International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-021-01334-4

ORIGINAL ARTICLE

Accelerating ReliefF using information granulation

Wei Wei[1] · **Da Wang**[1] · **Jiye Liang**[1]

Received: 3 September 2020 / Accepted: 13 April 2021

Abstract
Feature selection is an essential preprocessing requirement when solving a classification problem. In this respect, the Relief algorithm and its derivatives have been demonstrated to be a class of successful feature selectors. However, the computational cost of these algorithms is very high when large-scale datasets are processed. To solve this problem, we propose the fast ReliefF algorithm based on the information granulation of instances (IG-FReliefF). The algorithm uses K-means to granulate the dataset and selects the significant granules among them using the criteria defined by information entropy and information granulation, and then evaluates each feature on the dataset composed of the selected granules. Extensive experiments show that the proposed algorithm is more efficient than the existing representative algorithms, especially on large-scale data sets, and the proposed algorithm is almost the same as the comparison algorithm in terms of classification performance.

Keywords Feature selection · ReliefF · Information granulation · Information entropy

1 Introduction

Learning from high-dimensional data has always been a challenge when using data mining [1, 2]. The use of high-dimensional data has attracted considerable attention in many areas such as image processing, bioinformatics, and financial businesses. However, high-dimensional data usually contain hundreds or thousands of features, some of which are irrelevant and redundant, and which could deteriorate the performance of learning algorithms. Dimension reduction is only able to eliminate the adverse effects caused by high dimensionality. There are two main types of data dimensionality reduction methods, including feature extraction and feature selection. Feature extraction is the process of transforming instances of the original space by linear or non-linear mapping to obtain a small number of new features with better expressiveness for dimensionality reduction, and it has received much attention from scholars [3, 4]. Feature extraction focuses on representing features to transform raw data into features that can be recognized by machine learning algorithms, which can retain all the information in the original feature space, but the mapped new features lack practical meaning. The other method, feature selection [5, 6], aims to select a representative optimal feature subset from the original features. It retains the actual meaning of the original features and is easier to understand. Feature selection can be regarded as an essential preprocessing step in learning tasks, and we focus our attention on it in this paper.

Relief and its derivatives are successful feature selection methods and were first proposed by Kira and Rendell [7]. In this type of method, the weight of each feature is assigned according to the relevance between each feature and the class label. However, the original Relief can only accommodate datasets with binary classes. Kononenko [8] proposed the ReliefF algorithm for multi-classification problems, and then he and Robnik [9] introduced the RReliefF algorithm for the regression problem. Two drawbacks of these Relief algorithms were pointed out by Sun et al. [10]: the first is that the nearest neighbors are defined in the original feature space, which is highly unlikely to be those in the weighted space, and the other is the lack of a mechanism to process outlier data. To overcome these drawbacks, Sun et al.

✉ Jiye Liang
ljy@sxu.edu.cn

Wei Wei
weiwei@sxu.edu.cn

Da Wang
417109539@qq.com

[1] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China

Published online: 28 April 2021

[10, 11] proposed an iterative feature selection algorithm (I-Relief) and logistic I-Relief (LI-Relief), respectively. Based on these studies, Cai et al. [12] proposed a Relief feature selection algorithm based on a local hyperplane (LH-Relief), which can obtain the final feature weight by maximizing the expected interval between instances and the local hyperplane. Zhang et al. [13] proposed a feature selection algorithm based on I-Relief and LH-Relief, named DRNSR-Relief, and then presented logistic local hyperplane-based Relief (LLH-Relief) based on LI-Relief and LH-Relief [14]. In addition, Relief algorithms for semi-supervised learning were investigated [15, 16], and others for multiple-instance learning and imbalanced data classification were proposed [17, 18]. In another solution, the algorithm was combined with a convolutional neural network [19]. In recent years, Relief and its derivative algorithms have found many successful real applications, for example in the medical field [20, 21], and in other fields [22–27].

However, the original Relief algorithms are computationally inefficient in terms of the time they require when processing large-scale datasets, preventing them from being implemented in real applications. To solve this problem, researchers employed a method that randomly selects instances to calculate each feature's score, ReliefF-RS, which could also lead to bias in evaluating the features. Huang et al. [28] further proposed a modified version of ReliefF-RS, named FSSMC, in which a dataset is partitioned into several clusters, and the centers of these clusters are selected to calculate the score of each feature. Because the centers of clusters would be expected to be more representative than the randomly selected instances, Huang's method [28] outperforms ReliefF-RS. However, FSSMC is less efficient than ReliefF-RS, which randomly selects instances, owing to the preprocessing procedure involving clustering.

To improve the efficiency of ReliefF-RS and FSSMC, in this paper, we present a fast ReliefF algorithm based on information granulation (IG-FReliefF). When we process much complex information, we tend to divide much complex information into several simple blocks due to the limitation of cognitive ability, and each block is a granule. This process is called information granulation. Information granulation reflects how we process and store information. Granular computing is essential for solving our problems. It can help us better analyze and solve problems by abstracting and dividing complex problems into several relatively simple problems. There has been much exciting work in information granulation and granular computing [29–33].

The primary basis of our proposed algorithm comes from information granulation. The algorithm granulates a dataset by using K-means algorithms, and among all the granules, those that are significant are selected by an evaluation index that is defined by combining information granularity and the information entropy. Then, the selected granules are combined into a set of instances used to assign weights to the features. Because fewer instances are used to calculate the weights of features than all the instances in a dataset, the proposed algorithm is much more efficient than ReliefF-RS and FSSMC. Besides, the selected instances, which are close to the classification hyperplane, are more useful for evaluating the significance of a feature than those eliminated. Thus, the proposed algorithms' classification accuracy is as good as those obtained by ReliefF-RS and FSSMC, although only the selected instances are employed to evaluate features.

