Knowledge-Based Systems 84 (2015) 144–161

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Integration of graph clustering with ant colony optimization for feature selection

Parham Moradi *, Mehrdad Rostami

Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

ARTICLE INFO

Article history:
Received 18 August 2014
Received in revised form 16 January 2015
Accepted 6 April 2015
Available online 9 April 2015

Keywords:
Feature selection
Ant colony optimization
Filter method
Graph clustering

ABSTRACT

Feature selection is an important preprocessing step in machine learning and pattern recognition. The ultimate goal of feature selection is to select a feature subset from the original feature set to increase the performance of learning algorithms. In this paper a novel feature selection method based on the graph clustering approach and ant colony optimization is proposed for classification problems. The proposed method's algorithm works in three steps. In the first step, the entire feature set is represented as a graph. In the second step, the features are divided into several clusters using a community detection algorithm and finally in the third step, a novel search strategy based on the ant colony optimization is developed to select the final subset of features. Moreover the selected subset of each ant is evaluated using a supervised filter based method called novel separability index. Thus the proposed method does not need any learning model and can be classified as a filter based feature selection method. The proposed method integrates the community detection algorithm with a modified ant colony based search process for the feature selection problem. Furthermore, the sizes of the constructed subsets of each ant and also size of the final feature subset are determined automatically. The performance of the proposed method has been compared to those of the state-of-the-art filter and wrapper based feature selection methods on ten benchmark classification problems. The results show that our method has produced consistently better classification accuracies.

1. Introduction

In recent years, with the advance of science and technology, the amount of data has been growing rapidly and thus pattern recognition methods often deal with samples consisting of thousands of features. This problem is called curse of dimensionality and reduction of the datasets' dimensionality becomes crucial to make them tractable [7,45,61]. Dimensionality reduction methods provide a way of understanding the data better, improving prediction performance and reducing the computation time in pattern recognition applications. As a general rule, for a classification problem with D dimensions and C classes, a minimum of $10 \times D \times C$ training samples are required [9]. While it is practically impossible to acquire the required number of training samples, reducing features reduces the size of the training sample required and consequently helps to improve the overall performance of the classification algorithm.

A common way to deal with such problems is the feature selection technique. The feature selection methods can be classified into four categories including filter, wrapper, embedded, and hybrid models [51]. The filter approach requires the statistical analysis of the feature set without utilizing any learning algorithm. In contrast, wrapper-based feature selection methods apply a learning algorithm to evaluate the quality of feature subsets in the search space iteratively. In the embedded model the feature selection procedure is considered as a part of the process of building models. On the other hand, the goal of the hybrid-based methods is to use the computational efficiency of the filter model and the proper performance of the wrapper model.

In recent years many evolutionary and swarm-based methods such as Genetic Algorithm (GA) ([8,56], ant colony optimization (ACO) [1,11,17,54,60]), particle swarm optimization (PSO) [27,58] and Artificial Bee Colony (ABC) [36,44] and Harmony Search algorithm (HSA) [62] have been utilized to tackle the feature selection problem. Among the swarm intelligence-based methods, ACO has been successfully used in the feature selection area of research. The ACO is a metaheuristic algorithm for solving hard combinatorial optimization problems [35]. This algorithm has been successfully applied to a large number of difficult combinatorial

* Corresponding author. Tel.: +98 8733668513.
E-mail addresses: p.moradi@uok.ac.ir (P. Moradi), me.rostami@gmail.com (M. Rostami).

http://dx.doi.org/10.1016/j.knosys.2015.04.007
0950-7051/© 2015 Elsevier B.V. All rights reserved.

problems such as vehicle routing [47], graph coloring [16], and communication network [68]. ACO is a multi-agent system and it has some advantages such as positive feedback, the use of a distributed long-term memory, nature implementation in a parallel way, functions similar to those of reinforcement learning schemata, and a good global and local search capability due to stochastic and greedy components in the algorithm. Moreover, ACO has been successfully applied for feature selection problem [1,11,12,17,26,31,32,37,52,54,67]. Although, ACO has been shown as an effective approach to finding optimal (or near optimal) feature subsets while it suffers from several shortcomings which are listed as follows:

