Pattern Recognition 60 (2016) 121–129

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Feature selection using Forest Optimization Algorithm

Manizheh Ghaemi [a,*], Mohammad-Reza Feizi-Derakhshi [b]

[a] Faculty of Electrical and Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran
[b] Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Article history:
Received 17 August 2015
Received in revised form
26 March 2016
Accepted 11 May 2016
Available online 24 May 2016

Keywords:
Feature selection
Forest Optimization Algorithm (FOA)
KNN classifier
Dimension reduction
FSFOA

ABSTRACT

Feature selection as a combinatorial optimization problem is an important preprocessing step in data mining; which improves the performance of the learning algorithms with the help of removing the irrelevant and redundant features. As evolutionary algorithms are reported to be suitable for optimization tasks, so Forest Optimization Algorithm (FOA) – which is initially proposed for continuous search problems – is adapted to be used for feature selection as a discrete search space problem. As the result, Feature Selection using Forest Optimization Algorithm (FSFOA) is proposed in this article in order to select the more informative features from the datasets. The proposed FSFOA is validated on several real world datasets and it is compared with some other methods including HGAFS, PSO and SVM-FuzCoc. The results of the experiments show that, FSFOA can improve the classification accuracy of classifiers in some selected datasets. Also, we have compared the dimensionality reduction of the proposed FSFOA with other available methods.

1. Introduction

One of the inevitable steps in knowledge discovery is data mining and the knowledge obtained as the result of data mining is used in many trends; like business and medical use [6,15,20,37]. These days, there has been an increase in the number of collected and stored features in databases but not all the features are useful for data mining, so that some of the features are completely irrelevant or redundant [8,10,36,23]. These features not only have no use in the process of knowledge discovery, but also they increase the complexity and incomprehensibility of the results. So, feature selection helps to reduce the dimensionality of the datasets before data mining. In large databases with many features to handle, when there are n features, time complexity to evaluate all the subsets of features is exponential $(O(2^n))$ [31], which is practically impossible. So, feature selection methods are the bases for data mining to keep useful features for latter learning tasks alongside the ignoring of the most irrelevant and less important ones [11]. In fact feature selection techniques ignore the irrelevant features so, learning process can be done more efficiently. It is also proved that feature selection increases the classification accuracy of machine learning algorithms like KNN classifier [11].

Feature selection is the special case of feature weighting problem [34]. Many studies have shown the beneficial effect of feature weighting [1,9,27,30,32,33]. In feature weighting problem, features are assigned a value which shows their importance in the machine learning process but, in feature selection problem a feature is either retained or deleted and the weights are limited to just '0' and '1'. In fact, feature selection algorithms are a proper subset of feature weighting algorithms which use binary weights (i.e., 0 or 1).

It has been proved that feature selection has an impact on the accuracy and complexity of the classifiers [11]. The mostly used criterions for evaluating the selected feature subset is classification accuracy (CA) on new instances (test dataset). In fact, we expect that dimensionality reduction with the help of feature selection will increase classification accuracy or at least it remains the same.

The objective of this paper is to select the useful features of the datasets with the help of FOA as a new evolutionary algorithm. As FOA is reported to be suitable in continuous search spaces, in this article we have attempted to investigate the performance of FOA in feature selection (FS) as a discrete search problem and we have introduced a method named as Feature Selection using Forest Optimization Algorithm (FSFOA). In fact, FSFOA searches for the best feature subset with the objective of improving the classification accuracy of some classifiers as learning algorithms including KNN, C4.5 and SVM classifiers. The contribution of this paper is twofold: adapting FOA for solving discrete problems and also solving feature selection problem with the help of discrete FOA which leads to the proposed FSFOA method.

The rest of this paper is organized as follows. In Section 2, an overview of feature selection methods is presented. An overview of Forest Optimization Algorithm (FOA) is given in Section 3. In

* Corresponding author.
E-mail addresses: m_ghaemi@ee.kntu.ac.ir (M. Ghaemi), mfeizi@tabrizu.ac.ir (M.-R. Feizi-Derakhshi).

http://dx.doi.org/10.1016/j.patcog.2016.05.012
0031-3203/

Section 4, the application of FOA for feature selection (FSFOA) is presented and Section 5 is devoted to the experiments and results on the proposed FSFOA. Finally, Section 6 summarizes the main conclusions.

2. An overview of feature selection methods

Many researchers have addressed feature selection (FS) problem up to now and also more attempt is needed to further speed up the process of selecting informative and useful features in databases for data mining.

The earliest methods in FS literature based on the machine learning algorithms are filters [11,12]. In all the filters, heuristic techniques based on the general characteristics of data such as information gain and distance is used instead of learning algorithms. Another approach in feature selection is wrapper methods [11,19]. In contrary to filters, wrappers use learning algorithms to investigate the worthy of the selected features [41]. Generally, wrappers produce better results than filters; because while using wrapper approach, the relationship between the learning algorithm and the training data is considered. The well-known drawback of wrappers is that they are slower than filters; because the learning algorithm must be repeatedly executed for every selected feature subset. Sometimes a hybrid of filter-wrapper methods is used. Hybrid methods integrate feature selection within the learning algorithm in order to exploit the advantages of both wrappers and filters [11]. Ignoring the filter or wrapper approach for feature selection methods, they can match any of the following groups: complete search, heuristic search and meta-heuristic methods.

Almuallim and Dietterich presented FOCUS method which completely searches the search space up to reaching to the smallest set of features that divides the training data into pure classes [2,3]. But with n features to handle, there are $(2^n) - 1$ possible subsets of features so, evaluating all of the subsets is practically impossible in datasets with many features. As the result, complete search methods are seldom used for feature selection in large datasets with many features.

Heuristic methods of feature selection problem include greedy hill climbing algorithm [25,26], branch and bound method, beam search and best first algorithm. Greedy hill climbing algorithm evaluates all local changes in order to select the relevant features [11,25]. SFS (Sequential Forward Selection) and SBS (Sequential Backward Selection) are two kinds of hill climbing methods. SFS starts with an empty set of selected features and each step of the algorithm adds one of the informative features to the selected set; but, SBS starts will the full set of features and in each step, one of the redundant or irrelevant features is omitted. Bi-directional search is another method which considers both adding and deleting the features simultaneously [11]. The main drawback of both SFS and SBS algorithms is the "nesting effect" problem; which means that while a change is considered positive (either addition or deletion of a feature), there is no chance of re-evaluating that feature. Later in order to overcome the "nesting effect" of SFS and SBS algorithms, SFFS (Sequential Forward Floating Selection) and SBFS (Sequential Forward Floating Selection) were introduced [24]. Best first search is another method which like hill climbing considers local changes in the search space but, it allows backtracking in the search space unlike hill climbing methods [11].