The contributions to this paper are as follows:

- We propose a new sampling method: the original training set is granulated to contain as many granules as possible for selecting instances around the classification boundary to form a new training set, and a small number of instances that we consider valuable are extracted from this training set to guide the feature weighting process of the Relief algorithm.
- We propose a new evaluation criterion, based on information entropy and knowledge granularity, which can be used to evaluate the granulation results and the degree of confusion of each granule.
- Experiments on eight UCI datasets show that the proposed algorithm is more efficient than existing representative algorithms, especially on large-scale datasets, while maintaining almost the same classification performance as its counterpart.

The remainder of this paper is organized as follows: Sect. 2 briefly reviews relevant work, including knowledge granularity, information entropy, Relief, and ReliefF. The fast ReliefF algorithm based on information granulation (IG-FReliefF) is presented in Sect. 3. Section 4 describes experiments that were carried out to verify the performance of the proposed algorithm. Section 5 summarizes the study.

2 Preliminaries

2.1 Knowledge granularity

Knowledge granularity describes the uncertainty of a granular structure and the degree of coarseness of a knowledge structure. Generally speaking, knowledge granulation reflects the ability to distinguish knowledge in a granular space. The smaller the knowledge granularity of a particle space is, the stronger is its ability to distinguish knowledge. Wierman et al. [34] defined knowledge granularity based on the information entropy as follows:

Definition 1 Given an information system $S = (U, A)$, the knowledge granularity of the granular space derived from the attribute set A is defined as

$$G(A) = -\sum_{i=1}^{m} \frac{|X_i|}{|U|} log_2 \frac{|X_i|}{|U|}, \tag{1}$$

where X_i represents the equivalence class generated by the partition of the field U by attribute set A.

Liang et al. [35, 36] used complementary entropy to define a new knowledge granularity:

$$CG(A) = \sum_{i=1}^{m} \frac{|X_i|}{|U|} \left(1 - \frac{|X_i|}{|U|}\right). \tag{2}$$

2.2 Information entropy

In informatics, information entropy (i.e., Shannon entropy), which was introduced in 1948, is a measure of uncertainty proposed by Shannon. The information entropy is a popular measure in machine learning, and this metric has been studied extensively by academics [37, 38]. The information entropy describes the probability of the occurrence of discrete random events. The more orderly (chaotic) a system is, the lower (higher) its information entropy is.

For a classification system $X = (U, A \cup C)$, C is a category variable with values of $C_1, C_2, ..., C_{\mathcal{L}}$, and the probability of each of these categories is $P(C_1), P(C_2), ..., P(C_{\mathcal{L}})$. Further, $\mathcal{L}$ is the number of classes in the classification system, and the information entropy of the system is:

$$H(X) = -\sum_{i=1}^{\mathcal{L}} P(C_i) log_2 P(C_i). \tag{3}$$

For supervised learning, information entropy can be used to evaluate the degree of the chaos of the instance set, that is, the proportion of various categories of data in the instance set. When the information entropy of an instance set is higher, the instance set is considered to be more "chaotic".

2.3 Relief

The Relief algorithm [7] was first proposed by Kira and is restricted to binary classification. The main idea of Relief is to iteratively assign weights to features based on the correlation between features and class labels, which is based on the ability of features to distinguish between neighboring instances of different classes.

In Relief, given a training set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, in which $\mathbf{x}_i$ is an instance, $\mathbf{y}_i$ is the label of $\mathbf{x}_i$, and a randomly selected instance $\boldsymbol{R}_i$ in a certain iteration, and then the nearest neighbor from the instances in the same class of $\boldsymbol{R}_i$ (***Near Hit*** or ***NH***) and the nearest neighbors $\boldsymbol{M}$ from the instances in different classes of $\boldsymbol{R}_i$ (***NearMiss*** or ***NM***) can be obtained. The weights of the features are then updated according to the following rules: (1) If the distance between $\boldsymbol{R}_i$ and ***NH*** on a feature is less than that between $\boldsymbol{R}_i$ and ***NM***, then this feature is helpful to distinguish between instances that are close to each other, and thus the weight of this feature should be increased; (2) If the distance between $\boldsymbol{R}_i$ and ***NH*** on a feature is farther than that between $\boldsymbol{R}_i$ and ***NM***, then this feature is not helpful to distinguish between instances that are near each other, and thus the weight of the feature should be decreased. By repeating the above process m times, the final weight of each feature can be obtained. The greater the weight of a feature, the stronger the classification ability of the feature becomes. The following formula is used to update the weight of a feature j in each iterative cycle:

$$\mathcal{W}_j = \mathcal{W}_j - \frac{diff(j, \boldsymbol{R}_i, \boldsymbol{NH})}{m} + \frac{diff(j, \boldsymbol{R}_i, \boldsymbol{NM})}{m}, j = 1, 2, ..., I \tag{4}$$

where $diff(j, x_u, x_v)$ is the difference between x_u and x_v on feature j, I is the number of features.

For a symbolic feature, $diff(j, x_u, x_v)$ can be calculated by

$$diff(j, x_u, x_v) = \begin{cases} 0; & value(j, x_u) = value(j, x_v), \\ 1; & otherwise. \end{cases} \tag{5}$$

For a numerical feature, $diff(j, x_u, x_v)$ can be obtained by

$$diff(j, x_u, x_v) = \frac{|\, value(j, x_u) - value(j, x_v) \,|}{|\, max(V_j) - min(V_j) \,|}. \tag{6}$$

where V_j is the value set of the j-th feature.

