Applied Intelligence (2020) 50:101–118
https://doi.org/10.1007/s10489-019-01518-0

Feature selection with Symmetrical Complementary Coefficient for quantifying feature interactions

Rui Zhang[1] · Zuoquan Zhang[1]

Published online: 3 July 2019

Abstract

In the field of machine learning and data mining, feature interaction is a ubiquitous issue that cannot be ignored and has attracted more attention in recent years. In this paper, we proposed the Symmetrical Complementary Coefficient which can quantify feature interactions very well. Based on it, we improved the Sequential Forward Selection (SFS) algorithm and proposed a new feature subset searching algorithm called SCom-SFS which only needs to consider the feature interactions between adjacent features on a given sequence instead of all of them. Moreover, discovered feature interactions can speed up the process of searching for the optimal feature subset. In addition, we have improved the ReliefF algorithm by screening out representative samples from the original data set, and need not to sample the samples. The improved ReliefF algorithm has been proved to be more efficient and reliable. An effective and complete feature selection algorithm RRSS is obtained through the combination of the two modified algorithms. According to the experimental results, the proposed algorithm RRSS outperformed five classic and two latest feature selection algorithms in terms of size of resulting feature subset, Accuracy, Kappa coefficient, and adjusted Mean-Square Error (MSE).

Keywords Feature selection · ReliefF · Sequential Forward Selection · Feature interaction · Random Forest · Symmetrical Complementary Coefficient

1 Introduction

Feature selection is an important part of machine learning and data mining. Modern data sets are getting wider and bigger. But sometimes this is not a good thing for some tasks in machine learning such as classification and this problem was described by [12] in detail. And they roughly divided the methods of feature selection into three categories: filter methods such as mRMR algorithm [25], wrapper methods such as RFE algorithm [27], Boruta algorithm [20] and embedded methods such as L1 and L2 regularization [23] and different kinds of decision trees and their ensembles such as Random Forest [2]. And the elements and process of feature selection can be described as generation, evaluation, stopping criterion and validation [4]. In addition, there are many ways to classify feature selection methods. According to the evaluation of the relativity between features and the class variable, they can be divided into the consistency measure [4, 21, 38], the dependence measure [13, 35], the distance measure [19, 24] and the information theory measure [7, 18].

Furthermore, the current feature selection methods can be classified according to whether they can cope with different types of features which can be grouped into irrelevant features, redundant features and the interactive features. Irrelevant features do not help with learning tasks [16]. Searching and deleting irrelevant features has always been the major topic of feature selection. These methods often give feature weights and ordering by measuring the degree of association between features and class variable [28]. But most of them cannot remove redundant features, such as Relief [17] and its extension ReliefF [19]. Redundant features have a negligible effect on learning and forecasting because most information provided by them has already been presented by other features [36]. Some

✉ Zuoquan Zhang
zuoquanzhang@163.com

Rui Zhang
15600825195@163.com

[1] School of Science, Beijing Jiaotong University, Beijing, China

algorithms take it into consideration, such as CFS [13], FCBF [35], CMIM [8], ACE [32]. Whereas, they cannot recognize feature interactions.

A feature interaction is a ubiquitous issue that cannot be ignored and has attracted more attention in recent years [15]. A simple example of feature interactions is the concept of exclusive OR operation: $C = XOR(X_1, X_2)$ where C is a Boolean label, X_1 and X_2 are the Boolean features. X_1 and X_2 are irrelevant with Y separately, however they have a strong relevance to Y when they are combined together. Thus, Removing any one of them will have a bad effect. There are few researches on feature interaction currently. Jakulin and Bratko [14] introduced the concept of interactive gain and used interactive information as a way of qualifying the feature interactions. Wang et al. [34] provides a flexible framework to extract different types of information which can deal with feature interactions. Wang and Song [33] proposed FEAST algorithm which is based on the association rule mining and takes feature interactions into account. Zeng et al. [37] proposed the interaction weight factor which can reflect the information of whether a feature is redundant or interactive and then brought forward an Interaction Weight based Feature Selection(IWFS) algorithm. Gao et al. [10] took composition of feature relevancy into account and proposed the Composition of Feature Relevancy (CFR) feature selection method. Tang et al. [31] proposed the Five-dimensional Joint Mutual Information(FJMI) feature selection algorithm which took the higher-order feature interactions into account and adopted the 'maximum of the minimum' method. Whereas, there is no widely accepted method currently, in particular, a better way to measure feature interactions.

In this paper, we will propose a new method of quantifying feature interactions, namely, the Symmetrical Complementary Coefficient. The Symmetrical Complementary Coefficient is based on the Enhanced Complementary Coefficient which is an improved version of the Complementary Coefficient [30]. Based on the Symmetrical Complementary Coefficient, we will improve the Sequential Forward Selection (SFS) algorithm and propose a new feature subset search algorithm which is called SCom-SFS. In addition, we will propose an improved ReliefF algorithm by selecting representative instances. An effective and complete feature selection algorithm RRSS will be obtained through the combination of the two modified algorithms. And it will be compared with other five classic feature selection algorithms and two latest algorithms in terms of four evaluation metrics, i.e., size of resulting feature subset, Accuracy, Kappa coefficient, and adjusted Mean-Square Error (MSE).

The rest of the paper is organized as follows. In Section 2, we describe the preliminaries. In Section 3, we introduce the RF-efficient-ReliefF algorithm. In Section 4, we give the definition of the Symmetrical Complementary Coefficient and propose the SCom-SFS algorithm. In Section 5, we bring in the RRSS algorithm and analyze its time complexity and parameter's selection. In Section 6 and 7, we present the experiment, results and its analyses. Section 8 provides conclusions.

2 Preliminaries

This section introduces the knowledge associated with the topic. Firstly, let's introduce the famous ReliefF algorithm, then introduce the SFS algorithm for feature subset generation, and then introduce the concept of feature interaction. Finally, we introduce the Random Forest and its feature selection algorithm.

2.1 ReliefF algorithm

Relief algorithm is a classical filter method of feature selection [17]. It is a method of calculating feature weights for the binary classification problem. Kononenko [19] extended it to ReliefF algorithm, which can be used for multi-class problems. The basic idea of calculating feature weights is to assign weights to the features through distance measures. It believes that good features should make the distances between samples with the same class as short as possible, while the distances between samples with the different classes are as long as possible. The weight of feature A in each sampling is $W(A)$, m is the sampling times, S is the original data set, and n is the number of features. The details of the algorithm are in Algorithm 1.

According to the obtained features' weights, the threshold of the ordering and weights can be given, and the features with lower weights will be deleted. The ReliefF algorithm only needs to calculate the feature weights on the samples which are randomly sampled rather than on the entire data set. The time cost of this algorithm depends on the number of features and the sampling times.

2.2 Sequential Forward Selection

As a feature subset searching method, Sequential Forward Selection (SFS) is essentially a kind of greedy algorithm. The feature subset starts from an empty set, and features are added one by one if it makes the evaluation function better. So, evaluation function achieves the optimal in this way. When there are no features to be selected, the algorithm is terminated. The evaluation function can be the classifier-dependent metrics such as accuracy or some other classifier-independent metrics. The advantage of using classifier related metrics is higher accuracy, but often cost too much time. On the contrary, the advantage of using classifier-independent metrics is the faster speed, but the accuracy is not that high.

Algorithm 1 ReliefF algorithm.

```
Input the data set $S$ and the sampling times $m$. The weights of all features $W(\cdot)$ are initialized to 0.
for $i$ from 1 to $m$ do
    A sample denoted as $R$ is randomly selected from $S$;
    In $S$, find out $R's$ $k$ nearest neighbor samples $NH_j (j = 1, 2, \ldots, k)$ which has the same class with $R$ and find out $R's$ $k$ nearest neighbor samples $NM_j(c)(j = 1, 2, \ldots, k)$ whose class is $c$ and $c$ is different from the class of $R$;
    for Feature $A$ from 1 to n do
        Use following formula to calculate weights of $A$;
        $W(A) = W(A) - \sum_{i=1}^{k} diff(A, R, NH_j)/(mk) + \sum_{c \neq class(R)} \left[ \frac{p(c)}{1 - p(class(R))} \cdot \sum_{j=1}^{k} diff(A, R, NM_j(c)) \right] /(mk)$
        where
        $diff(A, R_1, R_2) = \begin{cases} \frac{|R_1[A]-R_2[A]|}{\max(A)-\min(A)} & \text{If } A \text{ is continuous;} \\ 0 & \text{If } A \text{ is discrete and } R_1[A] = R_2[A]; \\ 1 & \text{If } A \text{ is discrete and } R_1[A] \neq R_2[A]; \end{cases}$
        And class(R) is the class of the sample $R$
    end for
end for
Output features' weights.
```

2.3 Definition of feature interactions

Consider a supervised learning problem with one class variable and n features. Feature interaction represents that multiple features can be more effective than the sum of the effects of the single feature. According to this, an informal definition of feature interaction has been given [15].

k-way feature interaction works on k features $X_1, X_2, \ldots, X_k$. Define the function $e(C; X_1, X_2, \ldots, X_i)$ $(i = 1, 2, \ldots, k)$ which evaluates the contribution from feature(s) to the class variable C. If

$$e(C; X_1, X_2, \ldots, X_k) \geq \sum_{i=1}^{k} e(C; X_i)$$

we can deem that there is a k-way feature interaction between these k features. And these features are called the interactive features. The difference between them

$$e(C; X_1, X_2, \ldots, X_k) - \sum_{i=1}^{k} e(C; X_i)$$

can be seen as the strength of the existed feature interaction.

