Applied Intelligence (2021) 51:3899–3916
https://doi.org/10.1007/s10489-020-02028-0

Feature selection with multi-objective genetic algorithm based on a hybrid filter and the symmetrical complementary coefficient

Rui Zhang[1] · Zuoquan Zhang[1] · Di Wang[1] · Marui Du[1]

Accepted: 16 October 2020 / Published online: 23 November 2020

Abstract

With the expansion of data size and data dimension, feature selection attracts more and more attention. In this paper, we propose a novel feature selection algorithm, namely, Hybrid filter and Symmetrical Complementary Coefficient based Multi-Objective Genetic Algorithm feature selection (HSMOGA). HSMOGA contains a new hybrid filter, Symmetrical Complementary Coefficient which is a well-performed metric of feature interactions proposed recently, and a novel way to limit feature subset's size. A new Pareto-based ranking function is proposed when solving multi-objective problems. Besides, HSMOGA starts with a novel step called knowledge reserve, which precalculate the knowledge required for fitness function calculation and initial population generation. In this way, HSMOGA is classifier-independent in each generation, and its initial population generation makes full use of the knowledge of data set which makes solutions converge faster. Compared with other GA-based feature selection methods, HSMOGA has a much lower time complexity. According to experimental results, HSMOGA outperforms other nine state-of-art feature selection algorithms including five classic and four more recent algorithms in terms of kappa coefficient, accuracy, and G-mean for the data sets tested.

Keywords Feature selection · Feature interaction · Hybrid filter · Symmetrical complementary coefficient · Multi-objective genetic algorithm

1 Introduction

Feature selection plays an important role in many aspects of machine learning such as multivariate classification (including the binary classification) where each instance has just one class, and the multi-label classification [18, 19] where there is more than one class variable or each instance can belong to multiple classes at the same time, and sometimes there are dependencies between these classes. These classification tasks all need to learn the input data, and the features are used to characterize the data from different perspectives. Whereas, for a data set, sometimes many features of it are not helpful for learning and mining tasks, or even harmful [20]. Thus, correct features are essential. Nowadays, there is a growing requirement of feature selection, as data sets are getting bigger and wider.

1.1 Literature review

Features which need to be coped with can be divided into three types, i.e., irrelevant features, redundant features and interactive features. Irrelevant feature refers to the one which does not help with learning and mining tasks, as it has no connection or weak connection with the target [26]. Redundant feature refers to the one whose information or knowledge can be provided by other features [48]. Irrelevant features and redundant features need to be removed. Whereas, the interactive features need to be retained for further observation. Feature interaction refers to that multiple features contribute more than the sum of these individual features' contribution, and these features

This work is supported by the National Natural Science Foundation of China under Grant 51727813.

✉ Zuoquan Zhang
zuoquanzhang@163.com

Rui Zhang
15600825195@163.com

Di Wang
15118424@bjtu.edu.cn

Marui Du
17118446@bjtu.edu.cn

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

are interactive features [25]. A typical example of it is the exclusive OR operation, i.e., $Y = X_1 \oplus X_2$, and Y, X_1, X_2 are Boolean. A single feature X_1 or X_2 is irrelevant to Y, but when they are combined together, they have a strong correlation with Y. Therefore, removing any one of them will lead to a serious consequence. More and more scholars are beginning to pay attention to feature interactions [41, 43, 44, 49]. Recently, [50] proposed a new well-performed metric of feature interaction named Symmetrical Complementary Coefficient (SCom) which measures feature interaction based on classifiers at high speed. Whereas, they have to determine SCom's threshold artificially, which is troublesome.

Feature selection can be divided into four types. Firstly, filter method such as ReliefF [28] which determines the weight of features by calculating the distance between randomly selected instances and their nearest neighbors of the same and different classes, mRMR [34] which quantify candidate feature's relevancy and the redundancy between each candidate feature and the selected features based on the information theory, and CRF [14] which can find the composition of feature relevancy and maximize feature relevancy while minimizing feature redundancy based on the information theory. Secondly, wrapper method such as RFE [39] which builds the models iteratively, and then selects the best (or worst) features based on the fitness function, and Boruta [29] which iteratively creates new feature set composed of shadow features and original features, builds model, and then determines important or unimportant features. Thirdly, embedded method which evaluates features during the model building process such as L1, L2 regularization [33] and Random Forest [3]. Finally, hybrid method such as BBHFS algorithm [7] which incorporates some advantages of filter and wrapper methods through a natural stopping criterion and the concept of boosting, [22] combines filter and wrapper methods which takes advantage of both the filters and wrappers, [45] combines two different optimal filters and a component co-occurrence based feature relevance measure is proposed, [46] combines several filters and calculate the multi-filter wights and the multi-feature weights, then using a new Q-range based feature relevance calculation method and the greedy searching policy to get the final feature subset. Among them, the hybrid method is getting more and more popular, as it combines several different feature selection algorithms which belong to different kinds of methods or the same kind of method, and normally performs better.

Besides, these traditional methods are often combined with the evolutionary algorithms which showed promising results, [23]. Combined Genetic Algorithm (GA) with filters and wrappers, and both the parsimonious feature selection and excellent classification accuracy of the new algorithm are presented. [51] proposed PSO-based multi-objective feature selection algorithm which consists of a probability-based encoding technology and an effective hybrid operator can evolve a set of non-dominated solutions automatically. [15], combined six well-known filters and an enhanced GA in a wrapper approach. Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) algorithm [4, 12, 31, 37] could brilliantly complete the problem space exploration through the lens of chosen features, and generate a set of solutions which is both diverse and high-performing. [52], proposed two-archive multi-objective artificial bee colony algorithm (TMABC-FS) which contains two new operators, i.e., convergence-guiding search for employed bees and diversity-guiding search for onlooker bees, and two archives, i.e., the leader archive and the external archive. [16], proposed a wrapper-filter combination of Ant Colony Optimization (ACO) which can reduce computational complexity and performs well. [53], improved the binary differential evolution algorithm based on a new binary mutation operator, a new one-bit purifying search operator (OPS), and a novel efficient non-dominated sorting operator to solve the multi-objective feature selection problems. [40], proposed the variable-size cooperative coevolutionary particle swarm optimization algorithm (VS-CCPSO) which is competitive in solving the problem of high-dimensional feature selection problems. [24], proposed a novel two-step method called Information Gain Directed Feature Selection (IGDFS) algorithm, in which Information Gain reduces the feature space of possible configuration, and GA is combined with a wrapper to get the optimal solution. [21], proposed a tri-objective GA-based feature selection algorithm which considers the number of features, mutual information, and accuracy at the same time. [6], proposed a bi-objective GA-based feature selection algorithm whose two fitness functions are based on rough set theory and Kullback-Leibler divergence.

Although many scholars have combined hybrid methods with evolutionary algorithms, there are still some defects. Firstly, most of them [6, 15, 16, 21, 23, 24] use wrappers which means their methods are classifier-dependent in each generation. As evolutionary algorithms have many generations and there are many individuals in each generation, the time cost has increased significantly. Secondly, when generating the initial population, a large proportion of studies use random [15, 16, 21, 23, 24] or some pseudo-random pattern generator [6], which does not take full advantage of knowledge or information of data set used. Finally, at present, no research has combined feature interaction with GA-based feature selection algorithm.

1.2 Contribution

In this paper, a novel feature selection algorithm is proposed, which can solve these problems and performs better,

i.e., **H**ybrid filter and **S**com based **M**ulti-**O**bjective **G**enetic **A**lgorithm feature selection (**HSMOGA**). Main contributions of this paper are as follows.

- HSMOGA contains three new fitness functions including a hybrid filter which performs more balanced than these single filters, a new usage of the feature interaction metric SCom which solves the problem of [50] that needs to determine the threshold of SCom artificially, and a novel way to limit the feature subset's size. In this way, no wrapper method is contained in HSMOGA, and HSMOGA is classifier-independent in each generation. Which also makes it faster than other GA-based algorithms, and we are the first to combine GA-based feature selection algorithm with the feature interaction metric.
- A new Pareto-based ranking function which is more suitable than the traditional one in NSGA-ii for these three fitness functions is proposed.
- We make great improvements in the structure, whose superiority is obvious. HSMOGA starts with a new step called knowledge reserve, which precalculate the knowledge required for the calculation of fitness functions and the initial population generation. Compared with other GA-based feature selection methods, HSMOGA has a much lower time complexity and its parameter tuning can save a lot of time.
- Its initial population generation makes full use of the knowledge of data set, which makes solutions converge faster. Moreover, it is related to the third fitness function that controls the size of the feature subset, and jointly promotes HSMOGA to obtain better results.

We design three experiments to objectively evaluate the effectiveness of the two of three fitness functions and the Pareto-based ranking function proposed in this paper, and two experiments to objectively compare and assess the relative performance of HSMOGA in terms of classification performance and time cost. According to the experimental results, the fitness functions and ranking function proposed in this paper are satisfactory. For the data sets tested, HSMOGA outperforms other nine state-of-art feature selection algorithms including five classic and four more recent algorithms. Moreover, for the data sets tested, HSMOGA is much faster than other GA-based feature selection algorithm which is classifier-dependent in each generation, and even if the max number of generations increases a lot, the time cost will not increase much.

The structure of this paper is as follows. Section 2 introduces the preliminaries. In Section 3, we introduce HSMOGA step by step. Section 4 presents experiments, results, and analyses. Section 5 concludes our work.

2 Preliminaries

We introduce some methods related to this article. Firstly, the three feature filters. Secondly, the Symmetrical Complementary Coefficient. Finally, the Multi-objective genetic algorithm.

2.1 Three filters

2.1.1 Information gain

Considering the data set with N instances, n features, and 1 class variable, $S = (X, Y) = \{(x_{i1}, x_{i2}, \cdots, x_{in}, y_i)\}(i = 1, \cdots, N)$, where $X = (X_1, \cdots, X_n)$, $y_i \in \{1, 2, \cdots, C\}$, and C is the number of class.