2.4 ReliefF

ReliefF is an extension of Relief that can be used for datasets with multiple classes [8]. In contrast to Relief, in ReliefF, k nearest neighbors of randomly selected instances are used to evaluate a feature, which is useful for dealing with incomplete and noisy datasets and improves the robustness of ReliefF.

ReliefF randomly selects an instance $\boldsymbol{R}_i$ from the training data $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, and k nearest neighbors of $\boldsymbol{R}_i$ and $\boldsymbol{NH}_t, t = 1, ..., k$, are selected from the same class with $\boldsymbol{R}_i$. In addition, k nearest neighbors of $\boldsymbol{R}_i$, $\boldsymbol{NM}_t$, $t = 1, ..., k$, are selected from different classes with $\boldsymbol{R}_i$. The weight of a feature j is updated in each iterative cycle using the following formula:

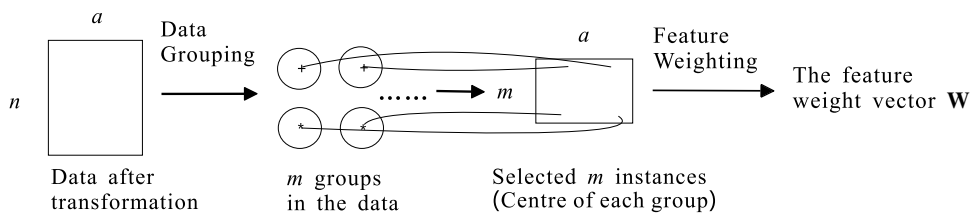


Fig. 1 Work flows of FSSMC

$$\mathcal{W}_j = \mathcal{W}_j - \frac{\sum_{t=1}^{k} diff(j, \boldsymbol{R}_i, \boldsymbol{NH}_t)}{m \times k} + \frac{\sum_{C \neq class(\boldsymbol{R}_i)} [\frac{P(C)}{1-P(class(\boldsymbol{R}_i))} \sum_{t=1}^{k} diff(j, \boldsymbol{R}_i, \boldsymbol{NM}_t(C))]}{m \times k}, j = 1, 2, ..., I, \quad (7)$$

where m is the number of randomly sampled instances, and $P(C)$ is the prior probability of class C (obtained from the training set), I is the number of features. To ensure that the sum of the weights of all the different classes is 1, we use $1 - P(class(\boldsymbol{R}_i))$ to divide the probability of each weight.

3 Fast ReliefF based on information granulation

This section first presents the disadvantages of ReliefF-RS and FSSMC [28], and describes the information granulation-based fast ReliefF algorithm (IG-FReliefF), which is proposed to overcome these shortcomings.

ReliefF-RS is problematic because it requires m (the number of instances that are randomly sampled) to be much smaller than n (the number of instances) to process a dataset with high efficiency when n is very large. However, random selection could result in severe information loss because it cannot make full use of the information of data, and it is difficult to determine how many sampled instances are "sufficient" to assign weights to features.

To address this drawback of ReliefF-RS, FSSMC (whose flowchart is shown in Fig. 1) improves the method of sampling instances by dividing the original dataset into m clusters and then extracting the centers of these m clusters to perform ReliefF-RS. Although the FSSMC algorithm has contributed to solving the random sampling problem, the following two problems still exist.

1. The FSSMC algorithm is an improvement of the ReliefF-RS algorithm. However, the algorithm's calculation cost is more than that of the ReliefF-RS algorithm because it includes a step to preprocess data before clustering.
2. All instances are treated equally by the FSSMC algorithm. However, it is difficult for a classification task to discern those instances that are close to the classification boundary in terms of their features.

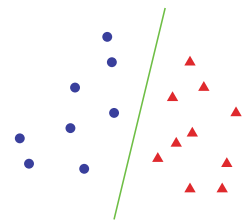
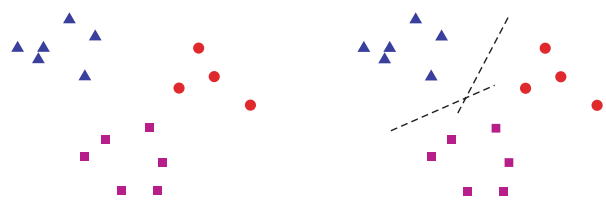


Fig. 2 Decision of binary classification



(a) Multi-class problem (b) Translation into binary classification

Fig. 3 Multi-class decision

To more effectively utilize the instances near the classification boundary, we need to clarify the following two problems:

1. When solving classification problems, it is crucial to distinguish instances that lie at the boundaries between classes correctly. Therefore, it is reasonable to assign larger weights to features that can discriminate instances that are at class boundaries. (See Fig. 2)
2. When solving a multi-class problem, it is essentially the same as solving multiple one-to-many problems and multiple ($\mathcal{L} - 1$) binary problems ($\mathcal{L}$ is the number of categories). Figure 3a shows a classification problem involving three classes, whereas Fig. 3b shows a problem that can be transformed into two dichotomies by taking the "triangle" as an example.