1. *Graph representation*: In many ACO-based feature selection methods the problem space is represented by a fully connected graph except a work [11] in which the problem space was represented by a directed graph with only $2n$ arcs where n denotes the number of features. In the case of fully connected graphs, in each step (i.e., step t) each ant should compute the probability rule for unselected features (i.e., $n-t+1$, where n denotes the number of features) which leads to increase the time complexity of the algorithm. For example if the ant needs to traverse m numbers of nodes in the graph, $(n)!/(n-m)!$ computations are needed; therefore one can reduce these computations.
2. *Updating pheromone*: Most ACO-based feature selection methods employed a learning model in their search process to evaluate a constructed feature subset, and thus they are classified as the wrapper model [1,37,60]. While, the wrapper based methods need high computational time especially on the datasets with a large number of features. On the other hand, only in three cases instead of the learning model, information theoretic measures are used to update the pheromone [32,48,52,54].
3. *Selecting redundant features*: In most of ACO-based feature selection methods, the possible dependency between the features is ignored in the search process [1,11,60]. These methods assume that the features are conditionally independent and thus, while the ant selects the next feature, the dependency of the feature on previously selected ones is ignored in the computations. Therefore, the constructed subset may contain the redundant features, which reduces the classifier performance.
4. *Final subset size*: The number of selected features which defines traverse path length of the ants, imposes another challenge on ACO-based methods. In most of the ACO-based feature selection methods the number of traversed nodes should be pre-determined before the ant starts their search processes [48,52,54]. Moreover the accuracy of these methods depends on optimally defining the size of feature subset.

To overcome the mentioned shortcomings, in this paper we propose a novel filter-based feature selection method based on ACO algorithm. The method attempts to select high-quality features within a reasonable time. The proposed algorithm which is called Graph Clustering based ACO feature selection method, in short GCACO, works in three steps. In the first step, the problem space is represented as a graph in which each node denotes a feature and the edges weights are similarities between features. In the second step, features are divided into several clusters by employing an efficient community detection algorithm [59]. Finally in the third step, a novel ACO-based search strategy is proposed for selecting the final feature subset. In this strategy an ant which is placed on a randomly selected cluster, in each step, decides to select the next position in the current cluster or move to another cluster. In the case of remaining in the same cluster, the probability values are computed only for the features of this cluster. In contrast for the other cases the probability values are computed for the features of the next selected cluster. This process is continued until all of the clusters are visited. Therefore, the number of features which are selected by each ant in each cycle and also the final feature subset can be automatically determined based on the number of clusters in the problem space.

This approach is quite different from those of the existing schemes [48,52,54], where the size of the constructed subset is defined by a fixed number. Furthermore, the aim of using community-based representations of the problem space is to group highly correlated features into the same cluster. Therefore the ACO-based search process is guided in such a way that relatively less correlated features are injected in a high proportion with respect to more correlated features to the consecutive iteration. Besides, the similarity between features is considered in computation of feature relevance, which minimizes the redundancy between selected features. Therefore, the clustering-based strategy of the proposed method has a high probability of identifying a subset of useful and independent features. Moreover, clustering the features in the problem space results in reduction of the computation complexity of probability values because when an ant is placed in a given cluster, the probability value is computed only for features in the current cluster. Furthermore, unlike most of the existing ACO-based feature selection methods which use a learning algorithm [1,11,37] to evaluate the constructed subsets, in this paper a feature subset is evaluated by means of a separability index matrix without using any learning models. Therefore the proposed method can be classified as a filter-based approach and thus it will be computationally efficient for high-dimensional datasets.

The rest of this paper is organized as follows. Section 2 reviews related works on feature selection. A detailed description of our proposed method, including the complexity analysis of the different steps, is presented in detail in Section 3. In Section 4 we compare the proposed algorithm with other existing feature selection methods. Finally, Section 5 summarizes the present study.

2. Related works

The main idea behind feature selection is to choose a subset of available features, by eliminating irrelevant features with little or no predictive information, as well as redundant features that are strongly correlated. To find the optimal feature subset one needs to enumerate and evaluate all the possible subsets of the features. The entire search space contains all the possible subsets of features, meaning that the search space size is 2^n where n is the dimensionality of the problem (i.e., the number of original features). Therefore, the problem of finding the optimal feature subset is NP-hard [10,19]. Since evaluating the entire feature subsets is computationally expensive, time consuming and also impractical even for moderate sized feature sets, the final solution should be found in a feasible computational time with a reasonable trade-off between the quality of the found solution and time–space cost. Therefore, many feature selection algorithms involve heuristic or random search strategies to find the optimal or near optimal subset of features in order to reduce the computational time. Feature selection is a fundamental research topic in machine learning with a long history since the 1970s, and there are a number of attempt to review the feature selection methods [10,51].