Entropy is a measure of variable's uncertainty [5]. Entropy of the class variable Y is in (1).

$$H(Y) = -\sum_{k=1}^{C} p_k \log_2 p_k \tag{1}$$

where p_k is the proportion of instances whose class is k.

According to values of a specific feature $A \in \{X_j\}, j = 1, \cdots, n$, instances are divided into V subsets. Therefore, class variable Y is divided into $\{Y^1, Y^2, \cdots, Y^V\}$. Information Gain (IG) of feature A is in (2).

$$IG(A) = H(Y) - \sum_{v=1}^{V} \frac{|Y^v|}{|Y|} H(Y^v) \tag{2}$$

IG quantifies the information shared by feature A and class variable Y, which is often used as a filter to measure whether a feature is good or not for the classification problem. One of its most famous applications is the ID3 decision tree algorithm [35]. Although it has a defect that favors the features with more values.

2.1.2 Information gain ratio

Information Gain Ratio (IGR) is an improvement of IG to overcome the above defect through adding a penalty term, as in (3).

$$IGR(A) = \frac{IG(A)}{-\sum_{v=1}^{V} \frac{|Y^v|}{|Y|} \log_2 \frac{|Y^v|}{|Y|}} = \frac{H(Y) - \sum_{v=1}^{V} \frac{|Y^v|}{|Y|} H(Y^v)}{-\sum_{v=1}^{V} \frac{|Y^v|}{|Y|} \log_2 \frac{|Y^v|}{|Y|}} \tag{3}$$

IGR is also known as a feature filter and is applied to the C4.5 decision tree [36]. Whereas, IGR favors the features with fewer values. So, at each split point of C4.5 decision tree, firstly, select those features whose IG are greater than the average, and then select the feature with the largest IGR.

2.1.3 ReliefF

ReliefF is a distance-based filter method [28]. It assigns higher weights to the features which make distance between instances with the same class shorter, while distance between instances with different classes longer, and vice versa. It calculates feature's weight iteratively through select m instances randomly, as in (4).

$$W(A) = W(A) - \sum_{i=1}^{k} diff(A, R, NH_j)/(mk) + \sum_{c \neq class(R)} \left[\frac{p(c)}{1 - p(class(R))} \cdot \sum_{j=1}^{k} diff(A, R, NM_j(c)) \right] (mk) \tag{4}$$

$$diff(A, R_1, R_2) = \begin{cases} \frac{|R_1[A] - R_2[A]|}{\max(A) - \min(A)} & \text{If } A \text{ is continuous;} \\ 0 & \text{If } A \text{ is discrete and } R_1[A] = R_2[A]; \\ 1 & \text{If } A \text{ is discrete and } R_1[A] \neq R_2[A]; \end{cases} \tag{5}$$

where $W(A)$ is the weight of feature A. R is the current selected instance. $NH_j(j = 1, 2, \ldots, k)$ are R's k nearest neighbor instances which have the same class with R. $NM_j(c)(j = 1, 2, \ldots, k)$ are R's k nearest neighbor instances whose class are c and c is different from the class of R. $class(R)$ is the class of R. $p(c)$ is the proportion of class c. $R[A]$ is the value of R on A.

2.2 Symmetrical complementary coefficient

Symmetrical Complementary Coefficient (SCom) [50] is an effective metric of feature interactions. Feature interaction refers to that multiple features contribute more than the sum of these individual features' contribution. k-way feature interaction acts on k features $A_1, A_2, \ldots, A_k$. Let $e(Y; A_1, A_2, \ldots, A_i)$ $(i = 1, 2, \ldots, k)$ be the contribution of feature(s) to the class variable Y. So, k-way feature interaction exists if (6) exists [25].

$$e(Y; A_1, A_2, \ldots, A_k) \geq \sum_{i=1}^{k} e(Y; A_i) \tag{6}$$

SCom measures feature interaction based on classifiers at a high speed. Considering two features X_i, X_j, firstly, they are used individually to represent the characteristics of the data set, namely, new data set $S_i = (X_i, Y)$, $S_j = (X_j, Y)$. Then, two classification models M_i, M_j are established through learning S_i, S_j. Then, get two models' corresponding misclassified instances D_i, D_j. Note that, in this step, reserved validation data set can be used, or use the OOB data [2] if the classifier are Bagging-based ensemble method such as Random Forest just like [50] did. Next, D_i, D_j are classified by M_j, M_i respectively. Define D_i's subset which are correctly classified by M_j is D_{ij}. Define D_j's subset which are correctly classified by M_i is D_{ji}.

Therefore, define $ECom(i, j)$ is the Enhanced Complementary Coefficient from X_j to X_i , as in (7). $ECom(j, i)$ is the Enhanced Complementary Coefficient from X_i to X_j, as in (8). $SCom(i, j)$ is the SCom between X_i and X_j, as in (9).

$$ECom(i, j) = \frac{|D_{ij}|}{|D_i|} \tag{7}$$

$$ECom(j, i) = \frac{|D_{ji}|}{|D_j|} \tag{8}$$

$$SCom(i, j) = SCom(j, i) = \frac{ECom(i, j) + ECom(j, i)}{2} = \frac{1}{2}\left(\frac{|D_{ij}|}{|D_i|} + \frac{|D_{ji}|}{|D_j|}\right) \tag{9}$$

The situation of two groups of features can be given similarly. However, it will cost much more time, while calculating SCom between two features takes very little time, as we only need to use one feature to create the classifier. This paper only considers the circumstances of two features. Note that, a large $SCom(i, j)$ does not represent X_i, X_j should be contained, but only represents X_i, X_j should be considered simultaneously, and we should combine SCom with two features' contribution to class variable to decide whether to keep them or not [50]. Thus, SCom can be combined with some feature evaluation methods and feature subset searching algorithms to obtain a better feature subset.

2.3 Multi-objective genetic algorithm

Genetic algorithm (GA) [8] is a kind of heuristic evolutionary algorithms based on language of natural genetics and biological evolution. In GA, each variable corresponds to a gene and one solution of these variables corresponds to a chromosome. There is a fitness function to evaluate chromosomes' performance by calculating their fitness values. According to these fitness values, selection, crossover, and mutation operation are implemented to these chromosomes to generate new individuals. These steps are cycled until the stopping criteria is reached. In this paper, 'chromosome', 'individual', and 'solution' are the same, and feature subset corresponds to them.

GA deals with just one fitness function. Whereas, a large proportion of real-world problems are multi-objective inherently, in which more than one objective needs to be satisfied

simultaneously. Moreover, sometimes these objectives are conflicting. Apparently, using a single fitness function will cause that the promising resolution cannot be found. Multi-objective genetic algorithm (MOGA) [9, 27] was proposed to solve these problems. MOGA contains more than one fitness function, i.e., $F_1(\cdot), F_2(\cdot), \cdots, F_m(\cdot)$. In each generation, these fitness functions' values of all solutions need to be calculated.

A solution ch_1 dominate another solution ch_2 if the following two condition are met:

1. $\forall i \in \{1, 2, \cdots, m\}, F_i(ch_1) \geq F_i(ch_2)$
2. $\exists j \in \{1, 2, \cdots, m\}, F_j(ch_1) > F_j(ch_2)$

The dominating relation is not reflexive and symmetric but transitive. Considering one generation's solution set R, its non-dominated solution set R', which is also called the Pareto front, refers to a subset whose members are not dominated by any member of R. Some of MOGA's operations are implemented based on it.

3 HSMOGA

In this section, a novel feature selection algorithm is proposed, i.e., **H**ybrid filter and **S**ymmetrical Complementary Coefficient based **M**ulti-**O**bjective **G**enetic **A**lgorithm feature selection (**HSMOGA**). Let's introduce HSMOGA step by step.

3.1 Knowledge reserve

Unlike the traditional GA and MOGA, we are not starting with generating the initial population, but starting with *knowledge reserve*. Knowledge reserve refers to calculation and storage of knowledge or information that will be used in subsequent steps including the initial population generation and calculation of fitness functions. Reserved knowledge is divided into two parts, features' hybrid weights, i.e., $HW = (hw_1, hw_2, \cdots, hw_n)$, and their SComs, i.e., $SC_{n\times n}$. The hybrid filter is made up with three classic filter, i.e., IG, IGR, and ReliefF. HW is the normalization of the average ranking of IG, IGR and ReliefF weights of these features, as in (10) and (11). $SC_{n\times n}$ is a symmetric matrix of SCom of these features, and each element of it is the SCom of the two corresponding features, as in (12).

$$HW_{temp} = \frac{rank(IG(X)) + rank(IGR(X)) + rank(W(X))}{3} \tag{10}$$

$$HW = \frac{HW_{temp} - \min(HW_{temp})}{\max(HW_{temp}) - \min(HW_{temp})} \tag{11}$$

$$SC = \begin{pmatrix} 0 & SCom(1,2) & SCom(1,3) & \cdots & SCom(1,n) \\ SCom(2,1) & 0 & SCom(2,3) & \cdots & SCom(2,n) \\ SCom(3,1) & SCom(3,2) & 0 & \cdots & SCom(3,n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ SCom(n,1) & SCom(n,2) & SCom(n,3) & \cdots & 0 \end{pmatrix} \tag{12}$$

where $X = (X_1, \cdots, X_n)$. $IG(\cdot), IGR(\cdot), W(\cdot), SCom(i, j)$ are calculated according to (2), (3), (4) and (9), respectively. $IG(X), IGR(X), W(X)$ are all vectors, and each of their elements represents weight of the corresponding feature. $rank(\cdot)$ means calculating the rank of every element in a vector. The lowest is assigned 1, the second lowest is assigned 2, etc. $min(\cdot), max(\cdot)$ represent the minimum and maximum values in a vector, respectively. Denote that $HW[k] = hw_k, SC[i, j] = SCom(i, j)$.

The reason for choosing IG, IGR and ReliefF is that they are commonly used filters which have good effects. Meanwhile, using both IG and IGR can solve the problem of their respective bias, which is somewhat similar to C4.5 [36].