According to the concept of feature interaction, some methods can be designed to describe it. Whereas there isn't a particularly good way to do it. It's not easy to search out interactive features, and multi-way interaction is complex [14, 33, 34, 37]. But the quantification of feature interactions is still very profitable and worth doing.

2.4 Random Forest

Random forest (RF) is an integrated machine learning algorithm proposed by [2]. RF is one of the hottest algorithms in the field of machine learning and data mining. It is data-oriented and does not need any assumption on the distribution of the data set. Compared with other models, it has a faster operating speed and a better performance on imbalanced data, multi-variable data, and big data. It is robust to outliers and noises, and without over-fitting problems. K sample sets $\{S_k\}_{k=1}^{K}$ are obtained from the initial data set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ through bootstrap sampling method. A decision tree h_k is built on each data set S_k. Each node of each tree randomly selects the same number of features for splitting. The combination of decision trees is the Random Forest. Then the result is a vote or an average given by these decision trees. It depends on whether the question is classification or regression. For classification problems, the results can be expressed by the following formula.

$$H(x) = \arg\max_y \sum_{k=1}^{K} I(h_k(x) = y)$$

where $h_k(x)$is the result given by the kth decision tree. $I(\cdot)$ is the indicator function.

In the process of constructing Random Forest, the ordering and the weights of features can be given. There are two ways to calculate the weight of each feature:

1. Mean Decrease Gini (MDG): Feature A's weight is measured by the decrease of Gini index caused by A. The more Gini index goes down, the greater the weight of A [29].
2. Mean Decrease Accuracy (MDA): Add noises to a certain feature in OOB data [1]. The greater the OOB error goes down, the more important this feature is.

After getting the feature weights, RF has its own feature selection algorithm: Random Forest-based Wrapper Feature Selection (RFWFS). The details of this algorithm are in Algorithm 2. Through the combination of MDG or MDA and RFWFS, reliable feature subset can be screened out.

Algorithm 2 RFWFS algorithm.

```
1: Rank the resulting feature weights from large to small;
2: Delete one feature with low weight in order at a time.
   Each deletion can get a feature subset;
3: Using each feature subset to build RF. The feature
   subset with highest accuracy is the optimal feature
   subset.
```

3 RF-efficient-ReliefF algorithm

Although ReliefF algorithm is quick and effective, there are still many deficiencies:

1. The resulting feature weights do not help remove redundant features.
2. Sampling times m will affect the final weights and ordering.
3. Although random sampling will reduce the time cost, the weights and ordering will change with the sampling results.
4. In the case of repeated sampling, repeated samples do not have any significance for the update of feature weights, but reduce the effectiveness and reliability of the results.
5. The point which is the most important in my view is that the data set has noisy data and bad data upon most occasions, which seriously affects the results.

In response to the above problems, this paper proposes a reliable and efficient algorithm: RF-efficient-ReliefF algorithm.

RF has a high accuracy, especially for the data far from the classification boundary. And RF has a fast operating speed. Based on these characteristics, first of all, we can establish a RF model on the original data set. For each sample, there are some decision trees which did not use it during the build process. So, we can give each sample the proportion of votes for each class according to the vote results of these trees. Based on the results of each sample, we first calculate the average proportions of votes for each class, and then filtered out those data whose proportion of votes for its real class is higher than the average proportion of votes for this class. ReliefF algorithm is used on the data subset which can be seen as the high quality data collection. The difference from the original algorithm is that we use all samples in the data subset instead of sampling. In addition, we adopted the same treatment as [26] used to assign k different nearest neighbors the weights, which indicates that k nearest instances have weights exponentially decreasing with increasing rank. The original data set is S. Suppose the classes for the data set are $c = 1, 2, \ldots, length(C)$ where $length(C)$ is the size of the class variable C. n is the number of features. The details of the algorithm are in Algorithm 3.

RF-efficient-ReliefF has these advantages:

1. The selected data subset does not include noisy data and bad data. It can be considered that the samples in it are more beneficial to distinguish the class variable. The feature weights derived from them are more effective.
2. The selected good data subset has roughly half of the original data set, which can shorten the time cost compared to using all the data.
3. No sampling is required, which eliminates the confusion caused by the determination of sampling times, and the results will not change with it.

In addition, we propose two alternatives:

- RF-e-R1: Also use the new data subset. For each sample d in the data subset S', let the normalized voting proportion of d for its true class be its weight. And during the implementation of the ReliefF algorithm, feature weights are multiplied by sample weights in each iteration. Still use all data subsets without sampling.
- RF-e-R2: Using original samples for feature weight calculation. Let each data's voting proportion for its true class be its weight. And during the implementation of the ReliefF algorithm, feature weights are multiplied by sample weights in each iteration. Still use all data subsets without sampling.

In our research, we found that the effects of these three algorithms are similar, but it is clear that the RF-efficient-ReliefF algorithm is more convenient. Therefore, it is chosen for further research in the subsequent sections.

4 Improved SFS algorithm based on Symmetrical Complementary Coefficient

In this section, firstly, we analyze some defects of the SFS algorithm we have discovered. Then we give the definition of Symmetrical Complementary Coefficient and its calculation formulas. And finally, the SFS algorithm is improved based on the Symmetrical Complementary Coefficient. This section gives two examples to help explain.

Algorithm 3 RF-efficient-ReliefF algorithm.

```
Input the data set. The weights of all features $W(\cdot)$ are initialized to 0. And data subset is $S'$, $|S'| = 0$.
Establish a RF model on $S$;
for $c$ from 1 to $length(C)$ do
    Find out the data subset $S_c$ whose samples' classes are $c$. Calculate the average voting proportion of $S_c$ for the class $c$, namely $mean(S_c, c)$;
    for each sample $d$ in $S_c$ do
        The voting proportion of $d$ for the class $c$ is $vote(d, c)$;
        if $vote(d, c) \geq mean(S_c, c)$ then
            Add $d$ to $S'$;
        end if
    end for
end for
for each sample $R$ in $S'$ do
    In $S'$, find out $R$'s $k$ nearest neighbor samples $NH_j(j = 1, 2, \ldots, k)$ which have the same class with $R$ and find out $R$'s $k$ nearest neighbor samples $NM_j(c)(j = 1, 2, \ldots, k)$ whose class are $c$ and $c$ is different from the class of $R$;
    for Feature $A$ from 1 to $n$ do
        Use the following formula to calculate $A$'s weight;
            $W(A) = W(A) - \sum_{i=1}^{k} diff(A, R, NH_j) \cdot d(R, NH_j)/|S'| + \sum_{c \neq class(R)} \left[ \frac{p(c)}{1 - p(class(R))} \cdot \sum_{j=1}^{k} diff(A, R, NM_j(c)) \cdot d(R, NM_j(c)) \right] /|S'|$
        where
            $diff(A, R, R_j) = \begin{cases} \frac{|R[A] - R_j[A]|}{\max(A) - \min(A)} & \text{If } A \text{ is continuous;} \\ 0 & \text{If } A \text{ is discrete and } R[A] = R_j[A]; \\ 1 & \text{If } A \text{ is discrete and } R[A] \neq R_j[A]; \end{cases}$
            $d(R, R_j) = \frac{d_1(R, R_j)}{\sum_{l=1}^{k} d_1(R, R_l)}$
            $d_1(R, R_j) = \exp\left\{ -\left( \frac{rank(R, R_j)}{\sigma} \right)^2 \right\}$
        $rank(R, R_j)$ is the rank of $R_j$ in the sequence of samples ordered by the distance from $R$;
        $\sigma$ is a defined parameter which controls the influence of the distance and its default value is 20 which is an empirical result;
        $|S'|$ is the number of data in $S'$;
        And class($R$) is the class of the sample $R$;
    end for
end for
Output features' weights.
```

Table 1 Feature weights and ordering. X_i represents the i-th feature of the original data set ($i = 1, 2, \dots, 20$)

Order	Feature	Weight
1	X1	0.57724
2	X3	0.18923
3	X2	0.15522
4	X5	0.09395
5	X12	0.09189
6	X7	0.08793
7	X4	0.07414
8	X6	0.07311
9	X9	0.07208
10	X15	0.06392
11	X13	0.06353
12	X14	0.06066
13	X19	0.03733
14	X11	0.03590
15	X8	0.03466
16	X17	0.02552
17	X16	0.02168
18	X18	0.01930
19	X20	0.00214
20	X10	0.00006

4.1 The defects of current SFS algorithm

The SFS algorithm starts from an empty feature subset and continuously adds features to it, as a consequence, the evaluation function maintains optimal status in every addition. So, you can only continue adding features to the future subset, but can't delete the features inside. The result obtained in this way is not reliable, and it is easy to fall into local optimum. But these defects can be alleviated by a pre-specified feature ordering, for example, the algorithm proposed in the previous section can be used to give the feature ordering. Then according to the feature ordering, implement the SFS algorithm.

Here another defect of SFS algorithm is presented, that is, feature interactions exist and SFS algorithm cannot recognize them. When an added feature cannot make the evaluation function optimal, it will be discarded. But in fact, it may interact with features that haven't been added yet, so deleting it is a very hasty behavior. Using the classic German credit data set gives the following example to illustrate the existence of this problem.

Example 1 German credit data set has 20 features, and two cases or states of the class variable. Table 1 shows the feature weights and ordering calculated by the RF-efficient-ReliefF algorithm. Features are sorted by weights from large to small.