Heuristic algorithms perform better than complete search methods while comparing time complexities, but recently meta-heuristic algorithms like Genetic Algorithm (GA), Particle Swarm Intelligence Optimization (PSO) and Ant Colony Optimization (ACO) show more desirable results. The main advantage of the meta-heuristic methods is their acceptable time complexity. Due to the random nature of meta-heuristic search methods, the application of genetic algorithms, particle swarm optimization algorithm and ant colony optimization in feature selection domain have shown promising results [18]; some of which are summarized in the following.

Hamdani et al. proposed a new algorithm based on hierarchical genetic algorithms with bi-coded chromosome representation and new evaluation function [13]. In order to minimize the computational cost and also speed up the convergence speed, they used a hierarchical algorithm with homogeneous and heterogeneous population. In another attempt, Zhu et al. proposed a new algorithm which is a combination of genetic algorithm and local search method [40]. At first, GA population is generated randomly, then local search is applied to all of the individuals of the population in order to improve the classification accuracy and speed up the searching process. Tan et al. used SVM (Support Vector Machine) based on wrapper approach [31] in GA. In their proposed algorithm, GA searches for the best feature subset and the classification accuracy of SVM guides the search process. Gheyas et al. combined both simulated annealing (SA) and GA to use the advantages of both SA and GA [10]. In their proposed SAGA, GA helps to escape from local optimum of SA with the crossover operator. Nemati et al. proposed a new hybrid algorithm of GA and ACO in order to use the advantages of both algorithms [22]. In their algorithm, ACO performs a local search, while GA is used to perform a global search. Sivagaminathan et al. used ACO which searches for near-optimum solution and ANN is used as a classifying function [28]. ElAlami et al. proposed an algorithm based on GA, which optimizes the output nodes of ANN [7]. In their method, ANN is used to give a weight to each of the features and GA finds the optimal relevant features. Kabir et al. proposed a new hybrid algorithm that combines GA with local search method (HGAFS) [17]. Their proposed method selects the feature subset with a limited size; which is the important aspect of their method. Their method is a wrapper based method that uses both GA and ANN. In another attempt, Tabakhi et al. presented an unsupervised feature selection method based on ant colony optimization, called UFSACO [29]. Their proposed UFSACO is a filter-based method and the search space is represented as a fully connected undirected weighted graph. Xue et al proposed a series of methods based on PSO with novel initialization and updating mechanisms [35]. In their proposed algorithm, three new initialization strategies and three new personal best and global best updating mechanisms in PSO are presented to develop novel feature selection approaches; in which, maximizing the classification performance, minimizing the number of features and reducing the computational time are the main goals.

Despite good progress in solving feature selection problem, more study is also welcomed to further optimize the solutions. In all the proposed methods, one should choose either computationally feasible or optimality of the selected features. Further research is needed to develop more promising methods for feature selection with the aim of providing very good results. In the present work, FSFOA algorithm is proposed to further optimize the results of feature selection methods in the case of improving classification accuracy.

3. An overview of the Forest Optimization Algorithm (FOA)

Forest Optimization algorithm is an evolutionary algorithm, which is inspired by the procedure of a few trees in the forests [9]. FOA is proposed to solve continuous search space problems, but in this article we have attempted to adjust it to use in discrete search space problems like feature selection. FOA involves three main stages: 1 – Local seeding of the trees, 2 – Population limiting, and 3

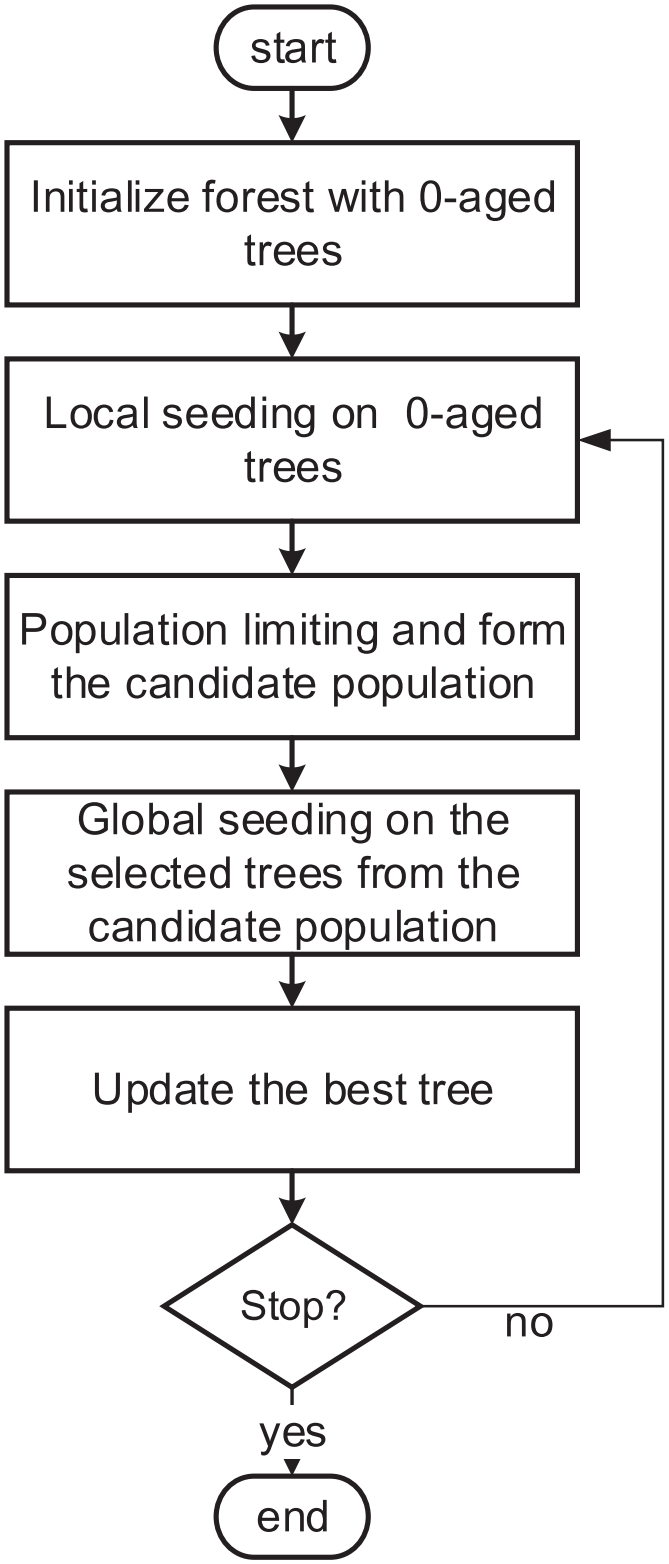

Fig. 1. Flowchart of FOA [9].