Given that a single filter may performs better on some data sets and worse on some other data sets [46], in this paper, we just want to get a hybrid filter that performs well on different kinds of data sets, that is, a more balanced hybrid filter. Of course, it can be replaced with other different kinds of filters or hybrid filters. Therefore, we choose to average the results of these three commonly used filters. Moreover, $rank(\cdot)$ can avoid the deviation caused by metrics of different filters. For example, there are a data set with two features A and B,

$$\begin{aligned} IG(A) = 0.19,\ IGR(A) = 0.1,\ W(A) = 0.1 \\ IG(B) = 0.1,\ IGR(A) = 0.12,\ W(A) = 0.12 \\ \frac{IG(A) + IGR(A) + W(A)}{3} = \frac{0.39}{3} \\ \frac{IG(B) + IGR(B) + W(B)}{3} = \frac{0.34}{3} \\ \frac{rank(IG(A)) + rank(IGR(A)) + rank(W(A))}{3} \\ = \frac{2+1+1}{3} = \frac{4}{3} \\ \frac{rank(IG(B)) + rank(IGR(B)) + rank(W(B))}{3} \\ = \frac{1+2+2}{3} = \frac{5}{3} \end{aligned}$$

By inspection, the latter is more reasonable. In the experiment section, we will design an experiment to demonstrate that this hybrid filter performs more balanced on different data sets, and always has a good performance.

3.2 Population generation

Generating the initial population is a crucial thing in GA, MOGA or some other evolutionary algorithms. Whereas, a large proportion of studies use random [15, 16, 21, 23] or some pseudo-random pattern generator [6]. Knowledge of the used data set should be fully utilized. Although, this bring in more time cost. Therefore, we introduced the knowledge reserve to share time cost with other steps.

Every generation contains M chromosomes. A chromosome $ch_s (s = 1, 2, \cdots, M)$ is a binary vector whose length is n. Each chromosome is a feature subset, in which '1' represents its corresponding feature is contained, and vice versa. When generating the first generation of individuals, each element of ch_s, i.e., $ch_s[k](k = 1, 2, \cdots, n)$ is '1' with a probability p_k and is '0' with a probability $1 - p_k$, as in (13). The formula of $p_k(k = 1, 2, \cdots, n)$ is in (14) and (15).

$$ch_s[k] = \begin{cases} 1 & p_k \\ 0 & 1 - p_k \end{cases} \tag{13}$$

$$p_{k,temp} = HW[k] + bias \tag{14}$$

$$p_k = \begin{cases} 0 & \text{if } p_{k,temp} \in [-1, 0] \\ p_{k,temp} & \text{if } p_{k,temp} \in (0, 1) \\ 1 & \text{if } p_{k,temp} \in [1, 2] \end{cases} \tag{15}$$

where $bias$ is a control term that controls the size of feature subsets when facing a specific data set. Range of $bias$ is $[-1, 1]$. Thus, we can control the size of feature subset by controlling the probability that each element of chromosome takes '1'. Meanwhile, it is guaranteed that the better the feature (the larger the $HW[k]$), the greater the probability that it appears in the feature subset.

3.3 Fitness functions

In this paper, three new fitness functions are combined with MOGA. They are proposed from three perspectives, i.e., average hybrid weight, average strength of feature interaction, and size of feature subset, as in (16), (17) and (18), respectively. Values of these three fitness functions are the larger the better.

$$F_1(ch_s) = \frac{1}{|K|} \sum_{k \in K} HW[k] \tag{16}$$

$$F_2(ch_s) = \frac{2}{|K|(|K| - 1)} \sum_{k_1 \in K} \sum_{k_2 \in K \wedge k_2 > k_1} SC[k_1, k_2] \tag{17}$$

where K is the collection of sequence numbers of elements that are not '0' in ch_s. $|K|$ is K's length, and $|K|$ is also the size of the feature subset which corresponds to ch_s. For example, if $ch_s = (1, 0, 1, 1, 0)$, then $K = \{1, 3, 4\}$, $|K| = 3$.

$$F_3(ch_s) = -\,||K| - en| \tag{18}$$

$$en = \sum_{k=1}^{n} p_k \tag{19}$$

where en represents the expected size of feature subset, as the probability that each feature appears in the initial feature subset is p_k. By searching for the optimal $bias$, we can get the optimal en and make the feature subsets of every generation closer to en.

$F_1(\cdot)$ is the average hybrid weight of a feature subset. $F_2(\cdot)$ is the average feature interaction strength of every two features. At present, no research has combined feature interaction with GA-based feature selection algorithm. So, we proposed the $F_2(\cdot)$ function to quantify the feature interaction degree of each feature subset generated by MOGA. $F_2(\cdot)$ solves the problem that [50] needs to artificially determine the threshold of SCom. Here we only need to calculate the average value and compare it with the values of other feature subsets. $F_3(\cdot)$ is the limitation of feature subset's size, and it can also be seen as the limitation of $F_1(\cdot)$ and $F_2(\cdot)$. The feature subset whose size is closer to en has a larger value of $F_3(\cdot)$.

These three functions are conflicting to some extent, which is suitable for MOGA. If there is no limitation, the best solution which maximize $F_1(\cdot)$ will be the feature subset with just one feature, and this feature has the largest hybrid weight. Similarly, the best solution which maximize $F_2(\cdot)$ will be the feature subset with only two features, and the SCom between these two features are the largest. We can be find that $F_1(\cdot)$ and $F_2(\cdot)$ conflict with $F_3(\cdot)$ respectively. As the $F_1(\cdot)$-optimized solution conflicts with $F_2(\cdot)$-optimized solution (One of two contains one feature and the other contains two features), $F_1(\cdot)$ conflicts with $F_2(\cdot)$. As $F_3(\cdot)$ exists, feature subset's size will be closer to en.

When dealing with multi-objective problems, the most commonly used approaches are weighted sum, altering objective functions, and Pareto-based ranking approaches [27]. In this paper, Pareto-based ranking approach is used, and we propose a new ranking function to assign solution ch_s $(s = 1, 2, \cdots, M)$ a rank within every generation, as in (20).

$$RANK(ch_s) = (M - n_{bedom}(ch_s) + n_{dom}(ch_s))/2 \tag{20}$$

where $n_{bedom}(ch_s)$ is the number of solutions which dominate ch_s, and $n_{dom}(ch_s)$ is the number of solutions which are dominated by ch_s within the current generation.

In the multi-objective optimization problem, the relationships between two individuals are more complicated, i.e. in addition to being completely superior to and completely inferior to, it may be unable to determine which one is better. In fact, $M - n_{bedom}(ch_s)$ and $n_{dom}(ch_s)$ can be seen as the upper and lower approximation limitation of ch_s's rank, where the former indicates the number of individuals that are not better than ch_s the latter indicates the number of individuals that are worse than ch_s, and the average of them can be seen as the rank of ch_s within the whole generation.

Figure 1 gives an example of $RANK(\cdot)$. There are two fitness functions. The dot represents ch_s, and triangles represent other solutions. The small ellipse represents the solutions dominated by ch_s, and the large ellipse represents the solutions which dominate ch_s. Therefore,

$$n_{dom}(ch_s) = 6,\ n_{bedom}(ch_s) = 6,\ M - n_{bedom}(ch_s) = 11$$
$$RANK(ch_s) = (6 + 11)/2 = 8.5$$

The reason for constructing such a new ranking function is that the commonly used fast non-dominated sorting and the crowding-distance estimation [27] are not suitable for the three fitness functions proposed in this paper. We want those solutions who perform well on all three functions are used to generate the next generation, and those edge points on different non-dominated fronts should be discarded. The fast non-dominated sorting and the crowding-distance estimation which are the methods used by NSGA-ii to solve multi-objective problems, cannot solve these above problems, as edge points will have higher ranks and will be used to form the next generation.

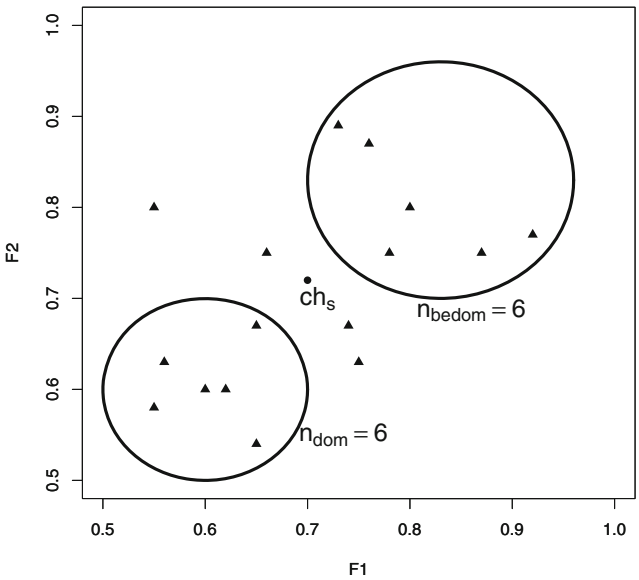


Fig. 1 An example of ranking function

In the experiment section, we will design several experiments to demonstrate the superiority and effectiveness of these fitness functions and the new ranking function.

The rank of each solution, instead of its fitness function values, will be used to perform the evolution operation, i.e., selection, crossover, and mutation.

3.4 Evolution operation

The flow chart of evolution operation we used is in Fig. 2. The solution with a larger rank has a greater chance of being selected. Chromosomes are selected through the roulette wheel strategy. Moreover, the chromosomes whose ranks are smaller than the average will not be selected.

When performing crossover operation, two chromosomes also called parents are selected at a time, and two children are generated. One-point crossover is used. Firstly, randomly select one point indicating the position of gene in chromosome. Then, parents exchange the gene sequences after the selected point mutually, and get two children.

When performing mutation operation, one chromosome is selected at a time. In this paper, each gene of the parent chromosome will mutate with a mutation rate r_m. The gene to be mutated will change from '0' to '1' or from '1' to '0'.

