10
The error rates (%) of the selected methods on UCI data sets ($\beta = 0.2$).

	KNN	CART	LSVM	FLA	WFLA	RWFLA
autos	78.54	74.15	75.61	75.12	**72.68**	76.10
car	68.40	68.52	**62.67**	71.82	69.73	66.15
glass	**67.29**	78.04	69.16	71.50	72.43	71.50
iris	39.33	42.00	37.33	37.33	37.33	**36.00**
sonar	49.52	51.44	50.00	47.12	47.12	**46.15**
ttt	48.43	48.12	49.69	54.49	49.48	**46.76**
ecoli	66.37	74.11	**60.71**	72.32	70.54	66.07
liver	47.54	50.43	49.28	48.99	46.96	**46.38**
tumor	48.25	**47.55**	49.65	52.10	52.10	47.90
parkinson	**37.95**	38.46	44.10	45.13	44.62	40.51
thoracic	44.04	43.19	42.13	43.83	44.26	**41.70**
wpbc	45.45	45.96	49.49	44.44	50.00	**43.94**
Avg.	53.43	55.16	53.32	55.35	54.77	**52.43**



(a) KNN (b) CART

(c) LSVM

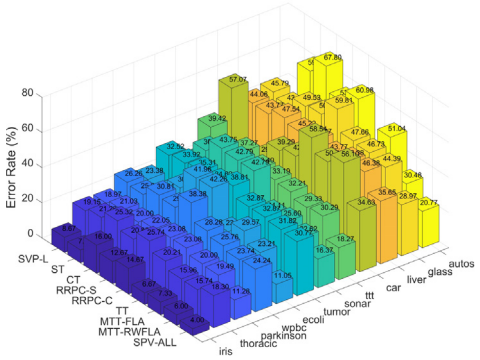
Fig. 5. The error rates (%) of the selected methods under the noise rate $\beta = 0.1$ (label rate $\alpha = 0.1$).

Among the three fuzzy rough sets-based classifiers FLA, WFLA, and RWFLA, FLA does not use sample weights and cannot capture sample structure information well, so its performance is worse. WFLA considers sample weights in computing lower approximation, which improves the performance by 13.00% over FLA. For RWFLA, it uses the technique of data editing to remove potential noise and takes into consideration the high-order neighborhood information of samples, thereby obtaining sample weights that can better reflect the sample distribution, which improves the performance by 18.00% over FLA. As the noise rate increases, the performance of both classifiers degrades drastically, but RWFLA decreases relatively slow and always achieves the best average results under different levels of noise. Take the noise rate $\beta = 0.10$ as an example, RWFLA gains the best performance on 6 of 12 data sets. Compared with other methods, RWFLA achieves an improvement of 4.68% over KNN, an improvement of 6.97% over CART, and an improvement of 3.49% over SVM. Also, RWFLA is improved over FLA and WFLA by 9.90% and 5.52%, respectively. These results further verify the effectiveness of the proposed weighted fuzzy rough set model.

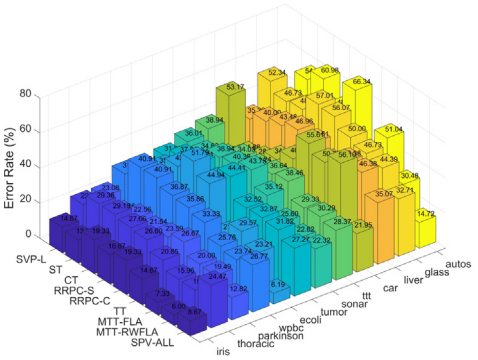
4.3. The effectiveness of the improved tri-training model and its application to medical diagnosis

To learn from partially labeled data, a weighted fuzzy rough sets-based multi-view tri-training is proposed, which takes into consideration both labeled data and unlabeled data. To verify the effectiveness, the proposed method is compared with some methods, including the supervised methods based on labeled data L (denoted by SPV-L), the semi-supervised self-training, co-training, tri-training, and RRPC (denoted by ST, CT, TT, and RRPC, respectively), and the supervised methods based on labeled data L and unlabeled data U with ground truth (denoted by SPV-ALL). Among them, RRPC is a semi-supervised attribute reduction algorithm using Pearson's correlation coefficient with the objective of max-relevance and min-redundancy. It selects important attributes that have strong label correlations in the supervised case and avoids attribute redundancy in the semi-supervised case. For RRPC, we also use the methods of self-training and co-training to train classifiers in the selected attribute set, denoted by RRPC-S and RRPC-C, respectively. The parameter settings of these methods are the same as Section 4.2. In the proposed method, a discretized view of data is employed for tri-training. In the experiments, we performed the discretization of equal frequency binning with three bins on numerical attributes. Specifically, all samples are sorted in ascending order by their attribute values, the first 1/3 of the samples are set to 0, the middle 1/3 is set to 0.5, and the bottom 1/3 of the samples are set to 1. To test the selected methods under different numbers of initial labeled data and different levels of noise, we set the label rate α to [0.1, 0.2, 0,3] and the noise rate β to [0, 0.05, 0.1, 0.15, 0.2] in the experiments. Also, three different classifiers, i.e., KNN, CART, and LSVM, are used as the base classifier for the selected methods. Given a noise rate $\beta = 0.1$, the experimental results under different label rates are shown in Figs. 5–7, where the results are averaged over 10-fold cross-validation.