– Global seeding of the trees. Fig. 1 shows the flowchart of FOA. FOA starts with the initial population of trees (solutions) which forms forest in this algorithm. Each tree represents a potential solution of the problem. A tree has a part that represents the "*Age*" of the related tree in addition to the values of the variables. The "*Age*" of each newly generated tree is set to '0' [9].

In the nature when seeding procedure of the trees begins, some seeds fall just beneath the parent tree and then they turn into young trees [9]; which is simulated by local seeding in FOA. After initialization of the trees, the local seeding stage will operate on trees with "*Age*" '0' to simulate the nearby seeds of the parent trees. Then all the trees, except new generated ones, get old and their "*Age*" increases by '1'. This stage simulates the local search of the algorithm.

Next stage is population limiting in which the trees with "*Age*" bigger than "*life time*" parameter will be omitted from the forest and they will form the candidate population [9]. Also in population limiting stage, the rest of the trees of the forest are sorted according to their fitness value and if the number of whole trees of the forest exceeds the pre-defined "*area limit*" parameter, the extra trees will join to the candidate population too. In the global seeding stage, a percentage of the candidate population is chosen. The selected trees from the candidate population will be used in the global seeding stage. Global seeding stage simulates the global search of FOA [9]. Next stage in FOA is updating the best tree in which the best solution is selected according to its fitness value and its "*Age*" is set to 0 in order to avoid the aging and afterward removing the best tree from the forest. These stages will continue iteratively until the termination criterion is met. Forest Optimization Algorithm has 5 parameters which should be initialized at the start of the algorithm [9]:

1. "Local Seeding Changes" or "*LSC*",
2. The limitation of the forest or "*area limit*",
3. The maximum allowed "*Age*" of a tree, which is named as "*life time*" parameter,
4. Percentage of the candidate population to be used in the global seeding stage or "*transfer rate*",
5. The number of the variables, whose values will be changed in the global seeding stage, is another parameter of the algorithm and is named as "Global Seeding Changes" or "*GSC*".

As FOA is proposed for continuous space problems, in this article we have adapted FOA to be suitable in discrete space problems like feature selection. The pseudo code of Forest Optimization Algorithm for feature selection, named as FSFOA, is illustrated as Algorithm 1. As it is shown in Algorithm 1, the needed changes in FOA to be suitable for feature selection problem should be in initialization, local seeding and global seeding stages. In the next section the stages of FSFOA to handle the discrete search space of the feature selection problem are explained in more details.

Algorithm 1. FSFOA (life time, LSC, GSC, transfer rate, area limit)

Input: life time, LSC, GSC, transfer rate, area limit
Output: The best feature set with the highest fitness
1: **Procedure FSFOA**
2: Initialize forest with random 0/1 trees
3: Each tree is a $(D+1)$-dimensional vector x (D is the number of all features).
4: The "Age" of each tree is initially zero.
While stop condition is not satisfied do

1: Perform local seeding on trees with Age 0
2: **For** $i=1$: "LSC" **do**
3: Randomly choose a variable of the selected tree
4: change from 0 to 1 or vice versa
5: **end for**
6: Increase the Age of all trees by 1
7: Population limiting
8: Global seeding
9: Choose "transfer rate" percent of the candidate population
10: **for** each selected tree **do**

11: Choose "GSC" variables of the selected tree randomly
12: change from 0 to 1 or vice versa
13: **end for**
14: Update the best so far tree
15: Sort trees according to their fitness value
16: Set the Age of the best tree to 0
End While
5: **end procedure**
Return the best tree which shows the best selected feature subset

4. The proposed feature selection using forest optimization algorithm (FSFOA)

The stages of FOA for feature selection problem are adapted as the following.

4.1. Initialize trees

The forest is initialized by randomly generated trees [9]. At first, each variable of each tree in FSFOA is initialized randomly with either '0' or '1'. If a dataset has n features, the size of each tree will be $1*(n+1)$; where one of the variables shows the "*Age*" of that tree. Each '1' in a tree indicates that the corresponding feature is selected and therefore is involved in the machine learning process and each '0' shows the exclusion of the related feature in the learning process. At first, the "*Age*" of each tree is considered to be '0', but local seeding in each iteration of the algorithm will increase the "*Age*" of all trees except new generated ones in the local seeding stage.

4.2. Local seeding

This stage adds some neighbors of each tree with "*Age*" 0 to the forest [9]. In order to simulate this stage in FSFOA, for each tree of the forest with "*Age*" 0, some variables are selected randomly ("*LSC*" parameter determines the number of the selected variables). Then the values of the selected variables are changed from 0 to 1 or vice versa. This procedure simulates local search in the search space; because each time the importance of one feature is evaluated by adding and removing that feature prior to learning algorithm. Fig. 2 shows an example of local seeding operator on one tree, where the number of the features of the dataset is 5 and the value of "*LSC*" is considered to be 2. After performing the local seeding stage, the "*Age*" of all trees except new generated ones, is increased by '1.

4.3. Population limiting

In this stage two series of trees will be omitted from the forest to form the candidate population: 1 – trees with "*Age*" bigger than "*life time*" parameter and 2 – the extra trees that exceed "*area limit*" parameter after sorting the trees according to their fitness value. This stage forms the candidate population and pre-defined percentage of the candidate population is used later in global seeding stage.

Fig. 2. An example of local seeding operation on one tree with "*LSC*"=2.

Fig. 3. An example of global seeding operator on one tree"*GSC*"=3.

Table 1
Summary of the selected datasets.

Dataset	#Features	#Instances	#Class
Heart-statlog	13	270	2
Vehicle	18	846	4
Cleveland	13	303	5
Dermatology	34	366	6
Ionosphere	34	351	2
Sonar	60	208	2
Glass	9	214	7
Wine	13	178	3
Segmentation	19	2310	7
SRBCT	2308	63	4
Hepatitis	19	155	2

Table 2
The value of "*LSC*" and "*GSC*" parameters for each dataset.