Besides, we make some minor changes to improve our work. In the crossover and mutation operation, child(ren) identical to parent(s) is prohibited. If such situation happens, we will repeat the crossover and mutation operation.

Moreover, we add the replace operation to guarantee that the next generation will not be worse than the current generation. If parents dominate their children, then children will be replaced by their parents. In other words, in this case, parents will take the place of children into the next generation.

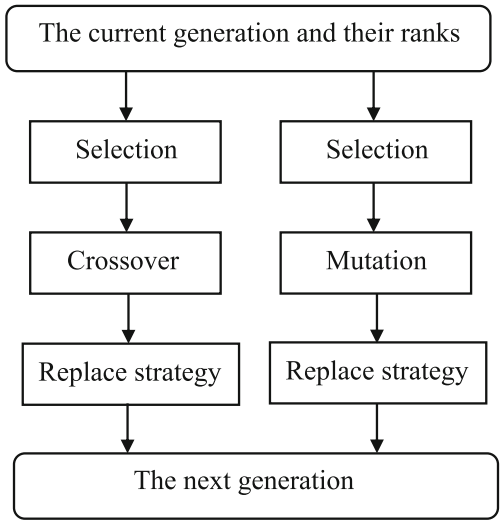


Fig. 2 Evolution operation flow chart

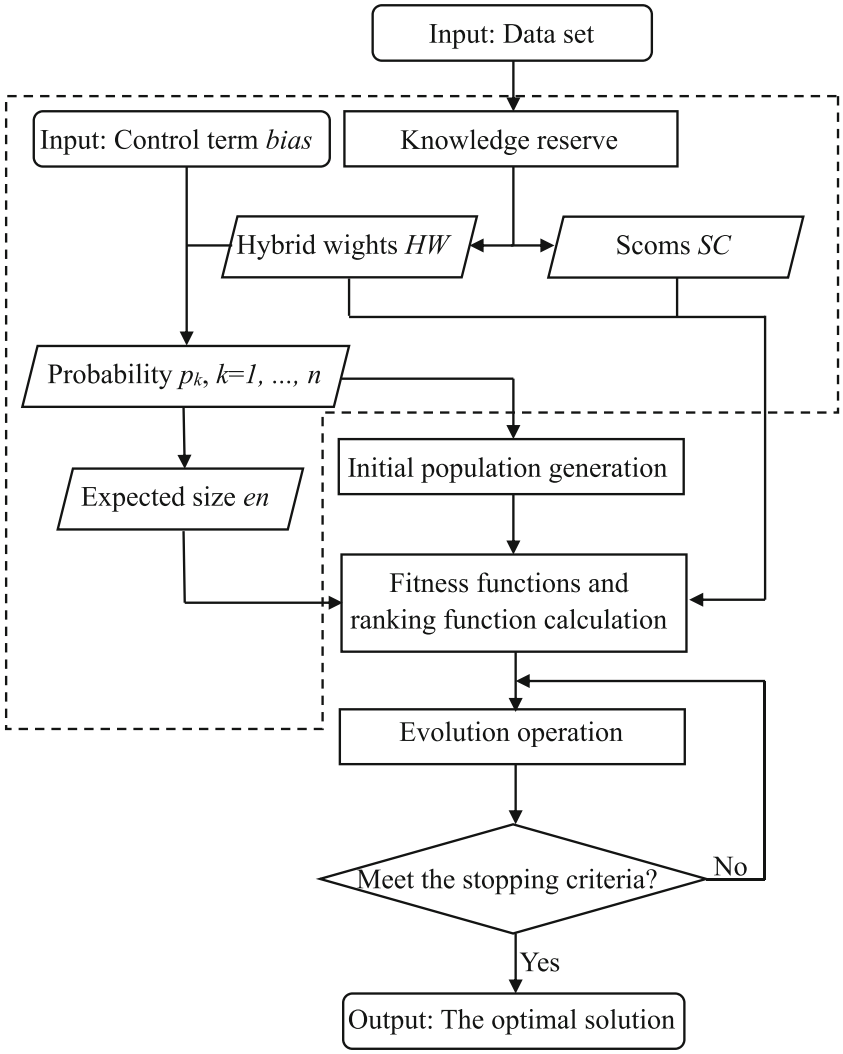
Individuals generated by crossover and mutation operation make up the next generation, and each generation has the same number of individuals.

3.5 HSMOGA algorithm

The proposed HSMOGA algorithm is made up with these previous parts. It contains three new fitness functions and a new ranking function. The three new fitness function are proposed from three conflicting perspectives, i.e., average hybrid feature weights, average feature interaction strength, and the limitation of feature subset's size. The new ranking function is Pareto-based and it considers the upper and lower approximation limitation of every solution's rank at the same time. HSMOGA's initial population generation makes full use of the data set's knowledge. Moreover, the knowledge or information needed for the fitness functions calculation and the initial population generation is precalculated, which indicates these parts will run at a very fast speed. Besides, we add some minor steps to improve the algorithm, as follows. Chromosomes whose ranks are smaller than the average will not be selected. In the crossover and mutation operation, child(ren) identical to parent(s) is prohibited. The replace operation is added to guarantee the next generation will not be worse than the current generation.

Flow chart of HSMOGA is in Fig. 3, where the components contained in the dotted box is the main part where HSMOGA differs from other GA-based algorithms. The pseudo-code of HSMOGA is in Algorithm 1. HSMOGA outputs individuals of the last generation. If you want to reduce time cost, you can choose the highest ranked individual as the final feature subset. In this paper, in order to get better results, the top 50% of individuals will be selected and compared by the classifier to get the best feature subset.

Fig. 3 HSMOGA flow chart



Algorithm 1 HSMOGA algorithm.

Input:
Data set: $S = \{(x_{i1}, x_{i2}, \cdots, x_{in}, y_i)\}(i = 1, \cdots, N)$
Population size: M
Maximum number of generations: G
Control term of feature subset's size: $bias$
Number of individuals generated by crossover operation: $2a$
Number of individuals generated by mutation operation: $b = M - 2a$
Mutation rate: r_m
Begin
Calculate HW and SC according to (11) and (12), respectively.
Calculate p_k with the control term $bias$ according to (15).
Generate the initial population $ch_s(s = 1, 2, \cdots, M)$ according to (13), and initialize $g = 1$.
Calculate fitness values and ranks of all chromosomes according to (16), (17), (18) and (20), respectively.
repeat
  for $i = 1$ to a **do**
    Select one couple of chromosomes.
    Perform the crossover operation, and get 2 children.
    Calculate and record children's fitness values.
    if Parent dominates the child(ren) **then**
      The dominated child(ren) is(are) replaced by this parent. The recorded fitness values are replaced by parent's fitness values.
    end if
  end for
  for $j = 1$ to b **do**
    Select a chromosome.
    Perform the mutation operation, and get 1 child.
    Calculate and record the child's fitness values.
    if Parent dominates the child **then**
      The dominated child is replaced by parent. The recorded fitness values are replaced by parent's fitness values.
    end if
  end for
  Calculate the ranks of the new generation which is made up with the individuals generated by the previous two steps.
  $g = g + 1$
until $g > G$
End
Output: Individuals of the last generation and their ranks.

3.6 Time complexity analysis

As the knowledge needed for the fitness function is pre-calculated, the calculation of fitness functions takes very little time. Even though the maximum number of generations is large, it won't take too much time. Therefore, the two parts that take the most time are the knowledge reserve and the final selection of the best feature subset using classifier.

As filter method of feature selection is very fast, main part of the former's time cost is the calculation of $SC_{n\times n}$. It can be separated into two parts, i.e., the establishment of n models and classification of the misclassified instances between each two classifiers. In this paper, Random Forest is selected as the classifier just like [50] did, as Random Forest has fast speed, high accuracy, and the OOB data as a validation set [2] brings great convenience in calculating the SCom and parameters' tuning.

Time complexity of establishing n model where each one contains just one feature is the same as that of establishing a Random Forest model which contains n features, namely, $O(nN\log_2 N)$. Time complexity of $D_i(i = 1, 2, \cdots, n)$ being classified by other $n - 1$ models is $O(n|D_i|)$. So, the time complexity of classification of the misclassified instances between each two classifiers is $O(n(|D_1| + |D_2| + \cdots + |D_n|))$. Given that the number of misclassified

instances is often not too much and normally $n < N$ exists. Therefore, time complexity of knowledge reserve is the same as that of establishing a Random Forest model with n features, i.e., $O(nN \log_2 N)$.

Final selection of the optimal feature subset indicates establishing $M/2$ models using $M/2$ feature subsets of the last generation, where M is the generation size. Let n' be the average size of these $M/2$ feature subsets. As $F_3(\cdot)$ limits the size of feature subset, the size of these last generation feature subset is close to the expected size en. So, time complexity of final selection is $O(\frac{1}{2}Mn'N \log_2 N)$.

So, time complexity of whole method is $O(\max\{nN \log_2 N, \frac{1}{2}Mn'N \log_2 N\})$. Normally, M is not very large. By inspection, the HSMOGA algorithm proposed in this paper has a fast speed. As HSMOGA is classifier-independent in each generation except the last generation, its time complexity is much lower than the time complexity of GA-based feature selection algorithm which is classifier-dependent in each generation, i.e., normally $O(MGn'N \log_2 N)$, which indicates that HSMOGA can use a larger maximum number of generations G without increasing much time cost. Moreover, as the existence of knowledge reserve, parameters' tuning does not have to go through all the steps, which significantly reduces time cost. In the experiment section, we will demonstrate that HSMOGA has a significant advantage in terms of time cost over those GA-based feature selection algorithms that are classifier-dependent in each generation.

