



# A novel filter feature selection algorithm based on relief

Xueting Cui<sup>1,2</sup> · Ying Li<sup>1,2</sup> · Jiahao Fan<sup>1,2</sup>  · Tan Wang<sup>3</sup>

Accepted: 1 July 2021 / Published online: 3 August 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

The Relief algorithm is a feature selection algorithm that uses the nearest neighbor to weight attributes. However, Relief only considers the correlation between features, which leads to a low classification accuracy on noisy datasets whose interaction effect is weak. To overcome the weaknesses of Relief, a novel feature selection algorithm, named Multidirectional Relief (MRelief), is proposed. The MRelief algorithm includes four improvements. First, the multidirectional neighbor search method, which finds all neighbors within a distance threshold from different orientations, is included to obtain regularly distributed neighbors. Therefore, the weights provided by MRelief are more accurate than those provided by Relief. Second, a novel objective function that incorporates the instances' force coefficients is introduced to reduce the influence of noise. Thus, the new objective function improves the classification accuracy of MRelief. Third, subset generation is introduced to the MRelief algorithm and combined with the maximum Pearson maximum distance (MPMD) to generate a promising candidate subset for feature selection. Finally, a novel multiclass margin definition is proposed and introduced to the MRelief algorithm to handle multiclass data. As demonstrated by extensive experiments on eleven UCI datasets and eleven real-world gene expression benchmarking datasets, MRelief is significantly better than other algorithms including LPLIR, ReliefF, LLH-Relief, MultiSURF, MSLIR-NN, MRMR, MPMD and STIR in our study.

**Keywords** Relief · ReliefF · Neighbor search · Feature selection · Classification

## 1 Introduction

An overwhelming amount of data is currently available [1, 2]. As the dimensionality of data increases, so does the number of features. Therefore, to select an optimal feature subset, feature selection is used [3, 4]. Feature selection [5, 6] is very effective in reducing dimensionality; and it is applied in numerous fields, such as pattern recognition [7], machine learning [8, 9], data mining [10] and some other fields [11]. Feature selection includes four processes, including subset generation, subset evaluation, stopping criterion and result validation.

Typical feature selection methods are divided into three classes: typical wrapper methods typical filter methods and embedded methods [12–14]. A typical wrapper method uses the performance of the classifier as an evaluation criterion while a typical filter method depends on the characteristics of the dataset to select a subset of features without evaluating any classifier [15]. Embedded methods have the advantage that they not only include the interaction with the classifier, but also take less computational time than wrapper methods [16]. Typical filter methods include the Maximum Relevance Minimum Redundancy (MRMR) [17] and ReliefF [18]. Typical wrapper methods include Particle Swarm Optimization (PSO) [19], the Genetic Algorithm (GA) [20], Bacterial Foraging Optimization (BFO) [21], Ant Colony Optimization (ACO) [22], Cuckoo Search (CS) [23], the Artificial Bee Colony (ABC) [24], the Whale Optimization Algorithm [25, 26] and the Dragonfly Algorithm (DA) [27]. Because the performance of a typical wrapper method heavily relies on the specified mining algorithm, typical filter methods are generally more popular than the wrapper methods.

Relief is an extensively studied filter method because it is simple and effective in high-dimensional feature space and nonparametric [28]. The improved Relief algorithms mainly

---

✉ Jiahao Fan  
jihfanfan@hotmail.com

<sup>1</sup> College of Computer Science and Technology, Jilin University, Changchun, China

<sup>2</sup> Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun, China

<sup>3</sup> Space Technology (Jilin) Limited Company, Jilin, China

address two aspects. First, the weight estimation function is improved. Second, different neighbor search methods are proposed.

In order to alleviate the deficiencies of Relief, improved algorithms with weight estimation functions, such as Relief-MM [29] and I-Relief [30], have been proposed. To improve the performance of the Relief algorithm, logistic local hyperplane-based Relief (LLH-Relief) has been proposed as a feature selection algorithm. LLH-Relief combines the advantages of logistic iterative-Relief (LI-Relief) and local hyperplane-Relief (LH-Relief). A multiclass semisupervised feature selection named the local preserving logistic I-Relief (LPLIR) algorithm that pays attention to the local characteristics of data has been proposed [31]. However, instance force coefficients are not considered in the above algorithms.

To avoid arbitrary selection of the nearest neighbors, recent research improves the number of selected nearest neighbors, such as using a fixed number of nearest neighbors in ReliefF, the constant neighborhood radius in SURF [32], the adaptive radius in MultiSURF [33] and a specific  $k$  for each feature in ReliefSeq [34]. However, none of the above algorithms solve the problem that neighborhoods of a center instance are irregularly distributed.

In summary, the above studies ignore four shortcomings. First, the neighborhoods of a center instance are irregularly distributed, thus, the neighborhoods that are far away from the center instance from different orientations are not involved in calculating the weights of Relief. Therefore, the weights of Relief are inaccurate for feature selection. Second, because the objective function assigns instances the same force coefficient based on their importance, noise and outliers affect the classification accuracy rate of Relief. Third, Relief ignores the correlation between features and classes. Furthermore, Relief does not consider the redundancy among features. Thus, Relief easily achieves the local optimal solution for classification. Finally, Relief was originally designed to solve two-class problems, and it is not applicable to multiclass data.

To address the above problems, a novel feature selection algorithm called MRelief is proposed. First, to solve the problem of irregularly distributed neighbors, MRelief finds all neighbors within a distance threshold in different orientations. The circle around the center instance within the distance threshold is split into specific uniform regions, and MRelief selects neighbors from all regions. Second, to improve the classification rate of MRelief, instance force coefficients are introduced to the margin-based objective function. This function integrates the Pearson correlation coefficient to represent the instance force coefficients. Third, a new subset generation process is proposed and combined with MPMD to obtain a promising candidate subset. Finally, to handle multiclass data, a novel multiclass margin definition is proposed.

Overall, we present four corresponding improvements to MRelief, as shown below.

1. A multidirectional neighbor search method is proposed to select regularly distributed neighbors in different orientations.
2. A novel objective function that incorporates instance force coefficients is formed to improve the classification accuracy rate.
3. A subset generation method is proposed to obtain the optimal candidate subset.
4. A multiclass margin definition is proposed to handle multiclass data.

The remainder of our research is organized as follows. In section 2, we introduce the theoretical background of our paper. In section 3, we propose a novel feature selection algorithm called MRelief. In section 4, we evaluate the performance of MRelief compared with ReliefF, LLH-Relief, LPLIR, MultiSURF and STIR. In section 5, the discussion and conclusion are given in detail.

## 2 Theoretical backgrounds

We introduce two types of filter algorithms. The filter algorithms are Relief-based algorithms and MPMD.

### 2.1 Relief-based algorithms

A number of Relief-based methods have been proposed. Relief-based algorithms output individual feature weights. In this section, some typical Relief-based methods, i.e., Relief, ReliefF, SURF, MultiSURF, and STIR, are briefly introduced.

#### 2.1.1 Relief and ReliefF

Relief is a feature weighting algorithm [35]. Relief evaluates the weights of features consistent with the difference between instances that are similar. Relief randomly selects an instance  $X$ . According to a random sampling of instances, Relief finds its nearest hit  $H$  from the same class and its nearest miss  $M$  from the other different class. Then, Relief calculates the difference between two similar instances in order to influence the weights of attributes and updates the quality estimation  $W[A_i]$  depending on their values for  $X$ ,  $M$  and  $H$  for all features  $A$ . In the same class, the feature has a negative influence if the difference between two similar instances is caused by the feature. In a different class, a feature has a positive influence if the difference between two similar instances is caused by the feature. The weight of feature  $A_i$  is calculated using the

formula given in (1).

$$W[A_i] = W[A_i] - \text{diff}(A_i, X_m, H) / T + \text{diff}(A_i, X_m, M) / T \quad (1)$$

Function  $\text{diff}(A_i, X_1, X_2)$  calculates the difference between two instances  $X_1$  and  $X_2$  for feature  $A_i$ .  $W[A_i]$  is the weight of feature  $A_i$ .  $T$  is number of the maximum iterative times.

The numerical attributes are calculated as follows.

$$\text{diff}(A_i, X_1, X_2) = \left| \text{value}(A_i, X_1) - \text{value}(A_i, X_2) \right| / (\max(A_i) - \min(A_i)) \quad (2)$$

where  $\text{value}(A_i, X_1)$  is the value of feature  $A_i$  for instance  $X_1$ .

The nominal attributes are calculated as follows.

$$\text{If } \text{value}(A_i, X_1) = \text{value}(A_i, X_2), \text{diff}(A_i, X_1, X_2) = 0; \quad (3)$$

$$\text{otherwise, diff}(A_i, X_1, X_2) = 1$$

The original Relief can handle both nominal and numerical attributes. However, it is limited to solving datasets that are two-class. In addition, the original Relief cannot handle incomplete data.

The ReliefF algorithm is able to handle problems with multiclass datasets that contain noisy data. Based on Relief, ReliefF finds its  $K$  nearest hits  $H_j$  from the same class and its  $K$  nearest misses  $M_j(\text{Class})$  from a different class according to a random sampling of instances. It updates the weight  $W[A_i]$  relying on  $M_j(\text{Class})$  and  $H_j$  for all features  $A$ .

$$W[A_i] = W[A_i] - \sum_{j=1}^K \text{diff}(A_i, X_m, H_j) / (T \times K)$$

$$+ \sum_{C \neq \text{Class}(X_m)} \left[ \frac{P(C)}{1 - P(C)} \sum_{j=1}^K \text{diff}(A_i, X_m, M_j(C)) \right] / (T \times K) \quad (4)$$

where  $P(C)$  is the prior probability of class  $C$ .  $T$  is the maximum number of iterations.  $\text{Class}(X_m)$  is the class of instance  $X_m$ .

### 2.1.2 SURF and MultiSURF

The SURF algorithm [32] is similar to the ReliefF algorithm. Different from ReliefF, SURF implements a new neighbor search method. Instead of using a fixed parameter  $k$ , SURF adopts a distance threshold  $T$  to determine which instances are selected. The radius  $T$  of a given target instance is calculated as the average of two instances.

