Journal of Mathematical Sciences: Advances and Applications
Volume 37, 2016, Pages 51-78
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/jmsaa_7100121627

A HYBRID FEATURE SELECTION METHOD BASED ON FISHER SCORE AND GENETIC ALGORITHM

MI ZHOU

School of Information Science and Technology
Jinan University
Guangzhou 510632
P. R. China
e-mail: 18819432258@163.com

Abstract

Fisher score and genetic algorithm are widely used for feature selection. However, some redundant features will be selected by Fisher score, and the convergence properties may be worse if the initial population of genetic algorithm is generated by a random manner. To improve the performance of feature selection by Fisher score and genetic algorithm, we propose a hybrid feature selection method, which merging the advantages of Fisher score and genetic algorithm together. It aims at utilizing the features' Fisher score to generate the initial population of genetic algorithm. To begin with, the Fisher scores of all the features will be mapped into a specific interval by a linear function, and then the rescaled Fisher scores will be utilized to generate the initial population of genetic algorithm. Finally, the initial population will be used in the subsequent procedure of genetic algorithm to perform feature selection with elitist strategy for reference. In this paper, we choose four data sets of Sonar, WDBC, Arrhythmia, and Hepatitis to test the performance of our algorithm. Feature subsets of the four data sets will be selected by our algorithm, and then the dimensionality of data sets will be reduced according to the selected feature subsets, respectively. 1-NN classifier is used to classify the dimensionality reduced data sets, and respectively, achieving the classification

2010 Mathematics Subject Classification: 68T10.
Keywords and phrases: feature selection, Fisher score, genetic algorithm, elitist strategy.
Received January 21, 2016

accuracy of 72.36%, 95.64%, 72.04%, and 87.83% with ten-fold cross validation method. The experiment results show that, compared to the performance of Fisher score, genetic algorithm and Fisher score genetic algorithm, our algorithm is fit for eliminate redundant features, and can select discriminative features. Above all, our method is effective in feature selection.

1. Introduction

In many research fields such as machine learning, pattern recognition, biomedicine and so on, researchers need to analysis vast amounts of data, which is described by high-dimensional vectors. Every component of a vector represents a kind of feature of the high-dimensional data. In reality, there are some meaningless and redundant features in the high-dimensional vectors. To improve the efficiency of subsequent procedure (such as classification and prediction) and save the dedicated space of device, we need to eliminate the meaningless and/or redundant features from those high-dimensional vectors, and select the features which are most relevant for our problems to form the optimal feature subset, the procedure is called to be feature selection, or feature subset selection (FSS) [1, 2]. Superficially, exhaustive search can select the optimal feature subset, however, it is a NP-hard problem that we adopt exhaustive search to select features, because each dataset has 2^n feature subsets. In the past decades, a number of feature selection methods have been proposed, and they are generally divided into two categories: filter-based methods and wrapper-based methods.

Filter-based methods rank the features according to a predefined criterion independent of the actual generalization performance of the learning machine, and select those features with high ranking scores, so a faster speed can usually be obtained. Mutual information (MI) [3]; Fisher score (FS) [4]; Relieff [5]; Laplacian score [6]; Hilbert Schmidt independence criterion (HSIC) [7]; and Trace ratio criterion [8] or so can be regarded as the criterion for filter-based feature selection, among which Fisher score is one of the most widely used criteria for filter-based feature selection due to its general good performance. The specific process of feature selection method based on Fisher score is like this: calculating

the Fisher scores of all the features, and for a given threshold θ, feature $\boldsymbol{f}_i$ is selected if $F(\boldsymbol{f}_i) > \theta$, otherwise, if $F(\boldsymbol{f}_i) \leq \theta$, feature $\boldsymbol{f}_i$ will not be selected. Selecting features by Fisher score can improve the accuracy of subsequent procedure (such as classification and prediction), and the process is simple, feasible and time saving. However, there are also some problems about Fisher score: (1) How to set the threshold θ such that an optimal feature subset is selected? (2) We cannot eliminate redundant features via Fisher score. For example, both the scores of feature $\boldsymbol{f}_i$ and feature $\boldsymbol{f}_j$ are very high, but they are highly correlated. In this case, the filter-based algorithm will select both $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, while either $\boldsymbol{f}_i$ or $\boldsymbol{f}_j$ should be eliminated without any loss in the subsequent classification performance. (3) Since the filter-based algorithm computes the Fisher score of each feature individually, it neglects the combination of features, which means evaluating two or more than two features together. For instance, it could be the case that the scores of feature $\boldsymbol{f}_i$ and feature $\boldsymbol{f}_j$ are both low, but the score of the combination of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ is very high. In this case, the filter-based algorithm will discard both $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, although they should be selected.

Wrapper-based methods use a predictor as a black box and the predictor performance as the objective function to evaluate the feature subset. A wide range of wrapper-based methods have been used including forward selection [9]; backward elimination [10]; hill-climbing [11]; branch and bound algorithms [12]; simulated annealing and genetic algorithms (GAs) [13, 14, 15, 16]. Kudo and Sklansky [17] made a comparison among many of the feature selection algorithms and explicitly recommended that GAs should be used for large-scale problems with more than 50 candidate variables. In many practical applications, the genetic algorithm can be reformed in different ways according to different situations. However, when the GA is adopted to select features, it is unreasonable that the initial population is generated by a random manner. Because every feature has the same chance to be selected into the initial population, the convergence properties of GA may be worse.

Yet, we just want to select the "best features" rapidly. Since our purpose is to select the "best features" rapidly, why not select the features of high Fisher score by setting a threshold θ to generate the initial population, and then adopt the genetic algorithm to achieve the optimal feature subset? It is reasonable to a extent, however, features of lowest Fisher score may have little chance to be selected.

To overcome the above problems, we consider that if the features of higher Fisher score have more probabilities to be selected to generate the initial population than features of lower Fisher score, and some chromosomes of best fitness are reserved to the next generation without any genetic operations, the classification accuracy after feature selection will be improved. The basic idea of this paper is just about this. In this paper, we present a hybrid feature selection method based on Fisher score and genetic algorithm. Our method utilize features' rescaled Fisher score to generate the initial population of genetic algorithm. In the first place, the Fisher scores of all the features will be mapped into a specific interval by a linear function, and then we utilize the rescaled Fisher scores to generate the initial population of genetic algorithm. Finally, the initial population will be used in the subsequent procedure of genetic algorithm to perform feature selection with elitist strategy for reference. Experiments on four benchmark data sets of Sonar, WDBC, Arrhythmia, and Hepatitis indicate that the proposed method outperforms Fisher score method and genetic algorithm, and it does well in eliminating redundant features. Features selected by our method are more discriminative than that some state of the art feature selection methods.

The remainder of this paper is organized as follows. In Section 2, we briefly review Fisher score and genetic algorithm. We present the hybrid feature selection method based on Fisher score and genetic algorithm in Section 3. The experiments on four data sets of Sonar, WDBC, Arrhythmia, and Hepatitis are demonstrated in Section 4. Finally, we draw a conclusion in Section 5.