The feature selection methods can be classified into four categories including filter, wrapper, embedded, and hybrid models. The filter approach requires only a statistical analysis on a feature set for solving the feature selection task without utilizing any learning algorithms. Therefore, the methods in this approach are typically fast. The filter-based feature selection methods can be classified into univariate and multivariate methods. In the univariate methods, the informativeness of each feature is evaluated individually, according to a specific criterion, such as the Information gain [33], Gain Ratio [39], Term Variance [55], Gini index [42], Laplacian Score (L-Score) [64] and Fisher Score (F-Score) [41]. This means that each feature is considered separately, thereby ignoring feature dependencies, which may lead to worse classification performance when compared to other types of feature selection methods. In order to overcome the problem of ignoring feature dependencies, a number of multivariate filter methods were introduced, aiming at the incorporation of feature dependencies to some degree. Multivariate approaches, on the contrary, evaluate the relevance of the features considering how they function as a group, taking into account their dependencies.

The wrapper model uses the learning algorithm as a predictor and the predictor performance as the objective function to evaluate the feature subset. The wrapper-based methods can be broadly classified into sequential feature selection algorithms (SFS) and heuristic search algorithms. The sequential forward (backward) selection algorithms start with an empty set (full set) and add features (remove features) until the maximum objective function is obtained. On the other hand, the heuristic search algorithms evaluate different subsets to optimize the objective function. Different subsets are generated either by searching around in a search space or by generating solutions to the optimization problem. The main drawback of the wrapper model is the number of required computations to obtain the feature subset. To evaluate the feature subset, a learning algorithm is trained for each subset and tested to obtain the classifier accuracy. Therefore, most of the execution time of the algorithm is spent on the training of the predictor when dealing with high-dimensional datasets.

The hybrid model is a combination of the filter and the wrapper models and attempts to take advantage of both approaches. The hybrid model mainly focuses on combining the filter and the wrapper-based methods to achieve the best possible performance with a particular learning algorithm with time complexity similar to that of the filter-based methods. Moreover, the embedded model tries to include the feature selection as a part of the classifier training process, like inherently binary decision tree classifiers do. In other words, the feature selection process is embedded into the training of the learning algorithm.

Feature selection methods are modeled using different sorts of optimization algorithms such as swarm intelligence (SI) or evolutionary algorithms (EAs). Swarm intelligence has become a research interest to many research scientists of related fields in recent years. Swarm intelligence is a computational intelligence-based approach which is made up of a population of artificial agents and inspired by the social behavior of animals in the real world. Each agent performs a simple task, while the colony's cooperative work will solve a hard problem. In this section, feature selection algorithms relying on the swarm intelligent methods such as particle swarm optimization (PSO), artificial bee colony optimization (ABC), differential evolution (DE), gravitational search algorithm (GSA), harmony search algorithm (HSA) and ant colony optimization (ACO) are reviewed and outlined in Table 1.

Particle swarm optimization is a powerful swarm-based metaheuristic method, proposed by Kennedy and Eberhart in 1995 [28]. PSO is motivated by social behaviors such as bird flocking and fish schooling. The particle swarm optimization method has recently gained more attention for solving the feature subset selection problem [6,25,27,58,65]. Unler et al. [58] proposed a hybrid method called maximum relevance minimum redundancy PSO (mr2PSO), which integrates the mutual information-based filter model within the PSO-based wrapper model. Another hybridized approach was proposed by Inbarani et al. [27] to solve the selection of appropriate features for the medical diagnosis problems. In this method, two hybrid supervised PSO-based feature selection methods called PSO-based Relative Reduct (PSO-RR) and PSO based Quick Reduct (PSO-QR) are proposed. Moreover, Huang and Dun [25] proposed a PSO–SVM model that hybridizes the PSO and support vector machines (SVMs) to improve the classification accuracy with a small and appropriate feature subset. Furthermore, Xue et al. [65] proposed three initialization strategies and several updating mechanisms in PSO to develop feature selection approaches. The goal of the method was to select smaller numbers of features as well as to achieve better classification performance.

Table 1
Outlining the reviewed papers.