4 Experiment

4.1 Data sets

In this paper, 17 real-world data sets in the UCI Machine Learning Repository [10] are selected, which are all available to evaluate the performance of machine learning methods. They come from different fields such as clinical research, purchasing intention research, spam email classification, sonar target classification, molecular research, etc. They are of different number of features, classes, instances, and ratios of discrete and continuous features, which can on behalf of a great part of real-world problems, so that they can evaluate the performance of a method accurately. Features of these data sets were normalized by Min-Max scaling before experiments. Details of these data sets are in Table 1. Their names are the first words or abbreviations of the items in UCI Repository and more detailed description of these data sets can be found there.

Table 1 Date set description. N is the number of instances. n is the number of features. Con is the number of continuous features. Dis is the number of discrete features. $Con + Dis = n$. C is the number of classes

Dataset	N	n	Con	Dis	C
CMC[a]	1473	9	2	7	3
Thoracic[b]	470	16	3	13	2
Online[c]	12330	17	10	7	2
SPECT[d]	267	22	0	22	2
Default[e]	30000	23	13	10	2
Dermatology[f]	358	34	1	33	6
SPECTF[g]	267	44	44	0	2
Spambase[h]	4601	57	0	57	2
Connectionist[i]	208	60	60	0	2
Quality[j]	287	69	62	7	3
Libras[k]	360	90	90	0	15
Musk1[l]	168	166	166	0	2
Musk2[m]	6598	166	166	0	2
LSVT[n]	126	309	309	0	2
Gastro1[o]	76	698	698	0	3
Gastro2[p]	76	698	698	0	3
Internet[q]	3279	1558	3	1555	2

[a]The 'Contraceptive method used' is the class variable
[b]The 'Risk1Y: 1 year survival period' is the class variable
[c]The 'Revenue' is the class variable
[d]The 'OVERALL_DIAGNOSIS' is the class variable
[e]The 'default payment' is the class variable
[f]The 'Dermatology' is the class variable
[g]The 'OVERALL_DIAGNOSIS' is the class variable
[h]The 'type spam' is the class variable
[i]The 'label associated with each record' is the class variable
[j]The 'consensus' is the class variable
[k]The 'class' is the class variable
[l]The 'class' is the class variable
[m]The 'class' is the class variable
[n]The 'Binary class' is the class variable
[o]The 'the type of lesion' is the class variable, and 'the type of light used' is 'white light (WL)'
[p]The 'the type of lesion' is the class variable, and 'the type of light used' is 'narrow band imaging (NBI)'
[q]The 'class' is the class variable

4.2 Experiment settings

We design three experiments to objectively evaluate the effectiveness of the two of three fitness functions and the Pareto-based ranking function proposed in this paper, and other two experiments to objectively compare and assess the relative performance of HSMOGA in terms of classification performance and time cost.

The first experiment is to evaluate the effectiveness of the hybrid filter through demonstrating that the three single filters, IG, IGR, and ReliefF, perform better on some data sets and worse on some other data sets, while the hybrid filter performs more balanced on different data sets, that is,

performs well on different data sets. In this experiment, all 17 data set will be used.

Details of evaluating the effectiveness of SCom and its superiority in combination with filters and feature subset generation algorithms can be found in [50]. So, the second experiment is to evaluate the effectiveness of $F_3(\cdot)$ function through demonstrating that the size of the feature subsets will be close to en which is the expected size of optimal feature subset and is controlled by $bias$. In this experiment, the 'Musk2' data set will be used, as it has a relatively large number of instances and features.

The third experiment is to evaluate the effectiveness of $RANK(\cdot)$ through comparison with other two methods, as follows. Note that, initial population generation method, three fitness functions, and the final optimal feature subset generation method are the same in all three cases. In this experiment, all 17 data set will be used.

- Case 1: HSMOGA.
- Case 2: $RANK(\cdot)$ is replaced by fast non-dominated sorting and the crowding-distance estimation. Binary tournament selection is used, and the replace operation is the same as HSMOGA, which is based on the three fitness values.
- Case 3: $RANK(\cdot)$ is replaced by fast non-dominated sorting and the crowding-distance estimation. Binary tournament selection is used, and the replace operation is replaced by the one in NSGA-ii, which is based on the non-dominated ranks and crowding distances. Actually, this kind of method is NSGA-ii [27].

The fourth experiment is to compare and assess the classification performance of HSMOGA through comparing it with nine state-of-art feature selection algorithms including five classic and four more recent algorithms. Algorithms for comparison are as follows. Among them, mRMR, ReliefF, MDG, CFR need to determine the size of feature subset artificially. So, searching algorithm are implemented in order to get their feature subsets containing the first n'' features. n'' is optimized by validation set. In this experiment, all 17 data set will be used.

1. mRMR [34], a commonly used filter method.
2. ReliefF [28], a commonly used filter method.
3. MDG [3], a commonly used embedded method, included in the establishment of Random Forest.
4. RFE [39], a commonly used wrapper method.
5. Boruta [29], a commonly used wrapper method.
6. CRF [14], a more recent filter method, which considers feature's relevance and redundancy at the same time.
7. RRSS [50], a more recent wrapper method, which takes the feature interactions into account.
8. IGDFS [24], a more recent GA-base method, which combines GA, a filter, and a wrapper.
9. FSBOGA [6], a more recent bi-objective GA-based feature selection algorithm whose two fitness functions are defined using rough set theory and the Kullback-Leibler divergence.
10. HSMOGA, the algorithm proposed in this paper.

The fifth experiment is to demonstrate that HSMOGA has a significant advantage over IGDFS and FSBOGA in terms of time cost, and the latter two stands for state-of-art GA-based feature selection algorithms which are classifier-dependent in each generation. In this experiment, data sets 'Online', 'Default', 'Musk2', and 'Internet' will be used, as they have a relatively large number of instances or features. The hardware environment used to run this experiment is equipped with 2.50-GHz i5-7300HQ CPU and 8 GB of RAM.

Package of R language is used to perform Random Forest. For easier comparison, the parameters of Random Forest are set to default values where the number of selected features at every splitting node is $mtry = \lfloor \log_2(\text{the number of features}) \rfloor + 1$, and number of trees is set to $ntree = 500$. On each data set, five times of five-fold cross-validation are performed, and the average values are taken as the final results.

Some of parameters used in HSMOGA are in Table 2. These parameters are not the optimal, as the time cost is reduced. $bias$ is the most important parameter in our view, because it not only controls the probability of occurrence of each feature in the initial population, but also limits the size of feature subset in each generation through the $F_3(\cdot)$ function. So, we search for its optimal value within $[-1, 1]$ to maximize the kappa coefficient. Note that, the test data does not appear in any part of the model establishment including the parameter tuning. The OOB data [2] of training set is used as the validation set.

Main parameters of comparison algorithms are as follows.

- The 'mRMRe' package of R language is used to implement mRMR algorithm. All parameters are their default values.

Table 2 Parameters of HSMOGA

Parameter	Value
Population size(M)	50
Maximum number of generations(G)	20
Number of individuals generated by crossover operation($2a$)	30
Number of individuals generated by mutation operation(b)	20
Mutation rate(r_m)	0.1

- The 'COREIearn' package of R language is used to implement ReliefF algorithm. The number of iterations is equal to the number of instances N. The number of selected neighbor is $k = 5$.
- The number of parameter values of MDG are the same of those in RF, i.e., $mtry = \lfloor \log_2(\text{the number of features}) \rfloor + 1, ntree = 500$.
- The 'caret' package of R language is used to implement RFE algorithm. $functions = rfFuncs, method =$ "cv", $number = 5$.
- The 'Boruta' package of R language is used to implement Boruta algorithm. $pValue = 0.01, mcAdj = TRUE, maxRuns = 100$.
- When implementing CFR, continuous features are discretized by Ameva algorithm [17].
- When implementing RRSS, the parameter values of RF-efficient-ReliefF are the same as those of ReliefF, and the thresholds of SCom-SFS are determined according to their piecewise function.
- For better comparison, population size and Maximum number of generations of IGDFS are equal to those of HSMOGA, i.e., 50 and 20. Other parameters of IGDFS are consistent with [24], that is, tournament size is 2, mutation rate is 0.1, and mutation type is uniform mutation.
- Similarly, population size and Maximum number of generations of FSBOGA are equal to those of HSMOGA, i.e., 50 and 20. Other parameters of HSMOGA are consistent with [6], that is, the probability of crossover is 0.9, and the probability of mutation is 0.15.

4.3 Evaluation metrics

When measuring the quality of a filter, i.e., rationality of feature ranking, one commonly used method is to use the first n' features of its resulting order to build the model, and n' changes from 1 to $n-1$, that is $n-1$ models. Then, we can compare two or more filters by calculating the rank of their models when facing different n'. In experiment one, the Average Rank (AR) is selected. One filter's AR on each data set is in (21).

$$AR_i = \frac{1}{4(n-1)} \sum_{n'=1}^{n-1} rank_{in'} \tag{21}$$

where $i = 1, 2, 3, 4$ represents IG, IGR, ReliefF, Hybrid Filter (HF), respectively. $rank_{in'}$ represents the i-th filter's rank when using the first n' features of its resulting order to build the model, and the best is ranked 1 and the worst is 4. The range of AR_i is [0, 1], and the lower, the better.

In this paper, kappa coefficient is the main evaluation metric of models' results. When facing a confusion matrix, accuracy only considers the instances in the diagonal direction which are classified correctly. Whereas, kappa coefficient also takes the misclassified and unidentified instances outside the diagonal into account, which indicates that kappa coefficient is more reasonable especially when facing the imbalanced data set. Its range is $[-1, 1]$, and the larger, the better. In this paper, parameter tuning of every algorithm is also based on the kappa coefficient.