2. A Brief Review of Fisher Score and Genetic Algorithm

2.1. Fisher score

Given dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, where $\boldsymbol{x}_i \in R^M$ and $y_i \in \{1, 2, \cdots, c\}$ represents the class which $\boldsymbol{x}_i$ belongs to. N and M are the number of samples and features, respectively. $\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_M$ denote the M features. Redundant and meaningless features may be included in the feature set when M is large, thus we want to select a subset from the M features in order to make the data set more productive. For example, we hope the selected features improve the classification accuracy for a classification problem.

If a feature is discriminative, the between-class variance of the feature should be large, while the within-class variance of the feature should be small. Let $\mu_{\boldsymbol{f}_i}$ denote the mean of feature $\boldsymbol{f}_i$, and $\mu_{\boldsymbol{f}_i}^k$ be the mean of feature $\boldsymbol{f}_i$ in class k. Fisher score [4] is defined as follows:

$$F(\boldsymbol{f}_i) = \frac{\sum_{k=1}^{c} n_k (\mu_{\boldsymbol{f}_i}^k - \mu_{\boldsymbol{f}_i})^2}{\sum_{k=1}^{c} \sum_{y_j = k} (\boldsymbol{f}_{j,i} - \mu_{\boldsymbol{f}_i}^k)^2}, \tag{2.1}$$

where n_k is the number of samples in class k, and $\boldsymbol{f}_{j,i}$ is the value of feature $\boldsymbol{f}_i$ in sample $\boldsymbol{x}_j$. From the Equation (2.1), we can conclude that features with higher Fisher score are more discriminative to some extent.

2.2. Genetic algorithm

The genetic algorithm (GA) [18] was originally developed by Holland [3] and his associates. The basic idea of GA is to construct a fitness function according to the objective function of the problem, and utilize the fitness function to evaluate a set of candidate solutions (every solution responds to a chromosome) called a population. Based on the Darwinian

principle of 'survival of the fittest', the GA obtains the optimal solution after a series of iterative computations. GA generates successive populations of alternate solutions that are represented by a chromosome, i.e., a solution to the problem, until acceptable result is obtained. Genetic algorithm is a method that based on group optimization, as described below:

Step 1: Generating initial population. The generating of initial population is random, and the specific way relays on the encoding of chromosomes. The 0-1 codes are generated as below: let $\boldsymbol{I}^{0,l} \in R^M$ denotes the l-th chromosome of the initial population, for $\boldsymbol{I}^{0,l} = \left(I_1^{0,l}, I_2^{0,l}, \cdots, I_M^{0,l}\right)$, generating a random floating number $\xi_i^l \in U(0, 1)$, if $\xi_i^l > 0.5$, then $\boldsymbol{I}_i^{0,l} = 1$; otherwise, if $\xi_i^l \leq 0.5$, then $I_i^{0,l} = 0\,(i = 1, 2, \cdots, M)$. The setting of population size N relays on the computing ability of a computer and the complexity of a algorithm. Generally, N is set to be 100-1000. It is the 0-th generation at present.

Step 2: Estimating stopping criterion. We adopt the number of maximum generation as the stopping criterion. The algorithm stops when the number of generations reaches the preset maximum generation.

Step 3: Computing the fitness. Choosing a proper fitness function $f(\boldsymbol{I}^{g,l})$ based on the objective function of optimizing issue, and calculating the fitness of every chromosome, respectively. The higher fitness a chromosome have, the more probability it will be selected to take part in the subsequent genetic operation.

Step 4: Selecting the chromosome with high fitness. The roulette-wheel selection scheme will be used to select N chromosomes to generate a new population.

Step 5: Genetic operation. The selected chromosomes need to undergo genetic operations, such as crossover and mutation. (1) Crossover: On the basis of the crossover rate P_c, the crossover operation generates new

chromosomes (offspring) out of their parents. (2) Mutation: The mutation is applied to the offspring. According to the mutation rate P_m, we perform the 1-0 and 0-1 conversions.

Step 6: Updating the population. We obtain the offspring after the crossover and mutation process, and the algorithm enters to the next generation. Return to Step 2.

3. A Hybrid Feature Selection Method Based on Fisher Score and Genetic Algorithm

To overcome the problems of Fisher score and genetic algorithm, we propose a hybrid feature selection method based on Fisher score and genetic algorithm. We calculate the Fisher score $[F(\boldsymbol{f}_1), F(\boldsymbol{f}_2), \cdots, F(\boldsymbol{f}_M)]$ of every feature, and then map all the Fisher scores into a closed interval $[\varepsilon, 1-\varepsilon]$ by a linear function, where ε is a relatively small positive real value. We utilize the rescaled Fisher scores to generate the initial population of genetic algorithm, and make a little modification in the subsequent operator of genetic algorithm. We aim to select a more optimal subset of features than Fisher score and genetic algorithm.

3.1. Generating initial population

Suppose we have Fisher scores $[F(\boldsymbol{f}_1), F(\boldsymbol{f}_2), \cdots, F(\boldsymbol{f}_M)]$ for all features, utilize a linear function $L(F(\boldsymbol{f}_i))$ to rescale $F(\boldsymbol{f}_i)$ into range $[\varepsilon, 1-\varepsilon]$ for $i = 1, 2, \cdots, M$, and the linear function $L(F(\boldsymbol{f}_i))$ is given as below:

$$L(F(\boldsymbol{f}_i)) = \frac{1-2\varepsilon}{F_{\max} - F_{\min}} F(\boldsymbol{f}_i) + \frac{\varepsilon(F_{\max} + F_{\min}) - F_{\min}}{F_{\max} - F_{\min}}, \qquad (3.1)$$

where ε is a relatively small positive real value, $0 < \varepsilon < 0.1$, and we ascertain the value of ε according to the practical situation of problems. $F_{\max} = \max\{F(\boldsymbol{f}_1), F(\boldsymbol{f}_2), \cdots, F(\boldsymbol{f}_M)\}$, $F_{\min} = \min\{F(\boldsymbol{f}_1), F(\boldsymbol{f}_2), \cdots, F(\boldsymbol{f}_M)\}$.

We adopt 0-1 coding scheme to represent individual in every generation. Let $\boldsymbol{I}^{g,l} \in R^M$ denote the l-th individual in the g-th generation. Then $\boldsymbol{I}^{g,l} = \left(I_1^{g,l}, I_2^{g,l}, \cdots, I_M^{g,l}\right)$, where $\boldsymbol{I}_i^{g,l} = 1$ if feature $\boldsymbol{f}_i$ is selected, otherwise, $\boldsymbol{I}_i^{g,l} = 0$, for $i = 1, 2, \cdots, M$.

Unlike genetic algorithm, we mainly utilize the rescaled Fisher scores $L(F(\boldsymbol{f}_i))$ to generate initial population in GA algorithm. Generating M real value of η_i^l by computer with a random manner, where $\eta_i^l \in U(0, 1)$, if $L(F(\boldsymbol{f}_i)) > \eta_i^1$, then $I_i^{0,l} = 1$; otherwise, if $L(F(\boldsymbol{f}_i)) \le \eta_i^l$, then $I_i^{0,l} = 0$, for $i = 1, 2, \cdots, M$.