Dataset	#Features	"*LSC*"	"*GSC*"
Heart-statlog	13	3	6
Vehicle	18	4	9
Cleveland	13	3	6
Dermatology	34	7	15
Ionosphere	34	7	15
Sonar	60	12	30
Glass	9	2	4
Wine	13	3	6
Segmentation	19	4	9
SRBCT	2308	460	700
Hepatitis	19	4	10

Table 3
Summary of the methods for our comparisons.

Method name	Dataset splitting	Description/year
SFS, SBS, SFFS	70–30	Greedy hill climbing methods[a][21] (2010)
NSM	10-fold	Neighbor soft margin [14]/2010
SVM-FuzCoc	70–30%	A novel SVM- based FS [21]/2010
HGAFS	2-fold	Hybrid genetic algorithm for FS [16]/2007
FS-NEIR	10-fold	Neighborhood effective information ratio based FS [40]/2013
UFSACO	70–30	Unsupervised FS algorithm based on ACO [29]/2014
PSO(4-2)	10-fold	Particle swarm optimization for feature selection [35]/2013

[a] Sequential Forward selection, Sequential Backward selection, Sequential Floating Forward selection method reported from [21]

4.4. Global seeding

In order to perform this stage in FSFOA, at first for each selected tree from the candidate population, some of the variables are selected randomly. The number of the selected variables is determined by the "*GSC*" parameter. Then, the value of each selected variable will be negated (changing from 0 to 1 or vice versa). But this time, adding or deleting some features are considered simultaneously and not just one feature at a time. This operator performs a global search in the search space. An example of performing this operator on one tree is shown as Fig. 3. In Fig. 3, the value of "*GSC*" parameter is considered to be 3 (3 variables are

Dataset	Wine		
	Accuracy (%)	DR (%)	Classifier
FSFOA (our work)	**96.06 (10-fold)**	21.42	J48
	98.87(10-fold)	42.58	3-NN
	98.07 (70%-30%)	50	1-NN
	96 (70%-30%)	57.14	J48
	99.2 (70%-30%)	30.76	5-NN
	96.06 (2-fold)	37.17	Rbf-svm
SVM-FuzCoc [21]	97.12 (70%-30%)	53.84	1-NN
UFSACO [29]	95.08 (70%-30%)	**61.53**	J48
SFS [21]	97.69 (70%-30%)	35.38	1-NN
SBS [21]	94.77(70%-30%)	46.15	1-NN
SFFS [21]	96.56(70%-30%)	36.92	1-NN
HGAFS [16]	**98.31 (2-fold)**	53.85	Rbf-svm
NSM [14]	98(10-fold)	53.84	3-NN
FS-NEIR [40]	95.04 (10-fold)	61.53	J48
PSO(4-2) [35]	95.26 (10-fold)	**51.6**	5-NN

Dataset	Ionosphere		
	AC (%)	DR (%)	Classifier
FSFOA (our work)	**93.16 (10-fold)**	68.57	J48
	92.3 (10-fold)	61.76	3-NN
	89.43 (10-fold)	54.28	5-NN
	94.58 (2-fold)	57.14	Rbf-svm
	89.52 (70%-30%)	54.28	1-NN
	95.12 (70%-30%)	47.05	J48
SVM-FuzCoc [21]	89.46 (70%-30%)	88.23	1-NN
HGAFS [16]	92.76 (2-fold)	82.35	Rbf-svm
SFS [21]	87.75 (50%-50%)	65.88	1-NN
SBS [21]	84.61 (50%-50%)	77.64	1-NN
SFFS [21]	88.32 (50%-50%)	75.29	1-NN
FS-NEIR [40]	92.59 (10-fold)	82.35	J48
NSM [14]	92 (10-fold)	88.23	3-NN
PSO(4-2) [35]	87.27 (70%-30%)	**90.41**	5-NN
UFSACO [29]	88.61 (70%-30%)	11.17	J48

Dataset	Sonar		
	Accuracy (%)	DR (%)	classifier
FSFOA (our work)	65.86 (2-fold)	54.09	RBF-SVM
	82.69 (10-fold)	52.45	J48
	85.43 (70%-30%)	57.37	1-NN
	86.98 (70%-30%)	44.26	5-NN
SVM-FuzCoc [21]	73.17 (70%-30%)	68.33	1-NN
SFS [21]	66.43 (50%-50%)	61.33	1-NN
SBS [21]	62.2 (50%-50%)	45.33	1-NN
SFFS [21]	64.55 (50%-50%)	61.33	1-NN
HGAFS [16]	**87.02 (2-fold)**	75	Rbf-svm
FS-NEIR [40]	75.97(10-fold)	**91.66**	J48
PSO(4-2) [35]	78.16 (70%-30%)	81.26	5-NN

Dataset	Heart-statlog		
	Accuracy (%)	DR (%)	Classifier
FSFOA (our work)	**85.15 (10-fold)**	48.07	J48
	85.18 (10-fold)	35.71	3-NN
	84.07 (2-fold)	50	Rbf-svm
NSM [14]	84 (10-fold)	69.23	3-NN
FS-NEIR [40]	79.86 (10-fold)	46.15	J48
HGAFS [16]	82.59 (2-fold)	**76.92**	Rbf-svm

Dataset	Dermatology		
	Accuracy (%)	DR (%)	Classifier
FSFOA (our work)	**96.99 (10-fold)**	21.42	J48
	97.27 (70%-30%)	45.71	1-NN
	90.09 (70%-30%)	44.11	J48
SFS [21]	94.02 (70%-30%)	44.7	1-NN
SBS [21]	91.78 (70%-30%)	58.23	1-NN
SFFS [21]	93.7 (70%-30%)	62.35	1-NN
FS-NEIR [40]	93.95 (10-fold)	**70.58**	J48
UFSACO [29]	**95.28 (70%-30%)**	26.47	J48
SVM-FuzCoc [21]	94.11 (70%-30%)	64.7	1-NN

Dataset	Glass		
	Accuracy (%)	DR (%)	Classifier
FSFOA (our work)	**75.7 (10-fold)**	50	J48
	71.88 (70%-30%)	40	1-NN
	68.22 (2-fold)	**60**	Rbf-svm
SVM-FuzCoc [21]	**73.36 (70%-30%)**	33.33	1-NN
HGAFS [16]	65.51 (2-fold)	44.44	Rbf-svm
SFS [21]	72.24 (70%-30%)	26.66	1-NN
SBS [21]	71.77 (70%-30%)	37.77	1-NN
SFFS [21]	71.77 (70%-30%)	37.77	1-NN
FS-NEIR [40]	68.53 (10-fold)	22.22	J48