MultiSURF defines a threshold  $T_i - \sigma_i/2$  to determine which instances are considered neighbors [32].  $T_i$  is the mean pairwise distance between the target instance and all others.  $\sigma_i$  is the standard deviation of the pairwise distances between the target instance and all others.

### 2.1.3 STIR

Combined with the pooled standard deviations, this novel Relief-based algorithm transforms the Relief-based score  $W_{STIR}$  into a pseudo t-statistic [36]. For feature  $A$ , the STIR weight is calculated as the difference of the means and the standard error.

$$W_{STIR}[A, M, H] = \frac{\overline{M}_A - \overline{H}_A}{S_p[M, H] \sqrt{1/|M| + 1/|H|}} \quad (5)$$

where  $|M|$  and  $|H|$  are the total numbers of nearest misses and hits, respectively. The pooled standard deviation is calculated by the following formula.  $\overline{M}_A$  is the mean difference for feature  $A$  averaged over of all pairs of nearest misses.  $\overline{H}_A$  is the mean difference for feature  $A$  averaged over of all pairs of nearest hits.

$$S_p[M, H] = \sqrt{\frac{(|M|-1)S_{M_A}^2 + (|H|-1)S_{H_A}^2}{|M| + |H| - 2}} \quad (6)$$

where  $S_{M_A}^2$  and  $S_{H_A}^2$  are the pooled standard deviations.

### 2.2 MPMD

Zheng et al. proposed a new filter algorithm called the MPMD [37]. The relevance between classes and features is calculated using the Pearson correlation coefficient, and the redundancy among features is calculated using the correlation distance.

The value of the maximum Pearson correlation coefficient between the feature set  $A = \{A_1, A_2, \dots, A_i, A_D\}$  and the class set  $\text{Class} = \{\text{Class}(X_1), \text{Class}(X_2), \dots, \text{Class}(X_j), \text{Class}(X_N)\}$  is calculated by the maximum of Pearson correlation coefficient between the feature and the class using (7).

$$MP(A, \text{Class}) = \max(PE(A_i, \text{Class})) \quad (7)$$

where  $PE(A_i, \text{Class})$  is calculated using (8).

$$PE(A_i, \text{Class}) = \frac{\text{cov}(A_i, \text{Class})}{\sigma(A_i) \times \sigma(\text{Class})} \quad (8)$$

where  $\text{cov}(A_i, \text{Class})$  is the covariance between feature  $A_i$

and class  $Class$  in (9),  $\sigma(A_i)$  is defined by (10) and  $\sigma(Class)$  is defined by (11).

$$\text{cov}(A_i, Class) = \frac{\sum_{j=1}^N (A_i^j - \bar{A}_i) (Class(X_j) - \overline{Class})}{N} \quad (9)$$

where  $A_i^j$  is value of  $j$ th instance for feature  $A_i$ ,  $\bar{A}_i$  is the average of feature  $A_i$ ,  $\overline{Class}$  is the average of class  $Class$ .

$$\sigma(A_i) = \sqrt{\frac{\sum_{j=1}^N (A_i^j - \bar{A}_i)^2}{N}} \quad (10)$$

$$\sigma(Class) = \sqrt{\frac{\sum_{j=1}^N (Class(X_j) - \overline{Class})^2}{N}} \quad (11)$$

where  $N$  is the number of instances.

The maximum correlation distance between remaining feature  $R_m$  and selected feature  $S_k$  is calculated by the maximum correlation coefficient in (12).

$$MD(S, R) = \max(D(S_k, R_m)) \quad (12)$$

where the remaining feature set is defined by  $R = \{R_1, R_2, \dots, R_m, R_M\}$ , the selected feature set is defined by  $S = \{S_1, S_2, \dots, S_k, S_W\}$  and  $D = M + W$ .  $D(S_k, R_j)$  is calculated using (13).

$$D(S_k, R_m) = \frac{\sum_{m=1}^M (1 - |PE(S_k, R_m)|)}{M} \quad (13)$$

MPMD is generated by the maximum Pearson correlation coefficient and maximum correlation distance. Two factors,  $r_1$  and  $r_2$ , are introduced to balance  $MP$  and  $MD$ .

$$MPMD = r_1 \times MP(A, Class) + r_2 \times MD(S, R) \quad (14)$$

where  $r_1$  and  $r_2$  are calculated using (15) and (16), respectively.

$$r_1 = \cos\left(\frac{t \times a}{T}\right) \times 10 \quad (15)$$

$$r_2 = \sin\left(\frac{t \times a}{T}\right) \times 10 \quad (16)$$

where  $t$  represents the current number of iterations,  $T$  represents the total number of iterations and  $a$  is a constant.

### 3 The proposed algorithm

In this part, we introduce the multidirectional neighbor search method, the new relief-feature weighting objective function, the subset generation and the multiclass extension.

#### 3.1 Multidirectional neighbor search method

Because neighbors far away from the center instance are useless, MRelief finds all neighbors within a distance threshold  $r$  instead of selecting a fixed number of neighbors. However, the neighborhoods of a center instance are irregularly distributed, and we endeavor to select neighbors in different orientations. The circle around the center instance within the distance threshold  $r$  is split into 5 uniform regions. In Fig. 1, each region represents a direction, and region has an azimuth angle  $\theta$  that is set to  $72^\circ$ . We select  $m$  nearest neighbor instances from a region to represent the region. For example, when  $m = 1, 2, 3, \dots, K = 5, 10, 15, \dots$ , where  $K$  is the number of selected neighbors. To avoid insufficient instances inside a region, we use iterative farthest point sampling (FPS) to choose the neighbor set  $\{X_{i_1}, X_{i_2}, \dots, X_{i_m}\}$  of the center instance. Given input instances  $\{X_1, X_2, \dots, X_N\}$ ,  $N$  is the total number of instances. Thus,  $X_{i_j}$  is the most distant point from the set  $\{X_{i_1}, X_{i_2}, \dots, X_{i_{j-1}}\}$  with regard to the remaining instances.

In Fig. 1, the circle is drawn with the center instance as the circle center and the search radius as the circle radius. The circle is divided into 5 uniform directions. The azimuth  $\theta$ , radius  $r$  and  $m$  selected instances are marked in one region. The existing methods use some neighbor search methods. ReliefF selects a fixed number of nearest neighbors. SURF uses a constant neighborhood radius to select neighbors. MultiSURF uses the adaptive radius for neighbor searches. ReliefSeq assigns a specific  $k$  to each

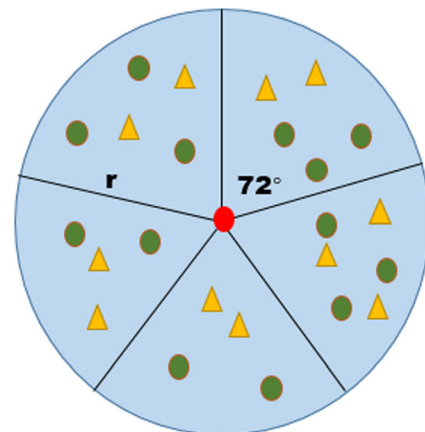


Fig. 1 Multidirectional search method

feature. However, none of the existing methods solve the problem that the neighborhoods of a center instance are irregularly distributed.

In Fig. 2, Relief finds the nearest neighbor in the same class (green circle) and the nearest neighbor in the other class (orange rectangle). ReliefF finds a specified number of neighbors (five in this example) that are used to calculate the weights. MRelief finds all neighbors that are divided into 5 uniform regions within a distance threshold, which is expressed as a circle. The center instance that selects neighbors to be used for weighting is expressed as a filled red circle. The neighbors from the same class are highlighted in green and the neighbors from the other class are highlighted in orange for each algorithm. Parts A, B, and C represent Relief, ReliefF and MRelief, respectively.

### 3.2 New relief-feature weighting objective function

In Relief, each instance in a given dataset is equally important for constructing optimization objective functions. However, a reasonable objective function should assign instances different force coefficients based on their importance. Thus, two instance force coefficients are introduced to generate a new objective function to reduce the influence of noise.

As shown in (17), when the degree of coincidence is large between the instance and the farthest miss, the instance needs to be given a small instance force coefficient  $u_m$  for  $X_m$

$$u_m = \frac{PE(X_m, M_1(Class(X_m)))^{r_1}}{\sum_{m=1}^N PE(X_m, M_1(Class(X_m)))^{r_1}} \tag{17}$$

*s.t.*  $\|u_m\| = 1, u_m \geq 0$

where  $M_1(Class(X_m))$  is the farthest miss of  $X_m$  from other

different classes, coefficient  $r_1$  is used to adjust  $u_m$ .  $PE$  is the Pearson correlation coefficient calculated by (8). The constraints  $\|u_m\| = 1$  prevent the maximization from increasing without bound, and  $u_m \geq 0$  ensures that the instance force coefficient induces a distance measure.

In (18), when the degree of coincidence between the instance and farthest hit is large, the instance needs to be given a large instance force coefficient.

$$h_m = \frac{PE(X_m, H_1(Class(X_m)))^{\frac{1}{r_2}}}{\sum_{m=1}^N PE(X_m, H_1(Class(X_m)))^{\frac{1}{r_2}}} \tag{18}$$

*s.t.*  $\|h_m\| = 1, h_m \geq 0$

where  $H_1(Class(X_m))$  is the farthest hit of instance  $X_m$ , coefficient  $r_2$  is used to adjust  $h_m$ .  $PE$  is the Pearson correlation coefficient calculated by (8). The constraint  $\|h_m\| = 1$  prevents the maximization from increasing without bound, and  $h_m \geq 0$  ensures that the instance force coefficient induces a distance measure.