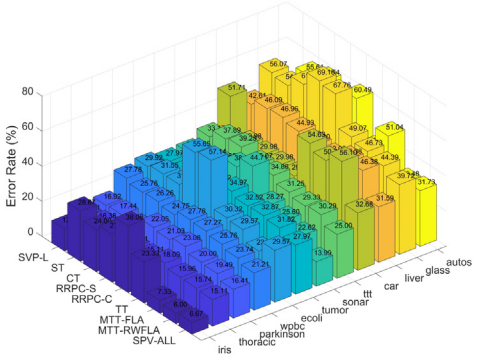
As shown in Figs. 5–7, the robust weighted fuzzy rough sets-based multi-view tri-training (MTT-RWFLA) achieves good performance under different label rates and noise rates. For the supervised methods SPV-L, they only gain mediocre performance. The reason may be the scarcity of labeled data, and consequently, the trained classifier is underfitting and biased. The semi-supervised methods ST, CT, RRPC-S, RRPC-C, and TT take into consideration both labeled data and unlabeled data. However, it is observed that some of their average performance gets worse after learning from unlabeled data, especially in the case of a low label rate. For example, on a label rate $\alpha = 0.1$, the average performance of CT, RRPC-S, RRPC-C, and TT under the KNN, CART,



(a) KNN

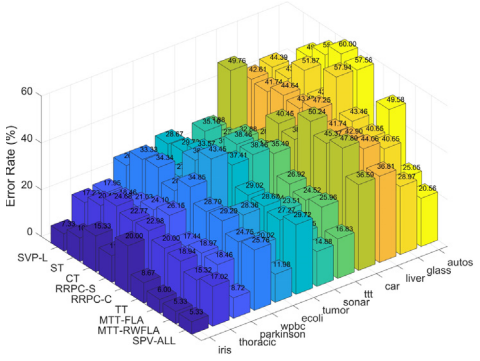


(b) CART

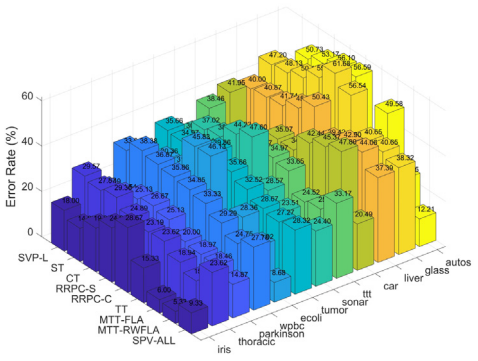


(c) LSVM

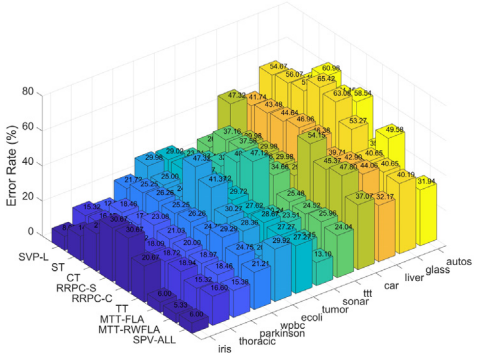
Fig. 6. The error rates (%) of the selected methods under the noise rate $\beta = 0.1$ (label rate $\alpha = 0.2$).



(a) KNN



(b) CART



(c) LSVM

Fig. 7. The error rates (%) of the selected methods under the noise rate $\beta = 0.1$ (label rate $\alpha = 0.3$).