3.2. Fitness function

The selection of a fitness function has a direct influence on the convergence rate of GA, and it also concerns whether the optimal solution will be found. Generally speaking, the fitness function relates to the objective function of a optimization problem. Therefore, the fitness function denoted by $f(\boldsymbol{I}^{g,l})$ is defined as classification accuracy of 1-NN classifier with 10-folds cross validation, and there is an additional term of $\lambda \cdot \sum_{i=1}^{M} I_i^{g,l}$ to penalize the number of features. Then $f(\boldsymbol{I}^{g,l})$ can be expressed as:

$$f(\boldsymbol{I}^{g,l}) = accuracy(\boldsymbol{I}^{g,l}) - \lambda \cdot \sum_{i=1}^{M} I_i^{g,l}, \tag{3.2}$$

where λ is a positive real parameter.

3.3. Elitism strategy

In traditional genetic algorithm, after a series of selection, crossover and mutation procedures, we get N (where N means population size) new chromosomes, and the N new chromosomes will replace all of their parents to form the next generation. Different from the traditional GA,

elitism strategy [19] considers two situations: if the new chromosome with best fitness is superior to the best fitness individual in the last population, all the individuals in the last population will be replaced; otherwise, the most inferior new chromosome is replaced by the best chromosome in the last generation. Elitism strategy guarantees the consistency of the optimal solution at present and in history, and makes the algorithm to be global convergence [20].

We take the idea of elitism strategy for reference, selecting $(N - m)$ chromosomes from the last population to participate in genetic operation, and retain those chromosomes corresponding to top m fitness values in the next generation without participating in any genetic operations. The specific operations are exemplified in Figure 3.1. This operator does not obstruct the creation of new chromosomes, and ensures the best fitness chromosome in the next generation never inferior to the one in the last generation, the process of evolution becomes a optimizing process.

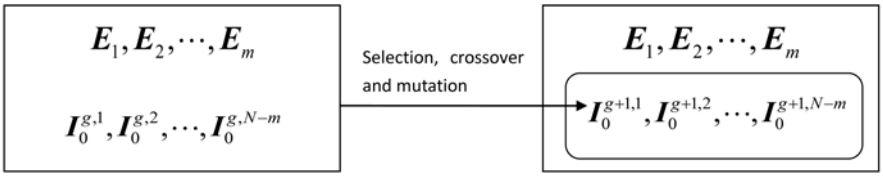


Figure 3.1. The elitism strategy operation in this paper.

3.4. Selection strategy

Let $P(\boldsymbol{I}^{g,l})$ denotes the probability that $\boldsymbol{I}^{g,l}$ is selected. Then we define $P(\boldsymbol{I}^{g,l})$ as follows:

$$P(\boldsymbol{I}^{g,l}) = \frac{f(\boldsymbol{I}^{g,l})}{\sum_{l=1}^{N} f(\boldsymbol{I}^{g,l})}, \tag{3.3}$$

where N means population size. Roulette wheel selection scheme is applied to select individuals for genetic operations. Let

$$PP_0 = 0, \tag{3.4}$$

$$PP_l = \sum_{j=1}^{l} P(\boldsymbol{I}^{g,j}). \tag{3.5}$$

Generate a random variable $\xi_k \sim U(0, 1)$, for $k = 1, 2, \cdots, N$; if $PP_{l-1} \leq \xi_k < PP_l$, then the individual $\boldsymbol{I}^{g,l}$ is selected. The selection process is repeated $(N - m)$ times.

3.5. Genetic operator and stopping criterion

Genetic operator mainly includes two procedures: crossover and mutation. The $(N - m)$ selected chromosomes will take part in the crossover and mutation procedure according to the probability of crossover and mutation, respectively. The crossover operation generates two new chromosomes (offspring) out of their parents, and the mutation operation slightly perturbs the offspring. The GA stops when the number of generations reaches the preset maximum generation "Maxiters".

3.5.1. Crossover

The crossover operator is performed according to the crossover probability denoted by P_c. The probability of crossover controls the frequency of crossover operator, and a higher probability is good for open up a new searching area, while a lower probability may lead to a bluntness state of GA algorithm. To get a considerable result, in this paper, we set the crossover probability as a function of the iteration times, and the expression of the crossover probability is given below:

$$P_c = \frac{1}{10 \cdot \ln(Maxiters)} \ln(iters) + 0.7, \tag{3.6}$$

where *iters* denotes the current iteration times and *Maxiters* is the maximum iteration times. Figure 3.2 shows the curve of crossover probability as a function of iteration times in 500 times of iteration.

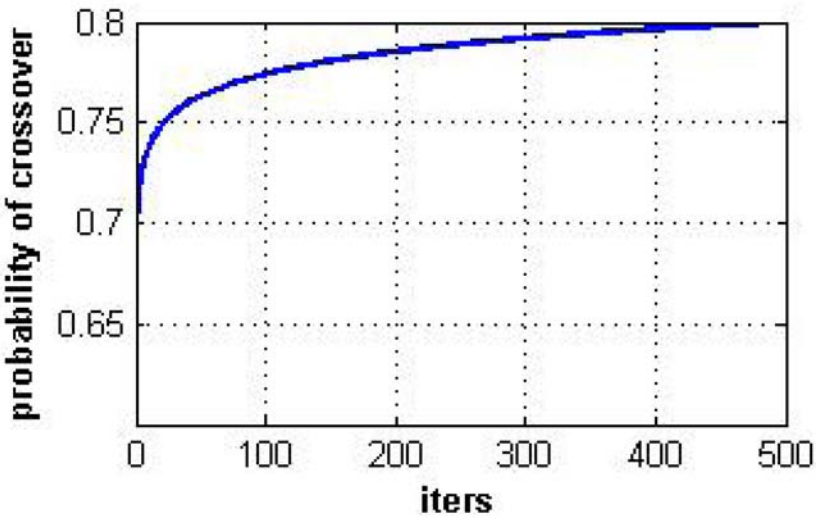


Figure 3.2. P_c changing with iteration times.

We adopt the single-point crossover manner, and select two chromosomes of $\boldsymbol{I}^{g,t} = \left(I_1^{g,t}, I_2^{g,t}, \cdots, I_M^{g,t}\right)$ and $\boldsymbol{I}^{g,t+1} = \left(I_1^{g,t+1}, I_2^{g,t+1}, \cdots, I_M^{g,t+1}\right)$ by Roulette wheel, choosing a cutting-point randomly and exchanging the part on the right of cutting-point to get two new chromosomes of $\boldsymbol{I}^{g+1,t} = \left(I_1^{g+1,t}, I_2^{g+1,t}, \cdots, I_M^{g+1,t}\right)$ and $\boldsymbol{I}^{g+1,t+1} = \left(I_1^{g+1,t+1}, I_2^{g+1,t+1}, \cdots, I_M^{g+1,t+1}\right)$. A specific example of single-point crossover is exemplified in Figure 3.3.

$$\begin{array}{lll} & \textit{cutting - point} & \\ \boldsymbol{I}^{g,t} = 10010|11 & \xrightarrow{\textit{single-point crossover}} & \boldsymbol{I}^{g+1,t} = 10010|01 \\ \boldsymbol{I}^{g,t+1} = 01001|01 & & \boldsymbol{I}^{g+1,t+1} = 01001|11 \end{array}$$

Figure 3.3. A specific example of single-point crossover.