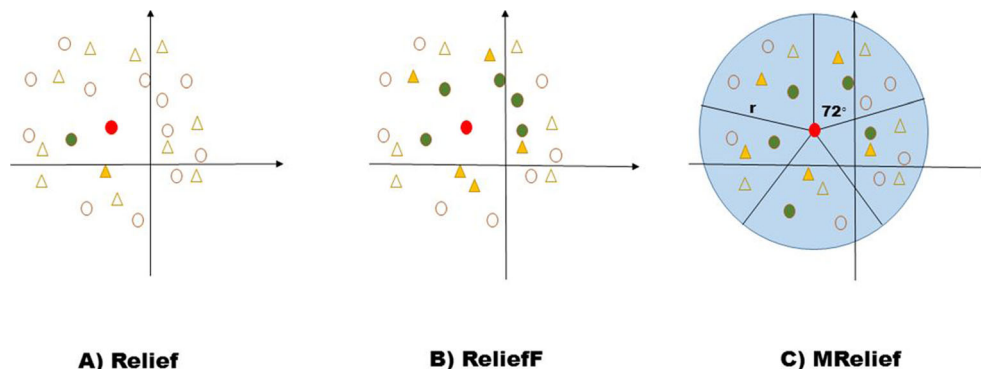
When the instance force coefficient is introduced, the objective function for instance  $X_m$  is expressed as follows:

$$\rho_m = u_m \sum_{j=1}^K \sum_{i=1}^D diff(A_i, X_m, M_j(Class(X_m))) - h_m \sum_{j=1}^K \sum_{i=1}^D diff(A_i, X_m, H_j) \tag{19}$$

where function  $diff(A_i, X_1, X_2)$  calculates the difference for feature  $A_i$  between two instances  $X_1$  and  $X_2$ .  $K$  is the number of nearest hits and misses.  $M_j(Class)$  is the selected nearest misses from other different classes.  $H_j$  is the selected nearest hits from the same class.  $N$  is the number of instances.  $Class(X_m)$  is the class of instance  $X_m$ .

$J(w)$  is the objective function for a given training dataset computed with respect to  $w$ . (20) is constructed to optimize  $u_m$  and  $h_m$ .

Fig. 2 The selected neighbors for three algorithms



$$J(w) = \sum_{m=1}^{N_1} w \cdot \rho_m = \sum_{m=1}^{N_1} \left( u_m \sum_{j=1}^K \sum_{i=1}^D w_i \text{diff}(A_i, X_m, M_j(\text{Class}(X_m))) - h_m \cdot \sum_{j=1}^K \sum_{i=1}^D w_i \text{diff}(A_i, X_m, H_j) \right) \tag{20}$$

*s.t.*  $\|w\|^2 = 1, w \geq 0$

where  $w$  is the weight vector.  $w$  is defined by  $w = \{w_1, w_2, \dots, w_i, w_D\}$ .  $\rho_m$  is calculated by (19).  $N_1$  is the number of instances in a given training dataset.

### 3.3 Subset generation method

Better classification can be achieved if some features with small but important weights are added in the process of generating feature subsets. Relief uses a fixed threshold to select attributes directly according to the feature weight. However, some features whose weight is small but important cannot be selected. Therefore, MPMD is used to select features in MRelief. MPMD balances the maximum Pearson correlation coefficient and the maximum correlation distance in order to enrich candidate subsets. In (21), the probability of a feature  $A_i$  being selected is expressed by the result of combining the weights of MRelief and the weights of MPMD.

$$PP[A_i] = \frac{(1 + \beta^2)MPMD[A_i]*W[A_i]}{\beta^2MPMD[A_i] + W[A_i]} \tag{21}$$

where  $\beta$  is a coefficient that is used to balance the degree of importance between MPMD and MRelief. When  $\beta$  is set to 1, it means that MPMD and MRelief are equally important. For example, when  $\beta$  is set to 0.5, it means that the force coefficient of MPMD is 0.25 while the force coefficient of MRelief is 1. Thus, MPMD is less important than MRelief. For example, when  $\beta$  is set to 1.5, it means that the force coefficient of MPMD is 1.25 while the force coefficient of MRelief is 1. Therefore, MPMD is more important than MRelief.

### 3.4 Multiclass extension

In order to handle multiclass data, the margin of an instance is calculated by MReliefF as follows:

$$\rho_m = u_m \cdot \sum_{\{C \in Y, C \neq y(X_m)\}} \frac{P(C)}{1-P(y(X_m))} \sum_{i=1}^K \sum_{j=1}^D \text{diff}(A_i, X_m, M_j(\text{Class}(X_m))) - h_m \cdot \sum_{i=1}^K \sum_{j=1}^D \text{diff}(A_i, X_m, H_j) \tag{22}$$

where  $Y \in \text{Class}$  is the set of class labels,  $y(X_m)$  is the label of instance  $X_m$ ,  $M_j(C)$  is the nearest miss of  $X_m$  from class  $C$ ,  $H_j$  is the selected nearest hits from the same class,  $h_m$  and  $u_m$  are the instance force coefficients,  $P(C)$  is the a priori probability of class  $C$  and  $\text{Class}(X_m)$  is the class of instance  $X_m$ .

As summarized by the pseudocode in Algorithm 1, MRelief randomly selects  $M$  instances in each cycle. For

each selected instance, MRelief finds its  $K$  nearest hits  $H_j$  and its  $K$  nearest misses  $M_j(\text{Class})$  according to the multidirectional neighbor search method. For each feature, MRelief uses a new Relief-Feature weighting objective function to update the weights. Moreover, the value of MPMD is calculated by (14). Finally, the probability of a feature is determined using (21) according to subset generation.

**Algorithm 1. Pseudocode for Mrelief**

**Input:** the feature set  $A = \{A_1, A_2, \dots, A_i, A_D\}$ , the maximum number of iterations  $T$ , the number of selected instances  $M$ , the number of features  $D$ , the number of instances  $N$ , the class of instance  $X_m$   $Class(X_m)$ , initialize all feature weights  $W[A] := 0$ ,  $MPMD[A] := 0$ ,  $r_1$  and  $r_2$ .

**Output:** Feature weight  $P$ .

**1 For**  $t := 1$  to  $T$  **do**

**2 For**  $m := 1$  to  $M$  **do**

**3** Random select an instance  $X_m$ ;

**4** Use the **multidirectional neighbor search method** to find its  $K$  nearest hits  $H_j$  from the same class and its  $K$  nearest misses  $M_j(Class)$  from another different class;

**5 For**  $i := 1$  to  $D$  **do**

**6** Use the **new Relief-Feature weighting objective function** to update  $W[A_i]$  in (22);

**7 End For**

**8 End For**

**9 End For**

**10** Obtain  $MPMD[A]$  using (14);

**11 Return** the probability of a feature using (21) according to **subset generation**;

## 4 Experiment results

### 4.1 Benchmark data

We selected eleven UCI datasets and eleven microarray datasets in the following experiments. UCI datasets are downloaded from <http://archive.ics.uci.edu/ml>. The microarray datasets are downloaded from <http://datam.i2r.a-star.edu.sg/datasets/krbd/>. The characteristics of the datasets, which contain numeric data, nominal data and mixed data, are introduced in Tables 1 and 2. As we can see from Tables 1 and 2, the number of instances varies from 72 to 10,992, the

number of features varies from 13 to 16,063 and the number of classes varies from 2 to 26.

### 4.2 Parameter settings

The experiments include state-of-the-art Relief-based methods and mutual-information-based feature selection methods. The detailed parameter values of each algorithm are described in Table 3. For ReliefF, the number of iterations is set to 10, and the number of selected instances is set to 5. For LLH-Relief, the number of iterations is set to 10, the number of selected instances is set to 5,  $\theta = 0.01$ ,  $\lambda_r = 0.001$ ,  $\lambda = 1$  and  $\eta = 0.1$ . For LPLIR, the number of iterations

**Table. 1** UCI datasets

Datasets	Instances	Features	Class	Abbreviation
Parkinson’s	195	22	2	Pa
Sonar	208	60	2	So
HEART_SPECTF	187	44	2	HE
Vehicle Silhouettes	846	18	4	Ve
Wine	178	13	3	Wi
Vowel	990	13	11	Vo
Secom	1567	590	2	Se
Arrhythmia	452	279	13	Ar
Waveform	5000	21	3	Wa
Pendigits	10,992	16	10	Pe
Musk	6598	168	2	Mu

is set to 10, the number of selected instances is set to 5,  $\lambda_1 = 1$  and  $\lambda_2 = 0.01$ . For MultiSURF, the number of iterations is set to 10, and the number of selected instances is set to 5. For MSLIR-NN, the number of iterations is set to 10, and the number of selected instances is set to 5. For MRMR, the number of iterations is set to 10. For MPMD, the number of iterations is set to 10. For STIR, the number of iterations is set to 10, and the number of selected instances varies for each instance. For MRelief, the number of iterations is set to 10, the number of selected instances is set to 5,  $r_1=1.1$  and  $r_2=0.9$ .

### 4.3 Experiments and analysis

The experiments are conducted by implementing ReliefF, LLH-Relief, LPLIR, MultiSURF, STIR, MSLIR-NN, MRMR, MPMD and MRelief using MATLAB R2014a. Each algorithm is run 10 times, and the average classification accuracy rate is calculated as the final result. The SVM

**Table. 2** Microarray datasets

Datasets	Instances	Features	Class	Abbreviation
LUNG_Cancer	203	12,600	5	LU
Leukemia2	72	7129	2	Le2
Leukemia3	72	7129	3	Le3
Lymphoma	45	4026	9	Ly
DLBCL	77	7129	2	DL
Colon	62	2000	2	Co
GCM	280	16,063	2	GCM
Pros1	102	12,600	2	Pr1
Pros2	88	12,625	2	Pr2
11-Tumors	174	12,533	11	11-T
14-Tumors	308	15,009	9	14-T

**Table. 3** Parameter settings

Algorithm	Parameters	Reference	Year
ReliefF	Number of iterations =10 Selected instances =5	[18]	2003
LLH-Relief	Number of iterations =10 Selected instances =5 $\theta=0.01$ $\lambda_r=0.001$ $\lambda=1$ $\eta=0.1$	[41]	2019
LPLIR	Number of iterations =10 Selected instances =5 $\lambda_1=1$ $\lambda_2=0.01$	[31]	2020
STIR	Number of iterations =10 $K$ varies for each instance	[36]	2018
MultiSURF	Number of iterations =10 Selected instances=5	[33]	2018
MSLIR-NN	Number of iterations =10 Selected instances ( $\theta=0.01$ )=5	[40]	2019
MRMR	Number of iterations =10	[17]	2005
MPMD	Number of iterations =10 $a=1.57$	[37]	2019
MRelief	Number of iterations =10 Selected instances =5 $r_1=1.1, r_2=0.9$	Our paper	

classifier is used to evaluate the classification performance for each filter algorithm [38, 39]. The kernel function of the SVM classifier is the radial basis function (RBF). Penalty parameter C and RBF parameter gamma are obtained through the grid search method. Ten-fold cross validation is applied to test the algorithms mentioned above. Ten-fold cross validation consists of 10 cycles. The datasets are divided into 10 folds for every cycle. Then, 9 folds are used for training, and the remaining fold is used for testing. In our experiments, two criteria are considered including the classification accuracy rate  $Acc$  and the  $F_1$  value.  $Acc$  and the  $F_1$  value are calculated in every cycle. Therefore, we obtain the average classification accuracy rate of 10 cycles. The  $Acc$  criterion is computed by the following formula.