Table 11
The average error rates (%) of the selected methods under the different noise rates ($\alpha = 0.1$, KNN).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	**31.05**	32.01	37.36	34.55	39.79	31.43	32.37	31.25	21.43
$\beta = 0.05$	**32.40**	33.68	38.27	36.08	39.34	33.96	33.81	32.74	21.34
$\beta = 0.1$	35.72	34.45	38.66	38.14	40.20	37.39	34.46	**33.50**	22.73
$\beta = 0.15$	35.93	34.83	40.04	38.77	44.93	36.57	36.28	**34.81**	21.34
$\beta = 0.2$	37.36	37.99	41.61	39.39	43.44	37.19	37.79	**35.77**	22.06

Table 12
The average error rates (%) of the selected methods under the different noise rates ($\alpha = 0.1$, CART).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	33.55	32.03	35.24	37.31	40.94	34.58	32.37	**31.25**	20.26
$\beta = 0.05$	35.26	35.45	39.48	39.54	41.39	35.41	33.81	**32.74**	22.11
$\beta = 0.1$	39.57	39.03	39.20	41.01	42.37	38.87	34.46	**33.50**	23.45
$\beta = 0.15$	39.72	39.64	40.88	40.24	45.35	38.67	36.28	**34.81**	21.84
$\beta = 0.2$	40.47	42.34	43.41	44.80	44.59	39.86	37.79	**35.77**	23.12

Table 13
The average error rates (%) of the selected methods under the different noise rates ($\alpha = 0.1$, LSVM).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	33.20	34.70	38.50	37.81	40.51	34.01	32.37	**31.25**	24.17
$\beta = 0.05$	35.73	35.62	37.63	39.41	41.68	37.05	33.81	**32.74**	23.73
$\beta = 0.1$	35.89	35.50	38.64	41.55	41.52	40.65	34.46	**33.50**	24.59
$\beta = 0.15$	36.35	35.47	37.98	40.02	43.96	37.89	36.28	**34.81**	24.13
$\beta = 0.2$	36.68	37.52	41.39	41.17	42.48	38.31	37.79	**35.77**	23.83

Table 14
The average error rates (%) of the selected methods under the different noise rates ($\alpha = 0.2$, KNN).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	27.97	28.05	32.28	33.12	35.54	29.18	29.72	**27.45**	20.92
$\beta = 0.05$	28.76	30.49	34.79	33.49	36.08	30.21	30.50	**28.07**	20.86
$\beta = 0.1$	30.72	30.55	36.07	35.60	39.04	30.99	31.77	**29.19**	21.19
$\beta = 0.15$	32.69	33.38	36.66	37.26	39.58	34.09	34.30	**31.11**	22.50
$\beta = 0.2$	34.49	34.92	39.74	37.48	41.69	34.09	35.40	**31.72**	22.06

Table 15
The average error rates (%) of the selected methods under the different noise rates ($\alpha = 0.2$, CART).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	28.25	29.59	33.00	35.14	36.02	28.72	29.72	**27.45**	20.93
$\beta = 0.05$	31.32	31.14	33.76	36.86	38.23	30.38	30.50	**28.07**	22.57
$\beta = 0.1$	32.47	33.62	37.41	37.67	40.32	32.90	31.77	**29.19**	21.78
$\beta = 0.15$	37.49	39.19	38.89	38.18	42.19	34.47	34.30	**31.11**	23.13
$\beta = 0.2$	38.47	39.41	41.76	39.80	44.93	37.17	35.40	**31.72**	22.65

Table 16
The average error rates (%) of the selected methods under the different noise rates ($\alpha = 0.2$, LSVM).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	30.03	30.52	34.01	36.32	38.26	31.33	29.72	**27.45**	23.81
$\beta = 0.05$	32.01	32.35	33.80	36.00	37.58	33.28	30.50	**28.07**	24.11
$\beta = 0.1$	31.51	33.41	34.68	37.09	39.56	33.11	31.77	**29.19**	24.30
$\beta = 0.15$	33.22	32.85	35.69	37.60	38.11	34.02	34.30	**31.11**	24.23
$\beta = 0.2$	33.03	33.79	38.44	38.84	41.07	35.15	35.40	**31.72**	25.16

and LSVM classifiers decreases by 4.78%, 8.57%, 11.61%, 5.15%, respectively. These results may be attributed to the low quality of the base classifiers. Self-training is a self-taught algorithm with only one view and is easy to be affected by noise. Co-training is a multi-view paradigm in disagreement-based methods, but its constraint on view is hard to satisfy because most data sets do not have naturally partitioned views. Although RRPC-S and RRPC-C can extract important attributes, the mechanism and strategy of selecting unlabeled samples is not good. Compared with similar algorithms that use all features, their performance is degraded. Tri-training uses the technique of resampling to generate initial labeled data for different base classifiers. The learned classifiers may have high redundancy and low diversity when the number of initial labeled data is small. As a result, tri-training may use unlabeled data with incorrect pseudo labels to update its base classifiers, and thus obtain worse performance. Instead of resampling, the proposed semi-supervised method MTT-RWFLA uses the different views of data to train its base classifiers, where high diversity is preserved. MTT-RWFLA under different label rates achieves the best average performance in comparison with the supervised methods on initial labeled data and other semi-supervised methods. Also, MTT-RWFLA significantly improves on the original fuzzy rough sets-based multi-view tri-training (MTT-FLA), indicating the effectiveness of the robust weighted fuzzy rough sets. On all label rates, the average performance of MTT-RWFLA is improved on SPV-L with the KNN, CART, and LSVM classifiers by 5.62%, 14.50%, and 7.76%, respectively. Compared with other semi-supervised methods, MTT-RWFLA achieves