3.5.2. Mutation

The mutation operator is performed according to the mutation probability denoted by P_m. The mainly purpose of mutation is to

maintain the diversities of population. In general, a lower mutation probability means less likely to lose important genes; while a higher mutation probability will lead the algorithm to be a random research process, so we need to carefully control the number of 1-0 and 0-1 conversions. In this paper, we set the mutation probability as a function of the iteration times, and the expression of the crossover probability is given below:

$$P_m = 0.04 \cos\left(\frac{\pi(iters-1)}{2(Maxiters-1)}\right) + 0.01, \tag{3.7}$$

where *iters* denotes the current iteration times and *Maxiters* is the maximum iteration times. Figure 3.4 shows the curve of mutation probability as a function of iteration times in 500 iterations.

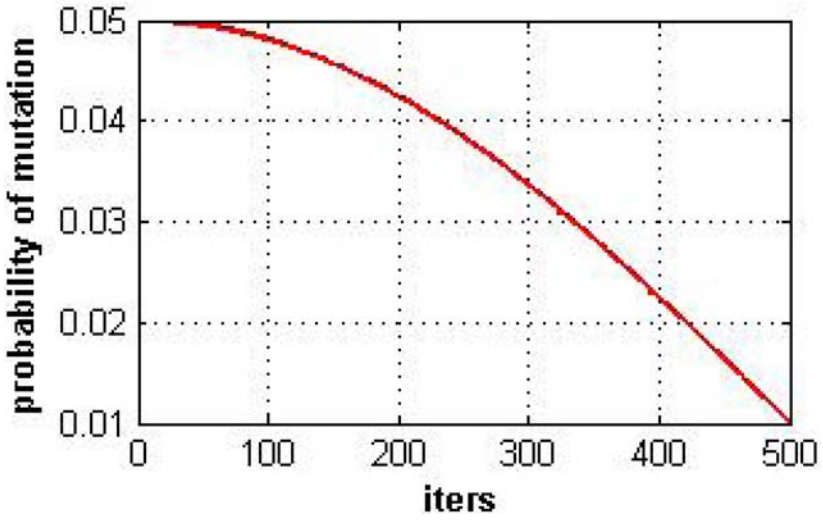


Figure 3.4. P_m changing with iteration times.

Based on the mutation probability, we select three gene sites, and perform the 1-0 and 0-1 conversions. A specific example of mutation is exemplified in Figure 3.5.

0110001011 → 0110110011

Figure 3.5. A specific example of mutation.

4. Experiments

4.1. Datasets

To test the efficiency of our proposed algorithm adequately, we use a subset of UCI machine learning benchmark data set [21], and they are Sonar, WDBC, Arrhythmia, and Hepatitis dataset, respectively. Table 4.1 summarizes the characteristics of the data sets used in our experiments. All datasets are standardized to be zero-mean and normalized by standard deviation for each dimension.

Table 4.1. Description of the data sets used in our experiments

Data sets	Size	Number of features	Classes
Sonar	208	60	2
WDBC	569	30	2
Arrhythmia	452	279	2
Hepatitis	155	19	2

Sonar dataset contains 208 vectors, each vector includes 60 components, which means the dataset has 60 features. We acquire the data from the feedback information of sonar. There are two classes in Sonar dataset, data of rock and data of mineral. After feature selection, 1-nearest neighbour (1-NN) classifier is used for classification, and ascertain whether a vector belongs to mineral according to the result of classification.

WDBC dataset contains 569 vectors, each vector includes 30 components, which means the dataset has 30 features. There are two classes in WDBC dataset, benign breast cancer and malignant breast cancer. After feature selection, 1-nearest neighbour (1-NN) classifier is used for classification, and ascertain whether a vector belongs to benign or malignant according to the result of classification.

Arrhythmia dataset contains 452 vectors, each vector includes 279 components, which means the dataset has 279 features. There are two classes in Arrhythmia dataset, normal heart rate and abnormal heart rate. After feature selection, 1-nearest neighbour (1-NN) classifier is used for classification, and ascertain whether a vector belongs to normal heart rate or abnormal heart rate according to the result of classification.

Hepatitis dataset contains 155 vectors, each vector includes 19 components, which means the dataset has 19 features. There are two classes in Hepatitis dataset, hepatitis patients and healthy person. After feature selection, 1-nearest neighbour (1-NN) classifier is used for classification, and ascertain whether a vector belongs to hepatitis patients or not according to the result of classification.

4.2. Experiments and parameters

In our experiments, we compare the proposed method to the state-of-art feature selection methods: Fisher score (FS), Genetic algorithm (GA), and Fisher score genetic algorithm (FSGA). We perform four experiments on every dataset respectively, and the parameters of 16 experiments are summarized in Table 4.2.

Table 4.2. The parameters in 16 experiments

Data sets	Algorithms	Parameters					
		λ	N	*Maxiters*	θ	ε	m
Sonar	FS	0.01	*	*	*	*	*
	GA	0.01	100	500	*	*	*
	FSGA	0.01	100	500	0.15	*	*
	HFSGA	0.01	100	500	*	0.05	2
WDBC	FS	0.01	*	*	*	*	*
	GA	0.01	100	500	*	*	*
	FSGA	0.01	100	500	1.5	*	*
	HFSGA	0.01	100	500	*	0.05	2
Arrhythmia	FS	0.01	*	*	*	*	*
	GA	0.001	100	500	*	*	*
	FSGA	0.001	100	500	0.5	*	*
	HFSGA	0.001	100	500	*	0.05	2
Hepatitis	FS	0.001	*	*	*	*	*
	GA	0.01	100	1000	*	*	*
	FSGA	0.01	100	1000	0.28	*	*
	HFSGA	0.01	100	1000	*	0.05	2

Experiment 1. Feature selection by Fisher score (FS). Whether a feature is selected or not is just related to the feature's Fisher score and the setting of threshold. When a threshold θ is setting, C_θ features of $F(\boldsymbol{f}_i) > \theta$ will be selected to construct a feature subset A, and then we reduce the dimensionality of original data set according to the selected feature subset A. 1-NN classifier is used to classify the dimensionality reduced data set. To ensure the comparability with other three methods, we also construct a fitness function of $f_0(A)$, which is expressed as the subtraction of the classification accuracy *accuracy*(A) with ten-fold cross validation method and a penalty item of λC_θ. The fitness of a feature subset A is defined as

$$f_0(A) = accuracy(A) - \lambda C_\theta, \tag{4.1}$$

where parameter λ is the same as that in formula (3.2). In the experiment of Sonar, WDBC, and Hepatitis, we set $\lambda = 0.01$, while in the experiment of Arrhythmia data set which has the most features, we set $\lambda = 0.001$. From formula (4.1), we can conclude that the feature subset with higher fitness is better.