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{23}$$

where  $TP$  ( $TN$ ) represents the number of positive (negative) instances that are classified correctly and  $FP$  ( $FN$ ) is the number of positive (negative) instances that are classified incorrectly.

The  $F_1$  value is calculated in every cycle. Therefore, we obtain the average  $F_1$  value of 10 cycles. The  $F_1$  criterion is



computed by the following formula.

$$F_1 = \frac{2 * TP_{rate} * PP_{rate}}{PP_{rate} + TP_{rate}} \tag{24}$$

where  $TP_{rate}$  and  $PP_{rate}$  are calculated by the following formulas.

$$TP_{rate} = \frac{TP}{TP + FN} \tag{25}$$

$$PP_{rate} = \frac{TP}{TP + FP} \tag{26}$$

Table 4 shows the average classification accuracies of ReliefF, ReliefF with subset generation in (21) and ReliefF with new relief-feature weighting objective function in (22) applied to ten datasets. The results show the effectiveness of the formulas in (21) and (22) is able to work better than the traditional method. The subset generation in (21) generates a promising candidate subset for feature selection combined with MPMD. The novel objective function that incorporates the instances' force coefficients in (22) improves the classification accuracy of MRelief.

Figure 3 indicates that MRelief performs the best on the general trend of the average classification accuracy. The result of Musk clearly shows that MRelief achieves the highest classification accuracy over the entire process. LLH-Relief obtains the local optimal classification accuracy other than that of MRelief. In addition, LLH-Relief achieves the highest classification accuracy in Wine and Vowel. As the Waveform results show, the classification accuracy of LPLIR is higher than those of the other algorithms except MRelief. The results of Sonar show that the classification accuracies of MRelief and STIR both achieve excellent performance. In Vowel and Pendigits, there are overlaps among curves. Furthermore, there are crosses in Parkinson's and Heart. Although the nine algorithms sometimes obtain the same accuracy rate, we know that MRelief has excellent performance among the nine

algorithms. The results of Fig. 3 show that the novel preprocessing method significantly promotes the classification accuracy of MRelief, and the improved objective function makes a huge contribution to jumping out of the local optimum for MRelief.

Table 5 and Table 6 show the mean accuracies and standard deviations (%) of the algorithms on the UCI and microarray datasets. The tables compare the average accuracies and standard deviations on the target sets among nine algorithms including ReliefF, LLH-Relief, LPLIR, MultiSURF, STIR, MSLIR-NN, MRMR, MPMD and MRelief. In Table 5, nine algorithms are tested on eleven UCI datasets. To compare the nine feature selection algorithms, 50-dimensional irrelevant data, which are obtained from a zero mean and unit variance Gaussian distribution, are added to the original UCI datasets. In Table 6, 200 top-ranked microarray features are selected for feature selection. Microarray datasets have redundant features, so we do not need to add extra noise to the original microarray datasets. The results show that MRelief achieves the highest average classification accuracy on most datasets except Vehicle Silhouettes, Wine and GCM. The results show that the instance force coefficient improves MRelief significantly, and subset generation greatly promotes MRelief. Moreover, the results show that MRelief obtains the lowest average standard deviation on six datasets. The multidirectional neighbor search method improves the stability of MRelief.

Friedman test are conducted with the corresponding post-hoc tests. The Bonferroni–Dunn test is selected as post-hoc tests, which is calculated by the differences between MRelief and other algorithms [42]. The performance of pairwise algorithms is significantly different if the corresponding differences between MRelief and other algorithms is higher than the critical difference ( $CD$ ). The  $CD$  is calculated by (27).

$$CD_{\alpha} = q_{\alpha} \sqrt{\frac{NA(NA + 1)}{6ND}} \tag{27}$$

**Table 4** The average classification accuracies (%) of ReliefF, ReliefF with subset generation, ReliefF with new relief-feature weighting objective function applied to ten datasets

Datasets	ReliefF	ReliefF with subset generation	ReliefF with new relief-feature weighting objective function
Pa	82.78	83.37	84.18
So	70.43	74.22	73.68
HE	75.25	80.13	76.86
Ve	84.15	84.19	85.14
Wi	97.16	98.24	98.27
LU	79.93	94.17	94.88
Le2	74.12	92.47	96.38
Le3	75.35	97.51	95.22
Ly	63.56	96.39	87.27
DL	73.93	86.75	94.28

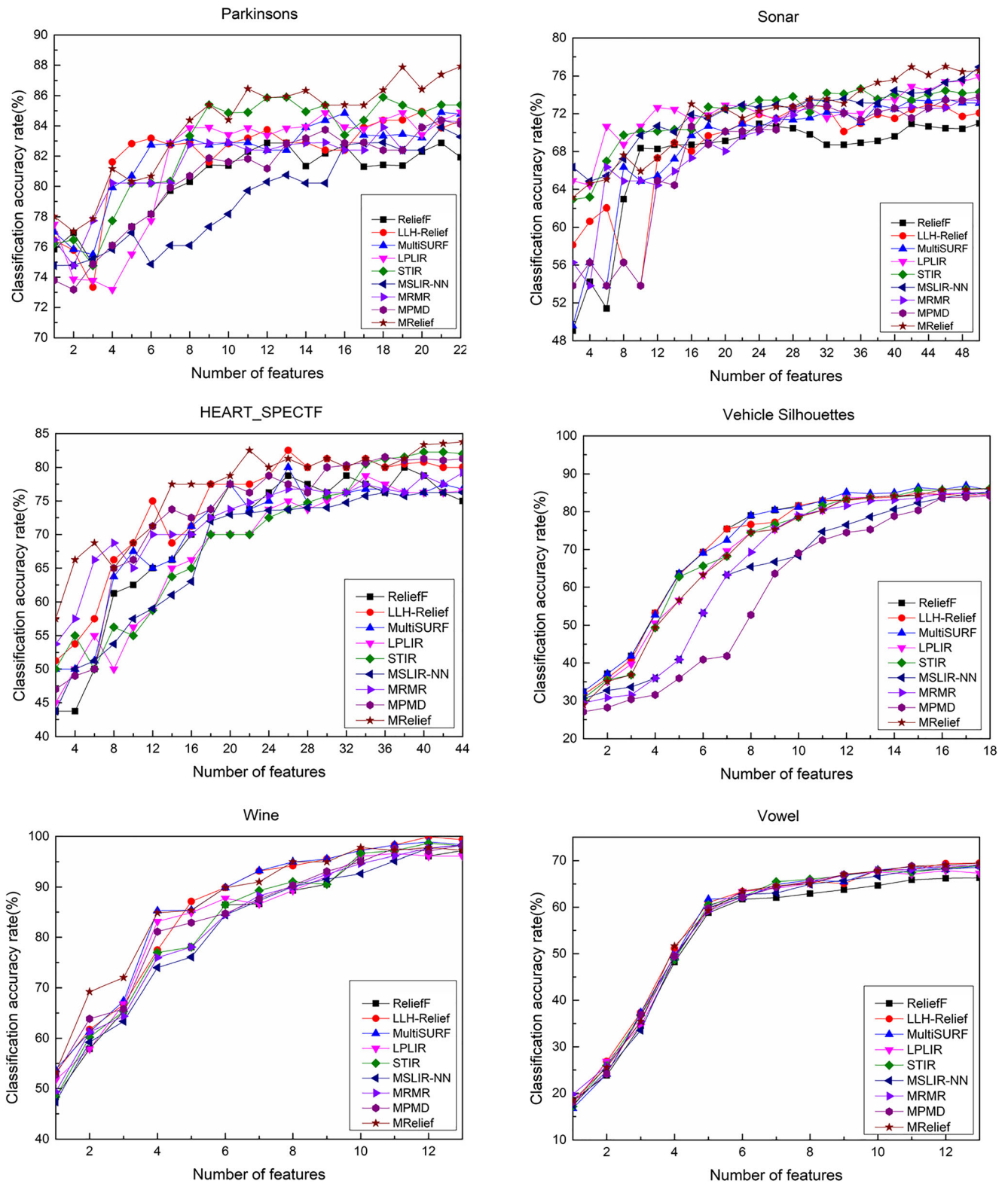


Fig. 3 The average classification accuracy achieved on eleven UCI datasets

where  $NA$  is the number of algorithms,  $ND$  is number of datasets. Generally,  $\alpha$  is set to 0.1,  $q_{0.10} = 2.326$ . In this paper,  $NA = 9$ ,  $ND = 11$ . Thus,  $CD_{0.1} = 2.72$ .