Table 17
The average error rates (%) of the selected methods under the different noise rates ($\alpha = 0.3$, KNN).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	26.78	27.17	31.18	32.73	34.65	28.64	26.76	**25.10**	21.56
$\beta = 0.05$	27.62	27.18	32.44	33.26	35.83	28.78	28.76	**25.67**	21.58
$\beta = 0.1$	27.60	29.45	33.36	34.61	38.31	27.85	29.73	**26.07**	21.10
$\beta = 0.15$	31.08	31.56	35.36	34.14	38.91	31.59	31.29	**27.40**	22.28
$\beta = 0.2$	32.47	31.72	37.07	36.82	39.86	33.42	34.73	**30.82**	22.93

Table 18
The average error rates (%) of the selected methods under the different noise rates ($\alpha = 0.3$, CART).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	26.65	27.20	29.75	34.11	36.38	27.17	26.76	**25.10**	20.45
$\beta = 0.05$	30.95	30.41	32.79	36.06	37.39	28.91	28.76	**25.67**	22.67
$\beta = 0.1$	31.77	32.67	34.96	38.79	39.97	31.05	29.73	**26.07**	23.22
$\beta = 0.15$	35.41	34.72	37.73	36.64	40.07	32.79	31.29	**27.40**	24.39
$\beta = 0.2$	36.65	37.33	39.68	39.14	41.33	35.40	34.73	**30.82**	24.43

Table 19
The average error rates (%) of the selected methods under the different noise rates ($\alpha = 0.3$, LSVM).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	29.04	29.05	31.39	34.54	35.62	29.66	26.76	**25.10**	24.10
$\beta = 0.05$	29.20	28.84	30.65	34.94	35.44	30.13	28.76	**25.67**	23.95
$\beta = 0.1$	28.83	30.98	33.30	37.07	37.24	30.89	29.73	**26.07**	24.57
$\beta = 0.15$	31.07	30.63	33.46	34.50	38.46	31.25	31.29	**27.40**	24.94
$\beta = 0.2$	30.94	31.25	35.14	36.90	37.50	32.82	34.73	**30.82**	24.74

Table 20
The average error rates (%) of the selected methods under all label rates (KNN).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	28.60	29.08	33.61	33.47	36.66	29.75	29.62	**27.93**	21.31
$\beta = 0.05$	29.59	30.45	35.17	34.28	37.08	30.98	31.02	**28.82**	21.26
$\beta = 0.1$	31.35	31.48	36.03	36.12	39.18	32.08	31.99	**29.58**	21.67
$\beta = 0.15$	33.23	33.26	37.35	36.72	41.14	34.09	33.96	**31.11**	22.04
$\beta = 0.2$	34.77	34.87	39.47	37.90	41.66	34.90	35.98	**32.77**	22.35

a performance improvement of 7.76%, 13.67%, 15.18%, 17.88%, 20.44%, and 16.75% over TT-KNN, TT-CART, TT-LSVM, CT-KNN, CT-CART, and CT-LSVM, respectively. And similar and even worse results can be found in ST, RRPC-S, and RRPC-C with different classifiers. It is worth mentioning that the proposed MTT-RWFLA method is favorable when compared to the fully supervised methods on labeled data and unlabeled data with ground truth. For example, on the data set ''iris'', MTT-RWFLA under a label rate $\alpha = 0.3$ achieves better performance than CART-ALL and SVM-ALL. These results show our proposed method MTT-RWFLA can effectively use unlabeled data to improve performance.

To verify the performance of the proposed method in the field of medical diagnosis, the experiments are conducted on some medical tasks shown in the last 6 data sets of Table 5. It is observed that MTT-RWFLA under all label rates is 3.44% higher than L-KNN and 13.75% higher than L-CART. Also, MTT-RWFLA achieves an improvement of 6.61% over TT-KNN, an improvement of 12.47% over TT-CART, and an improvement of 7.29% over TT-LSVM, respectively. Compared with other semi-supervised methods, MTT-RWFLA outperforms ST by 3.70%, and improves over CT, RRPC-S, and RRPC-C by 8.15%, 18.05%, and 19.54%, respectively. Interestingly, on the data sets ''tumor'' and ''thoracic'', the performance of MTT-RWFLA is the same to or even better than that of the full supervised methods ALL-KNN, ALL-CART, ALL-LSVM. These experimental results show that the proposed methods can achieve good results on medical diagnosis data sets as well, which further demonstrate the effectiveness of the proposed method for partially labeled data.