In addition, accuracy and G-mean are selected as the secondary evaluation metrics, where G-mean is the geometric mean of the recall rate of each class. The range of them is [0, 1], and the larger, the better. Table 3 is a typical confusion matrix of C classes. Formulae of accuracy, G-mean and kappa coefficient are in (22), (23) and (24), respectively. In experiment three, kappa coefficient is selected, as we're just going to demonstrate the superiority of the rank function. In experiment four, kappa coefficient, accuracy and G-mean are selected.

$$accuracy = \frac{\sum_{i=1}^{C} N_{i,i}}{N} \tag{22}$$

$$G-mean = \sqrt[C]{\prod_{i=1}^{C} \frac{N_{i,i}}{N_i}} \tag{23}$$

$$kappa\ coefficient = \frac{p_0 - p_e}{1 - p_e} \tag{24}$$

where

$$p_0 = accuracy \tag{25}$$

$$p_e = \frac{\sum_{i=1}^{C} N_i \cdot \hat{N}_i}{N^2} \tag{26}$$

4.4 Results and analyses

Experiments one to three are designed to objectively evaluate the effectiveness of the two of three fitness functions and the Pareto-based ranking function proposed in this paper, and experiments four and five are designed to objectively compare and assess the relative performance of HSMOGA in terms of classification performance and time cost. So, they are separated.

Table 3 A typical confusion matrix

	Actual class				
Predicted class	1	2	⋯	C	Total
1	$N_{1,1}$	$N_{1,2}$	⋯	$N_{1,C}$	$\hat{N}_1$
2	$N_{2,1}$	$N_{2,2}$	⋯	$N_{2,C}$	$\hat{N}_2$
⋯	⋯	⋯	⋯	⋯	⋯
C	$N_{C,1}$	$N_{C,2}$	⋯	$N_{C,C}$	$\hat{N}_C$
Total	N_1	N_2	⋯	N_C	N

4.4.1 Experiments one to three

In experiment one, AR results of 4 filters on 17 data sets are in Table 4. The last line of it is the average results of all data sets, and the best filter for each data set is highlighted in bold, the second best is highlighted in italic.

By inspection, we can find that single filters performed better on some data sets and worse on some other data sets, such as, IG performed well on Spambase data set but poorly on CMC data set, IGR performed well on CMC data set but poorly on SPECTF data set, ReliefF performed well on SPECTF data set but poorly on Spambase data set. Whereas, the hybrid filter is almost in the top two on all the 17 data sets, and its average AR is the best. It is apparently that the rank operation and average operation on these three single filters are effective. The performance of the hybrid filter is more balanced, and it is good on almost every data set.

In experiment two, in order to prove $F_3(\cdot)$ function is effective, we plot the average value of $F_3(\cdot)$ function of each generation, as in Fig. 4, where

Mean of F3 function $= 0$ means all the feature subsets in that generation achieve the size of en which is our expected size and it can be controlled by $bias$.

By inspection, we can find that as the number of generations increases, the average value of $F_3(\cdot)$ function increases. Apparently, $F_3(\cdot)$ function is effective. We can adjust the $bias$ term so that the size of the feature subsets is close to the expected size, and $F_1(\cdot)$ and $F_2(\cdot)$ functions are also optimized at the same time. Moreover, we can easily find that the average value of $F_3(\cdot)$ function of the initial population is almost in $[en-5, en+5]$. It indicates that our initial population generation method is very good, and the population will converge faster.

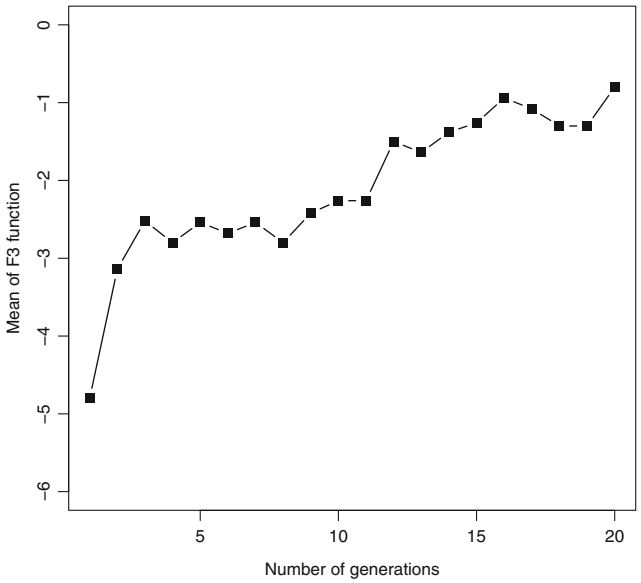


Fig. 4 Experiment two on Musk2 data set

In experiment three, in order to demonstrate the effectiveness of $RANK(\cdot)$ function, HSMOGA was compared with other two methods. Kappa coefficient results of these three cases on 17 data sets are in Table 5, and the best method on each data set is emphasized in bold.

Table 4 AR results of 4 filters on 17 data sets(%)

Dataset	IG	IGR	ReliefF	Hybrid filter
CMC	75.00	*54.17*	75.00	**45.83**
Thoracic	**52.88**	69.23	67.31	*60.58*
Online	*52.17*	84.78	72.83	**40.22**
SPECT	**53.98**	73.30	64.20	*58.52*
Default	66.18	*63.24*	76.47	**44.12**
Dermatology	61.40	70.59	*59.93*	**58.09**
SPECTF	69.32	88.07	**38.07**	*54.55*
Spambase	**42.98**	61.84	86.84	*58.33*
Connectionist	75.83	84.58	**35.83**	*53.75*
Quality	*58.33*	74.82	60.14	**56.70**
Libras	85.42	**32.64**	69.03	*62.92*
Musk1	**49.10**	61.75	77.86	*61.30*
Musk2	**37.20**	82.68	73.80	*56.33*
LSVT	*63.83*	65.45	66.02	**54.69**
Gastro1	**57.16**	63.61	71.80	*57.43*
Gastro2	**50.97**	*55.78*	82.51	57.75
Internet	**54.79**	63.52	71.09	*59.767*
Average	*59.21*	67.65	67.57	**55.34**

Table 5 Kappa coefficient results of three cases (%)

Dataset	Case1	Case2	Case3
CMC	**24.16**	20.45	17.07
Thoracic	**3.69**	3.24	-2.01
Online	**60.15**	57.63	58.87
SPECT	**49.06**	46.91	29.87
Default	**37.49**	37.17	36.50
Dermatology	**97.26**	95.78	95.72
SPECTF	**29.34**	28.40	25.54
Spambase	**90.63**	89.00	88.66
Connectionist	**66.19**	63.16	63.63
Quality	**93.20**	92.14	91.42
Libras	**80.73**	78.86	77.59
Musk1	**78.11**	76.71	75.72
Musk2	**90.87**	89.24	89.52
LSVT	**65.15**	62.98	63.19
Gastro1	**43.31**	39.33	42.43
Gastro2	**44.99**	40.14	44.24
Internet	**91.90**	91.81	91.04

According to the results, we can find that HSMOGA is better than the other two methods including the commonly used NSGA-ii method on 17 data sets. It indicates that the proposed $RANK(\cdot)$ function is more suitable for the three fitness functions proposed in this paper. The reason is that only those individuals who perform well on all the three fitness functions are what we want, and actually they represent better feature subsets. It seems like we focus on those individuals that are more central in the Pareto front, while NSGA-ii focuses more on finding the wider Pareto front.

4.4.2 Experiments four and five

In experiment four, Kappa coefficient is the main evaluation metric. Accuracy and G-mean are the secondary evaluation metrics. Results of them of these ten algorithms on 17 data sets are in Tables 6, 7, and 8, respectively. The best result for every data set is emphasized in bold.

For more intuitive comparison, we calculate the Friedman's ranks [13] of these three metrics of ten algorithms. Firstly, each algorithm's ranks on these 17 data sets are calculated. The best algorithm is ranked 1, the second-best algorithm is ranked 2, etc. Then average these ranks of every algorithm. Table 9 gives Friedman's ranks of these ten algorithms, and the best one is emphasized in bold.

We perform Analysis of Variance (ANOVA) to prove that there are significant differences between these ten algorithms. Then, Tukey HSD test [1] and Nemenyi test [32] are performed between every two algorithms. The significance level of all tests is set to $\alpha = 0.05$. If $p-value \leq 0.05$ exists, then we can hold that these ten algorithms or the corresponding two algorithms are not equivalent. According to Table 10, we can easily find that there are significant differences between these ten algorithms. Tables 11 and 12 are the two statistical tests' $p-value$ results of the pairs of algorithms with significant differences, respectively, and the best one of each pair is emphasized in bold.

From these tables, we can get several points:

- HSMOGA performs well in terms of kappa coefficient, accuracy, and G-mean. HSMOGA is significantly better than other nine algorithms for the data sets tested. It achieved the highest kappa coefficient on 12 of 17 data sets and it is almost the second or the third highest on the other 5 data sets. Although we did not optimize the model based on accuracy and G-mean, HSMOGA's accuracy and G-mean is still the highest overall. According to Friedman's ranks, HSMOGA is the best one in terms of all three metrics, and far exceeds the second place.
- The remaining nine algorithms have no significant difference with each other for the data sets tested. Among them, both IGDFS and FSBOGA performed moderately well, and IGDFS ranked second, which indicates the advantages of GA-based feature selection algorithms. Possible reasons why their results are worse than those of HSMOGA are as follows. The maximum number of generations or the population size we used are much smaller than those in [24] and [6]. IGDFS and

Table 6 Kappa coefficient results (%)