Experiment 2. Feature selection by genetic algorithm (GA). When adopt GA to perform feature selection in the four data sets respectively, we set the population size $N = 100$, and $\lambda = 0.01$ in the experiments of Sonar, WDBC and Hepatitis, while $\lambda = 0.001$ in the experiment of Arrhythmia. In the experiments of Sonar, WDBC, and Arrhythmia, we set *Maxiters* = 500, and *Maxiters* = 1000 in the Hepatitis data set experiment.

Experiment 3. Feature selection by Fisher score genetic algorithm (FSGA). The specific process of FSGA is like this: in the first place, set a threshold θ to select features whose $F(\boldsymbol{f}_i) > \theta$, then utilize the selected features to generate initial population of genetic algorithm by random manner (the features whose Fisher score is lower than threshold θ will not be selected to generate initial population), finally, the feature subset is selected by traditional genetic algorithm. We set the threshold to be 0.15, 1.5, 0.5, and 0.28, respectively, in data sets of Sonar, WDBC, Arrhythmia, Hepatitis, and the setting of parameter N, λ, and *Maxiters* are the same as Experiment 2.

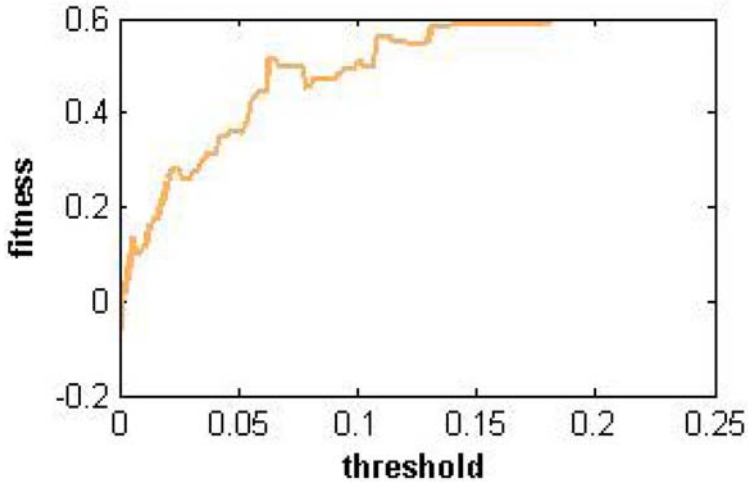
Experiment 4. Feature selection by the proposed hybrid feature selection method based on Fisher score and genetic algorithm (HFSGA). We set $N = 100$, $\varepsilon = 0.05$, $m = 2$ in experiments of data sets Sonar, WDBC, Arrhythmia, and Hepatitis, and the setting of parameter N, λ, and *Maxiters* are the same as Experiment 2.

4.3. Experimental results analysis

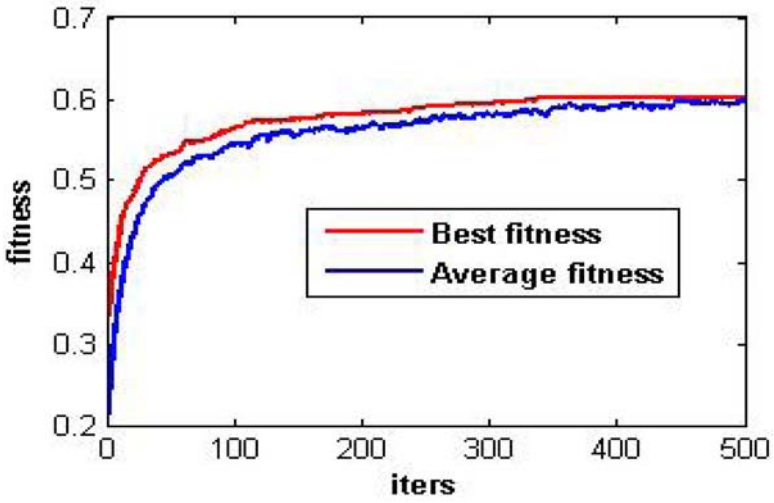
The results of four experiments in Sonar, WDBC, Arrhythmia, and Hepatitis data sets are shown in Figure 4.1, Figure 4.2, Figure 4.3, and Figure 4.4, respectively, and Table 4.3 shows the classification accuracy by ten-fold cross validation method.

4.3.1. Experimental results on Sonar

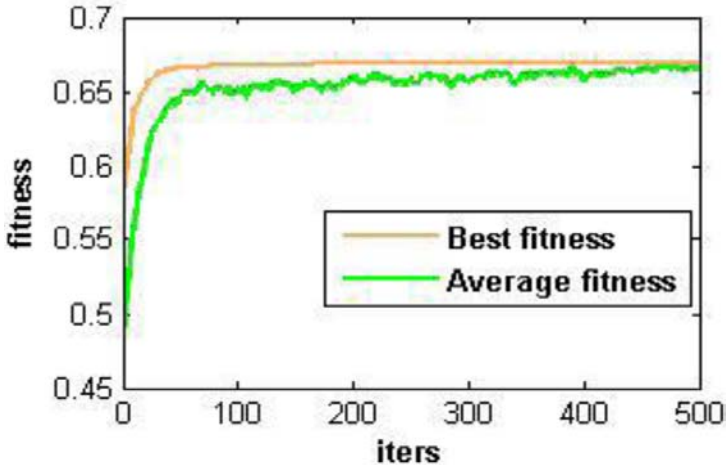
From Figure 4.1 and Table 4.3, we can conclude that our method achieved the highest classification accuracy of 72.36% in Sonar data set, and selected 4 features, so it can save the storage space and operation time in subsequent processing. In addition, our method reached the best fitness of 0.666786 just requiring 40 iterations, while the FSGA needs 75 iterations to get the best fitness of 0.666643, and the GA needs 338 iterations to get the best fitness of 0.6, thus, our method has the fastest convergence rate. In terms of classification accuracy, number of selected features and convergence rate, our proposed method is most effective for Sonar data set to perform feature selection.