To further test the effectiveness of the proposed methods, we conducted the experiments under different noise rates $\beta \in [0, 0.05, 0.1, 0.15, 0.2]$, the average results are shown in Tables 11–25, where the best results are boldfaced.

For Tables 11–25, as the noise rate increases, the performance of the selected methods decreases accordingly. However, among all selected semi-supervised methods, MTT-RWFLA always achieves the best results under different noise rates. It can be also seen that the overall average performance of the proposed method is slowly degraded as adding more noise to data sets, but its performance improvement over other supervised methods on labeled data and semi-supervised methods is increased gradually. At the same time, as the label rate increases, the selected supervised methods on labeled data and semi-supervised methods achieve better average performance across different noise rates, whereas the proposed methods still gain the best results. These results further verify that the proposed method could effectively deal with partially labeled data under different labeled rates and noise rates.

5. Conclusions

Classic fuzzy rough sets and their extensions are usually used to deal with uncertain labeled data or unlabeled data and are also sensitive to noise. In this paper, we proposed a robust weighted fuzzy rough set model, which utilizes the data editing technique of bad-points to remove noise and considers the high-order neighborhood structure information to maximize sample margin by optimizing sample weights. Moreover, to learn from partially labeled data, we introduced the proposed fuzzy rough sets into tri-training and developed a multi-view tri-training model, in which three diverse base classifiers are trained on different views of data. Experimental results on the selected benchmark and medical data sets show that the proposed method can achieve desirable results in supervised learning with noise and also can

Table 21
The average error rates (%) of the selected methods under all noise rates (KNN).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\alpha = 0.1$	34.49	34.59	39.19	37.39	41.54	35.31	34.94	**33.61**	21.78
$\alpha = 0.2$	30.92	31.48	35.91	35.39	38.38	31.71	32.34	**29.51**	21.51
$\alpha = 0.3$	29.11	29.42	33.88	34.31	37.51	30.06	30.25	**27.01**	21.89

Table 22
The average error rates (%) of the selected methods under all label rates (CART).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	29.48	29.61	32.66	35.52	37.78	30.16	29.62	**27.93**	20.55
$\beta = 0.05$	32.51	32.33	35.34	37.49	39.00	31.56	31.02	**28.82**	22.45
$\beta = 0.1$	34.60	35.11	37.19	39.16	40.89	34.28	31.99	**29.58**	22.82
$\beta = 0.15$	37.54	37.85	39.17	38.35	42.53	35.31	33.96	**31.11**	23.12
$\beta = 0.2$	38.53	39.69	41.62	41.25	43.62	37.48	35.98	**32.77**	23.40

Table 23
The average error rates (%) of the selected methods under all noise rates (CART).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\alpha = 0.1$	37.71	37.70	39.64	40.58	42.93	37.48	34.94	**33.61**	22.16
$\alpha = 0.2$	33.60	34.59	36.96	37.53	40.34	32.73	32.34	**29.51**	22.21
$\alpha = 0.3$	32.29	32.46	34.98	36.95	39.03	31.06	30.25	**27.01**	23.03

Table 24
The average error rates (%) of the selected methods under all label rates (LSVM).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\beta = 0$	30.76	31.42	34.63	36.23	38.13	31.67	29.62	**27.93**	24.02
$\beta = 0.05$	32.31	32.27	34.03	36.78	38.23	33.49	31.02	**28.82**	23.93
$\beta = 0.1$	32.08	33.30	35.54	38.57	39.44	34.88	31.99	**29.58**	24.49
$\beta = 0.15$	33.54	32.98	35.71	37.37	40.18	34.39	33.96	**31.11**	24.44
$\beta = 0.2$	33.55	34.19	38.33	38.97	40.35	35.42	35.98	**32.77**	24.57

Table 25
The average error rates (%) of the selected methods under all noise rates (LSVM).

	SVP-L	ST	CT	RRPC-S	RRPC-C	TT	MTT-FLA	MTT-RWFLA	SPV-ALL
$\alpha = 0.1$	35.57	35.76	38.83	39.99	42.03	37.58	34.94	**33.61**	24.09
$\alpha = 0.2$	31.96	32.58	35.32	37.17	38.92	33.38	32.34	**29.51**	24.32
$\alpha = 0.3$	29.82	30.15	32.79	35.59	36.85	30.95	30.25	**27.01**	24.46

capitalize on unlabeled data to improve performance in the semi-supervised scenario. It has been reported that the diversity of classifiers is beneficial to multi-view models. How to improve the diversity between the base classifiers in the proposed multi-view tri-training model is worthy of further investigation.