Dataset	**Cleveland**		
	Accuracy (%)	DR (%)	Classifier
FSFOA (our work)	55.55 (70%-30%)	**71.42**	1-NN
SVM-FuzCoc [21]	**61.01 (70%-30%)**	46.1	1-NN
SFS [21]	51.79 (70%-30%)	47.7	1-NN
SBS [21]	54.8 (70%-30%)	38.5	1-NN
SFFS [21]	49.55 (70%-30%)	53.8	1-NN

Dataset	Vehicle		
	Accuracy (%)	DR (%)	Classifier
FSFOA (our work)	**73.04 (10-fold)**	31.57	J48
	73.98 (70%-30%)	50	5-NN
	62.41 (2-fold)	47.22	Rbf-svm
HGAFS [16]	**76.36 (2-fold)**	38.89	Rbf-svm
FS-NEIR [40]	70.98 (10-fold)	50	J48
PSO(4-2) [35]	**85.3 (70%-30%)**	**68.4**	5-NN

Dataset	SRBCT		
	Accuracy (%)	DR (%)	Classifier
FSFOA (our work)	94.73(70%-30%)	49.06	1-NN
SVM-FuzCoc [21]	**98.88 (70%-30%)**	**98.57**	1-NN

Dataset	Hepatitis		
	Accuracy (%)	DR (%)	Classifier
FSFOA (our work)	**86.45 (10-fold)**	55	J48
	87.09 (10-fold)	42.1	3-NN
	84.4 (70%-30%)	45	J48
NSM [14]	**90 (10-fold)**	15.78	3-NN
FS-NEIR [40]	81.11 (10-fold)	68.42	J48
UFSACO [29]	78.87 (70%-30%)	**75**	J48

Dataset	Segmentation		
	Accuracy (%)	DR (%)	Classifier
FSFOA (our work)	**96.2(10-fold)**	30	3-NN
NSM [14]	95 (10-fold)	**63.15**	3-NN

Fig. 4. Comparison between the Classification Accuracy (Accuracy) and Dimension Reduction (DR) obtained by FSFOA and other available methods on "Heart-statlog", "Cleveland", "Vehicle", "Dermatology", "Sonar" "Ionosphere", "Glass", "Segmentation", "Hepatitis", "SRBCT" and "Wine" datasets.

Table 4
Summary of the configuration for the classifiers.

Classifier	Configuration
KNN	$K=1$, $K=3$, $K=5$
C4.5	J48
SVM	rbf kernel

negated to form a new tree).

4.5. Update the best tree

In this stage, after sorting the trees according to their fitness value, the tree with the highest fitness value is selected as the best tree and its "*Age*" will be set to '0'. These stages are performed iteratively until the stop condition is satisfied.

4.6. Fitness function

K-nearest Neighbor (KNN) is the bases of many lazy learning algorithms [33], but it has several drawbacks such as high storage requirements and sensitivity to noise. So, many researchers have attempted to address these drawbacks. Feature selection (FS), prototype generation/selection and feature weighting (FW) methods are all used to further improve the performance of KNN [9,33]. We have attempted to improve the performance of some classifiers including KNN ($K \in \{1, 3, 5\}$) with the help of feature selection using FOA. In other words, the classification accuracy of some classifiers is used as our fitness function in our experiments. K-Nearest-Neighbor (1-NN, 3-NN and 5-NN), Support Vector Machine (SVM) and C4.5 (J48) classifiers are selected in this need. More explanation about the classifiers' configuration is presented in Section 5.3.

Because of the fact that partitioning the datasets with different percentages for the training and testing datasets may affect the results [42] especially in datasets with small number of instances, so in our experiments datasets are partitioned according to different methods. These methods include 10-fold cross validation method; 70% for the training and 30% for the testing dataset, and also 2-fold cross validation. The results of the experiments in Section 5.3 are reported according to these methods where needed in comparisons. Reporting the results of the training datasets is not a good indicator for the performance of the selected features because of the high probability of overfitting problem; the same is true for reporting just the testing accuracy. A good cure for the problem of overfitting is to partition the dataset into 3 sets: training, validation, and testing datasets. As the result, the validation set will be used to prevent the overfitting problem. But this method could be problematic in datasets with small number of samples; because some extent of the dataset will be ignored and is used as a validation set. Due to the small selected datasets in this article, after putting away the testing dataset, we have not put away the validation set and instead, the training dataset is itself trained using 10-fold cross validation; where the training set is divided into 10 distinct sets and 9 parts of 10 parts is used as the training set and the rest is used for the validation need and this process is repeated 10 times and at last the average of 10 runs is reported.

After that the training phase have finished, using an unseen testing dataset apart from the training dataset is inevitable and at last all the candidate feature subsets are evaluated on the same testing dataset. In other words at last the average results of 10 runs due to the testing dataset is reported as the final result. The classification accuracy (CA) and dimension reduction (DR) of the experiments on the unseen testing datasets are reported in the experiments of Section 5. Classification accuracy is an effective way for feature selection validation [38] and it is defined as Eq. (1); where N_CC is the number of correct classifications and N_AS denotes the number of all samples of the dataset. Dimension Reduction ratio is calculated as in Eq. (2), where N_SF is the number of selected features and N_AF is the number of all features of each dataset [5].

$$CA = N\_CC/N\_AS \tag{1}$$

$$DR = 1 - (N\_SF/N\_AF) \tag{2}$$

Zhao et al. used fitness function which is the combination of classification accuracy, the number of selected features and the feature costs [39]. Another objective function for choosing the informative features considers the internal relation of the features [23], which measures the within-class and between-class correlation of the features. Also, the combination of the objective functions can be used in this need. In this article, we have just considered the classification accuracy as our fitness function but, we have also compared the dimensionality reduction of our proposed FSFOA method with other methods. Also when comparing the results with each other, we have used the same partitioning and the same classifiers with other methods which will be mentioned in the following.

5. Experiments and results

The proposed FSFOA is validated with 11 datasets. Ten datasets are obtained from the UCI machine learning repository [4] and also a high dimentional dataset ("SRBCT") from microarray datasets. In our experiments, we used publicly available package WEKA, which is a java-based machine learning toolkit. KNN, SVM and J48 classifiers of WEKA software are used in our experiments. All the experiments are performed on an ASUS machine with Intel Core i3 CPU (2.40 GHz) and 4 GB of RAM and the main programming language is in Java.