Dataset	mRMR	ReliefF	MDG	RFE	Boruta	CFR	RRSS	IGDFS	FSBOGA	HSMOGA
CMC	17.68	22.49	23.57	22.33	19.98	22.58	23.69	23.25	23.23	**24.16**
Thoracic	-0.10	1.69	1.72	2.12	0.32	0.96	1.88	3.49	0.21	**3.69**
Online	58.77	59.13	59.63	58.40	55.91	59.60	59.28	58.40	58.77	**60.15**
SPECT	39.19	41.09	37.82	41.50	32.64	41.99	41.20	39.43	43.50	**49.06**
Default	37.32	37.35	37.12	36.97	36.25	36.81	36.91	36.83	36.29	**37.49**
Dermatology	96.56	97.19	96.76	97.05	96.99	96.84	96.99	96.63	97.20	**97.26**
SPECTF	26.89	28.93	26.91	25.44	26.17	22.24	**29.97**	26.13	22.82	29.34
Spambase	89.76	89.34	89.66	89.90	89.07	89.95	89.71	89.44	89.62	**90.63**
Connectionist	65.38	64.59	65.08	65.16	58.29	64.44	59.91	**66.97**	65.10	66.19
Quality	**93.32**	93.19	92.78	92.67	93.11	92.35	92.15	92.80	93.20	93.20
Libras	80.61	80.65	79.30	80.57	80.51	80.67	80.53	79.95	80.61	**80.73**
Musk1	77.86	76.94	77.88	76.14	76.14	76.82	77.11	**78.77**	74.94	78.11
Musk2	90.60	90.64	90.69	90.82	90.76	90.65	90.08	91.02	**91.43**	90.87
LSVT	64.01	58.81	57.00	61.51	61.43	61.13	64.76	63.76	61.78	**65.15**
Gastro1	38.42	40.02	38.68	38.28	39.04	30.91	32.89	32.45	40.65	**43.31**
Gastro2	41.90	35.90	33.10	35.40	42.97	37.61	36.11	39.01	35.27	**44.99**
Internet	91.14	90.28	**91.90**	90.94	88.53	91.14	91.04	91.14	89.41	**91.90**

Table 7 Accuracy results (%)

Dataset	mRMR	ReliefF	MDG	RFE	Boruta	CFR	RRSS	IGDFS	FSBOGA	HSMOGA
CMC	78.22	**78.32**	77.76	77.42	77.91	77.91	78.11	77.76	78.21	78.11
Thoracic	84.21	84.00	84.26	83.15	83.91	82.89	**84.30**	83.32	84.04	84.26
Online	90.04	90.11	90.28	90.05	89.45	**90.37**	90.21	90.14	90.33	**90.37**
SPECT	81.43	81.88	80.67	82.92	79.63	81.95	81.50	80.52	82.40	**84.56**
Default	81.77	81.76	81.82	81.75	**81.90**	81.64	**81.90**	81.84	81.69	81.79
Dermatology	97.26	97.77	97.43	97.65	97.61	97.49	97.61	97.32	97.78	**97.82**
SPECTF	80.08	80.19	79.39	79.56	79.03	79.40	**81.21**	79.41	79.41	80.89
Spambase	95.18	94.94	95.10	95.20	94.81	95.22	95.16	94.98	95.07	**95.67**
Connectionist	82.89	82.49	82.76	82.79	79.40	82.40	80.18	**83.65**	82.71	83.28
Quality	**95.55**	95.46	95.19	95.12	95.41	94.90	94.77	95.21	95.47	95.47
Libras	81.90	81.94	80.68	81.87	81.81	81.96	81.83	81.30	81.90	**82.02**
Musk1	89.20	88.78	89.20	88.36	88.37	88.70	88.79	**89.64**	87.81	89.34
Musk2	97.68	97.68	97.70	97.73	97.71	97.68	97.55	97.77	**97.91**	97.74
LSVT	84.76	82.81	82.07	83.81	83.51	83.77	85.10	84.96	83.92	**85.35**
Gastro1	65.13	64.98	64.17	64.37	64.42	61.17	61.32	62.05	65.92	**67.03**
Gastro2	**67.80**	63.60	62.28	63.23	66.38	65.28	64.35	65.87	63.75	67.60
Internet	97.66	97.45	**97.88**	97.66	97.03	97.66	97.66	97.66	97.23	**97.88**

Table 8 G-mean results(%)

Dataset	mRMR	ReliefF	MDG	RFE	Boruta	CFR	RRSS	IGDFS	FSBOGA	HSMOGA
CMC	40.67	47.19	49.95	49.07	44.73	48.32	49.27	49.60	47.79	**50.87**
Thoracic	6.93	12.03	13.23	14.51	9.21	14.22	14.11	**15.29**	7.89	15.09
Online	75.12	75.13	75.23	74.33	72.95	75.31	75.19	73.98	73.33	**75.56**
SPECT	63.88	65.39	63.84	64.01	57.79	66.53	65.38	66.17	67.59	**69.09**
Default	59.29	59.32	58.85	58.84	56.99	58.90	57.69	58.29	58.11	**59.63**
Dermatology	96.23	96.71	96.45	96.52	96.54	96.25	96.51	96.01	**96.85**	96.69
SPECTF	50.96	51.90	52.26	48.83	52.68	45.56	**52.95**	51.92	47.82	52.79
Spambase	94.78	94.53	94.61	94.80	94.35	94.83	94.77	94.60	94.70	**95.36**
Connectionist	82.00	81.75	81.94	81.95	78.58	81.71	79.40	**83.04**	82.23	82.71
Quality	**95.43**	95.36	95.08	94.95	95.25	94.68	94.68	95.09	95.35	95.35
Libras	78.24	76.93	76.51	78.49	77.96	78.45	77.89	77.17	70.40	**78.55**
Musk1	88.62	88.04	88.67	87.75	87.69	88.07	88.25	**89.13**	87.14	88.69
Musk2	92.76	92.90	92.86	93.06	92.97	92.98	92.50	93.63	**94.35**	93.11
LSVT	79.71	76.37	75.17	77.78	78.06	77.46	79.77	78.54	77.52	**80.79**
Gastro1	18.50	37.91	35.30	31.06	38.32	18.05	35.52	24.00	37.80	**38.45**
Gastro2	29.16	26.46	25.89	30.20	44.51	14.28	26.85	26.05	17.12	**45.91**
Internet	94.33	93.64	**94.45**	93.17	92.22	94.33	93.75	94.33	92.93	**94.45**

Table 9 Friedman's ranks

Evaluation metric	mRMR	ReliefF	MDG	RFE	Boruta	CFR	RRSS	IGDFS	FSBOGA	HSMOGA
Kappa coefficient	5.47	5.65	5.82	6.00	7.47	6.18	5.65	5.35	5.71	**1.35**
Accuracy	4.59	5.71	6.41	6.12	7.12	6.35	5.35	5.53	5.12	**1.94**
G-mean	5.94	5.81	6.25	5.44	6.75	6.13	5.75	5.13	6.19	**1.63**

Table 10 Results of ANOVA

	Df	Sum Sq	Mean Sq	F value	p-value
Groups	9	373.8	41.54	6.399	9.72×10^{-8}
Residuals	160	1038.5	6.49		

Table 11 p-values of Tukey HSD test on the pairs of algorithms which have significant differences

Algorithm	p-value
HSMOGA vs. mRMR	2.23×10^{-4}
HSMOGA vs. ReliefF	9.38×10^{-5}
HSMOGA vs. MDG	3.83×10^{-5}
HSMOGA vs. RFE	1.52×10^{-5}
HSMOGA vs. Boruta	2.97×10^{-9}
HSMOGA vs. CFR	5.9×10^{-6}
HSMOGA vs. RRSS	9.38×10^{-5}
HSMOGA vs. IGDFS	3.92×10^{-4}
HSMOGA vs. FSBOGA	6.98×10^{-5}

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

Algorithm	p-value
HSMOGA vs. mRMR	1.34×10^{-3}
HSMOGA vs. ReliefF	6.2×10^{-4}
HSMOGA vs. MDG	2.8×10^{-4}
HSMOGA vs. RFE	1.2×10^{-4}
HSMOGA vs. Boruta	3.4×10^{-8}
HSMOGA vs. CFR	5.2×10^{-5}
HSMOGA vs. RRSS	6.4×10^{-4}
HSMOGA vs. IGDFS	2.18×10^{-3}
HSMOGA vs. FSBOGA	4.9×10^{-4}

Table 13 Time cost of IGDFS, FSBOGA, and HSMOGA on 'Online', 'Default', 'Musk2', and 'Internet' data sets (minute)

	'Online'			'Default'			'Musk2'			'Internet'		
G	IGDFS	FSBOGA	HSMOGA	IGDFS	FSBOGA	HSMOGA	IGDFS	FSBOGA	HSMOGA	IGDFS	FSBOGA	HSMOGA
$G = 10$	76.44	84.37	4.25	383.40	445.94	39.69	177.40	218.71	21.37	942.48	1401.42	494.16
$G = 20$	150.68	162.83	4.25	750.14	860.48	39.70	375.17	422.95	21.38	1835.89	2719.95	501.51
$G = 30$	225.99	241.74	4.25	1121.65	1278.65	39.70	560.43	632.83	21.39	2715.37	4039.74	502.02
$G = 40$	302.05	321.12	4.25	1483.32	1699.14	39.70	741.23	837.82	21.39	3603.18	5357.88	507.75
$G = 50$	377.68	402.15	4.25	1851.31	2112.72	39.70	948.10	1045.80	21.41	4481.85	6662.81	508.60
$G = 60$	451.06	478.36	4.26	2219.60	2531.32	39.70	1116.94	1251.25	21.42	5374.72	7975.13	507.52
$G = 70$	526.09	556.18	4.26	2599.21	2950.38	39.70	1327.43	1460.35	21.42	6254.29	9299.68	503.38
$G = 80$	600.30	635.60	4.26	2970.08	3361.71	39.70	1511.04	1679.19	21.43	7142.69	10628.18	507.78
$G = 90$	676.31	715.57	4.26	3353.36	3774.19	39.70	1682.78	1878.98	21.43	8023.63	11954.49	504.32
$G = 100$	750.98	797.00	4.26	3733.57	4190.51	39.71	1861.30	2082.47	21.44	8910.99	13297.90	510.79

FSBOGA did not take feature interaction into account. They have no means to control the size of the feature subset.

– For the experiments performed in this paper, HSMOGA is better than IGDFS and FSBOGA when using the same maximum number of generations and the population size. It indicates that HSMOGA works better when using small G and M. This is precisely because when generating the initial population, HSMOGA makes full use of the knowledge of data set through controls the occurrence of each feature according to the hybrid weights and the control term $bias$, which makes the results converge faster and performs better.