CRediT authorship contribution statement

Jinming Xing: Conceptualization, Methodology, Software, Data curation, Data analysis, Writing – original draft. **Can Gao:** Conceptualization, Methodology, Data analysis, Draft modification. **Jie Zhou:** Methodology, Data analysis.

Declaration of competing 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.

Acknowledgments

The authors would like to thank Editor-in-Chief, editor, and anonymous reviewers for their valuable comments and helpful suggestions. This work is supported in part by the National Natural Science Foundation of China (Nos. 61806127, 62076164), the Natural Science Foundation of Guangdong Province, China (No. 2021A1515011861), Shenzhen Science and Technology Program (No. JCYJ20210324094601005), and Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

[1] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (1982) 341–356.
[2] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, Dordrecht, 1991.
[3] Z. Pawlak, A. Skowron, Rudiments of rough sets, Inform. Sci. 177 (2007) 3–27.
[4] C. Gao, Z. Lai, J. Zhou, C. Zhao, D. Miao, Maximum decision entropy-based attribute reduction in decision-theoretic rough set model, Knowledge-Based Syst. 143 (2018) 179–191.
[5] Y.Y. Yao, S.K.M. Wong, A decision theoretic framework for approximating concepts, Int. J. Man. Mach. Stud. 37 (1992) 793–809.
[6] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, Int. J. Gen. Syst. 17 (1990) 191–209.
[7] W. Ding, J. Wang, J. Wang, Multigranulation consensus fuzzy-rough based attribute reduction, Knowl. Based Syst. 198 (2020) 105945.
[8] L. Sun, T. Yin, W. Ding, Y. Qian, J. Xu, Feature selection with missing labels using multilabel fuzzy neighborhood rough sets and maximum relevance minimum redundancy, IEEE Trans. Fuzzy Syst. (2021) http://dx.doi.org/10.1109/TFUZZ.2021.3053844.
[9] L. Sun, T. Wang, W. Ding, J. Xu, Y. Lin, Feature selection using fisher score and multilabel neighborhood rough sets for multilabel classification, Inform. Sci. 578 (2021) 887–912.
[10] W. Xu, Y. Guo, Generalized multigranulation double-quantitative decision-theoretic rough set, Knowl. Based Syst. 105 (2016) 190–205.
[11] W. Li, W. Xu, X. Zhang, J. Zhang, Updating approximations with dynamic objects based on local multigranulation rough sets in ordered information systems, Artif. Intell. Rev. (2021) http://dx.doi.org/10.1007/s10462-021-10053-9.
[12] J. Dai, H. Hu, W.Z. Wu, Y. Qian, D. Huang, Maximal-discernibility-pair-based approach to attribute reduction in fuzzy rough sets, IEEE Trans. Fuzzy Syst. 26 (2018) 2174–2187.
[13] E.C.C. Tsang, D. Chen, D.S. Yeung, X.Z. Wang, J.W.T. Lee, Attributes reduction using fuzzy rough sets, IEEE Trans. Fuzzy Syst. 16 (2008) 1130–1141.