5.1. Datasets

The selected benchmark datasets include "Ionosphere", "Glass", "Segmentation", "Hepatitis", "SRBCT", "Heart-statlog", "Cleveland", "Vehicle", "Dermatology", "Sonar" and "Wine" datasets. These datasets cover the examples of small, medium and large dimensional datasets. Summary of the selected datasets is presented in Table 1. Table 1 contains the number of features (#features), number of classes (#class) and the number of instances of each dataset (#instances). In feature selection problem, datasets are of small scale, medium scale, or large scale if number of features belongs to [0, 19], [20, 49], or [50, ∞], respectively [30]. So, 6 datasets among 11 ones are small scale datasets, 3 of them are medium scale datasets and "Sonar" and "SRBCT" datasets are large scale ones.

5.2. Parameters of FSFOA for our experiments

The parameters of FSFOA are defined as the following. Among the parameters, the value of "*life time*" parameter, "*area limit*" and "*transfer rate*" does not depend on the size of the datasets [9]; so we will consider the following values for these parameters: "*life time*"=15, "*area limit*"=50, "*transfer rate*"=5%. Because the value of "*LSC*" and "*GSC*" parameters depend on the number of the features of each dataset, so the value of these parameters are shown separately as Table 2. According to the experiments in [9], we will set the value of "*LSC*" parameter to 1/5 of the dimension of each dataset.

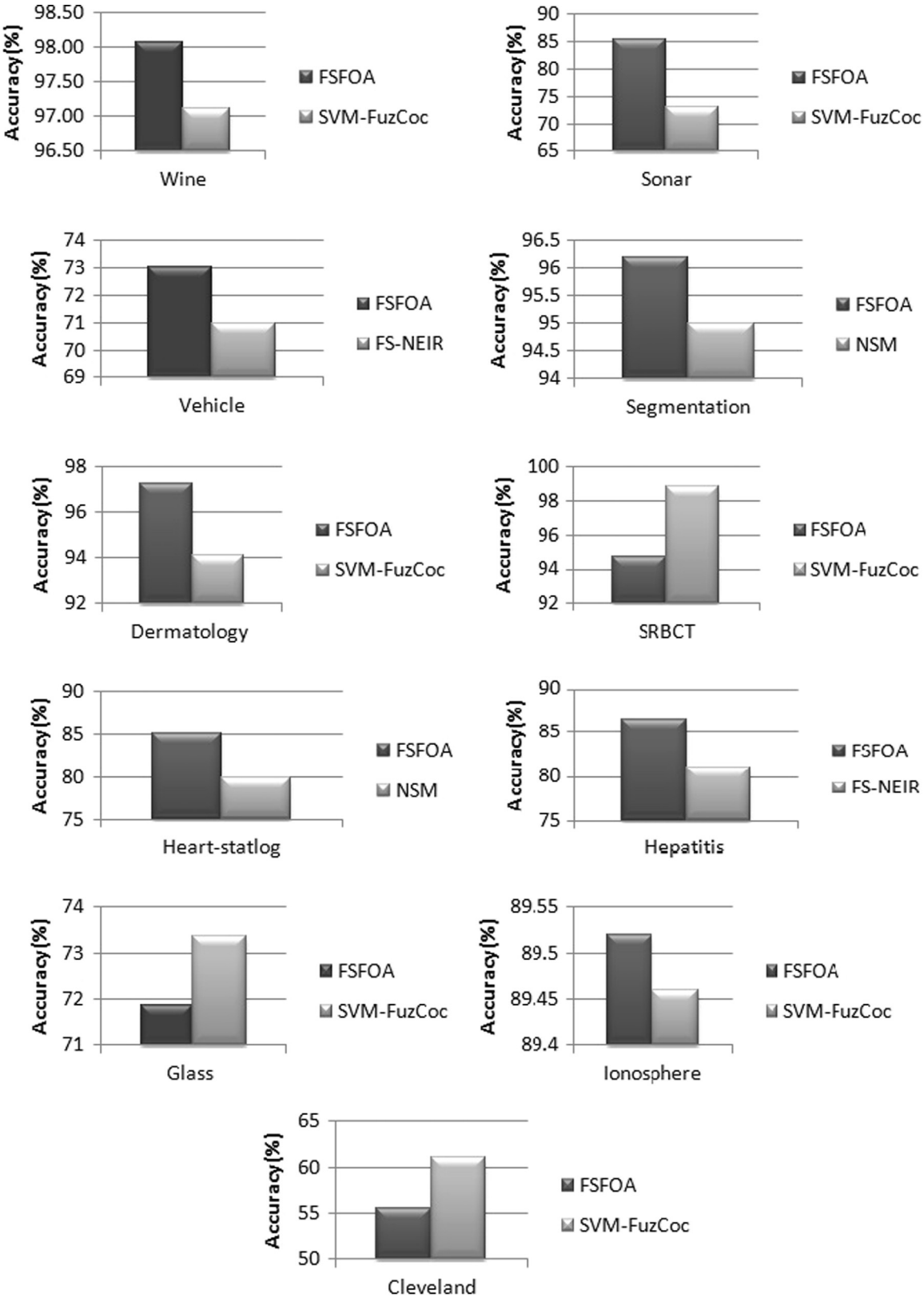

Fig. 5. Graphical comparisons according to Accuracy for each data sets. "Hepatitis" and "Vehicle" are compared according to J48 and the others are compared due to KNN classification accuracy.

5.3. Results and comparisons

We have compared our proposed FSFOA method with some other methods. All the results of our experiments are reported with 95% confidence interval. Feature selection algorithms selected for comparisons are: Neighborhood soft margin (NSM) method proposed by Hu et al. [14], SVM-FuzCoc by Moustakidis et al. [21], hybrid genetic algorithm for FS (HGAFS) by Huang [16],

FS-NEIR which uses a different feature evaluation criterion by Zhu et al. [40], an unsupervised feature selection algorithm based on ant colony optimization (UFSACO) proposed by Tabakhi et al. [29] and PSO(4-2) which is a PSO based method by Xue et al. [35].

Among the methods, HGAFS uses support vector machine. SFS, SBS and SFFS are among greedy methods and are chosen from the article of [21]. SVM-FuzCoc, PSO(4-2) and NSM use 1NN, 5NN and 3NN classifiers respectively. UFSACO and FS-NEIR reported the classification accuracy of J48 classifier. The summary of these methods is shown in Table 3. Also, Table 3 shows the way each method have used the datasets (10 fold cross-validation, 70% training and 30% testing, or 2-fold cross validation for training and testing).