Table 7 and Table 8 show the Friedman test results. In UCI datasets, LPLIR has significant difference compared with ReliefF, LLH-Relief, LPLIR,

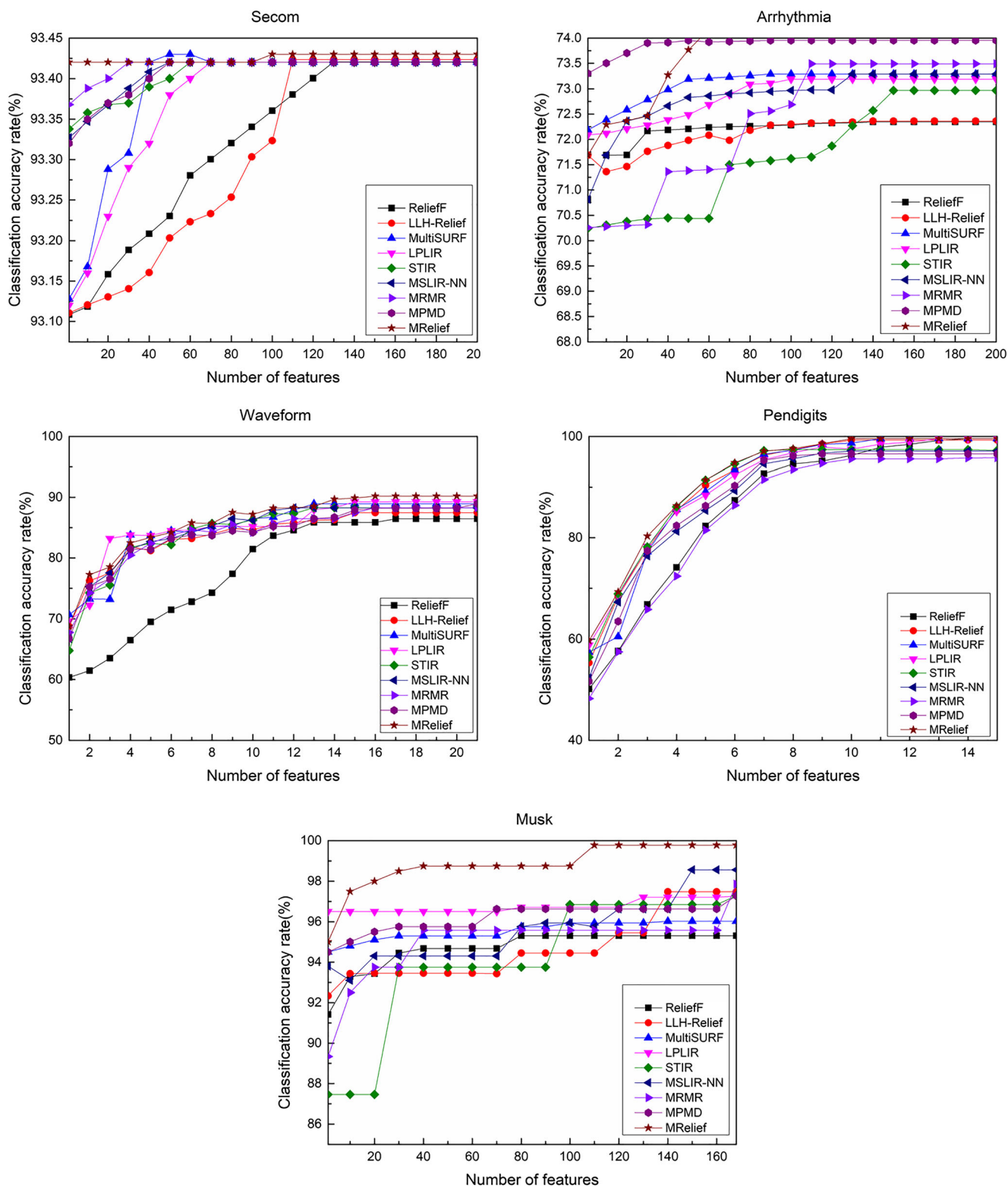


Fig. 3 (continued)

MSLIR-NN and MRMR. In microarray datasets, LPLIR has significant difference compared with other eight algorithms.

Figure 4 shows that MRelief has a performance advantage over the compared Relief-based methods in average classification accuracy. According to the results of LUNG,

**Table 5** The means, standard deviations and ranks of the classification accuracies (%) of nine algorithms applied to eleven UCI datasets

Datasets		ReliefF	LLH-Relief	MultiSURF	LPLIR	STIR	MSLIR-NN	MRMR	MPMD	MRelief
Pa	mean	82.78	84.34	84.86	84.89	85.39	83.28	84.25	84.10	<b>87.92</b>
	std.	3.16	2.86	2.65	<b>0.15</b>	3.21	1.28	1.36	1.49	1.25
	rank	9	5	4	3	2	8	6	7	1
So	mean	70.43	72.83	73.31	75.98	74.52	76.14	73.18	73.57	<b>76.26</b>
	std.	1.52	1.24	1.26	0.41	1.35	1.16	0.89	0.26	<b>0.17</b>
	rank	9	8	5	3	4	2	7	5	1
HE	mean	75.25	79.62	76.75	76.50	76.96	76.25	79.21	80.13	<b>83.33</b>
	std.	4.76	4.23	3.54	0.53	5.24	1.27	1.25	1.94	<b>0.28</b>
	rank	9	3	6	7	5	8	4	2	1
Ve	mean	84.15	85.05	85.70	84.05	<b>85.75</b>	85.15	85.19	84.19	85.70
	std.	1.32	1.09	0.77	0.31	0.93	0.68	0.49	0.37	<b>0.24</b>
	rank	8	6	2.5	9	1	5	4	7	2.5
Wi	mean	97.16	<b>99.53</b>	98.33	98.06	98.33	98.24	98.35	98.24	97.36
	std.	3.93	3.79	2.63	<b>0.42</b>	2.32	1.28	1.38	1.67	1.18
	rank	9	1	3.5	<b>7</b>	3.5	5.5	2	5.5	8
Vo	mean	66.30	<b>69.49</b>	68.99	67.29	68.89	68.57	68.98	69.17	<b>69.49</b>
	std.	2.93	2.46	2.23	0.58	2.32	1.35	0.78	1.04	<b>0.26</b>
	rank	9	1.5	4	8	6	7	5	3	1.5
Se	mean	93.42	93.42	<b>93.43</b>	93.42	93.42	93.42	93.42	93.42	<b>93.43</b>
	std.	<b>0</b>	<b>0</b>	0.01	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	0.01
	rank	6	6	1.5	6	6	6	6	6	1.5
Ar	mean	72.24	72.36	73.29	73.19	72.97	73.29	73.49	73.95	<b>74.77</b>
	std.	1.23	1.47	0.47	1.68	1.04	0.67	0.85	0.39	<b>0.27</b>
	rank	9	8	4.5	6	7	4.5	3	2	1
Wa	mean	86.47	87.46	88.94	89.28	88.29	88.27	88.28	88.95	<b>90.21</b>
	std.	1.98	0.94	0.87	0.69	0.38	<b>0.27</b>	0.36	0.48	0.46
	rank	9	8	4	2	5	<b>7</b>	6	3	1
Pe	mean	<b>99.56</b>	99.29	<b>99.56</b>	<b>99.56</b>	97.28	97.14	95.79	96.56	<b>99.56</b>
	std.	0.06	0.05	0.03	0.05	0.06	0.06	0.04	0.03	<b>0.02</b>
	rank	2.5	5	2.5	2.5	6	7	9	8	2.5
Mu	mean	95.31	97.48	96.02	97.26	97.28	98.57	97.86	97.29	<b>99.77</b>
	std.	1.68	1.37	1.56	1.27	1.03	0.48	0.53	0.35	<b>0.18</b>
	rank	9	4	8	7	6	2	3	5	1
Mean Rank		8.05	5.05	4.23	5.50	4.68	5.64	5.00	4.86	2.00

Leukemia3, Pros1, Pros2, 11-Tumors, and 14-Tumors, it is clear that MRelief achieves the highest classification accuracy over the entire process. In Leukemia2, Lymphoma, DLBCL and Colon, there are crosses among curves. Furthermore, there are overlaps in GCM. In the case of microarray datasets, MRelief is able to achieve the best performance among the compared methods. This implies that MRelief is able to extract useful information from a large number of features. Thus, MRelief achieves better classification performance. The results of Fig. 4 show that the novel subset generation method significantly promotes the classification accuracy rate of MRelief.

Table 9 shows that MRelief achieves superior  $F_1$  results in ten out of eleven datasets. Table 10 shows that MRelief outperforms other algorithms over all datasets. In conclusion, MRelief is competitive among Relief-based methods, regardless of what criteria are adopted.

Tables 11 and 12 show the Friedman test results with the  $F_1$  values. In UCI datasets, MRelief has significant difference compared with ReliefF, LPLIR, MRMR and MPMD. In microarray datasets, LPLIR has significant difference compared with other eight algorithms.

The computational complexity of MRelief is  $O(T_1N^2D + T_2N^2D)$ , which consists of two parts. In the first part, the

**Table 6** The means, standard deviations and ranks of the classification accuracies (%) of nine algorithms applied to the top 200 features selected from eleven microarray datasets