Authors	SI approach	Evaluation criteria
Unler et al. [58]	PSO	Wrapper
Inbarani et al. [27]	PSO	Wrapper
Xue et al. [65]	PSO	Wrapper
Raymer et al. [46]	GA	Wrapper
Ho et al. [24]	GA	Wrapper
Ros et al. [50]	GA	Wrapper
Kabir et al. [38]	GA	Wrapper
Uğuz [57]	GA	Wrapper
Schiezaro and Pedrini [36]	ABC	Wrapper
Forsati et al. [44]	BCO	Wrapper
Al-Ani et al. [3]	DE	Wrapper
Han et al. [21]	GSA	Hybrid
Wang et al. [62]	HS	Wrapper
Kabir et al. [37]	ACO	Wrapper
Vieira et al. [60]	ACO	Wrapper
Li et al. [67]	ACO	Wrapper
Chen et al. [11]	ACO	Wrapper
Forsati et al. [17]	ACO	Wrapper
Ke et al. [32]	ACO	Filter
Tabakhi et al. [54]	ACO	Filter

Genetic algorithm (GA) is one of the most widely used techniques for feature selection problem [2,13–15,29,38,43,46,50,53,56,57,63,66]. In [46], a genetic algorithm is used simultaneously for feature selection and extraction and training classifier. Ho et al. [24] designed an intelligent genetic algorithm (IGA) to tackle both instance and feature selection problems simultaneously by introducing a special orthogonal cross operator. Again, in Ramirez-Cruz et al. [43] GA and evolution strategies are combined to select instances and weight the features. Similarly, in Ros et al. [50] a hybrid genetic approach is proposed which treats feature and instance selection problems as a single optimization problem. Ahn and Kim [2] used a genetic algorithm method to simultaneously optimize feature weighting and instance selection for case-based reasoning in the bankruptcy prediction problem. In Kabir et al. [38] a hybrid genetic algorithm with a specific local search is proposed for feature selection. In Uğuz [57] a two-stage feature selection method is proposed for text categorization by using information gain, principal component analysis and genetic algorithm. A more comprehensive study of feature selection algorithms based on genetic algorithm is provided in Tsai et al. [13].

The artificial bee colony algorithm is one of the recently introduced swarm-based algorithms. This algorithm was proposed to simulate the intelligent foraging behavior of honey bee swarms. This method is also used for solving feature selection tasks in several studies [36,44]. Schiezaro and Pederini [36] proposed a feature selection method for data analysis based on the ABC algorithm that can be used in several knowledge domains through wrapper and forward strategies. Moreover, in Forsati et al. [44] the feature selection task is formulated as an optimization problem and a feature selection procedure based on Bee Colony Optimization (BCO) is proposed in order to achieve better classification results. The Differential Evolution (DE) algorithm is a well-known population-based algorithm which has been successfully applied to a variety of pattern recognition applications as well as the feature selection problem. Al-Ani et al. [3] proposed a wrapper-based feature selection method using differential evolution. The aim of the method is to reduce the search space using a simple, yet powerful, procedure that involves distributing the features among a set of wheels. Moreover, Han et al. [21] presented a feature subset selection search procedure using a modified gravitational search algorithm (GSA). Recently, a feature selection method based on harmony search algorithm (HS) is proposed for email classification [62].

In recent years, some ACO-based methods for feature selection have been reported. Aghdam et al. [1] proposed an ACO-based feature selection algorithm to improve the performance of the text categorization problems. In their method, the classifier performance and the length of the selected feature subset are used to adopt the heuristic information of the ACO. Moreover, Kabir et al. [37] proposed a hybrid ACO-based feature selection algorithm, called ACOFS, which uses the information gain method to define the heuristic information of the ACO for each feature and the neural network predictor to evaluate the results of each ant. Furthermore, Vieira et al. [60] proposed an ACO-based feature selection algorithm which uses two cooperative ant colonies. The goal of the first colony is to determine the number of features while the aim of the second one is to select the features based on cardinality given by the first colony. In Li et al. [67], a two-stage ACO-based feature selection called ACO-S was proposed to apply to microarray datasets. In the first stage of the method an ant system is used to filter the non-significant genes and in the second stage an improved ant colony system is applied to gene selection. Most of the existing ACO-based feature selection methods need to traverse a complete graph with $O(n^2)$ edges where n is the number of original features. However, Chen et al. [11] present a different feature ACO-based algorithm in which the artificial ants traverse on a directed graph with only $O(2n)$ arcs. Recently, in Forsati et al. [17], a new variant of ACO, called enRiched Ant Colony Optimization (RACO) was proposed for the feature selection task. In this method the information contained in the traversals of the previous iterations is modeled as a rich source that guides the ant's further path selection and pheromone updating stages. The mentioned ACO-based methods are based on the wrapper model which needs a learning algorithm to evaluate the results of each ant. On the other hand, in Ke et al. [32] a filter ACO-based algorithm called ACOAR was proposed to deal with attribute reduction in rough set theory. Recently, in Tabakhi et al. [54] a filter method based on the ACO algorithm, called UFSACO, was proposed. The method seeks the optimal feature subset through several iterations without using any learning algorithms.