[14] A.M. Radzikowska, E.E. Kerre, A comparative study of fuzzy rough sets, Fuzzy Sets and Systems 126 (2002) 137–155.
[15] J.S. Mi, W.X. Zhang, An axiomatic characterization of a fuzzy generalization of rough sets, Inform. Sci. 160 (2004) 235–249.
[16] X.Z. Wang, Y. Ha, D.G. Chen, On the reduction of fuzzy rough sets, in: 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 2005, pp. 3174–3178.
[17] Q. Hu, D. Yu, W. Pedrycz, D. Chen, Kernelized fuzzy rough sets and their applications, IEEE Trans. Knowl. Data Eng. 23 (2011) 1649–1667.
[18] A. Mieszkowicz-Rolka, L. Rolka, Variable precision fuzzy rough sets, in: Transactions on Rough Sets, Springer, 2004, pp. 144–160.
[19] Q. Hu, Z. Xie, D. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, Pattern Recognit. 40 (2007) 3509–3521.
[20] S. Zhao, E.C.C. Tsang, D. Chen, The model of fuzzy variable precision rough sets, IEEE Trans. Fuzzy Syst. 17 (2009) 451–467.
[21] C. Cornelis, N. Verbiest, R. Jensen, Ordered weighted average based fuzzy rough sets, in: International Conference on Rough Sets and Knowledge Technology, Berlin, Heidelberg, 2010, pp. 78–85.
[22] Y. Lin, Y. Li, C. Wang, J. Chen, Attribute reduction for multi-label learning with fuzzy rough set, Knowl. Based Syst. 152 (2018) 51–61.
[23] J. Dai, J. Chen, Feature selection via normative fuzzy information weight with application into tumor classification, Appl. Soft Comput. 92 (2020) 106299.
[24] L. Sun, L. Wang, W. Ding, Y. Qian, J. Xu, Feature selection using fuzzy neighborhood entropy-based uncertainty measures for fuzzy neighborhood multigranulation rough sets, IEEE Trans. Fuzzy Syst. 29 (2021) 19–33.
[25] W. Ding, J. Wang, J. Wang, Multigranulation consensus fuzzy-rough based attribute reduction, Knowl. Based Syst. 198 (2020) 105945.
[26] W. Ding, S. Chakraborty, K. Mali, S. Chatterjee, J. Nayak, A.K. Das, S. Banerjee, An unsupervised fuzzy clustering approach for early screening of COVID-19 from radiological images, IEEE Trans. Fuzzy Syst. (2021) http://dx.doi.org/10.1109/tfuzz.2021.3097806.
[27] C. Wang, Y. Qian, W. Ding, X. Feng, Feature selection with fuzzy-rough minimum classification error criterion, IEEE Trans. Fuzzy Syst. (2021) http://dx.doi.org/10.1109/tfuzz.2021.3097811.
[28] K. Yuan, W. Xu, W. Li, W. Ding, An incremental learning mechanism for object classification based on progressive fuzzy three-way concept, Inform. Sci. 584 (2022) 127–147.
[29] X. Chen, W. Xu, Double-quantitative multigranulation rough fuzzy set based on logical operations in multi-source decision systems, Int. J. Mach. Learn. Cybern. (2021) http://dx.doi.org/10.1007/s13042-021-01433-2.
[30] W. Xu, J. Yu, A novel approach to information fusion in multi-source datasets: A granular computing viewpoint, Inform. Sci. 378 (2017) 410–423.
[31] W. Xu, W. Li, Granular computing approach to two-way learning based on formal concept analysis in fuzzy datasets, IEEE Trans. Cybern. 46 (2016) 366–379.
[32] W. Ding, C.T. Lin, Z. Cao, Deep neuro-cognitive co-evolution for fuzzy attribute reduction by quantum leaping PSO with nearest-neighbor memeplexes, IEEE Trans. Cybern. 49 (2019) 2744–2757.
[33] W. Ding, W. Pedrycz, I. Triguero, Z. Cao, C.T. Lin, Multigranulation supertrust model for attribute reduction, IEEE Trans. Fuzzy Syst. 29 (2021) 1395–1408, http://dx.doi.org/10.1109/TFUZZ.2020.2975152.
[34] W. Ding, M. Abdel-Basset, H. Hawash, W. Pedrycz, Multimodal infant brain segmentation by fuzzy-informed deep learning, IEEE Trans. Fuzzy Syst. (2021) http://dx.doi.org/10.1109/TFUZZ.2021.3052461.
[35] Q. Hu, L. Zhang, S. An, D. Zhang, D. Yu, On robust fuzzy rough set models, IEEE Trans. Fuzzy Syst. 20 (2012) 636–651.
[36] A. Kolcz, C.H. Teo, Feature weighting for improved classifier robustness, in: The Sixth Conference on Email and Anti-Spam, Mountain View, California, USA, 2009, pp. 1–9.
[37] W. Ding, C.T. Lin, Z. Cao, Shared nearest-neighbor quantum game-based attribute reduction with hierarchical coevolutionary spark and its application in consistent segmentation of neonatal cerebral cortical surfaces, IEEE Trans. Neural Networks Learn. Syst. 30 (2019) 2013–2027.
[38] Y. Lin, Q. Hu, J. Liu, J. Duan, Multi-label feature selection based on max-dependency and min-redundancy, Neurocomputing 168 (2015) 92–103.
[39] L.I. Kuncheva, Fuzzy rough sets: Application to feature selection, Fuzzy Sets and Systems 51 (1992) 147–153.
[40] R. Jensen, Q. Shen, Fuzzy-rough data reduction with ant colony optimization, Fuzzy Sets and Systems 149 (2005) 5–20.