Classification accuracy and dimensionality reduction of FSFOA and the other methods of Table 3 are reported in the tables of Fig. 4. The results reported for FSFOA in Fig. 4 are over 10 independent runs. For each dataset the best classification accuracy and the best dimension reduction (DR) are highlighted in bold form. Dimension Reduction (DR) in Fig. 4 is calculated by Eq. (2). In order to provide fair comparisons, for each dataset multiple results according to dataset splitting with different percentages are reported and they are considered in our comparisons. Also, for each method the used classifier- i.e. KNN, SVM or J48 is presented in each table. Summary of the configuration parameters of the classifiers is presented as Table 4; which indicates that KNN classifier is used with different values for K where needed for comparisons ($K \in \{1, 3, 5\}$) and J48 classifier of Weka is used as a decision tree based method. The kernel function for SVM classifier is radial basis function (Rbf-svm).

As it is obvious from the tables of Fig. 4, Classification Accuracy (CA) of "Heart-statlog", "Ionosphere" and "Segmentation" datasets have improved in comparison with all the selected methods. Among the selected methods there are GA_based and ACO_based methods. This shows that FOA could improve the performance of KNN, J48 and SVM classifier by reducing the redundant features in these datasets. In datasets "Sonar", "Wine", "Hepatitis", "cleveland", "Dermatology", and "Glass", FSFOA could outperform in almost all the selected methods for comparisons; FSFOA has the second rank in these datasets where it couldn't outperform. In "Vehicle" dataset, FSFOA could outperform just one of the methods with the same partitioning and classifier. FSFOA did not show a good performance in "SRBCT" dataset. "SRBCT" dataset is the only dataset where the number of features are much more than the number of samples; this makes it difficult to select the proper features for prediction. As the number of samples are not sufficient for selecting the more informative features and applying the traditional methods yields poor results. Also, this dataset is partitioned to 70%-30% training and testing dataset; this makes the problem worse; because some extend of the dataset is ignored during the training phase. This shows that feature selection in large datasets with many features and where the number of samples is limited is a challenging problem and deserves more research.

Comparing the DR of the methods in Fig. 4, it is obvious that FSFOA couldnot outperform the selected methods; because as we mentioned before, the number of the selected features is not involved in the fitness evaluation of each potential solution and classification accuracy is considered as the fitness function. For better performance illustration, we have shown the results graphically in the charts of Fig. 5. For datasets "Hepatitis" and "Vehicle" the two selected methods are compared according to J48 classifier and for datasets "Dermatology", "Sonar", "SRBCT", "Wine", "Heart-statlog", "Ionosphere", "Glass", "Cleveland" and "Segmentation" KNN classifier is chosen in the graphical comparisons of Fig. 5.

While comparing the results of Fig. 4 and also charts of Fig. 5, FSFOA performs absolutely better than the other methods in 3 datasets ("Heart-statlog", "Ionosphere" and "Segmentation") of 11 ones according to classification accuracy. In datasets "Dermatology", "Sonar", "Wine", "Glass", "Cleveland" and "Hepatitis", FSFOA outperforms many of the other methods except for one method where it has the second rank. In the other two datasets FSFOA couldn't have the desirable performance. Among the selected methods for comparison, there are methods which employ GA, PSO, and ACO; which are well-known algorithms. These results show that FSFOA has acceptable performance in solving feature selection as a real optimization problem.

6. Conclusion

Feature selection is considered to be an important preprocessing step in machine learning and pattern recognition. Many heuristic and meta-heuristic methods have been proposed to address this problem.

In this article, we have attempted to use Forest Optimization Algorithm (FOA) for solving feature selection problem. As FOA is reported to be suitable for continuous search space problems, so we have adjusted the stages of FOA for discrete search space of feature selection problem and proposed FSFOA algorithm.

In order to investigate the performance of FSFOA, we have selected some well-known datasets from the UCI repository and compared the results of FSFOA with other methods. Among the selected methods for comparison, there are GA, ACO and PSO based algorithms. The results of the experiments showed the superiority of our method in most of the selected datasets. In this article, we have used KNN, SVM and J48 classifiers of WEKA software to evaluate the fitness of each potential solution and classification accuracy is considered as our fitness function.

This study shows that FOA is an effective search technique for feature selection problems but further research is also welcomed. For our future research, we will attempt to investigate the performance of FSFOA in very large datasets with huge number of features (e.g. over 10,000); because the size of datasets both in the number of features and instances grows these days and data mining in very large datasets is a big concern. Also, involving the number of the selected features in fitness function with the aim of improving the dimension reduction rate (DR) will be our feature attempt. This can be implemented by multi-objective fitness function which takes into account the classification accuracy and the number of selected features simultaneously.

Conflict of interest

None declared.

References

[1] W. Aha David, Feature weithing for lazy learning algorithms, in: Huan Liu, Hiroshi, Motoda (Eds.), Feature Extraction Construction and Selection: a Data Mining Perspective, Kluwer Academic Publishers, Massachussetts, 1998, 13–32.
[2] Almuallim Hussein, Thomas G. Dietterich, Learning Boolean concepts in the presence of many irrelevant features, Artif. Intell. 69 (1) (1994) 279–305.
[3] H. Almuallim, T.G. Dietterich, Learning with many irrelevant features, in: Proceedings of the AAAI, vol. 91, 1991 July 14, 547–552.
[4] C. Blake, E. Keogh, C.J. Merz, UCI Repository of machine learning databases, University of California, Irvine, ⟨http://www.ics.uci.edu/~mlearn/MLRepository.html⟩.
[5] M. Jose Cadenas, M. Carmen Carrido, Raquel Martinez, Feature subset selection filter-wrapper based on low quality data, Expert Syst. Appl. 40 (2013) 6241–6252.
[6] K.J. Cios, G. William Moore, Uniqueness of medical data mining, Artif. Intell. Med. 26 (1) (2002) 1–24.
[7] M.E. ElAlami, A filter model for feature subset selection based on genetic