Datasets		ReliefF	LLH-Relief	MultiSURF	LPLIR	STIR	MSLIR-NN	MRMR	MPMD	MRelief
LU	mean	79.93	99.35	77.86	94.84	83.04	86.29	90.24	94.79	<b>99.87</b>
	std.	3.42	3.17	1.92	<b>1.14</b>	2.18	2.14	2.05	2.87	1.54
	rank	8	2	9	3	7	6	5	4	1
Le2	mean	74.12	98.59	97.14	98.15	96.82	91.46	92.46	92.76	<b>98.99</b>
	std.	7.62	3.34	2.24	<b>0.61</b>	3.43	2.46	1.47	2.89	1.13
	rank	9	2	4	3	5	8	7	6	1
Le3	mean	75.35	96.36	95.64	95.83	98.00	95.67	96.81	97.34	<b>99.71</b>
	std.	6.25	3.55	1.07	0.47	4.53	3.47	2.18	4.67	<b>0.19</b>
	rank	9	5	8	6	2	7	4	3	1
Ly	mean	63.56	82.36	83.22	87.36	93.78	94.85	95.18	96.46	<b>98.89</b>
	std.	6.27	4.67	2.14	<b>0.39</b>	3.43	1.76	2.75	1.56	0.58
	rank	9	8	7	6	5	4	3	2	1
DL	mean	73.93	98.40	94.25	99.20	83.01	98.40	92.46	86.49	<b>100</b>
	std.	1.54	1.26	0.53	0.07	0.43	0.76	0.95	2.46	<b>0</b>
	rank	9	3.5	5	2	8	3.5	6	7	1
Co	mean	70.38	90.45	87.39	93.55	93.47	90.25	90.58	91.49	<b>93.74</b>
	std.	5.67	5.48	5.27	3.38	3.89	3.57	4.28	3.79	<b>2.48</b>
	rank	9	6	8	2	3	7	5	4	1
GCM	mean	71.45	<b>90.89</b>	82.46	83.97	87.59	85.69	82.19	82.59	90.78
	std.	6.98	<b>1.08</b>	3.67	2.84	2.39	2.49	3.98	5.28	1.47
	rank	9	1	7	5	3	4	8	6	2
Pr1	mean	79.42	96.89	92.14	93.56	94.57	92.09	73.26	74.68	<b>98.57</b>
	std.	5.89	3.02	6.37	2.89	4.27	3.28	2.78	3.43	<b>2.33</b>
	rank	7	2	5	4	3	6	9	8	1
Pr2	mean	65.44	86.36	65.78	68.47	70.46	72.48	75.39	76.44	<b>88.49</b>
	std.	7.48	3.56	3.87	4.56	5.87	6.48	6.98	7.28	<b>3.46</b>
	rank	9	2	8	7	6	5	4	3	1
11-T	mean	90.32	90.44	90.33	91.47	91.59	92.37	92.57	93.14	<b>94.78</b>
	std.	7.37	3.54	4.68	4.76	6.26	5.14	3.77	4.36	<b>3.28</b>
	rank	9	7	8	6	5	4	3	2	1
14-T	mean	64.53	71.24	69.36	70.22	70.47	70.88	72.24	73.13	<b>76.22</b>
	std.	6.47	<b>1.28</b>	7.58	3.96	3.77	4.52	6.38	6.24	2.31
	rank	9	4	8	7	6	5	3	2	1
Mean Rank		8.73	3.86	7.00	4.64	4.82	5.41	5.18	4.27	1.09

computational complexity of MRelief is  $O(T_1 N^2 D)$ .  $T_1$  represents the maximum number of iterations of MRelief without subset generation.  $D$  represents the number of features.  $N$

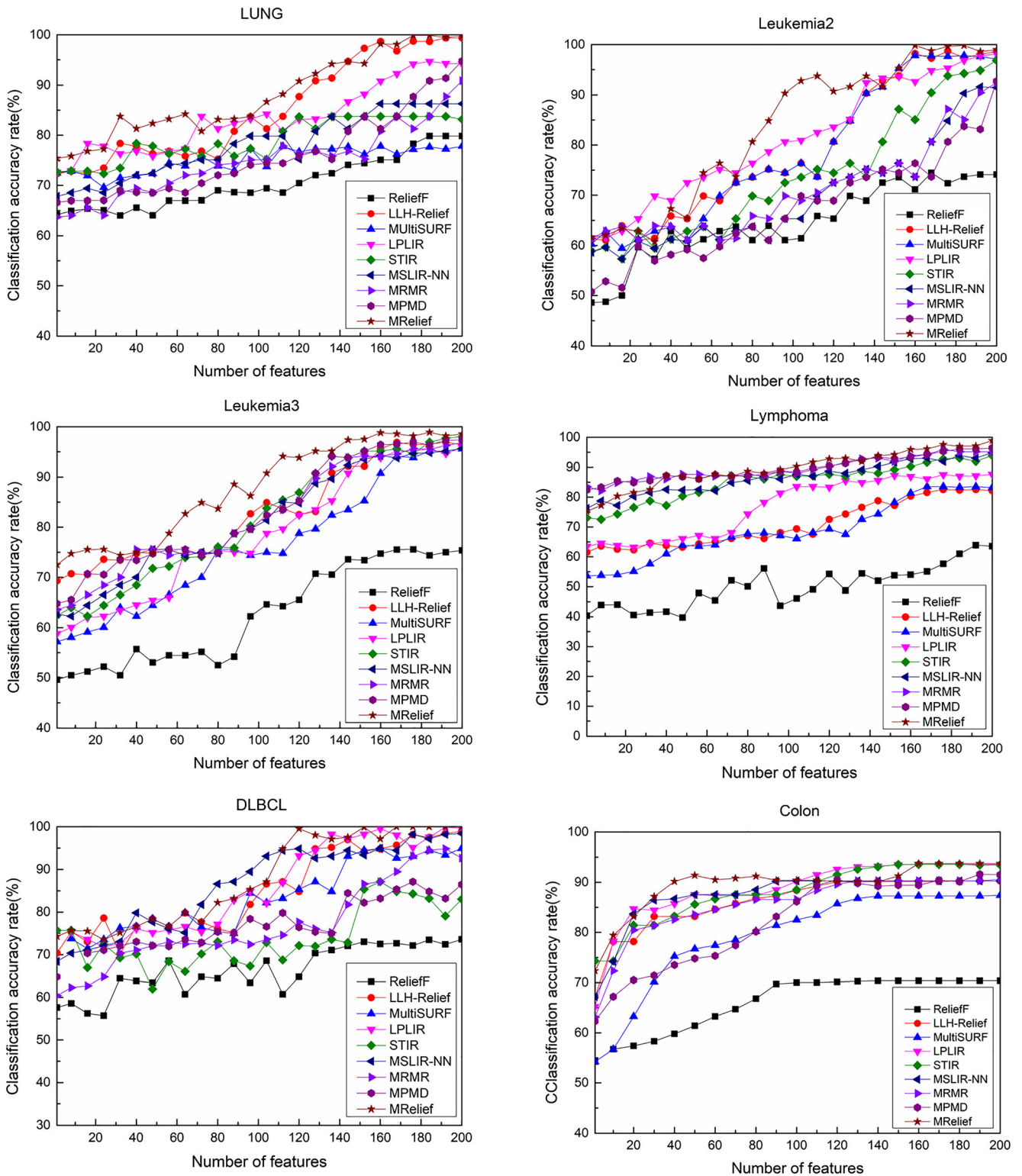
represents the number of instances. The computational complexity is the same as ReliefF because the multidirectional neighbor search method, new relief-feature weighting

**Table 7** Friedman tests with the classification accuracies to compare MRelief with other eight methods applied to eleven UCI datasets

$CD_{0.1}$	ReliefF	LLH-Relief	MultiSURF	LPLIR	STIR	MSLIR-NN	MRMR	MPMD
2.72	6.05	3.05	2.23	3.50	2.68	3.64	3.00	2.86

**Table 8** Friedman tests with the classification accuracies to compare MRelief with other eight methods applied to eleven microarray datasets

$CD_{0.1}$	ReliefF	LLH-Relief	MultiSURF	LPLIR	STIR	MSLIR-NN	MRMR	MPMD
2.72	7.64	2.77	5.91	3.55	3.73	4.32	4.09	3.18



**Fig. 4** The average classification accuracy achieved on eleven microarray datasets

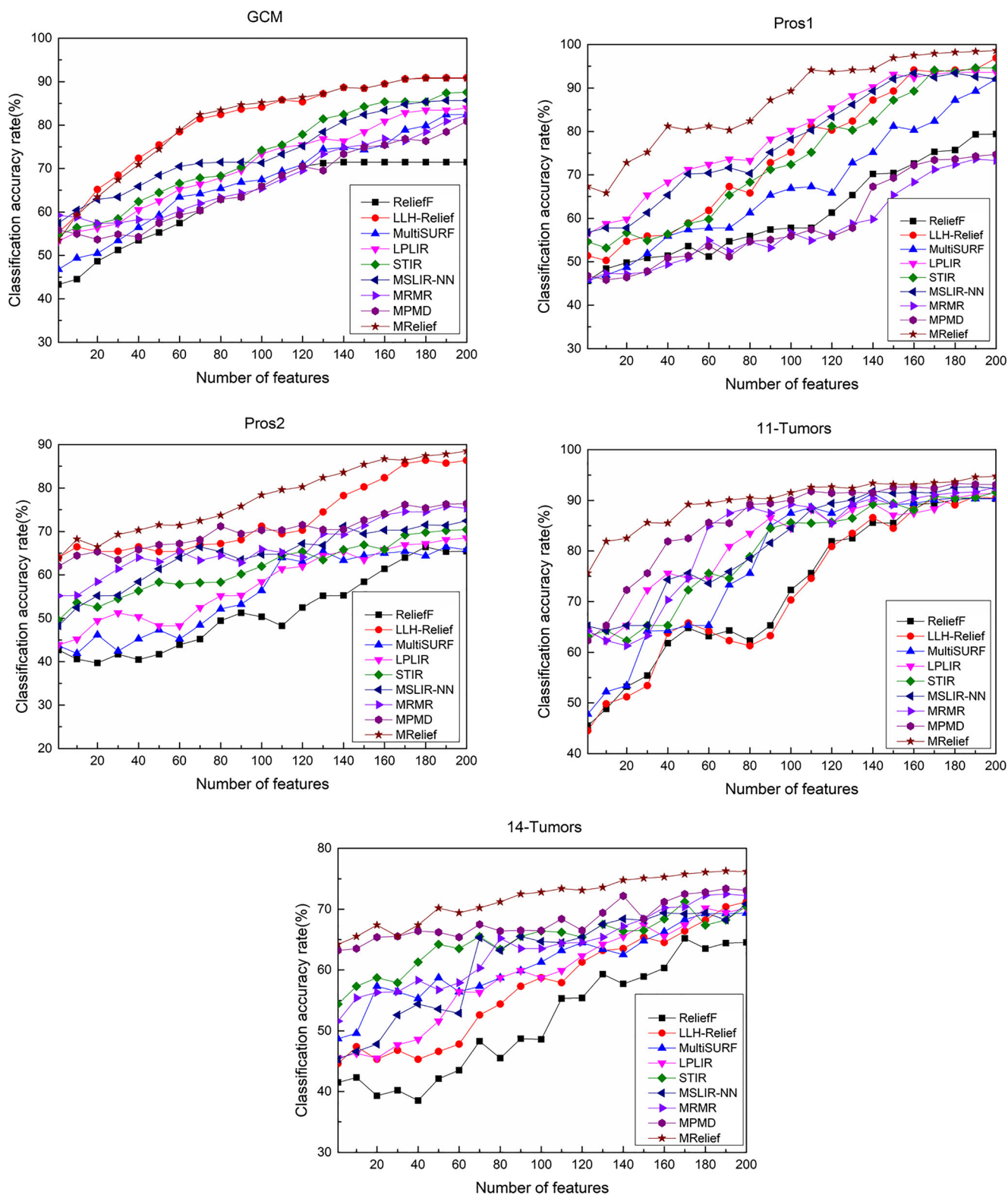


Fig. 4 (continued)

objective function and multiclass extension do not increase the computational complexity of MRelief. In the second part, the computational complexity of MRelief is  $O(T_2N^2D)$  because subset generation is combined with MPMD.  $T_2$  represents

the maximum number of MPMD. The experiments include state-of-the-art relief-based methods and mutual-information-based methods. The representative relief-based method is ReliefF, and the representative