We then use the accuracy of RF as an evaluation function to implement the SFS algorithm in order to find the optimal feature subset and its corresponding accuracy in this case. As a comparison, we selected the first eight features in Table 1 as a feature subset to calculate the accuracy. The two cases are shown in Table 2.

From Table 2, we can see that although the result of using the first 8 features is better, the SFS algorithm cannot search out all of them, but instead added many features ranked behind. Now we can see that the feature interactions cannot be recognized by SFS.

4.2 Symmetrical Complementary Coefficient

As mentioned in the previous introduction, currently, there is no widely accepted method. And, multi-way interaction is complex and it's not easy to search out all the interactive features. Here we present a new method for the quantification of feature interactions between small batch features. Its definition and calculation are as follows.

Firstly, let us introduce the Complementary Coefficient. The Complementary Coefficient was proposed by [30]. It has some good properties such as the contribution of feature interactions to the classification task has been taken into consideration and it's calculation is fast. However, at first it did not attract enough attention. It was not used to describe feature interactions but merely to search for important features. It is not symmetrical and the situation of multiple features has not been considered. And we made some changes to it.

The two features X_i, X_j of a given data set S are used to characterize the entire data set respectively. And use the classifier to classify the two data set which just have one feature and get two models M_i, M_j. Then get their corresponding misclassified sample sets D_i, D_j. Note that, you can use the reserved test set, just like cross-validation. In this paper, we take RF as the classifier and use the OOB data to find the D_i and D_j. Mark D_i's subset whose samples are classified correctly by M_j as D_{ij}. Mark D_j's subset whose samples are classified correctly by M_i as D_{ji}. Note that $D_{ij} \neq D_{ji}$.

Table 2 The resulting feature subsets and their accuracies

Case	Feature subset	Accuracy
1	X1, X3, X2, X7, X6, X15, X13, X17, X18, X10	0.7702333
2	X1, X3, X2, X5, X12, X7, X4, X6	0.7773333

Case 1 represents the result given by SFS algorithm; Case 2 represents the result given by using the first eight features. To avoid randomness, repeat 10 times each time the accuracy is calculated and then do the average. And the accuracy is calculated with OOB data which is a kind of test data

The Complementary Coefficient from X_j to X_i is $Com(i, j)$. The Complementary Coefficient from X_i to X_j is $Com(j, i)$.

$$Com(i, j) = \frac{|D_{ij}|}{|S|}$$
$$Com(j, i) = \frac{|D_{ji}|}{|S|}$$

where $| \cdot |$ is the number of the samples in the date set.

And in this paper, we put forward the Enhanced Complementary Coefficient. The Enhanced Complementary Coefficient from X_j to X_i is $ECom(i, j)$. The Enhanced Complementary Coefficient from X_i to X_j is $ECom(j, i)$.

$$ECom(i, j) = \frac{|D_{ij}|}{|D_i|} \tag{1}$$
$$ECom(j, i) = \frac{|D_{ji}|}{|D_j|} \tag{2}$$

The Complementary Coefficient and Enhanced Complementary Coefficient are different. The denominator changed from $|S|$ to $|D_i|$ and $|D_j|$. Although the contribution of feature interactions to the whole classification task has been taken into consideration in the case of using $|S|$ as the denominator, it's unfair. Supposing X_i is a very good feature, X_j is a very bad feature, and X_k is another common feature. Apparently, we have that $|D_i| \leq |D_j|$. So, under normal circumstances we have $|D_{ik}|/|S| \leq |D_{jk}|/|S|$. Absolutely it's unfair.

Let's suppose that

$$|S| = 1000,\ |D_i| = 10,\ |D_j| = 100,\ |D_{ik}| = 9,\ |D_{jk}| = 9,$$
$$\text{so} \quad |D_{ik}|/|S| = |D_{jk}|/|S| = 0.009,$$
$$\text{but} \quad |D_{ik}|/|D_i| = 0.9, \quad |D_{jk}|/|D_j| = 0.09$$

Obviously, the latter reflects more real and reliable feature interactions.

We are the first to use the Enhanced Complementary Coefficient to quantify feature interactions, and we have slightly expanded it to obtain the Symmetrical Complementary Coefficient. Firstly, let's consider the situation of two features. The Symmetrical Complementary Coefficient between X_i and X_j is $SCom(i, j)$.

$$\begin{aligned} SCom(i, j) = SCom(j, i) &= \frac{ECom(i, j) + ECom(j, i)}{2} \\ &= \frac{1}{2} \cdot \left(\frac{|D_{ij}|}{|D_i|} + \frac{|D_{ji}|}{|D_j|} \right) \end{aligned} \tag{3}$$

Symmetrical Complementary Coefficient can quantify the strength of the interaction between the two features. It has the following properties.

- $SCom(i, j) \in [0, 1]$
- $SCom(i, j) = 0$ indicates that there is no interaction between the two features.
- $SCom(i, j) > 0$ indicates that there is a positive interactive relationship between two features, that is, a synergistic relationship.
- The bigger the $SCom(i, j)$, the stronger the interaction between the two features.

The situations of one-to-many, many-to-many can be given similarly. Let U, V be the collections of feature subscripts. They may contain many features or only one features. Similarly there are D_{UV}, D_{VU}. So, the feature interaction between two sets of features can be obtained.

$$\begin{aligned} SCom(U, V) &= SCom(V, U) \\ &= \frac{ECom(U, V) + ECom(V, U)}{2} \\ &= \frac{1}{2} \cdot \left(\frac{|D_{UV}|}{|D_U|} + \frac{|D_{VU}|}{|D_V|} \right) \end{aligned}$$

However, it will take more time to do that. Computation of Symmetrical Complementary Coefficient between two features only need to use each feature to complete the classification task. So it won't take much time. In this paper, we just care about the situation of two features.

Let's observe the actual effect of the Symmetrical Complementary Coefficient by the following example which still use the German Credit Data set, and compare the Symmetrical Complementary Coefficient with the Interaction Information which is a classic quantification method of the feature interaction relationship proposed by [14].

Interaction Information between feature X_i and X_j is $I(X_i; X_j; C)$ whose formula is as follows. C is the class variable.

$$I(X_i; X_j; C) = I(X_i; X_j|C) - I(Xi; Xj)$$

where $I(X_i; X_j|C)$ is the conditional mutual information between X_i and X_j under C. $I(X_i; X_j)$ is the mutual information between X_i and X_j. $I(X_i; X_j; C) = 0$ indicates that there is no interaction between the two features. $I(X_i; X_j; C) > 0$ indicates that there is a positive interactive relationship between two features. $I(X_i; X_j; C) < 0$ indicates that there is a negative interactive relationship which is also called the redundancy between two features.

Example 2 Feature ordering has been given by RF-efficient-ReliefF algorithm. Compute the Symmetrical Complementary Coefficient and the Interaction Information between the kth feature and the $k + 1$st feature. The results of $SCom(i, j)$ and $I(X_i; X_j; C)$ are shown in Table 3.

It can be found from the table that under the feature ordering obtained by RF-efficient-ReliefF algorithm, the Interaction Information between each feature and its next feature are all negative and close to 0. It indicates that

Table 3 Results of $SCom(i, j)$ and $I(X_i; X_j; C)$

Feature	$SCom(i, j)$	$I(X_i; X_j; C)$
X1	0.15194	−0.00086
X3	0.20682	−0.01580
X2	0.73519	−0.03259
X5	0.71500	−0.01407
X12	0	−0.01797
X7	0	−0.02878
X4	0	−0.02706
X6	0	−0.01285
X9	0	−0.00545
X15	0.03777	−0.01093
X13	0.03932	−0.01476
X14	0	−0.00137
X19	0	−0.00330
X11	0	−0.01368
X8	0	−0.01239
X17	0	−0.00500
X16	0	−0.00201
X18	0	−0.00156
X20	0	−0.00309
X10	0	0

Each value represents the feature interaction between the corresponding feature and its next feature in the ordering. Since the last feature does not have the next one, its value is 0

the interaction relations between features quantified by the Interaction Information are all redundancy. Or there are no interactions at all. Whereas, the results given by Symmetrical Complementary Coefficient indicates that there are feature interactions among six pairs of features, and four of them are obvious.

As can be seen in Table 3, the Symmetrical Complementary Coefficient between $X5$ and $X12$ is 0.71500. From Table 2 in Example 1, the original SFS cannot search out the $X5$ and $X12$. It is precisely because SFS cannot recognize the feature interaction between $X5$ and $X12$ by adding features one by one. It happened that $X5$ and $X12$ cannot improve the evaluation function respectively. As a consequence, $X5$ and $X12$ have not been added to the optimal feature subset.

The results of Example 2 show that the Symmetrical Complementary Coefficient proposed in this paper is effective. Moreover, by comparison, It is more efficient than Interaction Information, at least in the German credit data set.

Note that the higher the Symmetrical Complementarity Coefficient between the two features, does not represent that the two features are more powerful for classification, but only shows that the combination of these two features is more powerful than they are used for classification respectively. If one or two features are to be judged as good correctly, besides observing the Symmetrical Complementary Coefficient between them, we must also look at their performance in the classifier, because feature interaction is only an auxiliary means and it should be combined with features' performance in the classifier which is absolutely accurate and effective. If you choose to save the time cost of feature selection, some classifier-independent indicators can also be used.