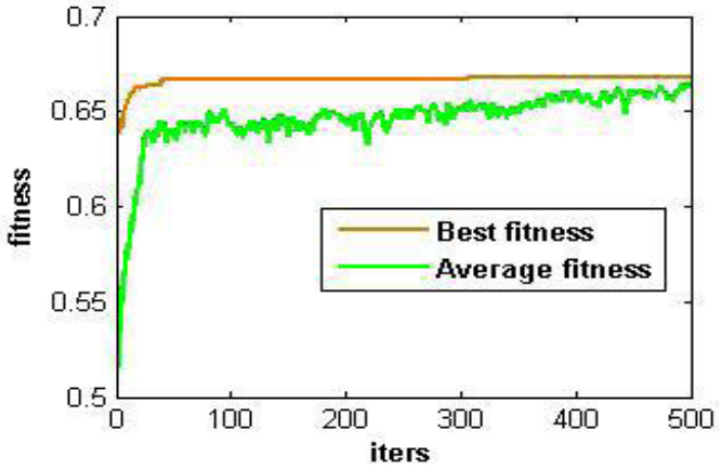
(a) Results of feature selection by FS



(b) Results of feature selection by GA



(c) Results of feature selection by FSGA



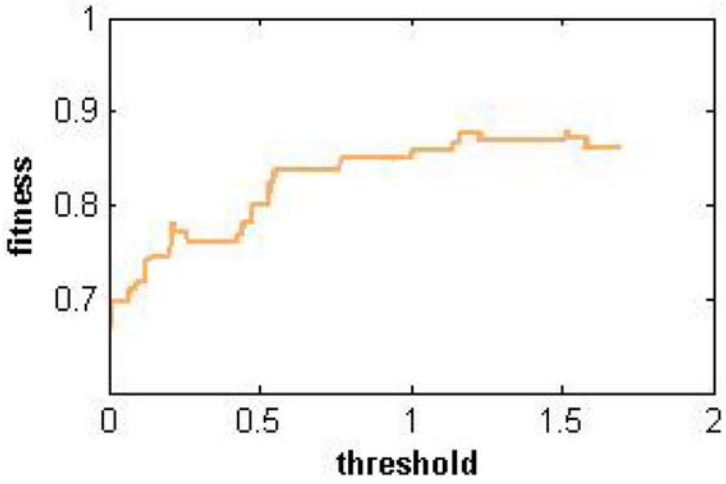
(d) Results of feature selection by HFSGA

Figure 4.1. Four experimental results on Sonar data set.

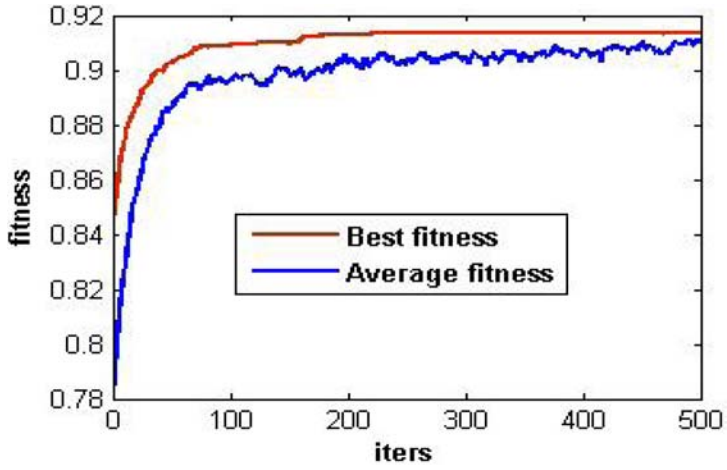
4.3.2. Experimental results on WDBC

Figure 4.2 and Table 4.3 show that the classification accuracy of performing feature selection on WDBC data set by the four methods are 92.70%, 94.10%, 94.05%, and 95.66%, respectively, which indicates that the discrimination between relevant features and irrelevant features is obvious. The FS selected 5 features, while other three methods selected 3

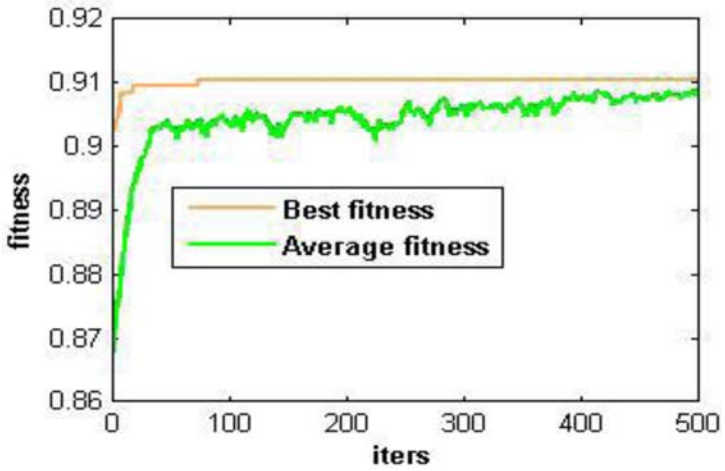
features respectively, we can infer that there may be 2 redundant features with high Fisher score. Our method achieved the highest classification accuracy of 95.66%, which demonstrated that the proposed method does well in eliminating redundant features.



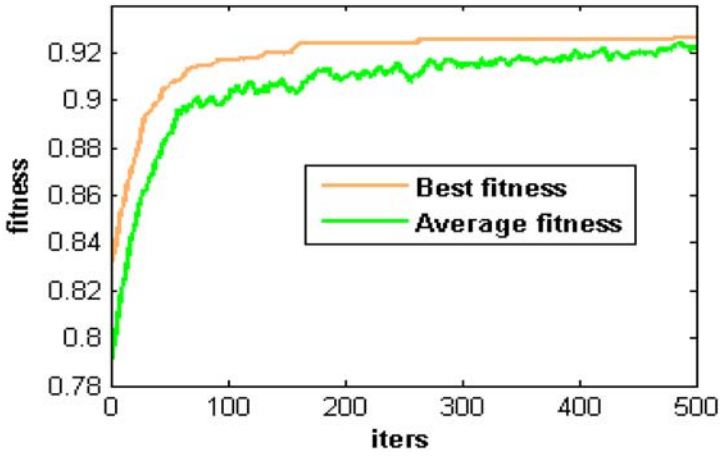
(a) Results of feature selection by FS



(b) Results of feature selection by GA



(c) Results of feature selection by FSGA



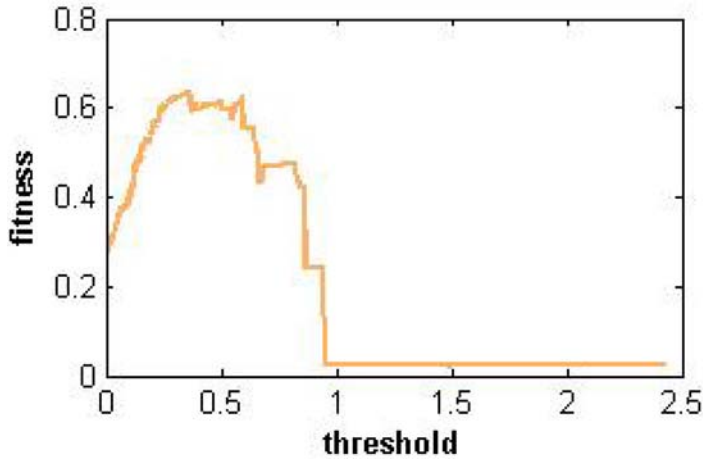
(d) Results of feature selection by HFSGA

Figure 4.2. Four experimental results on WDBC data set.