**Table 9** The mean  $F_1$  values and ranks of nine algorithms applied to the top 200 features selected from eleven UCI datasets

Datasets	ReliefF	LLH-Relief	MultiSURF	LPLIR	STIR	MSLIR-NN	MRMR	MPMD	MRelief
Pa	0.83(8)	0.86(2.5)	0.85(5)	0.85(5)	0.86(2.5)	0.84(7)	0.85(5)	0.81(9)	<b>0.88(1)</b>
So	0.72(8)	0.74(4.5)	0.74(4.5)	0.73(6.5)	0.75(3)	<b>0.78(1.5)</b>	0.71(9)	0.73(6.5)	<b>0.78(1.5)</b>
HE	0.71(9)	0.75(6)	0.73(7.5)	0.73(7.5)	<b>0.85(1.5)</b>	0.80(4)	0.82(3)	0.78(5)	<b>0.85(1.5)</b>
Ve	0.86(3.5)	0.85(5.5)	0.81(8.5)	0.85(5.5)	0.86(3.5)	0.87(2)	0.82(7)	0.81(8.5)	<b>0.88(1)</b>
Wi	0.95(9)	<b>0.99(3)</b>	<b>0.99(3)</b>	0.97(7.5)	<b>0.99(3)</b>	0.98(6)	<b>0.99(3)</b>	<b>0.99(3)</b>	0.97(7.5)
Vo	0.71(4.5)	<b>0.75(1)</b>	0.70(6.5)	0.68(8)	0.65(9)	0.72(3)	0.71(4.5)	0.70(6.5)	0.74(2)
Se	<b>0.92(5)</b>	<b>0.92(5)</b>	<b>0.92(5)</b>	<b>0.92(5)</b>	<b>0.92(5)</b>	<b>0.92(5)</b>	<b>0.92(5)</b>	<b>0.92(5)</b>	<b>0.92(5)</b>
Ar	0.70(9)	0.74(4.5)	<b>0.75(2)</b>	0.74(4.5)	0.71(8)	<b>0.75(2)</b>	0.72(7)	0.73(6)	<b>0.75(2)</b>
Wa	0.88(5.5)	0.87(8.5)	0.89(2.5)	0.87(8.5)	0.89(2.5)	0.88(5.5)	0.88(5.5)	0.88(5.5)	<b>0.92(1)</b>
Pe	<b>0.99(4)</b>	<b>0.99(4)</b>	<b>0.99(4)</b>	<b>0.99(4)</b>	<b>0.99(4)</b>	<b>0.99(4)</b>	0.85(8)	0.81(9)	<b>0.99(4)</b>
Mu	0.94(9)	0.97(3.5)	0.97(3.5)	0.96(6)	0.96(6)	0.98(2)	0.95(8)	0.96(6)	<b>0.99(1)</b>
Mean rank	6.77	4.36	4.72	6.18	4.36	3.81	5.91	6.36	2.50

mutual-information-based method is MRMR. Thus, ReliefF and MRMR are selected as comparison algorithms. The computational complexity of MRMR and ReliefF is  $O(N^2D)$ . Therefore, the computational complexity of MRelief is the highest compared to ReliefF and MRMR.

In Tables 13 and 14, without calculating the execution time of the classifier, the execution time consumption is calculated in one iteration. Table 13 shows the mean running times of nine algorithms on the top 200 features selected from eleven UCI datasets. Table 14 shows the mean running times of nine algorithms on the top 200 features selected from eleven microarray datasets. It can be observed that the mean running times of MRelief is higher than the other two algorithms in both UCI and microarray datasets. Since MRelief combines

ReliefF and MRMR, the running time of MRelief is almost equal to the sum of the running time of ReliefF and MRMR.

Table 15 shows the results of eight pairs of Wilcoxon signed-rank tests performed on the Colon dataset. The results show that the performance of MRelief is statistically significantly better than those of ReliefF, LLH-Relief, MultiSURF, LPLIR, STIR, MSLIR-NN, MRMR and MPMD at a significance level of 0.05. MRelief has demonstrated its robustness in handling noisy datasets whose interaction effect is weak. MRelief is able to extract useful information from a large number of features. Therefore, the multidirectional neighbor search method, new relief-feature weighting objective function, subset generation and multiclass extension make contribution to MRelief.

**Table 10** The mean  $F_1$  values and ranks of nine algorithms applied to the top 200 features selected from eleven microarray datasets by nine algorithms

Datasets	ReliefF	LLH-Relief	MultiSURF	LPLIR	STIR	MSLIR-NN	MRMR	MPMD	MRelief
LU	0.75(8)	<b>0.99(1.5)</b>	0.75(8)	0.92(3)	0.75(8)	0.88(6)	0.91(4)	0.90(5)	<b>0.99(1.5)</b>
Le2	0.70(9)	<b>0.99(1.5)</b>	0.98(3.5)	0.98(3.5)	0.95(5)	0.87(8)	0.89(7)	0.93(6)	<b>0.99(1.5)</b>
Le3	0.77(9)	0.95(5)	0.92(8)	0.94(7)	0.96(3)	0.98(2)	0.95(5)	0.95(5)	<b>0.99(1)</b>
Ly	0.65(9)	0.85(7)	0.81(8)	0.91(5)	0.90(6)	0.95(4)	0.96(3)	0.97(2)	<b>0.99(1)</b>
DL	0.70(9)	0.98(3.5)	0.92(5)	0.99(2)	0.81(8)	0.98(3.5)	0.88(6)	0.85(7)	<b>1(1)</b>
Co	0.65(9)	0.94(4)	0.90(6)	<b>0.95(2)</b>	<b>0.95(2)</b>	0.87(7)	0.86(8)	0.93(5)	<b>0.95(2)</b>
GCM	0.73(9)	<b>0.92(1.5)</b>	0.85(4.5)	0.81(6)	0.85(4.5)	0.86(3)	0.78(8)	0.79(7)	<b>0.92(1.5)</b>
Pr1	0.82(9)	0.97(2)	0.90(8)	0.94(6.5)	0.95(4.5)	0.94(6.5)	0.95(4.5)	0.96(3)	<b>0.99(1)</b>
Pr2	0.68(9)	0.88(2)	0.69(8)	0.71(7)	0.74(5)	0.73(6)	0.77(4)	0.78(3)	<b>0.92(1)</b>
11-T	0.85(9)	0.87(8)	0.91(7)	0.92(5.5)	0.92(5.5)	0.93(3.5)	0.93(3.5)	0.94(2)	<b>0.96(1)</b>
14-T	0.58(9)	0.67(8)	0.72(5.5)	0.72(5.5)	0.75(2.5)	0.69(7)	0.74(4)	0.75(2.5)	<b>0.78(1)</b>
Mean rank	8.91	4.00	6.50	4.82	4.91	5.14	5.18	4.32	1.23



**Table 11** Friedman tests with the  $F_1$  values to compare MRelief with other eight methods applied to eleven microarray datasets

$CD_{0.1}$	ReliefF	LLH-Relief	MultiSURF	LPLIR	STIR	MSLIR-NN	MRMR	MPMD
2.72	4.27	1.86	2.22	3.68	1.86	1.31	3.41	3.86

**Table 12** Friedman tests with the  $F_1$  values to compare MRelief with other eight methods applied to eleven microarray datasets

$CD_{0.1}$	ReliefF	LLH-Relief	MultiSURF	LPLIR	STIR	MSLIR-NN	MRMR	MPMD
2.72	7.68	2.77	5.27	3.59	3.68	3.91	3.95	3.09

### 5 Discussion

Tables 5 and 6 show that MRelief achieves the highest average classification accuracy rate on most datasets except Vehicle Silhouettes, Wine and GCM. Although MRelief does not obtain the best learning accuracies compared with other algorithms on all datasets, the learning accuracies of MRelief are better than those of other algorithms on ten microarray datasets except GCM. MRelief is more suitable for high-dimensional datasets. In addition, Tables 5 and 6 show that MRelief achieves the highest Rank compared with other algorithms on both UCI datasets and microarray datasets. Furthermore, Figs. 3 and 4 show that MRelief performs the best on the general trend with respect to the average classification accuracy. The results show that the instance force coefficient improves MRelief significantly, and subset generation greatly promotes MRelief. Moreover, Tables 5 and 6 show that MRelief obtains the lowest average standard deviation in most datasets. Therefore, the proposed multidirectional neighbor search method is able to select representative neighbors for feature selection.

### 6 Conclusion and further work

Relief is a feature selection method that reduces the number of features. Relief is the only individual evaluation filter algorithm that is capable of detecting feature dependencies. Due to the low classification accuracy rate of Relief, a novel filter feature selection algorithm named MRelief is proposed in this paper. First, the multidirectional neighbor search method with a distance threshold is designed. Second, to improve the classification accuracy, a reasonable objective function assigns instances different force coefficients based on instances' contribution to classification. Third, subset generation is proposed to obtain the optimal candidate subset. Last, the MRelief algorithm implements a multiclass margin definition to handle multiclass data.