[41] A. Ganivada, S.S. Ray, S.K. Pal, Fuzzy rough sets and a granular neural network for unsupervised feature selection, Neural Netw. 48 (2013) 91–108.
[42] F. Xiao, W. Ding, Divergence measure of pythagorean fuzzy sets and its application in medical diagnosis, Appl. Soft Comput. 79 (2019) 254–267.
[43] J. Hamidzadeh, E. Rezaeenik, M. Moradi, Predicting users' preferences by fuzzy rough set quarter-sphere support vector machine, Appl. Soft Comput. 112 (2021) 107740.
[44] L. Yang, K. Qin, B. Sang, W. Xu, Dynamic fuzzy neighborhood rough set approach for interval-valued information systems with fuzzy decision, Appl. Soft Comput. 111 (2021) 107679.
[45] C. Wang, Y. Huang, W. Ding, Z. Cao, Attribute reduction with fuzzy rough self-information measures, Inform. Sci. 549 (2021) 68–86.
[46] C. Wang, Y. Qian, W. Ding, X. Feng, Feature selection with fuzzy-rough minimum classification error criterion, IEEE Trans. Fuzzy Syst. (2021) http://dx.doi.org/10.1109/TFUZZ.2021.3097811.
[47] Q. Hu, P. Zhu, Y. Yang, D. Yu, Large-margin nearest neighbor classifiers via sample weight learning, Neurocomputing 74 (2011) 656–660.
[48] P. Zhu, Q. Hu, Y. Yang, Weighted nearest neighbor classification via maximizing classification consistency, in: Lecture Notes in Computer Science, vol. 6086 LNAI, 2010, pp. 347–355.
[49] G. Du, J. Zhang, S. Li, C. Li, Learning from class-imbalance and heterogeneous data for 30-day hospital readmission, Neurocomputing 420 (2021) 27–35.
[50] Z. Fan, M. Ni, Q. Zhu, E. Liu, Weighted sparse representation for face recognition, Neurocomputing 151 (2015) 304–309.
[51] E. Arazo, D. Ortego, P. Albert, N.E. O'Connor, K. McGuinness, Unsupervised label noise modeling and loss correction, in: The 36th International Conference on Machine Learning, Long Beach, California, USA, 2019 pp. 465–474.
[52] W. Zhang, W. Ouyang, W. Li, D. Xu, Collaborative and adversarial network for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 3801–3809.
[53] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, D. Meng, Meta-weight-net: Learning an explicit mapping for sample weighting, Adv. Neural Inform. Proces. Syst. 32 (2019) 1–12.
[54] A. Ghosh, A. Lan, Do we really need gold samples for sample weighting under label noise? in: 2021 IEEE Workshop on Applications of Computer Vision, 2021, pp. 3921–3930.
[55] A.T. Kalai, R.A. Servedio, Boosting in the presence of noise, J. Comput. System Sci. 71 (2005) 266–290.
[56] B. Frenay, M. Verleysen, Classification in the presence of label noise: A survey, IEEE Trans. Neural Networks Learn. Syst. 25 (2014) 845–869.
[57] J.E. van Engelen, H.H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2020) 373–440.
[58] N. Mac Parthalin, R. Jensen, Fuzzy-rough set based semi-supervised learning, in: IEEE International Conference on Fuzzy Systems, Taipei, Taiwan, 2011, pp. 2465–2472.
[59] Y. Qian, X. Liang, G. Lin, Q. Guo, J. Liang, Local multigranulation decision-theoretic rough sets, Internat. J. Approx. Reason. 82 (2017) 119–137.
[60] Q. Wang, Y. Qian, X. Liang, Q. Guo, J. Liang, Local neighborhood rough set, Knowl. Based Syst. 153 (2018) 53–64.
[61] Y. Guo, E.C.C. Tsang, W. Xu, D. Chen, Local logical disjunction double-quantitative rough sets, Inform. Sci. 500 (2019) 87–112.
[62] C. Gao, J. Zhou, D. Miao, X. Yue, J. Wan, Granular-conditional-entropy-based attribute reduction for partially labeled data with proxy labels, Inform. Sci. 580 (2021) 111–128.
[63] C. Gao, J. Zhou, D. Miao, J. Wen, X. Yue, Three-way decision with co-training for partially labeled data, Inform. Sci. 544 (2021) 500–518.
[64] X. Chen, G. Yu, Q. Tan, J. Wang, Weighted samples based semi-supervised classification, Appl. Soft Comput. 79 (2019) 46–58.
[65] Z. Ren, R.A. Yeh, A.G. Schwing, Not all unlabeled data are equal: Learning to weight data in semi-supervised learning, Adv. Neural Inform. Process. Syst. 33 (2020) 21786–21797.
[66] Z. Yuan, H. Chen, P. Xie, P. Zhang, J. Liu, T. Li, Attribute reduction methods in fuzzy rough set theory: An overview, comparative experiments, and new directions, Appl. Soft Comput. 107 (2021) 107353.
[67] W. Wu, J. Mi, W. Zhang, Generalized fuzzy rough sets, Inform. Sci. 151 (2003) 263–282.
[68] K. Nigam, R. Ghani, Analyzing the effectiveness and applicability of co-training, in: Proceedings of the 9th ACM International Conference on Information and Knowledge Management, McLean, VA, 2000, pp. 86–93.
[69] Z. Zhou, M. Li, Tri-training: Exploiting unlabeled data using three classifiers, IEEE Trans. Knowl. Data Eng. 17 (2005) 1529–1541.