What needs to be pointed out is that feature interaction and feature redundancy are two opposite things. Therefore, the redundant relationship between two features can be ignored when the Symmetrical Complementary Coefficient between them is large, and even if there is a strong redundancy relationship between them, these two features also deserve to be preserved. In next part, we will propose an improved SFS algorithm based on the Symmetrical Complementary Coefficient, which is called SCom-SFS.

4.3 SCom-SFS algorithm

For the reasons that the multi-way interaction is complex and it is hard to search out all interactive features, we choose to search for feature interactions in sequence. This idea is very similar to the principle of SFS algorithm. Moreover, the process of discovering feature interactions can speed up the features' search process. So we gathered them together without strain. The new algorithm is called SCom-SFS algorithm which can screen out the beneficial features at a faster speed while identifying and quantifying feature interactions.

Its main idea is that under a given feature ordering, firstly for each feature X_i which is the ith feature in the ordering, calculate the Symmetrical Complementary Coefficient between it and its next feature in the ordering and let this value correspond to X_i like what we did in Example 2. And if their symmetrical complementary coefficient is greater than or equal to the threshold β, the two features are seemed as partner of each other and then they are going to be packaged. In the process of implementing SFS algorithm, these two features will advance and retreat together. Then if the symmetrical complementary coefficient between the 2nd and the 3rd features is greater than or equal to β, the 1st, the 2nd and the 3rd features are seemed as partner of each other and then they are packaged. This process will continue similarly until the symmetrical complementary coefficient between the kth and the $k+1$st features is less than β. So the 1st, 2nd, 3rd ,. . ., kth features will be packaged and they will advance and retreat together. Next consider the $k+1st$ and $k+2$, etc. The defect that the original SFS algorithm cannot recognize feature interactions will be solved in this way. And the discovered feature interactions will speed up the SFS algorithm. The details of the SCom-SFS algorithm are in Algorithm 4.

Algorithm 4 SCom-SFS algorithm.

```
$\{X_i\}_{i=1}^n$ are the features. And the weights and the ordering of the features are given. $X_{(1)}$ is the best feature, and $X_{(n)}$ is the worst feature;
for each feature $X_{(i)} (i = 1, 2, \ldots, n)$ do
    $X_{(i)}$ is the $i$th feature in the ordering. $X_{(i)}$ is used to characterize the entire data set. And use the classifier to classify the dataset which just have only one feature $X_{(i)}$ and get the model $M_i$. Then get it's corresponding misclassified sample set $D_i$;
end for
for $i$ from 1 to $n - 1$ do
    Mark $D_i$'s subset whose samples are classified correctly by $M_{i+1}$ as $D_{i(i+1)}$. Mark $D_{i+1}$'s subset whose samples are classified correctly by $M_i$ as $D_{(i+1)i}$. The Enhanced Complementary Coefficient from $X_{(i+1)}$ to $X_{(i)}$ is $ECom(i, i + 1)$. The Enhanced Complementary Coefficient from $X_{(i)}$ to $X_{(i+1)}$ is $ECom(i + 1, i)$. They are calculated using (1) and (2);
    Using (3) to calculate the Symmetrical Complementary Coefficient $SCom(i, i+1)$ between $X_{(i)}$ and $X_{(i+1)}$;
    if $SCom(i, i + 1) \geq \beta$ then
        if $X_{(i)}$ is already in a package then
            Add $X_{(i+1)}$ to this package;
        else
            $X_{(i)}$ and $X_{(i+1)}$ are packaged;
        end if
    end if
end for
$P_1, P_2, \ldots, P_M$ are the resulting packages. $P_i (i = 1, 2, \ldots, M)$ may contains many features or only one feature;
$F$ is the feature subset to be output. The number of features in $F$ is 0;
$\{E_j\}_{j=1}^M$ are the results of the evaluation function which can be some classifier-dependent metrics such as accuracy or some other classifier-independent metrics. Initialize them to 0;
for $j$ from 1 to $M$ do
    $F'$ is a temporary feature subset. Assign $F$ to $F'$;
    Add $P_j$ to $F'$;
    According to the evaluation function, get $F'$'s result $E_j$;
    if $j > 1$ then
        if $E_j > E_{j-1}$ then
            Assign $F'$ to $F$;
        else
            Assign $E_{j-1}$ to $E_j$;
        end if
    else
        Assign $F'$ to $F$;
    end if
end for
Output the feature subset $F$.
```

5 Combination of RF-efficient-ReliefF algorithm and SCom-SFS algorithm

Firstly, we will introduce this new algorithm. And then explain how to determine the threshold β and the time complexity of this algorithm.

5.1 RRSS algorithm

From the former two sections, the RF-efficient-ReliefF algorithm can solve some defects of the original ReliefF algorithm, and obtain more reliable feature weights and ordering. Whereas it determines neither how many features should be deleted nor what features should be deleted. Simultaneously, under a given feature weights and ordering, the SCom-SFS algorithm can recognize the feature interactions and search out the optimal feature subset. After considering the above, we combined these two algorithms to get a new and complete feature selection algorithm. We named it RRSS Algorithm. In this algorithm, RF-efficient-ReliefF can provide SCom-SFS a more reliable search sequence so that it is not easy to fall into a local optimum, and SCom-SFS can be seen as a correction to the feature ordering provided by RF-efficient-ReliefF and get the optimal feature subset. This algorithm's flow chart is shown in Fig. 1.

5.2 Threshold's determination and time complexity of the algorithm

N is the number of samples in the data set, and n is the number of features in the data set. The time complexity of establishing a Random Forest is $O(knNlog_2N)$ where k is the number of decision trees. The time complexity of the original ReliefF algorithm is $O(Nnm)$ where m is the number of random samples. However, in the RF-efficient-ReliefF algorithm, we use probably half of the original data set instead of sampling. So, the time complexity of RF-efficient-ReliefF algorithm is $O(knNlog_2N + \frac{N^2}{4}n)$.

In the process of calculating the Symmetrical Complementary Coefficient, it is necessary to establish Random Forest for each feature. At the same time, their corresponding misclassified samples' sets need to be tested on one or two other RF models. Given that it costs very little time to run an established RF model and the number of misclassified samples is often not too much, the time complexity of this part can be ignored. So, the time complexity of calculating the Symmetrical Complementary Coefficient and packaging features is $O(n \cdot kNlog_2N + (n - 1)) = O(knNlog_2N)$. It is the same time complexity as building a Random Forest.

The time complexity of searching the optimal feature subset needs to take the number of packages and the specific evaluation function into account. In this paper, we select

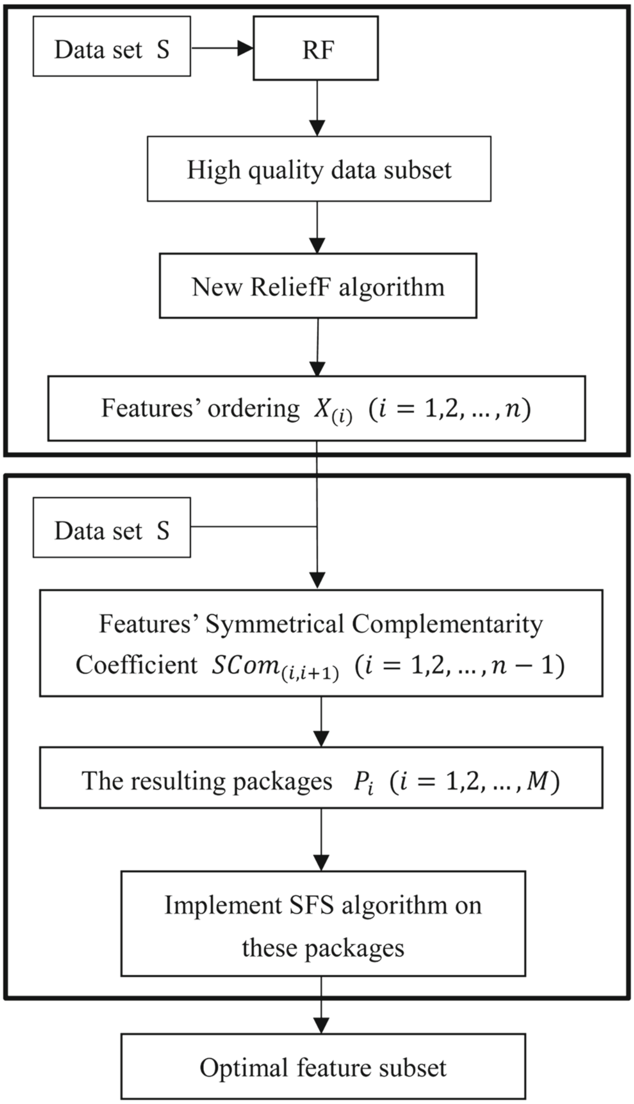
Data set S
RF
High quality data subset
New ReliefF algorithm
Features' ordering $X_{(i)}$ $(i = 1,2,\ldots,n)$
Data set S
Features' Symmetrical Complementarity Coefficient $SCom_{(i,i+1)}$ $(i = 1,2,\ldots,n-1)$
The resulting packages P_i $(i = 1,2,\ldots,M)$
Implement SFS algorithm on these packages
Optimal feature subset

Fig. 1 The RRSS algorithm

the accuracy of Random Forest as the evaluation function in order to make the resulting feature subsets good enough. Suppose that the number of features in the feature subset after each addition is $\{n_i\}_{i=1}^{M}$, where $n_1 \leq n_2 \leq \ldots \leq n_M \leq n$. So, the time complexity of searching the optimal feature subset is $O(n_1 kNlog_2 N + n_2 kNlog_2 N + \ldots + n_M kNlog_2 N)$.