algorithm, Knowl.-Based Syst. 22 (5) (2009) 356–362.
[8] E. Gasca, J.S. Sanchez, R. Alonso, Eliminating redundancy and irrelevance using a new MLP-based feature selection method, Pattern Recognit. 39 (2006) 313–315.
[9] Manizheh Ghaemi, Mohammad-Reza Feizi-Derakhshi, Forest optimization algorithm, Expert Syst. Appl. 41 (15) (2014) 6676–6687.
[10] A. Gheyas Iffat, S. Smith Leslie, Feature subset selection in large dimensionality domains, Pattern Recognit. 43 (2010) 5–13.
[11] A. Hall Mark, Correlation-based feature selection for machine learning (Ph.D. thesis), Hamilton, NewZealand, 1999.
[12] A. Hall Mark, Correlation-based feature selection for discrete and numeric class machine learning, in: Proceedings of 17th International Conference on Machine Learning, 2000, 359–366.
[13] M. Hamdani Tarek, Jin-Myung Won, M. Alimi Adel, Karray Fakhri, Hierarchical genetic algorithm with new evaluation function and bi-coded representation for the selection of features considering their confidence rate, Appl. Soft Comput. 11 (2011) 2501–2509.
[14] Q. Hu, X. Che, L. Zhang, D. Yu, Feature evaluation and selection based on neighborhood soft margin, Neurocomputing 73 (10) (2010) 2114–2124.
[15] Q.H. Hu, D. Yu, J.F. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Inf. Sci. 178 (2008) 3577–3594.
[16] J. Huang, Y. Cai, X. Xu, A hybrid genetic algorithm for feature selection wrapper based on mutual information, Pattern Recognit. Lett. 28 (2007) 1825–1844.
[17] Md. Kabir Monirul, Md. Shahjahan, Kazuyuki Murase, A new local search based hybrid genetic algorithm for feature selection, Neurocomputing 74 (2011) 2914–2928.
[18] Md. Kabir Monirul, Md. Shahjahan, Kazuyuki Murase, A new hybrid ant colony optimization algorithm for feature selection, Expert Syst. Appl. 39 (2012) 3747–3763.
[19] Ron Kohavi, H. John George, Wrappers for feature subset selection, Artif. Intell. 97 (12) (1997) 273–324.
[20] N. Lavrac, Selected techniques for data mining in medicine, Artif. Intell. Med. 16 (1) (1999) 3–23.
[21] S.P. Moustakidis, J.B. Theocharis, SVM-FuzCoC: a novel SVM based feature selection method using a fuzzy complementary criterion, Pattern Recognit. 43 (2010) 3712–3729.
[22] Shahla Nemati, Mohammad Ehsan Basiri, Nasser Ghasem Aghaee, Mehdi Hosseinzadeh Aghdam, A novel ACO-GA hybrid algorithm for feature selection in protein function prediction, Expert Syst. Appl. 36 (2009) 12086–12094.
[23] G.A. Papakostas, A.S. Polydoros, D.E. Koulouriotis, V.D. Tourassis, Evolutionary Feature Subset Selection for Pattern Recognition Applications, INTECH Open Access Publisher, 2011.
[24] P. Pudil, J. Novovicov, J. Kittler, Floating search methods in feature selection, Pattern Recognit. Lett. 15 (1994) 1119–1125.
[25] Bart Selman, Carla P. Gomes, Hill-climbing Search, Encyclopedia of Cognitive Science, 2006.
[26] B. Selman, H.J. Levesque, D.G. Mitchell, A new method for solving hard satisfiability problems, in: Proceedings of the AAAI, vol. 92, July 12, 1992, 440–446.
[27] O. Seral, S. Gunes, Attribute weighting via genetic algorithms for attribute weighted artificial immune system (AWAIS) and its application to heart disease and liver disorders problems, Expert Syst. Appl. 36 (2009) 386–392.
[28] Rahul Karthik Sivagaminathan, Sreeram Ramakrishnan, A hybrid approach for feature subset selection using neural networks and ant colony optimization, Expert Syst. Appl. 33 (2007) 49–60.
[29] S. Tabakhi, P. Moradi, F. Akhlaghian, An unsupervised feature selection algorithm based on ant colony optimization, Eng. Appl. Artif. Intell. 32 (2014) 112–123.
[30] M.A. Tahir, A. Bouridane, F. Kurugollu, Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K nearest neighbor classifier, Pattern Recognit. Lett. 28 (2007) 438–446.
[31] K.C. Tan, E.J. Teoh, Q. Yu, K.C. Goh, A hybrid evolutionary algorithm for attribute selection in data mining, Expert Syst. Appl. (2009) 8616–8630.
[32] A. Tosun, B. Turhan, A.B. Bener, Feature weighting heuristics for analogy-based effort estimation models, Expert Syst. Appl. 36 (2009) 10325–10333.
[33] I. Triguero, J. Derrac, S. Garca, F. Herrera, Integrating a differential evolution feature weighting scheme into prototype generation, Neurocomputing 97 (2012) 332–343.
[34] Dietrich Wettschereck, David W. Aha, Takao Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, Artif. Intell. Rev. 11 (1997) 273–314.
[35] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2013) 261–276.
[36] Zhi-Min Yang, Jun-Yun He, Yuan Hai Shao, Feature selection based on linear twin support vector machine, Proc. Comput. Sci. 17 (2013) 1039–1046.
[37] J.Y. Yeh, T.H. Wu, C.W. Tsao, Using data mining techniques to predict hospitalization of hemodialysis patients, Decis. Support Syst. 50 (2) (2011) 439–448.
[38] Y. Zhang, A. Yang, C. Xiong, T. Wang, Z. Zhang, Feature selection using data envelopment analysis, Knowl.-Based Syst. 64 (2014) 70–80.
[39] Mingyuan Zhao, Chong Fu, Luping Ji, Ke Tang, Mingtian Zhou, Feature selection and parameter optimization for support vector: a new approach based on genetic machines algorithm with feature chromosomes, Expert Syst. Appl. 38 (5) (2011) 5197–5204.
[40] Wenzhi Zhu, Si Gangquan, Zhang Yanbin, Wang Jingcheng, Neighborhood effective information ratio for hybrib feature evaluation and selection, Neurocomputing 99 (2013) 25–37.
[41] Zexuan Zhu, Yew-soon Ong, Manoranjan Dash, Wrapper-filter feature selection algorithm using a memetic framework, IEEE Trans. Syst., Man, Cybern. 37 (2007) 70–76.
[42] Alexandros Kalousis, Julien Prados, Melanie Hilario, Stability of feature selection algorithms: a study on high-dimensional spaces, Knowl. Inf. Syst. 12 (1) (2007) 95–116.

Manizheh Ghaemi received her B.S. and M.S. degrees in Computer Science from the University of Tabriaz, Iran. She is now a Ph.D. student for Artificial Intelligence in K.N. Toosi University of Technology, Iran. Her research interests include nature-based evolutionary algorithms, optimization and machine learning algorithms.

Mohammad-Reza Feizi-Derakhshi received his B.S. in Software Engineering from the University of Isfahan. He received his M.S. and Ph.D. in AI from the Iran University of Science and Technology. He is currently a faculty member at the University of Tabriz. His research interests include: NLP, optimization algorithms and intelligent databases.