According to the analysis mentioned above, to improve the algorithms ReliefF-RS and FSSMC, it would be

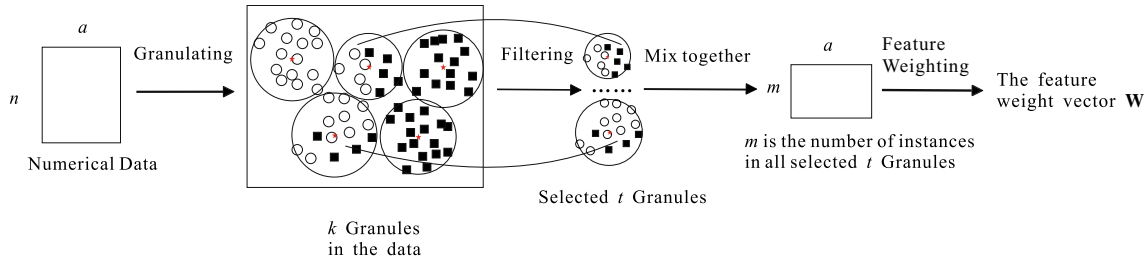


Fig. 4 Work flows of IG-FReliefF

necessary to select the instances close to the classification hyperplane for weighting features. To realize this idea, a new algorithm, IG-FReliefF, is proposed to accelerate ReliefF-RS and address the shortcomings of FSSMC. A flowchart of the proposed algorithm is shown in Fig. 4. The algorithm consists of the following three steps: (1) Granulate datasets by using K-means and obtain a certain number of granules of instances; (2) Calculate the quality of the granules obtained in Step 1, and select the granules that satisfy certain predefined conditions; and (3) construct a new dataset by collecting the selected granules, and calculate the weight of each feature on the new dataset.

Details of the method are as follows:

In Step (1), the training set $\mathcal{D}$ is divided into K granules by using K-means. Note that the value of K should be larger than the number of classes ($\mathcal{L}$). The instances in the same granule are close to each other if the value of K is too large, and the cost of preprocessing using K-means is also very high. Therefore, the value of K needs to be much smaller than the number of instances in the training set N [39]. To select a suitable K, we set the step size to perform the clustering process multiple times within the range of the K value based on experience.

In Step (2), an evaluation index is needed to evaluate the results of each clustering step to select the useful granules for feature selection. When clustering is used for information granulation, the number of granules is related to the granule's consistency and the classes. The number of granules should be as small as possible to increase the likelihood of the instances in the granule belonging to the same class. Conversely, the instances in a larger number of granules are more likely to fall into different classes.

Table 1 Comparison of time complexity

ReliefF	ReliefF-RS	FSSMC	IG-FReliefF
$O(N^2 \cdot I)$	$O(m \cdot N \cdot I)$	$O(m \cdot N \cdot I)+$ $O(N \cdot K \cdot T)$	$O(m \cdot \check{N} \cdot I)+$ $O(N \cdot K \cdot T) + O(N)$

For our task of evaluating the granulation results, a larger number of granules would be expected to enable more pure granules to be selected. Thus, we can employ knowledge granularity (Formula (2)) to obtain the desired granulation result. However, all the granules become pure if the number of granules is sufficiently large, which fails to select the instances at the boundaries of classes. To solve this problem, information entropy, which can characterize the degree of chaos in granulation results, is incorporated into the evaluation index of granulation so that the number of instances of different classes in each granule can be ensured to be approximately equal. Therefore, we propose an evaluation index of information granulation results by combining information entropy and granularity as follows.

$$Q(\pi) = G(\pi) \times \sum_{i=1}^{K} H(G_i), \tag{8}$$

where $H(G_i) = -\sum_{j=1}^{\mathcal{L}} P(C_j) \log_2 P(C_j)$.

In Step (3), all the selected granules are used to construct a new training dataset $\check{\mathcal{D}}$, which replaces the original training dataset $\mathcal{D}$, and is used to obtain the score of each feature. We list the time complexity of each algorithm in Table 1. It is well known that the time complexity of ReliefF is $O(N^2 \cdot I)$, and ReliefF-RS is $O(m \cdot N \cdot I)$, where m is the number of randomly selected instances, N is the number of instances, and I is the number of features. It is easy to see that m affects the efficiency of the ReliefF-RS algorithm. The time complexity of FSSMC is $O(m \cdot N \cdot I) + O(N \cdot K \cdot T)$, it is less efficient than ReliefF-RS, owing to the preprocessing procedure involving clustering, but it is more efficient than ReliefF. Since the time complexity of the K-means algorithm is $O(N \cdot K \cdot T)$ and the time complexity of evaluating all the granules is $O(N)$, we can get the time complexity of IG-FReliefF is $O(m \cdot \check{N} \cdot I) + O(N \cdot K \cdot T) + O(N)$, where $\check{N}$ is the instances number of the new dataset $\check{\mathcal{D}}$. Because the number of instances in $\check{\mathcal{D}}$ is much smaller than that in $\mathcal{D}$, the time efficiency of IG-FReliefF becomes higher than that of ReliefF-RS and FSSMC.

Of course, the algorithm proposed in this paper also has certain shortcomings. For example, there is a possibility that

the sampling method proposed in this paper cannot guarantee to cover all the instances in all categories, but we can try to avoid it by adjusting the parameter K value of granularity. When K value is larger, the more particles are produced, and the fewer instances in the particles, the more likely they belong to the same category. In this case, the more "chaotic" granules are chosen to try to cover all classes of instances, even those with a tiny number of instances.

Based on the above analysis, we design a new algorithm, IG-FReliefF, detailed in Algorithm 1.