It has a lot to do with M. However, M depends on threshold β. Consider two extremes. When β is small enough, $M = 1$ which means all features were packed together. Although the time complexity is greatly reduced, this makes no sense for feature selection. When β is large enough, $M = n$ which means all features were not packed with other features. This will not only lead to a high time complexity, but also will also fail to recognize feature interactions. Thus, threshold β's determination is very important. A good threshold should screen out the important feature interactions which is quantified by Symmetrical Complementary Coefficient. Here comes the question. How large is the Symmetrical Complementary Coefficient worth paying attention to? At first, we considered giving a fixed threshold. Whereas, it is difficult to determine the threshold, because the threshold depends on the size of the data set, the specifics of features, and the problems studied. Solution to this problem is what we will study further in the future. In this paper, we will give a method which is actually a piecewise function of determining the threshold that can get good results.

Because using the results of the classification will lead to small deviations, firstly let's define the Symmetrical Complementary Coefficient to be *valid* if it is greater than 0.05. The identification of feature interactions should be treated with caution because it is the opposite of removing features. Taking the mean of the *valid* symmetrical complementary coefficients as a threshold is a good choice. At the same time, we note that 0.5 is an important demarcation point. It distinguishes the overall levels of symmetrical complementary coefficients in different data sets. Thresholds need to be determined more carefully when symmetrical complementary coefficients are generally large. We then subdivide the value of the mean further. The threshold β can be given by the following equation.

$$\beta = \begin{cases} mean(SCom_{valid}) & \text{If } mean(SCom_{valid}) \in [0, 0.4) \\ 0.5 & \text{If } mean(SCom_{valid}) \in [0.4, 0.5) \\ mean(SCom_{valid}) & \text{If } mean(SCom_{valid}) \in [0.5, 0.8) \\ 1 & \text{If } mean(SCom_{valid}) \in [0.8, 1] \end{cases} \tag{4}$$

where $mean(SCom_{valid})$ is the mean of the *valid* symmetrical complementary coefficients. This piecewise function is just an empirical result which can make the results of this paper's experimental data sets generally good. For a specific data set, perhaps some search strategies yield better results, but they will bring in more time cost.

6 Experiment

6.1 Data set

In this section, 10 real-world data sets from the UCI repository [6] will be used. These data sets cover a wide range of situations, i.e., different number of instances, features, classes, and proportions of discrete features and

Table 4 Date set description

Dataset	N	n	Con	Dis	C
Breast-cancer	277	9	9	0	2
Germen-credit	1000	20	7	13	2
Chess-kr-vs-kp	3196	36	36	0	2
Soybean-small	47	35	0	35	4
Spambase	4601	57	0	57	2
Statlog-vehicle	846	18	0	18	4
Student	1044	32	16	16	2
Tic-tac-toe	958	9	9	0	2
Wine	178	13	0	13	3
Zoo	101	16	16	0	7

Column 'N' is the number of instances in these data sets, column 'n' is the number of features, column 'Con' is the number of continuous feature variables, column 'Dis' is the number of discrete feature variables, column 'C' is the number of cases or states of the class variable. For the student data set [3], we put two different data together and take the first column as the class variable

continuous features, which can represent a large proportion of real-world applications. Their features are normalized through Min-Max scaling before the experiment. And their details are shown in Table 4.

6.2 Evaluation metric

The number of features in the resulting feature subset, accuracy, Kappa coefficient and mean-square error (MSE) were selected as the evaluation metrics.

The Kappa coefficient is calculated with the confusion matrix. The overall accuracy can only reflect the case of samples that are correctly classified in the diagonal direction. Whereas, Kappa coefficient also considers the non-recognized and misclassified samples outside the diagonal direction. Its value usually be between 0 and 1 and the larger, the better.

Table 5 gives a typical resulting confusion matrix for a problem with C cases or status of class variable, where $n_{i,j}$ denotes the number of observations of a class with actual list status j that are classified as having listing status i $(i = 1, \cdots, C, j = 1, \cdots, C)$.

Table 5 A typical resulting confusion matrix

Predicted class	Actual class				
	l_1	l_2	$\cdots$	l_C	Total
l_1	$n_{1,1}$	$n_{1,2}$	$\cdots$	$n_{1,C}$	$\hat{n_1}$
l_2	$n_{2,1}$	$n_{2,2}$	$\cdots$	$n_{2,C}$	$\hat{n_2}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
l_C	$n_{C,1}$	$n_{C,2}$	$\cdots$	$n_{C,C}$	$\hat{n_C}$
Total	n_1	n_2	$\cdots$	n_C	N

$$\begin{aligned} accuracy &= \frac{\sum_{i=1}^{C} n_{i,i}}{N} \\ p_0 &= accuracy \\ p_e &= \frac{\sum_{i=1}^{C} n_i \cdot \hat{n_i}}{N^2} \\ kappa\ coefficient &= \frac{p_0 - p_e}{1 - p_e} \end{aligned}$$

MSE has some good properties, it can be expressed as the sum of the variance and the square of the deviation, and the smaller the MSE, the better. Since the MSE is for regression, we make some adjustments. The voting proportion p of the sample x_i for its real class is taken as the predicted value of the RF model. And 1 is taken as the true value of x_i, as follows.

$$\hat{f}(x_i) = p \qquad f(x_i) = 1$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{f}(x_i) - f(x_i))^2$$

6.3 Comparison algorithms

In Section 4, we illustrated the defects of original SFS algorithm and how the SCom-SFS algorithm solved them. Therefore, it is not necessary to compare SCom-SFS algorithm with SFS algorithm.

We perform a comparison of RRSS, RF-efficient-ReliefF, five classic feature selection algorithm and two latest feature selection algorithms considering feature interactions. The algorithms used for comparison in this experiment are as follows.

1. MDG [2], the embedded method, which is included in the process of building the RF.
2. ReliefF [19], one of the most famous feature selection algorithms.
3. mRMR [25], one of the most famous feature selection algorithms, which pays attention to the irrelevant features and redundant features at the same time.
4. RFE [27], the wrapper methods.
5. Boruta [20], the wrapper methods.
6. CFR [10], one of the latest feature selection algorithms.
7. FJMI [31], one of the latest feature selection algorithms, which takes high dimensional feature interactions into account.
8. RF-efficient-ReliefF, the improved version of ReliefF, proposed in this paper.

Table 6 The numbers of selected samples to implement the new ReliefF algorithm

Dataset	N	N'
Breast-cancer	277	151
Germen-credit	1000	559
Chess-kr-vs-kp	3196	2142
Soybean-small	47	29
Spambase	4601	3286
Statlog-vehicle	846	456
Student	1044	606
Tic-tac-toe	958	522
Wine	178	115
Zoo	101	74

Column 'N'' is the number of selected representative samples

9. RRSS, the combination of RF-efficient-ReliefF and the SCom-SFS proposed in this paper.

Among them, RF-efficient-ReliefF, ReliefF, MDG, mRMR, CFR, and FJMI need to artificially determine the number of features selected. So, we implement RFWFS to get their optimal feature subsets.

6.4 Experimental settings

These seven algorithms are tested on 10 data sets, using Random Forests as the classifier which is provided by R language. For the convenience of comparison, RF models use the default parameters where the number of trees is 500 and the number of randomly chosen attributes at each node is equal to the first integer less than $\log_2(\text{number of features}) + 1$. In order to avoid randomness, repeat the calculation of each evaluation metric for 30 times and take the average then get the results. And the results were calculated with OOB data. Breiman [1] has proved that using OOB data to estimate the results is unbiased. Compared with the cross-validation, the calculation of OOB estimation is simpler and faster, and its results are similar to cross-validation.

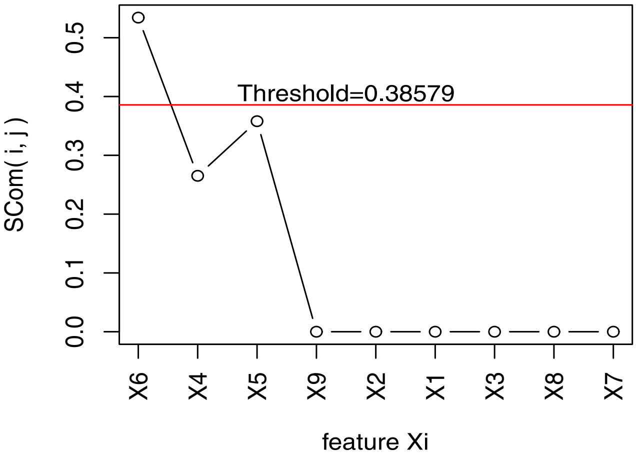

Fig. 2 Breast-cancer data set

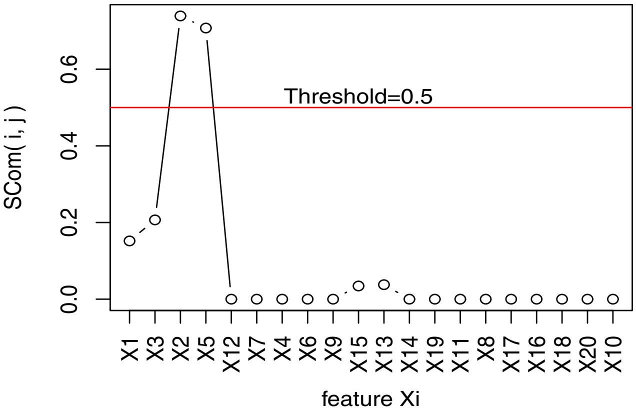

Fig. 3 Germen-credit data set

We use the 'CORElearn' package in the R language to implement ReliefF algorithm. The number of randomly selected samples is $m = 1000$. The number of selected nearest neighbor samples is $k = 5$.