To demonstrate the effectiveness of MRelief, experiments were conducted on 11 UCI and 11 microarray datasets. MRelief is compared with LPLIR, ReliefF, LLH-Relief, MultiSURF, MSLIR-NN, MRMR, MPMD and STIR. The results show that the multidirectional neighbor search method,

**Table 13** The mean running times (s) of nine algorithms on the top 200 features selected from eleven UCI datasets

Datasets	ReliefF	MRMR	MRelief
Pa	0.03	0.01	0.05
So	0.08	0.11	0.27
HE	0.03	0.01	0.05
Ve	0.08	0.03	0.14
Wi	0.04	0.01	0.05
Vo	0.10	0.02	0.13
Se	4.92	15.16	20.23
Ar	3.15	8.28	12.43
Wa	0.53	0.04	0.61
Pe	1.12	0.05	1.32
Mu	5.16	7.56	12.47

**Table 14** The mean running times of nine algorithms on the top 200 features selected from eleven microarray datasets

Datasets	ReliefF	MRMR	MRelief
LU	5.27	9.57	14.86
Le2	3.81	8.61	11.89
Le3	4.13	8.36	12.65
Ly	2.13	8.47	11.57
DL	4.12	7.94	12.53
Co	0.76	8.27	9.26
GC	5.65	9.86	15.78
Pr1	4.38	8.46	13.36
Pr2	4.03	8.57	12.44
11-T	6.23	9.27	15.83
14-T	7.35	10.35	18.56

**Table. 15** Comparison based on the Wilcoxon signed-rank test on the Colon dataset

Algorithm1	MRelief	MRelief	MRelief	MRelief	MRelief	MRelief	MRelief	MRelief
Algorithm2	ReliefF	LLH-Relief	MultiSURF	LPLIR	STIR	MSLIR-NN	MRMR	MPMD
<i>p</i> value	0.007	0.038	0.003	0.015	0.021	0.001	0.031	0.047

new relief-feature weighting objective function, subset generation and multiclass extension significantly improve MRelief.

In the future, we will improve running time of the MRelief in handling with largescale datasets and high dimensional datasets, which is important to their application to gene expression. Since MRelief combines ReliefF and MRMR, the running time of MRelief is consist of the running time of ReliefF and MRMR. To deal with this shortage, we will improve mRMR to save the execution time. Moreover, we will improve the classification accuracy rate of MRelief combined with wrapper method. Furthermore, we robust MRelief to deal with multiple types of datasets, such as missing datasets, unlabeled datasets and muti-label datasets.

**Acknowledgments** This research was supported by the Department of Science and Technology of Jilin Province (No. 20190303135SF).

## References

- Craven M, DiPasqua D, Freitagb D, McCalluma A, Mitchella T, Nigama K, Slattery S (2000) Learning to construct knowledge bases from the world wide web. *Artif Intell* 118(1–2):69–113
- Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326
- Kushwaha N, Pant M (2018) Link based BPSO for feature selection in big data text clustering. *Future Gener Comput Syst* 82:190–199
- Blum AL, Rivest RL (1992) Training a 3-node neural networks is NP-complete. *Neural Netw* 5(1):117–127
- Men M, Zhong P, Wang Z, Lin Q (2020) Distributed learning for supervised multiview feature selection. *Appl Intell* 50(9):2749–2769
- Xiang S, Shen XT, Ye JP (2015) Efficient nonconvex sparse group feature selection via continuous and discrete optimization. *Artif Intell* 224:28–50
- Zhang P, Gao WF, Liu GX (2018) Feature selection considering weighted relevancy. *Appl Intell* 48(12):4615–4625
- Cohen JP, Ding W, Kuhlman C, Chen AJ, Di LP (2016) Rapid building detection using machine learning. *Appl Intell* 45(2):443–457
- Xiao J, Cao HW, Jiang X, Jiang XY, Gu X, Xie L (2017) GMDH-based semi-supervised feature selection for customer classification. *Knowledge-Based Syst* 132:236–246
- Belkoura S, Zanin M, LaTorre A (2019) Fostering interpretability of data mining models through data perturbation. *Expert Syst Appl* 137:191–201
- Ji CY, Li Y, Fan JH, Lan SM (2019) A novel simplification method for 3D geometric point cloud based on the importance of point. *IEEE ACCESS* 7:129029–129042
- Zheng YF, Li Y, Wang G, Chen YP, Xu Q, Fan JH, Cui XT (2018) A novel hybrid algorithm for feature selection. *Pers Ubiquit Comput* 22(5–6):971–985
- Chen YP, Li Y, Wang G, Zheng YP, Xu Q, Fan JH, Cui XT (2017) A novel bacterial foraging optimization algorithm for feature selection. *Expert Syst Appl* 83:1–17
- Turabieh H, Mafarja M, Li XD (2019) An introduction to variable and feature selection. *Expert Syst Appl* 3:1157–1182
- Cui XT, Li Y, Fan JH, Wang T, Zheng YF (2020) A hybrid improved dragonfly algorithm for feature selection. *IEEE ACCESS* 8:155619–155629
- Guyon I, Gunn S, Nikravesh M, Zadeh LA (2006) Feature extraction, foundations and applications. Springer
- Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 27(8):1226–1238
- Urbanowicz RJ, Meeker M, Cava WL, Olson RS, Moore JH (2018) Relief-based feature selection: introduction and review. *J Biomed Informat* 85:189–203
- Li Y, Wang G, Chen HL, Dong H, Zhu X, Wang S (2011) An improved particle swarm optimization for feature selection. *J Bionic Eng* 8(2):191–200
- Nazemi A, Dehghan M (2015) A neural network method for solving support vector classification problems. *Neurocomputing* 152:369–376
- Kim DH, Abraham A, Cho JH (2007) A hybrid genetic algorithm and bacterial foraging approach for global optimization. *InfSci* 177(18):3918–3937
- Kashef S, Nezamabadi-pour H (2015) An advanced ACO algorithm for feature subset selection. *Neurocomputing* 147:271–279
- Mohapatra P, Chakravarty S, Dash PK (2015) An improved cuckoo search based extreme learning machine for medical data classification. *Swarm Evol Compu* 24:25–49
- Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif Intell Rev* 42(1):21–57
- Chen H, Li WD, Yang X (2020) A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems. *Expert Syst Appl* 158:113612
- Abdel-Basset M, Mohamed R, Mirjalili S (2021) A novel whale optimization algorithm integrated with Nelder-Mead simplex for multi-objective optimization problems. *Knowledge-Based Syst* 212:106619
- Mafarja M, Aljarah I, Heidari AA, Faris H, Fournier-Viger P, Li XD, Mirjalili S (2018) Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowledge-Based Syst* 161:185–204
- Reyes O, Morell C (2015) Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context. *Neurocomputing* 161:168–182
- Yilmaz T, Yazici A, Kitsuregawa M (2014) RELIEF-MM: effective modality weighting for multimedia information retrieval. *Multimedia Syst* 20(4):389–413

30. Sun Y (2007) Iterative RELIEF for feature weighting: algorithms, theories, and applications. *IEEE Trans Pattern Anal Mach Intell* 29 (6):1035–1051
31. Tan B, Zhang L (2020) Local preserving logistic I-relief for semi-supervised feature selection. *Neurocomputing* 399:48–64
32. Greene CS, Penrod NM, Kiralis J, Moore JH (2009) Spatially uniform relief (SURF) for computationally-efficient filtering of gene-gene interactions. *BioData Min* 2(1):5
33. Urbanowicz RJ, Olson RS, Schmit P, Meeker M, Moore JH (2018) Benchmarking relief-based feature selection methods for bioinformatics data mining. *J Biomed Informat* 85:168–188
34. McKinney BA, White BC, Grill DE, Li PW, Kennedy RB, Poland GA, Oberg AL (2013) ReliefSeq: a gene-wise adaptive-K nearest-neighbor feature selection tool for finding gene-gene interactions and main effects in mRNA-Seq gene expression data. *PLoS One* 8: e81527
35. Robnik-Sikonja M, Kononenko I (2003) Theoretical and empirical analysis of ReliefF and RReliefF. *Mach Learn* 53(1–2):23–69
36. Le TT, Urbanowicz RJ, Moore JH, Mckinney BA (2018) Statistical inference relief (STIR) feature selection. *Bioinformatics* 35(8): 1358–1365
37. Zheng YF, Li Y, Wang G, Chen YP, Xu Q, Fan JH, Cui XT (2019) A novel hybrid algorithm for feature selection based on whale optimization algorithm. *IEEE ACCESS* 7:14908–14923
38. Cortes C, Vapnik V (1995) Support-vector network. *Mach Learn* 20(3):273–297
39. Leopold E, Kindermann J (2001) Text categorization with support vector machines. How to represent texts in input space? *Mach Learn* 46(1–3):423–444
40. Tang BG, Zhang L (2019) Multi-class semi-supervised logistic i-relief feature selection based on nearest neighbor. In: Pacific-Asia conference on knowledge discovery and data mining, pp 281–292
41. Zhang L, Huang X, Zhou W (2019) Logistic local hyperplane-relief: a feature weighting method for classification. *Knowledge-Based Syst* 181:104741
42. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30

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



**Xueting Cui** received the B.S. degree in software engineering from the China University of Jilin. She is currently pursuing the Ph.D. degree in the College of Computer Science and Technology in Jilin University. Her research interests include object detection and feature selection.



3D visual modeling, 3D image processing, machine vision and machine learning.

**Ying Li** received the B.S., M.S., and Ph.D. degrees from Jilin University. From 2000 to 2006, she was an Associate Professor with the Department of Space Information Processing, Jilin University. Since 2006, she has been a Professor in the computer application technology with the Jilin University. She is currently a fellow of the China Computer Federation. She has published over 60 papers in journals and international conference. Her research interests include big data,



**Jiahao Fan** received the B.S. degree in Computer Science and Technology College from the Jilin University, Chang Chun, China, in 2015. From 2015 to 2017, he was a graduate student in Jilin University. He is currently pursuing the Ph.D. degree in the Jilin University. His research interests include swarm intelligence algorithm, machine learning, image processing, data mining, and 3D data processing.



**Tan Wang** received the B.S. degree in journalism and communication from the Jilin University of Finance and Economics, Changchun, China, in 2017, and M.S. degree in the Northeast Normal University in 2020. She is currently working part-time in Space technology (Jilin) Limited Company. Her research interests include media technology ethics, network ecological and feature selection.