4 Experiments

4.1 Datasets and parameter settings

This section empirically evaluates our proposed IG-FReliefF with two comparative algorithms on eight UCI datasets (listed in Table 2). For ReliefF, ReliefF-RS, FSSMC, and our proposed algorithm, the nested feature subsets are generated by the weights derived from these three algorithms. To fully evaluate the performance of the proposed feature selection algorithm, we also used two classifiers, C4.5 and

Algorithm 1 Information granulation based fast ReliefF algorithm IG-FReliefF

Input: The training data $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, the number of granulations K and the range [b,e] and the step size s, the screening threshold of the number of granules θ, the number of nearest neighbors k. Based on a large number of experiments, the general range in which to set the value of k was determined to be 10~20 [8]:

Output: The vectors $\mathbf{W}$

```
for K = b : s : e do
    Grain the training set D into K grains by K-means
    Calculate the value of the evaluation index Q(π) of the current clustering result with formula (8)
end for
Rank the evaluation index of the clustering results. The best granulation results are selected as the best granulation scheme.
Select a certain number of grains using the threshold θ and integrate these grains into the new training set D̆
Initialize the weight of all features to W_j = 0, j = 1, 2, ..., I
for i = 1 to m do
    Randomly select a instance R_i from D̆
    Find k nearest neighbor instances NH_t, t = 1, 2, ..., k among the instances of the same category as R_i
    for each classC ≠ class(R_i) do
        Find k nearest neighbor instances NM_t, t = 1, 2, ..., k in instances different from R_i
    end for
    for j = 1 to I do
        Update W_j with formula (7)
    end for
end for
return The vectors W
```

Table 2 Description of dataset

Dataset	Abbreviations	Number of instances	Number of features	Number of classes
User knowledge modeling	User	403	5	4
Breast cancer wisconsin	Breast	683	9	2
Banknote authentication	Banknote	1372	4	2
Wine quality-white	Wine	4898	11	5
Waveform	Waveform	5000	21	3
Page blocks	Page	5473	10	5
Electrical grid stability simulated	Electrical	10000	13	2
MAGIC gamma telescope	Magic	19020	10	2

Table 3 Classification accuracies of the eight UCI datasets with C4.5 classifiers

Dataset	Baseline	ReliefF-RS	FSSMC	IG-FReliefF
User	*0.9405*	**0.9430**	*0.9405*	*0.9405*
Breast	0.9353	0.9282	*0.9356*	**0.9370**
Banknote	0.9665	*0.9672*	0.9665	**0.9679**
Wine	0.4918	*0.5233*	**0.5253**	**0.5253**
Waveform	0.7520	*0.7546*	0.7538	**0.7552**
Page	0.9276	**0.9353**	0.9325	*0.9344*
Electrical	**0.9999**	**0.9999**	**0.9999**	**0.9999**
Magic	*0.8002*	**0.8126**	**0.8126**	**0.8126**

The highest accuracy in each row ishighlighted in bold, and the second highest appears in italics

Table 4 Classification accuracies of eight UCI datasets with KNN classifiers

Dataset	Baseline	ReliefF-RS	FSSMC	IG-FReliefF
User	0.8759	*0.9129*	0.9111	**0.9250**
Breast	0.9713	*0.9727*	0.9715	**0.9737**
Banknote	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Wine	0.4500	0.5287	**0.5385**	*0.5350*
Waveform	0.8237	0.8263	**0.8307**	*0.8293*
Page	0.9527	0.9586	*0.9589*	**0.9591**
Electrical	0.7893	*0.9124*	0.8830	**0.9463**
Magic	0.7907	0.8125	**0.8160**	*0.8134*

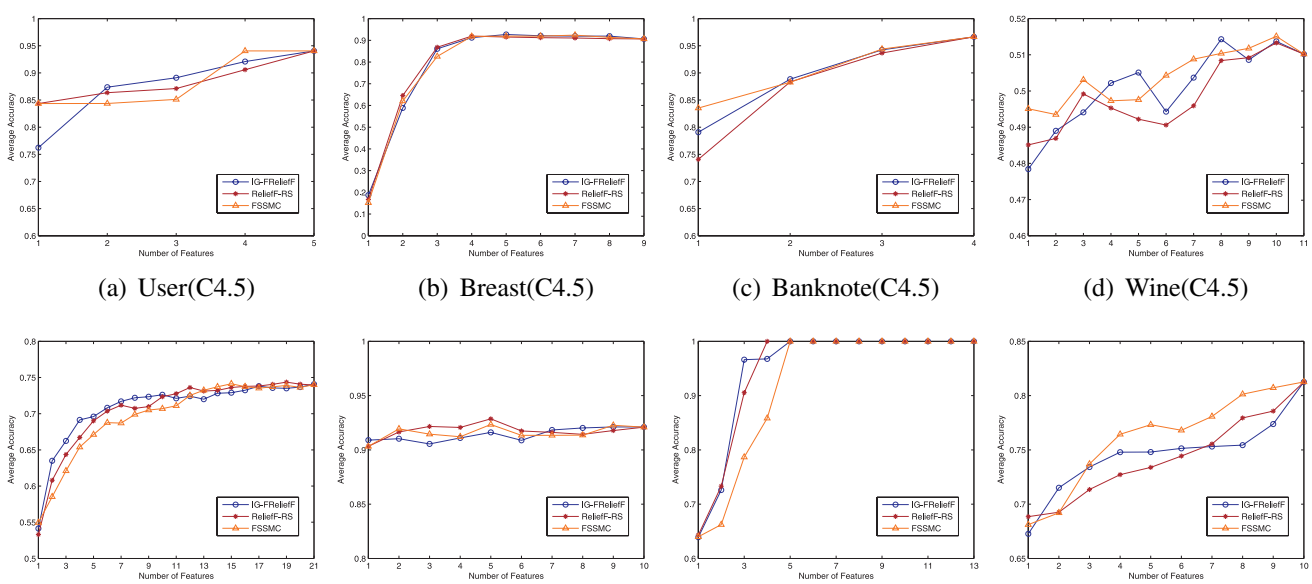
The highest accuracy in each row is highlighted in bold, and the second highest appears in italics

KNN, to evaluate the classification accuracy of the selected feature subsets. Then, the tenfold cross-validation method is employed to verify the performance of these algorithms.