We use the 'mRMRe' package in the R language to implement mRMR algorithm. All parameters are their default values.

We use the 'caret' package in the R language to implement RFE algorithm. $functions = rfFuncs$, $method = \text{“}cv\text{”}$, $number = 5$.

We use the 'Boruta' package in the R language to implement Boruta algorithm. $pValue = 0.01, mcAdj = TRUE, maxRuns = 100$.

In the process of implementing CFR and FJMI, continuous features are discretized by the Ameva algorithm [11]. The discretized features are only used for feature selection, and the classification process still uses these original features.

In the process of implementing algorithm RF-efficient-ReliefF algorithm, no sampling required, and the number of selected nearest neighbor samples is $k = 5$. Table 6 shows

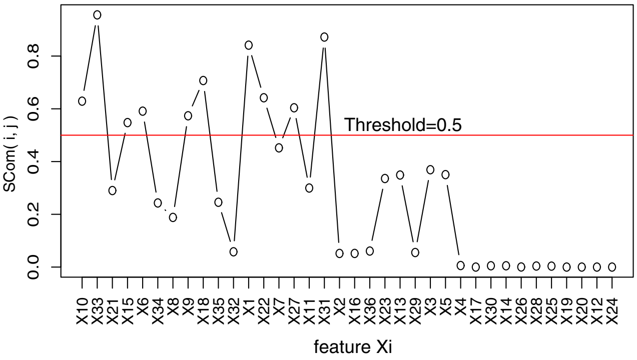

Fig. 4 Chess-kr-vs-kp data set

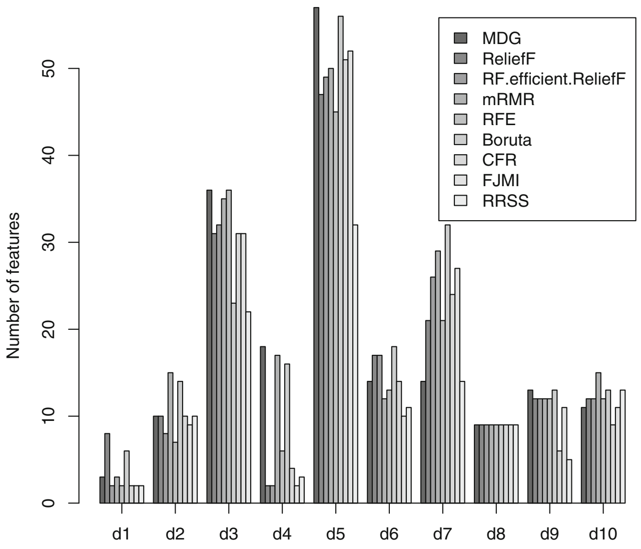

Fig. 5 Number of features obtained by these 9 algorithms

how many samples were selected to implement the new ReliefF algorithm. There exists randomness when building the RF classifier, so these numbers are derived from one RF.

In the process of implementing the RRSS algorithm, the Symmetrical Complementary Coefficients were calculated. Let $SCom(i, i+1)(i = 1, 2, \cdots, n-1)$ be the Symmetrical Complementary Coefficients between feature X_i and its next feature in the ordering like what we did in the Example 2. And Since the last feature does not have the next one, its default value is 0. Based on these coordinate points $\{(X_i, SCom(i, i+1))\}_{i=1}^{n-1}$, we can draw figures for each data set. Figures 2, 3 and 4 show the Symmetrical Complementary Coefficients of the first three data sets and their thresholds. The threshold is represented by a horizontal line. The figures of the remaining seven data sets, Figs. 6, 7, 8, 9, 10, 11 and 12, can be found in the Appendix.

In Fig. 2 the mean of the *valid* symmetrical complementary coefficients is 0.38579. So the threshold $\beta = 0.38579$ according to (4). And the feature X_6 and X_4 are packaged.

In Fig. 3 the mean of the *valid* symmetrical complementary coefficients is 0.45128. So the threshold $\beta = 0.5$ according to (4). And the feature X_2, X_5 and X_{12} are packaged.

In Fig. 4 the mean of the *valid* symmetrical complementary coefficients is 0.41456. So the threshold $\beta = 0.5$ according to (4). And the feature X_{10}, X_{33} and X_{21} are packaged. Feature X_{15}, X_6 and X_{34} are packaged. Feature X_9, X_{18} and X_{35} are packaged. Feature X_1, X_{22}, X_7 are packaged. Feature X_{27} and X_{11} are packaged. Feature X_{31} and X_2 are packaged.

7 Results and analyses

In the experiment we selected four evaluation metrics, i.e., number of features in the resulting feature subset, Accuracy, Kappa coefficient and MSE. Their results on the 10 data sets and the average of the results are shown in Tables 7, 8, 9, and 10, respectively. In these tables, the best algorithm for each data set is emphasized with bold fonts, the second best is marked with italic fonts.

RRSS achieved a lower number of features while maintaining the best ranking, especially on the 'Chess-kr-vs-kp', 'Spambase', and 'Wine' data set. The number of features obtained by these algorithms on ten data sets is shown in Fig. 5.

It seems unfair to average the results directly because the average is affected by larger values greatly. So we gave Friedman's ranks [9] of these results. For each algorithm, its rank in each data set is obtained first, and then average them. The best performing algorithm is assigned the rank of 1, the second best performing algorithm is assigned the rank of 2, etc. Table 11 shows the Friedman's ranks of the four

Table 7 The number of features in the resulting feature subsets

Dataset	MDG	ReliefF	RF-efficient-ReliefF	mRMR	RFE	Boruta	CFR	FJMI	RRSS
Breast-cancer	*3*	8	**2**	3	**2**	6	2	2	**2**
Germen-credit	10	10	*8*	15	**7**	14	10	9	10
Chess-kr-vs-kp	36	31	32	35	36	*23*	31	31	**22**
Soybean-small	18	**2**	**2**	17	6	16	4	**2**	*3*
Spambase	57	47	49	50	*45*	56	51	52	**32**
Statlog-vehicle	14	17	17	12	13	18	14	**10**	*11*
Student	**14**	*21*	26	29	*21*	32	24	27	**14**
Tic-tac-toe	**9**	**9**	**9**	**9**	**9**	**9**	**9**	**9**	**9**
Wine	13	*12*	*12*	*12*	*12*	13	6	11	**5**
Zoo	*11*	12	12	15	12	13	**9**	*11*	13
Average	18.5	16.9	16.9	19.7	16.3	20	*16*	16.4	**12.1**

Table 8 Accuracy results(%)

Dataset	MDG	ReliefF	RF-efficient-ReliefF	mRMR	RFE	Boruta	CFR	FJMI	RRSS
Breast-cancer	76.474	75.499	75.499	*77.593*	**77.617**	75.668	**77.617**	**77.617**	**77.617**
Germen-credit	77.430	76.413	*77.760*	75.693	76.653	76.757	77.383	77.223	**78.100**
Chess-kr-vs-kp	*98.689*	98.655	**98.700**	98.330	*98.689*	98.540	98.486	98.182	98.657
Soybean-small	*99.929*	**100.000**	**100.000**	**100.000**	98.683	99.504	99.787	**100.000**	**100.000**
Spambase	95.454	95.460	*95.487*	95.466	95.405	95.478	95.358	95.402	**95.490**
Statlog-vehicle	74.862	74.921	74.815	*75.737*	74.614	74.689	74.811	74.758	**75.800**
Student	88.646	87.372	87.414	88.560	*89.202*	87.369	87.037	87.333	**90.134**
Tic-tac-toe	**96.173**	**96.173**	**96.173**	**96.173**	**96.173**	**96.173**	**96.173**	**96.173**	**96.173**
Wine	98.071	98.057	98.057	98.071	98.034	*98.090*	97.715	97.959	**98.783**
Zoo	96.766	96.436	*96.766*	96.403	96.700	95.743	96.964	*97.558*	**97.558**
Average	*90.249*	89.899	90.067	90.067	90.177	89.801	90.133	90.207	**90.824**

Table 9 Kappa coefficient results(%)

Dataset	MDG	ReliefF	RF-efficient-ReliefF	mRMR	RFE	Boruta	CFR	FJMI	RRSS
Breast-cancer	31.609	30.214	30.214	*35.010*	**35.099**	30.970	**35.099**	**35.099**	**35.099**
Germen-credit	41.951	38.413	41.953	35.699	39.757	39.364	*41.974*	41.521	**43.197**
Chess-kr-vs-kp	97.510	*97.661*	**97.668**	96.652	97.510	97.073	96.964	96.355	97.307
Soybean-small	99.611	**100.000**	**100.000**	**100.000**	97.460	99.320	*99.710*	**100.000**	**100.000**
Spambase	90.404	90.444	*90.502*	90.462	90.326	90.482	90.235	90.322	**90.510**
Statlog-vehicle	66.381	66.641	66.364	*67.650*	66.151	66.250	66.414	66.311	**67.739**
Student	68.131	62.771	62.988	66.420	*68.768*	62.378	61.663	62.544	**71.990**
Tic-tac-toe	**91.302**	**91.302**	**91.302**	**91.302**	**91.302**	**91.302**	**91.302**	**91.302**	**91.302**
Wine	97.106	*97.124*	*97.124*	97.077	97.020	97.106	96.538	96.906	**98.154**
Zoo	95.655	95.158	95.655	95.250	95.644	94.377	*95.987*	**96.603**	**96.603**
Average	*77.966*	76.973	77.377	77.552	77.904	76.862	77.589	77.696	**79.190**