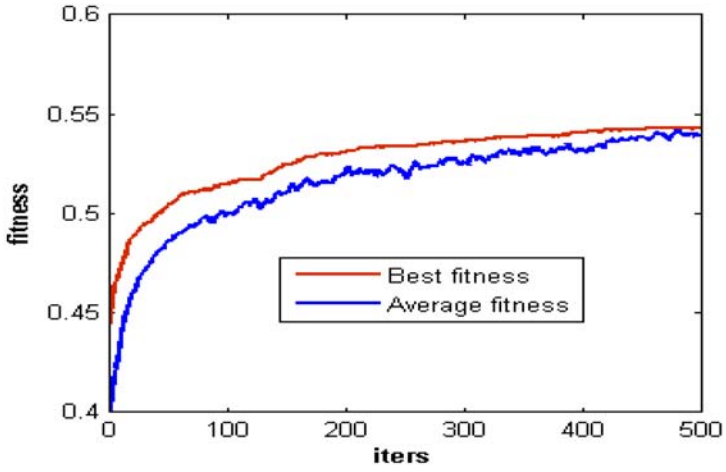
4.3.3. Experimental results on Arrhythmia

From Figure 4.3(a), when Fisher score is adopted to feature selection, in the beginning, the fitness becomes higher and higher with the increase of threshold, and when the threshold is higher than 0.5, the fitness becomes lower and lower with the increase of threshold. We can infer that, on Arrhythmia data set, there are more redundant features and less

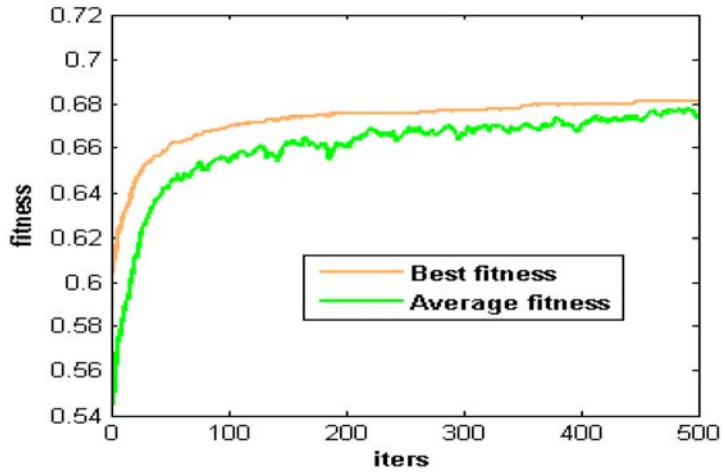
meaningless features. The classification accuracy of proposed method is 70.54%, lower than the FSGA method of 72.34%, while higher than genetic algorithm and Fisher score of 66.57% and 67.49%, respectively. This result indicates that, on the one hand, Fisher score is a favorable criterion to determine the relevance of features for Arrhythmia data set; on the other hand, the FAGA method does well in eliminating irrelevant features, while the proposed HFAGA method is good at eliminating redundant features.



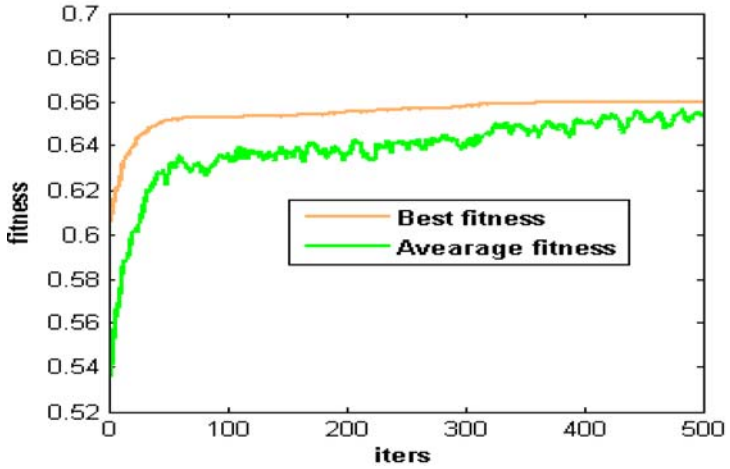
(a) Results of feature selection by FS



(b) Results of feature selection by GA



(c) Results of feature selection by FSGA



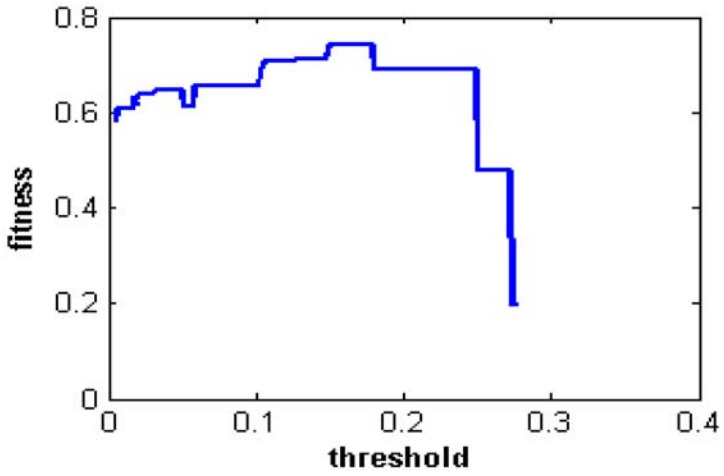
(d) Results of feature selection by HFSGA

Figure 4.3. Four experimental results on Arrhythmia data set.

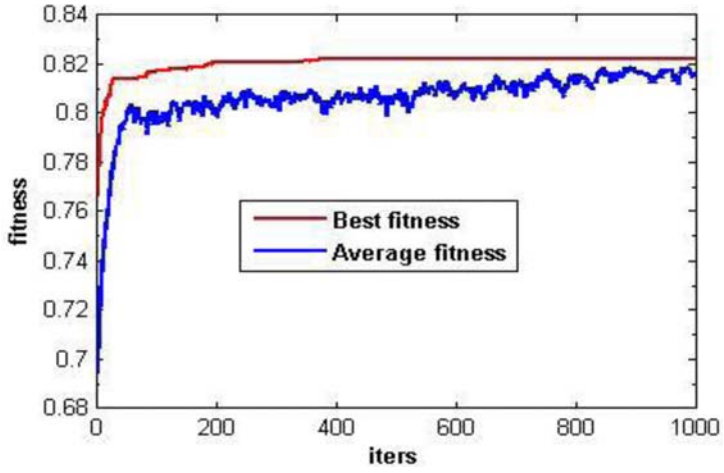
4.3.4. Experimental results on Hepatitis

From Figure 4.4 and Table 4.3, we can conclude that our method achieved the highest classification accuracy of 87.83% on Hepatitis data set, and selected 5 features, so it can save the storage space and

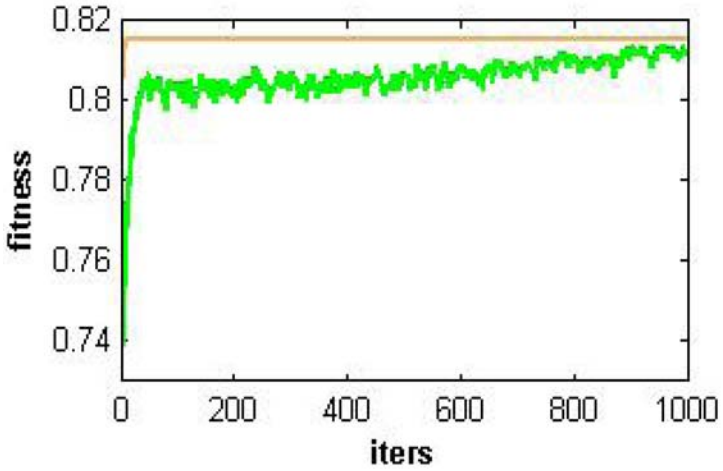
operation time in subsequent processing. Furthermore, our method has the fastest convergence rate. Above all, our proposed method is most effective for Hepatitis data set to perform feature selection.