All experiments were performed on personal computers with 3.6-GHz Intel Core processors and 8 GB of memory, and all the algorithms were implemented by using MATLAB R2016a. The experiment aims to illustrate that IG-FReliefF is much more efficient than these comparative algorithms and that the classification accuracy obtained by IG-FReliefF is similar to those obtained by ReliefF-RS and FSSMC.

For our proposed algorithm, the number of granules, K, is an important parameter. The larger the value of K is, the greater the number of granules. However, if the value of K is too large, each granule's size could become very small. Under these circumstances, the operation of information granulation would reduce the time consumed by the proposed algorithms by an insignificantly small amount. Therefore, the value of K should be set to be less than the number of instances (N). In the experiment, the value of K was specified to be $\{10, 15, 20, 25, 30\}$. Based on these five different K values, each dataset was granulated five times, and IG-FReliefF obtained five feature selection results. Among the classification accuracy derived from the five results, the highest was considered representative of the performance of IG-FReliefF.

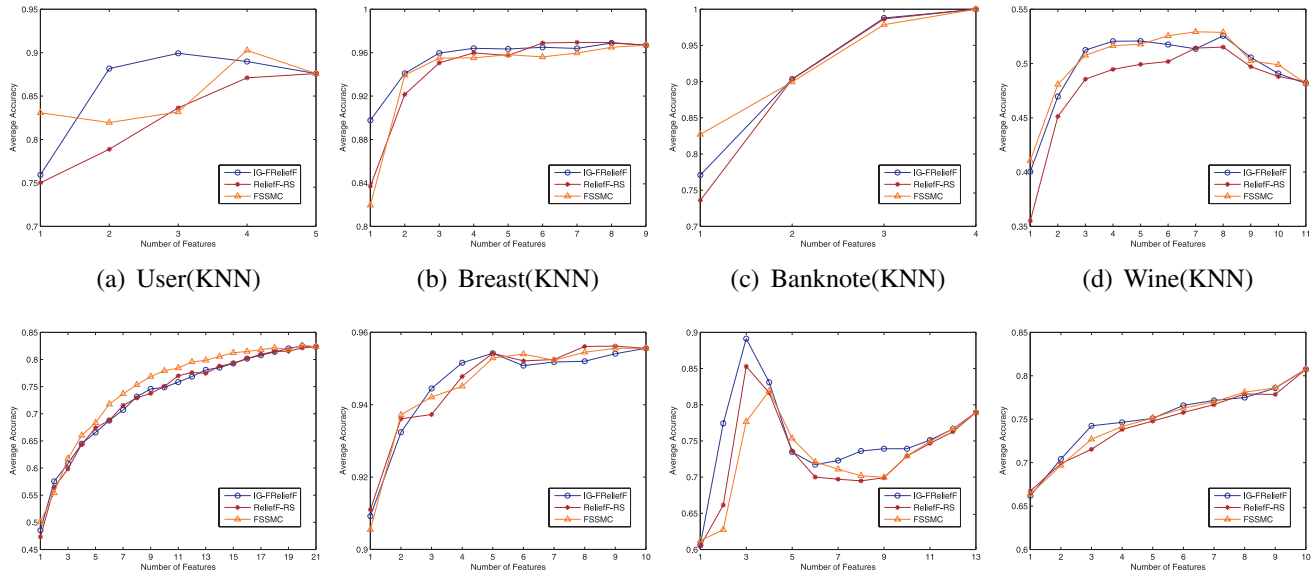
Another parameter of the proposed algorithm is the threshold (θ), which determines the number of selected granules and these granules' size. If the value of θ is too large, the number of selected granules will be very small, with the result that the selected particles lack representation. Conversely, the selected granules' size is very small



(a) User(C4.5) (b) Breast(C4.5) (c) Banknote(C4.5) (d) Wine(C4.5)

(e) Waveform(C4.5) (f) Page(C4.5) (g) Electrical(C4.5) (h) Magic(C4.5)

Fig. 5 Comparison of the average of 10 test results for each dataset on the C4.5 classifier



(a) User(KNN) (b) Breast(KNN) (c) Banknote(KNN) (d) Wine(KNN)

(e) Waveform(KNN) (f) Page(KNN) (g) Electrical(KNN) (h) Magic(KNN)

Fig. 6 Comparison of the average of 10 test results of each dataset on the KNN classifier

if the value of θ is too small, which increases the computational time of the proposed algorithm. In our experiments, the threshold θ was set to 1/4 of the number of granules. The number of sampled instances (m) is used in ReliefF-RS, FSSMC and IG-FReliefF. The m affects the efficiency of these algorithms, and the value of m is related to the size of the training set and is usually set to 5% or 10%.

4.2 Experimental results and analysis

For each dataset in Table 2, we generate 10 training sets and 10 corresponding test sets by the ten-fold verification method, and 10 nested feature subsets on each of the 10 training sets by ReliefF, ReliefF-RS, FSSMC, and IG-FReliefF, where we designate ReliefF as the baseline. The performance of IG-FReliefF was verified by using the classifiers C4.5 and KNN. The average classification accuracy on the test sets with the obtained nested feature subsets is provided in Tables 3 and 4.