Table 10 MSE results(%)

Dataset	MDG	ReliefF	RF-efficient-ReliefF	mRMR	RFE	Boruta	CFR	FJMI	RRSS
Breast-cancer	20.220	20.970	20.970	*19.495*	**19.458**	20.320	**19.458**	**19.458**	**19.458**
Germen-credit	16.432	16.510	16.406	16.616	16.852	16.343	**16.022**	*16.298*	16.400
Chess-kr-vs-kp	*1.583*	1.780	1.680	2.083	**1.581**	1.894	1.892	2.029	1.884
Soybean-small	*0.921*	**0.440**	**0.440**	**0.440**	1.583	0.924	1.632	1.117	**0.440**
Spambase	3.779	3.755	*3.751*	3.766	3.781	3.770	3.780	3.766	**3.701**
Statlog-vehicle	17.220	**17.130**	**17.130**	17.935	17.221	*17.191*	17.215	18.569	18.381
Student	9.008	10.439	10.436	9.496	*8.581*	10.414	10.798	10.515	**7.968**
Tic-tac-toe	**5.461**	**5.461**	**5.461**	**5.461**	**5.461**	**5.461**	**5.461**	**5.461**	**5.461**
Wine	2.770	2.737	2.737	2.729	*2.725*	2.774	3.034	**2.637**	2.733
Zoo	*4.555*	4.683	*4.555*	6.157	4.723	5.104	5.295	5.368	**4.421**
Average	*8.195*	8.391	8.357	8.418	8.197	8.420	8.459	8.522	**8.085**

Table 11 Friedman's ranks of the four evaluation metrics of different algorithms

Evaluation metric	MDG	ReliefF	RF-efficient-ReliefF	mRMR	RFE	Boruta	CFR	FJMI	RRSS
Number	5.3	3.9	3.6	6	3.4	6.7	3.3	*2.9*	**2.3**
Accuracy	3.8	4.9	*3.4*	4.5	5.1	5.7	5.6	5	**1.3**
Kappa coefficient	4.3	4.4	*3.3*	4.7	5.1	6.1	5.1	4.8	**1.4**
MSE	4.4	4.1	*3.4*	5.1	4.4	5.2	5.5	4.9	**2.7**
Average	4.45	4.325	*3.425*	5.075	4.5	5.925	4.875	4.4	**1.925**

Table 12 p-values of the Nemenyi test on the pairs of algorithms which have significant differences

Algorithm	p-value
RRSS vs. MDG	0.00056
RRSS vs. ReliefF	0.00192
RRSS vs. mRMR	0.0000089
RRSS vs. RFE	0.00072
RRSS vs. Boruta	0.0000000052
RRSS vs. CFR	0.000062
RRSS vs. FJMI	0.0022
RF-efficient-ReliefF vs. Boruta	0.00582

evaluation metrics of the nine algorithms and then average them again. The best algorithm on these ten data sets is emphasized with bold fonts, the second best is marked with italic fonts.

In addition, all these 9 algorithms were compared to each other by Nemenyi test [22] which is a non-parametric test. All the tests were carried out with $\alpha = 0.05$ which is the level of significance. We put the ranks of the four evaluation metrics together to implement the tests. Nemenyi test rejects the hypotheses that the algorithms are equivalent if the corresponding $p\text{-}value \leq 0.05$. When there is a significant difference, the best algorithm is marked with bold fonts. Table 12 shows the p-values of the Nemenyi test on the pairs of algorithms which have significant differences.

Then we performed the Wilcoxon signed rank test [5] on RRSS algorithm and RF-efficient-ReliefF algorithm, and on RF-efficient-ReliefF algorithm and ReliefF algorithm. Table 13 shows the p-values. When there is a significant difference, the best algorithm is marked with bold fonts. We can find that the corresponding $p\text{-}values$ are ≤ 0.05. So we can reject the hypotheses that the two pairs of algorithms are equivalent and hold that RRSS algorithm is better than RF-efficient-ReliefF algorithm and RF-efficient-ReliefF algorithm is better than ReliefF algorithm.

From these results, several points can be obtained.

- The RRSS algorithm which is the combination of RF-efficient-ReliefF and SCom-SFS algorithm ranked first among the nine algorithms. Overall, the RRSS algorithm have much higher accuracy, Kappa coefficient and smaller MSE than other algorithms while possessing the smallest number of features in the resulting feature subsets. Although it slightly performed bad on the Chess-kr-vs-kp data set, it significantly reduced the number of features compared to other algorithms.
- The RF-efficient-ReliefF algorithm proposed in this paper is better than the original ReliefF algorithm. And the combination of it and RFWFS has a positive statistical significant difference with Boruta algorithm. The rest algorithms have no significant differences.
- The SCom-SFS algorithm proposed in this paper is very successful. The combination of it and RF-efficient-Relief is much better than the combination of RF-efficient-Relief and RFWFS algorithm. The number of features in the resulting subset selected by SCom-SFS algorithm is much less than that of other algorithms, because it can minimize the number of feature subset by adding features instead of deleting features. Moreover, SCom-SFS considers feature interactions. Based on the feature ordering given by RF-efficient-ReliefF algorithm, it can search out truly effective features, and it is difficult to fall into local optimum. At the same time, SCom-SFS can be seen as a correction to the results of RF-efficient-ReliefF algorithm.

Table 13 p-values of the Wilcoxon signed rank test

Algorithm	p-value
RRSS vs. RF-efficient-ReliefF	0.005772
RF-efficient-ReliefF vs. ReliefF	0.02411

8 Conclusion

After analyzing the shortcomings of ReliefF algorithm, this paper combines it with Random Forests and proposed the RF-efficient-ReliefF algorithm. It can screen out representative samples that are more useful for the identification of class labels, and the feature weights obtained are more reliable. Moreover, it does not require sampling.

This paper proposed a new method of quantifying feature interactions, namely, the Symmetrical Complementary Coefficient. And we illustrated the superiority of it by using an example.

We proposed the SCom-SFS which is based on the Symmetrical Complementary Coefficient. It can recognise the feature interactions and takes the strong feature interactions into account in the subset searching process through the given threshold. So as to search out the truly effective feature subset at a faster speed by packaging these features which have strong feature interactions.

We combined the RF-efficient-ReliefF algorithm with the SCom-SFS algorithm to obtain a complete feature selection algorithm, that is, the RRSS algorithm. It was tested on 10 data sets and compared with the other 8 algorithms including 5 classic and 2 latest feature selection algorithms, and found that the feature subsets obtained by it have the best performance while having the minimum number of features.

In the future, we will find a better method to determine the threshold more accurately, and combine the symmetrical complementary coefficients with different algorithms to make the feature interaction play a greater role.

Acknowledgements Thanks to the data sets provided by the UCI repository. And The breast cancer domain was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data. The Statlog-Vehicle data set was from the Turing Institute, Glasgow, Scotland. Also thanks to *R* language and the authors of different packages.

Appendix A: Symmetrical complementary coefficients and thresholds of the remaining seven data sets