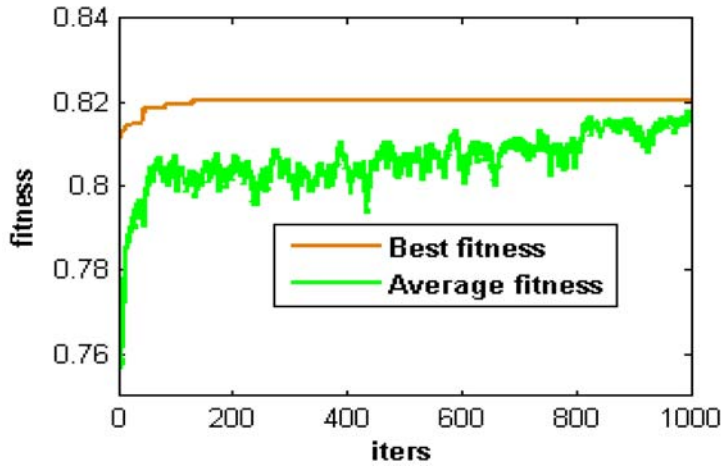
(a) Results of feature selection by FS



(b) Results of feature selection by GA



(c) Results of feature selection by FSGA



(d) Results of feature selection by HFSGA

Figure 4.4. Four experimental results on Hepatitis data set.

Table 4.3. Classification accuracy and number of selected features of 16 experiments

Data sets	Algorithms	Best fitness	Classification Accuracy	Number of selected features
Sonar (60)	FS	0.597143	0.607143	1
	GA	0.601571	0.711571	11
	FSGA	0.669286	0.719286	5
	HFSGA	**0.683571**	**0.723571**	4
WDBC (30)	FS	0.877033	0.927033	5
	GA	0.911030	0.941030	3
	FSGA	0.910495	0.940495	3
	HFSGA	**0.926352**	**0.956352**	3
Arrhythmia (279)	FS	0.635941	0.674941	39
	GA	0.542747	0.665747	123
	FSGA	**0.681358**	**0.720358**	39
	HFSGA	0.660379	0.705379	45
Hepatitis (19)	FS	0.740000	0.780000	4
	GA	0.821833	0.861833	4
	FSGA	0.815500	0.835500	2
	HFSGA	**0.828333**	**0.878333**	5

4.4. Discussion

The proposed HFSGA method is superior to other three feature selection methods of FS, GA, and FSGA methods in terms of classification accuracy, and it does well in eliminating redundant features with a fast convergence rate. From the experiments on Arrhythmia data set, we can discover that the FSGA method is good at eliminating irrelevant features. In a word, our method is effective in feature selection.

Certainly, our proposed HFSGA method requires to set many parameters, so we can do some further researches to explore the influence of parameters on experiment results, and then determine the

optimal parameters. In addition, the selection of classifier also influences the classification accuracy, thus, the match of classifier and feature selection method is worthy of further research.

5. Conclusion

In this paper, we presented a hybrid feature selection method based on Fisher score and genetic algorithm. It utilizes features' rescaled Fisher score to generate the initial population of genetic algorithm. On the one hand, our method avoids the conundrum of how to set a threshold, and it does well in eliminating redundant features, thus, it can select the feature subset with better performance. On the other hand, unlike genetic algorithm generating initial population with a random manner, our method has a faster convergence rate. Contrast experiments on four benchmark data sets of Sonar, WDBC, Arrhythmia, and Hepatitis indicate that the proposed method outperforms Fisher score method and genetic algorithm, and it is effective in feature selection.

References

[1] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003), 1157-1182.

[2] Y. Yang and J. O. Pedersen, A comparative study on feature selection in text categorization, In ICML (1997), 412-420.

[3] D. Koller and M. Sahami, Toward optimal feature selection, In ICML (1996), 284-292.

[4] P. E. H. R. O. Duda and D. G. Stork, Pattern Classification, Wiley-Inter Science Publication, 2001.

[5] M. Robnik-Sikonja and I. Kononenko, Theoretical and Empirical Analysis of ReliefF and RReliefF, Machine Learning 53(1-2) (2003), 23-69.

[6] X. He, D. Cai and P. Niyogi, Laplacian score for feature selection, In NIPS, 2005.

[7] L. Song, A. J. Smola, A. Gretton, K. M. Borgwardt and J. Bedo, Supervised feature selection via dependence estimation, In ICML (2007), 823-830.

[8] F. Nie, S. Xiang, Y. Jia, C. Zhang and S. Yan, Trace ratio criterion for feature selection, In AAAI (2008), 671-676.

[9] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Trans. Neural Networks 5(4) (1994), 537-550.

[10] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

[11] R. Caruana and D. Freitag, Greedy Attribute Selection, In: Proc. of the 11th Internet. Conf. on Machine Learn., New Brunswick, NJ, USA, (1994), 28-36.

[12] P. Somol, P. Pudil and J. Kittler, Fast branch and bound algorithms for optimal feature selection, IEEE Trans. Pattern Anal. Machine Intell. 26(7) (2004), 900-912.

[13] J. H. Yang and V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intell. Systems 13(2) (1998), 44-49.

[14] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn and A. K. Jain, Dimensionality reduction using genetic algorithms, IEEE Trans. Evolut. Comput. 4(2) (2000), 164-171.

[15] B. Bhanu and Y. Lin, Genetic algorithm based feature selection for target detection in SAR images, Image Vision Comput. 21(7) (2003), 591-608.

[16] F. Zhu and S. Guan, Feature selection for modular GA-based classification, Appl. Soft Comput. J. 4(4) (2004), 381-393.

[17] M. Kudo and J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recognition 33(1) (2000), 25-41.

[18] John H. Holland, Adaptation in Natural and Artificial Systems, 1975.

[19] K. A. De Jong, An Analysis of the Behavior of a Class of Generic Adaptive Systems, University of Michigan, 1975.

[20] B. Fogel David, Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, 2nd Edition, Wiley-IEEE Press, 1999.

[21] A. Asuncion and D. Newman, UCI Machine Learning Repository, 2007.

■