In experiment five, we compared the time cost of IGDFS, FSBOGA and HSMOGA. Table 13 shows the time cost of the two algorithms when maximum number of generations $G = 10, 20, \cdots, 100$. Note that, HSMOGA includes the knowledge reserve part, MOGA part, and the final feature subset selection part.

By inspection, we can easily find that HSMOGA has a significant advantage over IGDFS and FSBOGA in terms of time cost for the data sets tested. Moreover, as G increases, time cost of HSMOGA does not increase much. For example, when dealing with 'Musk2' data set, among the whole HSMOGA, the knowledge reserve part costs nearly 16.8 minutes, the final feature subset selection part costs nearly 4.5 minutes, and time cost of MOGA part changes from 6.9 seconds to 10.8 seconds as G changed from 10 to 100. This is because the three fitness functions we proposed are all independent of the classifier, and we precalculated the knowledge or information required through the knowledge reserve step. It indicates that HSMOGA not only has the better results, but also run faster. Most parameters of HSMOGA, such as G and $bias$, can be tuned at high speed.

5 Conclusion

We proposed a novel MOGA-based feature selection algorithm, i.e., HSMOGA. It contains three new fitness functions $F_1(\cdot), F_2(\cdot), F_3(\cdot)$ and a new Pareto-based ranking function $RANK(\cdot)$. $F_1(\cdot)$ is the average hybrid weight according to a new hybrid filter which performs more balanced on different data sets. $F_2(\cdot)$ is the average SCom of every two features which is a measure of feature interaction strength. $F_2(\cdot)$ solves the problem that [50] needs to determine the threshold of SCom artificially. $F_3(\cdot)$ is the limitation of $F_1(\cdot)$ and $F_2(\cdot)$ through ensuring that the size of feature subset is close to an expected size. Compared to traditional fast non-dominated sorting and the crowding-distance estimation, the new ranking function $RANK(\cdot)$ is more suitable for the three fitness functions proposed in this paper. When generate the initial population, HSMOGA makes full use of the knowledge of data set instead of some random or pseudo-random pattern generator. The probability of the occurrence of each feature is derived from its hybrid weight and a control term. We proposed a new step called knowledge reserve. It calculates features' hybrid weights and their SComs before any other steps of HSMOGA, which makes other steps including initial population generation, the calculation of fitness functions, and the parameter's tuning run faster.

We designed three experiments to objectively evaluate the effectiveness of the two of three fitness functions and the Pareto-based ranking function proposed in this paper, and other two experiments to objectively compare and assess the relative performance of HSMOGA in terms of classification performance and time cost. According to the experimental results, the fitness functions and ranking function proposed in this paper are efficient, and HSMOGA outperforms other nine state-of-art feature selection algorithms in terms of kappa coefficient, accuracy, and G-mean for the data sets tested. Moreover, HSMOGA has a significant time-saving advantage over other GA-based feature selection algorithms, as its three fitness functions are classifier-independent and their knowledge needed is calculated before, and even if the maximum number of generations increases a lot, the time cost will not increase much.

In the future, we will try to construct a better hybrid filter and combine HSMOGA with ensemble methods and Neural Networks.

Acknowledgements Thanks to the data sets provided by the UCI repository. The Thoracic data set was from [54]. The Online data set was from [38]. The Default data set was from [47]. The Quality data set was from [11]. The LSVT data set was from [42]. The Gastro1 and Gastro2 data sets were from [30].

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Abdi H, Williams LJ (2010) Tukey's honestly significant difference (hsd) test. Encyclopedia of Research Design 3:583–585
2. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
3. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
4. Colas C, Madhavan V, Huizinga J, Clune J (2020) Scaling map-elites to deep neuroevolution. In: GECCO, vol 2020, pp 67–75
5. Cover TM, Thomas JA (2012) Elements of information theory. Wiley, Berlin
6. Das AK, Pati SK, Ghosh A (2019) Relevant feature selection and ensemble classifier design using bi-objective genetic algorithm. Knowl Inf Syst 62(2):423–455
7. Das S (2001) Filters, wrappers and a boosting-based hybrid for feature selection. In: ICML 2001, vol 1, pp 74–81

8. Davis L (1991) Handbook of genetic algorithms. CUMINCAD
9. Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: PPSN VI. Springer, pp 849–858
10. Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
11. Fernandes K, Cardoso JS, Fernandes J (2017) Transfer learning with partial observability applied to cervical cancer screening. In: IbPRIA 2017. Springer, pp 243–250
12. Fioravanzo S, Iacca G (2019) Evaluating map-elites on constrained optimization problems. arXiv:190200703
13. 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
14. Gao W, Hu L, Zhang P, He J (2018) Feature selection considering the composition of feature relevancy. Pattern Recogn Lett 112:70–74
15. Ghareb AS, Bakar AA, Hamdan AR (2016) Hybrid feature selection based on enhanced genetic algorithm for text categorization. Expert Syst Appl 49:31–47
16. Ghosh M, Guha R, Sarkar R, Abraham A (2019) A wrapper-filter feature selection technique based on ant colony optimization. Neural Comput Appl 32(12):7839–7857
17. Gonzalez-Abril L, Cuberos FJ, Velasco F, Ortega JA (2009) Ameva: an autonomous discretization algorithm. EXPERT SYST APPL 36(3):5327–5332
18. González-López J, Ventura S, Cano A (2019) Distributed selection of continuous features in multilabel classification using mutual information. IEEE T Neur Net Lear
19. González-López J, Ventura S, Cano A (2020) Distributed multi-label feature selection using individual mutual information measures. Knowl-Based Syst 188:105052
20. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(6):1157–1182
21. Hammami M, Bechikh S, Hung CC, Said LB (2019) A multi-objective hybrid filter-wrapper evolutionary approach for feature selection. Memet Comput 11(2):193–208
22. Hsu HH, Hsieh CW, Lu MD (2011) Hybrid feature selection by combining filters and wrappers. Expert Syst Appl 38(7):8144–8150
23. Huang J, Cai Y, Xu X (2007) A hybrid genetic algorithm for feature selection wrapper based on mutual information. Pattern Recogn Lett 28(13):1825–1844
24. Jadhav S, He H, Jenkins K (2018) Information gain directed genetic algorithm wrapper feature selection for credit rating. Appl Soft Comput 69:541–553
25. Jakulin A, Bratko I (2004) Testing the significance of attribute interactions. In: ICML, vol 2004, pp 409–416
26. John GH, Kohavi R, Pfleger K (1994) Irrelevant features and the subset selection problem. In: ML 94. Elsevier, pp 121–129
27. Konak A, Coit DW, Smith AE (2006) Multi-objective optimization using genetic algorithms: a tutorial. Reliab Eng Syst Safe 91(9):992–1007
28. Kononenko I (1994) Estimating attributes: analysis and extensions of relief. In: ECML-94, pp 171–182
29. Kursa MB, Rudnicki WR, et al. (2010) Feature selection with the boruta package. J STAT SOFTW 36(11):1–13
30. Mesejo P, Pizarro D, Abergel A, Rouquette O, Beorchia S, Poincloux L, Bartoli A (2016) Computer-aided classification of gastrointestinal lesions in regular colonoscopy. IEEE T Med Imaging 35(9):2051–2063
31. Mouret JB, Clune J (2015) Illuminating search spaces by mapping elites. arXiv:150404909
32. Nemenyi P (1963) Distribution-eree multiple comparison. PhD thesis
33. Ng AY (2004) Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In: ICML 2004. ACM, p 78
34. Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE T Pattern Anal 27(8):1226–1238
35. Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106
36. Quinlan JR (2014) C4. 5: programs for machine learning. Elsevier
37. Quinonez B, Pinto-Roa DP, García-Torres M, García-Díaz ME, Núnez-Castillo C, Divina F (2019) Map-elites algorithm for features selection problem. In: AMW
38. Sakar CO, Polat SO, Katircioglu M, Kastro Y (2019) Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks. Neural Comput Appl 31(10):6893–6908
39. Shieh MD, Yang CC (2008) Multiclass SVM-RFE for product form feature selection. Expert Syst Appl 35(1):531–541
40. Song X, Zhang Y, Guo Y, Sun X, Wang Y (2020) Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. IEEE T Evolut Comput 24(5):882–895
41. Tang X, Dai Y, Xiang Y (2019) Feature selection based on feature interactions with application to text categorization. Expert Syst Appl 120:207–216
42. Tsanas A, Little MA, Fox C, Ramig LO (2013) Objective automatic assessment of rehabilitative speech treatment in Parkinson's disease. IEEE T Neur Sys Reh 22(1):181–190
43. Wang G, Song Q (2012) Selecting feature subset via constraint association rules. In: PAKDD, vol 2012, pp 304–321
44. 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
45. Wang Y, Feng L (2018) Hybrid feature selection using component co-occurrence based feature relevance measurement. Expert Syst Appl 102:83–99
46. Wang Y, Feng L (2019) A new hybrid feature selection based on multi-filter weights and multi-feature weights. Appl Intell 49:4033–4057
47. Yeh IC, Lien Ch (2009) The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Syst Appl 36(2):2473–2480
48. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5(12):1205–1224
49. Zeng Z, Zhang H, Zhang R, Yin C (2015) A novel feature selection method considering feature interaction. Pattern Recogn 48(8):2656–2666
50. Zhang R, Zhang Z (2020) Feature selection with symmetrical complementary coefficient for quantifying feature interactions. Appl Intell 50:101–118
51. Zhang Y, Gong DW, Cheng J (2015) Multi-objective particle swarm optimization approach for cost-based feature selection in classification. IEEE ACM T Comput Bi 14(1):64–75
52. Zhang Y, Cheng S, Shi Y, wei Gong D, Zhao X (2019) Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm. Expert Syst Appl 137:46–58
53. Zhang Y, Gong D, Gao X, Tian T, Sun X (2020) Binary differential evolution with self-learning for multi-objective feature selection. Inform Sciences 507:67–85
54. Zikeba M, Tomczak JM, Lubicz M, Świkatek J (2014) Boosted svm for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. Appl Soft Comput 14:99–108