In Tables 3 and 4, the highest accuracy in each row is highlighted in bold, and the second highest appears in italics. The results in Table 3 indicate that the classification accuracies corresponding to IG-FReliefF are the highest on six of the datasets, and those on the other two datasets are the second highest. The results in Table 4 show that the classification accuracies corresponding to IG-FReliefF are the highest on five of the datasets, and those on the other three datasets are the second highest. Thus, the performance of IG-FReliefF is comparable to that of ReliefF-RS and FSSMC in most cases.

Table 5 Comparison of the average of 10 runs of ReliefF-RS, FSSMC, and IG-FReliefF

Dataset	ReliefF-RS time(s)	FSSMC time(s)	IG-FReliefF				Average multiple of accelerate
			Total time(s)	Granulate and evaluate time(s)	Filter time(s)	ReliefF time(s)	
User	1.72	2.00	**1.07**	0.55	<0.01	0.52	≈1
Breast	1.38	1.84	**0.54**	0.35	<0.01	0.20	≈2
Banknote	16.23	21.84	**5.40**	1.06	<0.01	4.33	≈3
Wine	47.00	53.37	**15.50**	3.67	0.02	11.81	≈2
Waveform	51.10	60.92	**23.05**	11.12	<0.01	11.92	≈2
Page	58.23	80.67	**26.30**	2.18	<0.01	23.81	≈2
Electrical	792.87	981.27	**63.81**	24.64	0.01	39.16	≈14
Magic	2084.18	2860.40	**133.71**	18.20	0.05	115.46	≈21

Among ReliefF-RS, FSSMC and IG-FReliefF algorithms, the data with the least running time is indicated in bold

ReliefF-RS and FSSMC from the perspective of the nested feature subsets obtained by these feature selection algorithms. The classification accuracy of C4.5 and KNN is compared in Figs. 5 and 6, respectively. The horizontal axis in these two tables represents the number of features in the nested feature subsets, and the vertical axis represents the average classification accuracy corresponding to a certain size of the nested feature subsets. The results in Figs. 5 and 6 show that the classification accuracies corresponding to the proposed algorithm, IG-FReliefF, are similar to those corresponding to the compared algorithms, Relief-RS and FSSMC. Therefore, the results indicate that the proposed algorithm's performance is almost equal to that of the compacted algorithms.

The sequential processing efficiency of the proposed algorithm in terms of the computational time it requires was compared with Relief-RS and FSSMC. Table 5 compares the computational time of the three algorithms and lists the average of 10 computations. The computational time required for each stage of the IG-FReliefF algorithm is listed to show details of its efficiency. In terms of the experimental parameters, three key points should be noted. First, the time consumed by the process of multiple granulation and evaluation is presented in the column "Granulate and Evaluate". Second, the ratio of selected granules to all granules (θ) is set to 1/4, so the size of the actual training set is about 1/4 of the original training set. The total running time of the IG-FReliefF algorithm is $1 \sim 21$ times shorter than that of the other two algorithms. Third, the number of sampled instances (m) affects the running time of the ReliefF-RS algorithm; thus, the same m was adopted in all three algorithms.

Based on the experimental analysis, we can conclude that the computational efficiency of the IG-FReliefF algorithm is superior, whereas the classification performance is similar to that of the comparative algorithms, especially for large datasets.

5 Conclusion

In this paper, we propose the information granulation-based fast ReliefF algorithm to enhance the efficiency of the algorithms ReliefF-RS and FSSMC. The proposed algorithm first uses K-means to granulate the datasets to obtain a certain number of instance granules. The granules' quality is calculated to select the granules that satisfy certain pre-defined conditions, and a new dataset is constructed by collecting the selected granules to calculate the weight of features on the constructed dataset. Overall, the proposed algorithm evaluates features by sampling instances close to the classification boundary in training set, thus overcoming the drawback of weighting features with all instances in training set in ReliefF-RS and FSSMC. The experimental results demonstrated that the proposed algorithm is significantly more efficient than ReliefF-RS and FSSMC, and performs almost as well as ReliefF-RS and FSSMC with respect to classification accuracy.

Since the information granulation-based sampling method proposed in this paper is a general one, its ideas are expected to be used for more data types, such as applying the algorithm to mixed data, unbalanced data, or high-dimensional data, which can be interesting future work.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Nos. 61772323, 61976184, 61876103).