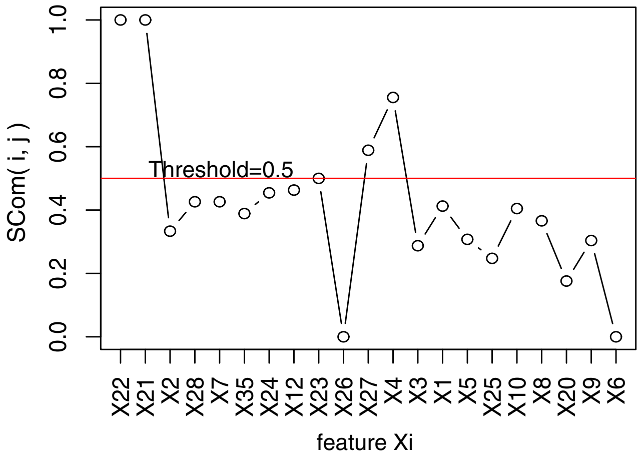

Fig. 6 Soybean-small data set

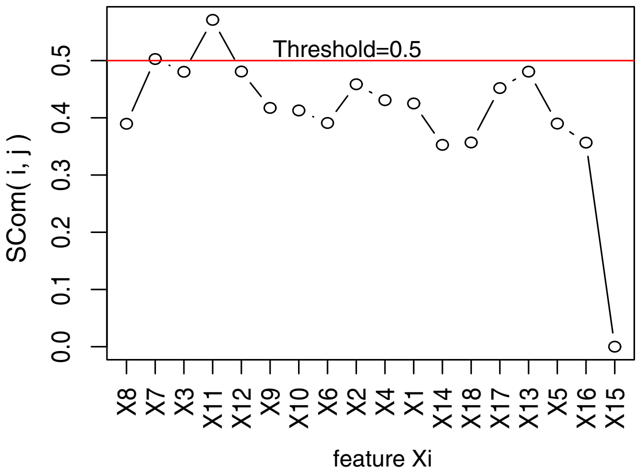

Fig. 8 Statlog-vehicle data set

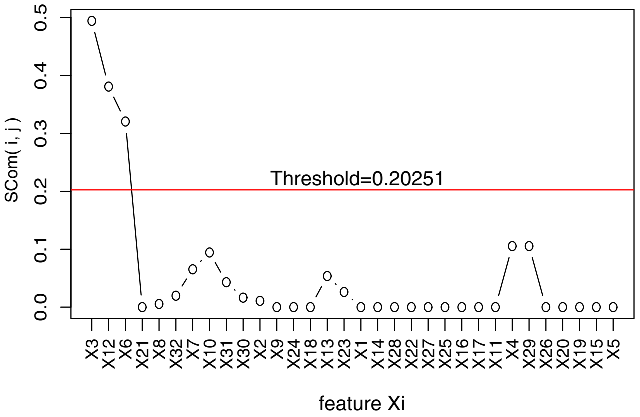

Fig. 9 Student data set

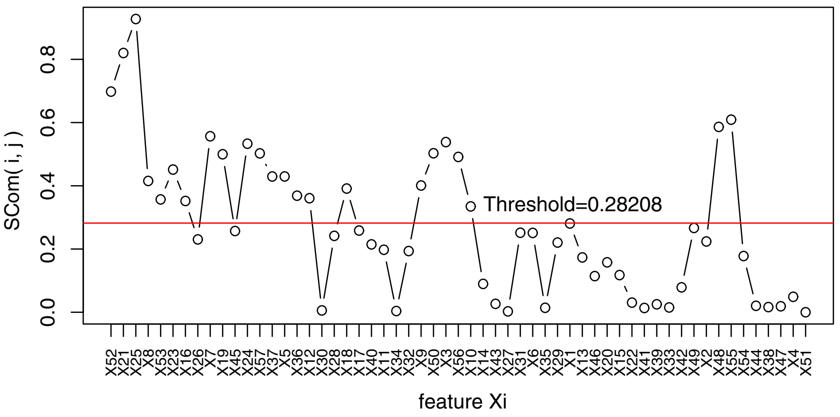

Fig. 7 Spambase data set

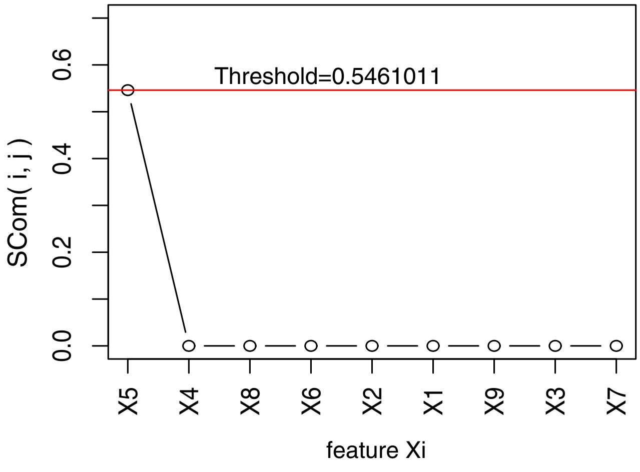

Fig. 10 Tic-tac-toe data set

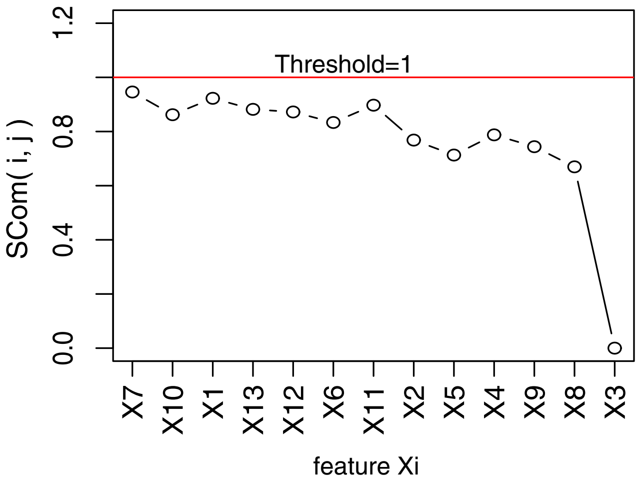

Fig. 11 Wine data set

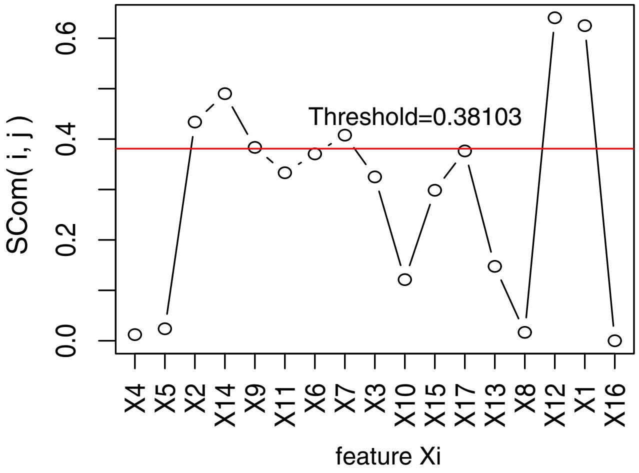

Fig. 12 Zoo data set

References

1. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
2. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
3. Cortez P, Silva AMG (2008) Using Data Mining to Predict Secondary School Student Performance. In: Brito A, Teixeira J (eds) Proceedings of 5th future business technology conference, pp 5–12
4. Dash M, Liu H (2003) Consistency-based search in feature selection. Artif Intell 151(1):155–176
5. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(1):1–30
6. Dheeru D, Karra Taniskidou E (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
7. Estevez PA, Tesmer M, Perez CA, Zurada JM (2009) Normalized mutual information feature selection. IEEE Trans Neural Netw 20(2):189–201
8. Fleuret F (2004) Fast binary feature selection with conditional mutual information. J Mach Learn Res 5(3):1531–1555
9. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701
10. Gao W, Hu L, Zhang P, He J (2018) Feature selection considering the composition of feature relevancy. Pattern Recognit Lett 112:70–74
11. Gonzalez-Abril L, Cuberos FJ, Velasco F, Ortega JA (2009) Ameva: an autonomous discretization algorithm. Expert Syst Appl 36(3):5327–5332
12. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(6):1157–1182
13. Hall MA (2000) Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the seventeenth international conference on machine learning, pp 359–366
14. Jakulin A, Bratko I (2003) Analyzing attribute dependencies. In: European conference on principles of data mining and knowledge discovery. Springer, pp 229–240
15. Jakulin A, Bratko I (2004) Testing the significance of attribute interactions. In: Proceedings of the 21st international conference on machine learning, pp 409–416
16. John GH, Kohavi R, Pfleger K (1994) Irrelevant features and the subset selection problem. In: Machine learning proceedings 1994. Elsevier, pp 121–129
17. Kira K, Rendell LA (1992) The feature selection problem: traditional methods and a new algorithm. In: Tenth national conference on artificial intelligence, pp 129–134
18. Koller D, Sahami M (1996) Toward optimal feature selection. In: Thirteenth international conference on international conference on machine learning, pp 284–292
19. Kononenko I (1994) Estimating attributes: analysis and extensions of relief. In: European conference on machine learning on machine learning, pp 171–182
20. Kursa MB, Jankowski A, Rudnicki WR (2010) Boruta—a system for feature selection. Fund Inform 101(4):271–285
21. Liu H, Setiono R (1996) A probabilistic approach to feature selection—a filter solution. In: International conference on machine learning, pp 319–327
22. Nemenyi P (1963) Distribution-eree multiple comparison. PhD thesis
23. Ng AY (2004) Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In: Proceedings of the twenty-first international conference on machine learning. ACM, p 78
24. Park H, Kwon HC (2008) Extended relief algorithms in instance-based feature filtering. In: International conference on advanced language processing and web information technology, pp 123–128
25. 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

26. Robnik-Šikonja M, Kononenko I (2003) Theoretical and empirical analysis of relieff and rrelieff. Mach Learn 53(1–2):23–69
27. Shieh MD, Yang CC (2008) Multiclass SVM-RFE for product form feature selection. Expert Syst Appl 35(1):531–541
28. Song L, Smola A, Gretton A, Bedo J, Borgwardt K (2012) Feature selection via dependence maximization. J Mach Learn Res 1(1):1393–1434
29. Strobl C, Boulesteix AL, Augustin T (2007) Unbiased split selection for classification trees based on the gini index. Comput Stat Data Anal 52(1):483–501
30. Su YX, Fu Y, Li X (2007) A feature selection method based on relieff evaluation and complementary coefficient. Electron Opt Control 14(3):12–15
31. Tang X, Dai Y, Xiang Y (2019) Feature selection based on feature interactions with application to text categorization. Expert Syst Appl 120:207–216
32. Tuv E, Borisov A, Runger G, Torkkola K (2009) Feature selection with ensembles, artificial variables, and redundancy elimination. J Mach Learn Res 10(3):1341–1366
33. Wang G, Song Q (2012) Selecting feature subset via constraint association rules. In: Pacific-Asia conference on advances in knowledge discovery and data mining, pp 304–321
34. Wang H, Lo SH, Zheng T, Hu I (2012) Interaction-based feature selection and classification for high-dimensional biological data. Bioinformatics 28(21):2834–2842
35. Yu L, Liu H (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Twentieth international conference on international conference on machine learning, pp 856–863
36. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5(12):1205–1224
37. Zeng Z, Zhang H, Zhang R, Yin C (2015) A novel feature selection method considering feature interaction. Pattern Recogn 48(8):2656–2666
38. Zhao Z, Liu H (2009) Searching for interacting features in subset selection. Intell Data Anal 13(2):207–228