References

1. Fawley WJ, PiatetskyShapiro G, Matheus CJ (1992) Knowledge discovery in databases: an overview. Ai Mag 13(3):3–16
2. Han JW, Kamber M (2006) Data mining: concepts and techniques. Data Min Conc Mod Methods Algorithms Sec Ed 5(4):1–18
3. Zhang C, Li HX, Chen CL, Zhou XZ (2020) Nonnegative representation based discriminant projection for face recognition. Int J Mach Learn Cybern (10)
4. Li HX, Zhang LB, Huang B, Zhou XZ (2020) Cost-sensitive dual-bidirectional linear discriminant analysis. Inform Sci 510:283–303
5. Destrero A, Mosci S, Mol CD, Verri A, Odone F (2009) Feature selection for high-dimensional data. Comput Manag Sci 6(1):25–40
6. Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng 17(4):491–502
7. Kira K, Rendell LA (1992) The feature selection problem: traditional methods and a new algorithm. AAAI 2:129–134
8. Kononenko I (1994) Estimating attributes: analysis and extensions of Relief. Mach Learn ECML 94:171–182
9. Robnik M, Kononenko I (2003) Theoretical and empirical analysis of ReliefF and RReliefF. Mach Learn 53(1–2):23–69
10. Sun YJ (2007) Iterative RELIEF for feature weighting: algorithms, theories, and applications. IEEE Trans Pattern Anal Mach Intell 29(6):1035–1051
11. Sun YJ, Todorovic S, Goodison S (2008) A feature selection algorithm capable of handling extremely large data dimensionality. In: Proceedings of the SIAM International Conference on Data Mining, Atlanta, Georgia, USA 530–540
12. Cai H, Ruan P, Ng M, Akutsu T (2014) Feature weight estimation for gene selection: a local hyperlinear learning approach. BMC Bioinform 15(1):1–13
13. Huang XJ, Zhang L, Wang BJ, Zhang Z, Li FZ (2018) Feature weight estimation based on dynamic representation and neighbor sparse reconstruction. Pattern Recogn 81(9):388–403
14. Zhang L, Huang XJ, Zhou WD (2019) Logistic local hyperplane-Relief: a feature weighting method for classification. Knowl Based Syst 181:104741
15. Liu XM, Tang JS, Liu J, Feng ZL (2008) A Semi-Supervised Relief based feature extraction algorithm. In: 2nd International Conference on Future Generation Communication and Networking Symposia. Piscataway NJ: IEEE Computer Society 3:3–6
16. Cheng YB, Cai YP, Sun YJ, Jian L (2008) Semi-supervised feature selection under logistic I-RELIEF framework. In: IEEE the 19th International Conference on Pattern Recognition. Piscataway NJ: 1–4

17. Zafra A, Pechenizkiy M, Ventura S (2012) ReliefF-MI: an extension of ReliefF to multiple instance learning. Neurocomputing 75(1):210–218
18. Song Y, Si WY, Dai FF, Yang GS (2020) Weighted ReliefF with threshold constraints of feature selection for imbalanced data classification. Concurr Comput Pract Exp 32(14):1–13
19. Kilicarslan S, Adem K, Celik M (2020) Diagnosis and classification of cancer using hybrid model based on ReliefF and convolutional neural network. Med Hypoth 137:109577
20. Jin LL, Zeng QR, He JZ, Feng YJ, Zhou SQ, Wu Y (2019) A ReliefF-SVM-based method for marking dopamine-based disease characteristics: a study on SWEDD and parkinson''s disease. Behav Brain Res 356:400–407
21. Praveena HD, Subhas C, Naidu KR (2020) Automatic epileptic seizure recognition using ReliefF feature selection and long short term memory classifier. J Ambient Intell Hum Comput.
22. Wang Z, Zhang Y, Chen ZC, Yang H, Sun YX, Kang JM, Yang Y, Liang XJ (2016) Application of ReliefF algorithm to selecting feature sets for classification of high resolution remote sensing image. In: 2016 IEEE International Geoscience and Remote Sensing Symposium 755–758
23. Dou DY, Wu WZ, Yang JG, Zhang Y (2019) Classification of coal and gangue under multiple surface conditions via machine vision and Relief-SVM. Powder Technol 356:1024–1028
24. Zhou ZB, Wang YF, He XR, Zhang XC (2020) Optimization of random forests algorithm based on ReliefF-SA. IOP Conf Ser Mater Sci Eng 768:072065
25. Baskar SS, Arockiam L (2014) C-LAS Relief-An improved feature selection technique in data mining. Int J Comput Appl 83(13):33–36
26. Liu Y, Tang F, Zeng Z (2015) Feature selection based on dependency margin. IEEE Trans Cybern 45(6):1209–1221
27. Shi SB, Li GN, Chen HX, Liu JY, Hu YP, Xing L, Hu WJ (2017) Refrigerant charge fault diagnosis in the VRF system using bayesian artificial neural network combined with ReliefF filter. Appl Thermal Eng 112:698–706
28. Huang Y, Mccullagh PJ, Black ND (2009) An optimization of ReliefF for classification in large datasets. Data Knowl Eng 68(11):1348–1356
29. Yao YY (2009) Interpreting concept learning in cognitive informatics and granular computing. IEEE Trans Syst Man Cybern Part B 39(4):855–866
30. Niu JJ, Huang CC, Li JH, Fan M (2018) Parallel computing techniques for concept-cognitive learning based on granular computing. Int J Mach Learn Cybern 9(11):1785–1805
31. Mi YL, Shi Y, Li JH, Liu WQ, Yan MY (2020) Fuzzy-based concept learning method: exploiting data with fuzzy conceptual clustering. IEEE Tran Cybern 42(1):1–12
32. Yao YY (2020) Tri-level thinking: models of three-way decision. Int J Mach Learn Cybern 11:947–959
33. Liu D, Yang X, Li TR (2020) Three-way decisions: beyond rough sets and granular computing. Int J Mach Learn Cybern 11:989–1002
34. Wierman MJ (1999) Measuring uncertainty in rough set theory. Int J Gen Syst 28(4–5):283–297
35. Liang JY, Qian YH (2008) Information granules and entropy theory in information systems. Sci China (Ser F Inform Sci) 10:29–46
36. Qian YH, Liang JY (2008) Combination entropy and combination granulation in rough set theory. Int J Uncert Fuz Knowl Based Syst 16(2):179–193
37. Qian YH, Liang JY, Wu WZ et al (2011) Information granularity in fuzzy binary GrC model. IEEE Trans Fuzzy Syst 19(2):253–264
38. Beaubouef T, Petry FE, Arora G (1998) Information-theoretic measures of uncertainty for rough sets and rough relational databases. Inform Sci 109(1):185–195
39. Bai L, Chen XQ, Liang JY, Shen HW, Guo YK (2017) Fast density clustering strategies based on the k-means algorithm. Pattern Recogn 71:375–386