3. Proposed method

In this section a novel method is described which can efficiently and effectively deal with both irrelevant and redundant features. The proposed method consists of three steps: (1) Graph representation of the problem space, (2) Feature clustering and (3) Searching for the optimal feature subset based on ACO. In the first step the feature set is represented as a graph in which each node in the graph denotes a feature and each edge weight indicates the similarity value between its corresponding features. In the second step, the features are divided into several clusters using a community detection method [59]. The goal of features clustering is to group most correlated features into the same cluster. In the third step a novel feature selection algorithm based on the search strategy of the ACO is proposed to select the final feature subset. Fig. 1 illustrates the overall schema of the proposed three-step method. In Fig. 1(a) the feature space is represented as a weighted graph in which the nodes represent the features, and also the edges denote that the similarity value between the corresponding two features. After applying the graph clustering method, the features were grouped into three clusters. Fig. 1(b) shows the clustering result for the graph. Finally Fig. 1(c) shows that the ants search for the optimal feature subset in the different groups of the features. For example, as can be seen from Fig. 1(c) the ant starts to move from cluster 1 and selects feature F_1 from this cluster. It should be noted that the features in the clusters are selected based on their probability values which can be obtained by applying the fisher score measure and considering the similarity with the previously selected features. Then in the next step, the ant has two choices: remaining in the same cluster or going to another cluster. In the case of remaining in the same cluster, the ant starts to choose remaining features based on their probability values. On the other hand while the ant wants to go to another cluster, the next feature will be selected in this cluster. It can be seen from Fig. 1(c) that the ant goes to the cluster 2 and selects feature F_6 from this cluster. Moreover, in the third step the ant decides to select the next position in the current cluster (i.e., cluster 2) and selects feature F_5. Then in the fourth step feature F_8 from cluster 3 is selected. Finally, all of the clusters have been traversed and the ant has constructed the feature subset (i.e., $\{F_1, F_6, F_5, F_8\}\}$). The obtained subset is evaluated using the separability index matrix method and there is no need for any learning models. The additional details are described in the corresponding subsections.

3.1. Graph representation

In general, to apply the ACO algorithm, the search space of the feature selection problem should be represented by a fully connected undirected graph. Thus, we attempt to model the feature selection problem using a graph theoretic representation. To this end, the feature set is mapped into its equivalent graph $G = (F, E, w_F)$, where $F = \{F_1, F_2, \ldots, F_n\}$ is a set of original features, $E = \{(F_i, F_j) : F_i, F_j \in F\}$ denotes the edges of the graph and w_{ij} indicates the similarity between two features F_i and F_j connected by edge (F_i, F_j). The methods for measuring the similarity values (i.e., edge weights) critically determine the performance of the subsequent graph-based feature selection algorithm. There are different similarity measures that can be used to determine the edge weights and different methods may lead to different results. Therefore, we need to carefully select the most suitable measure. Generally, the Euclidean distance and *Pearson's correlation coefficient* are both widely used as similarity measures. In this work, the Pearson correlation coefficient measure is used to measure the similarity value between different features of a given training set. The correlation between two features F_i and F_j is defined as follows:

$$w_{ij} = \left| \frac{\sum_p (x_i - \overline{x_i})(x_j - \overline{x_j})}{\sqrt{\sum_p (x_i - \overline{x_i})^2} \sqrt{\sum_p (x_j - \overline{x_j})^2}} \right| \tag{1}$$

where x_i and x_j denote the vectors of features F_i and F_j, respectively. Variables $\overline{x_i}$ and $\overline{x_j}$ represent the mean values of vectors x_i and x_j, averaged over p samples. It is clear that the similarity value between two features which are completely similar will be equal to 1, and on the other hand for completely dissimilar features this value will be equal to 0. In most cases, the similarity values between features in these datasets are so close to each other. To overcome this situation, a nonlinear normalization method called softmax scaling [55] is used to scale the edge weight into the range [0 1] as follows:

$$\hat{w}_{ij} = \frac{1}{1 + \exp\left(-\frac{w_{ij} - \bar{w}}{\sigma}\right)} \tag{2}$$

where w_{ij} the similarity value between features F_i and F_j, $\bar{w}$ and σ are respectively the mean and variance of indicates all of the

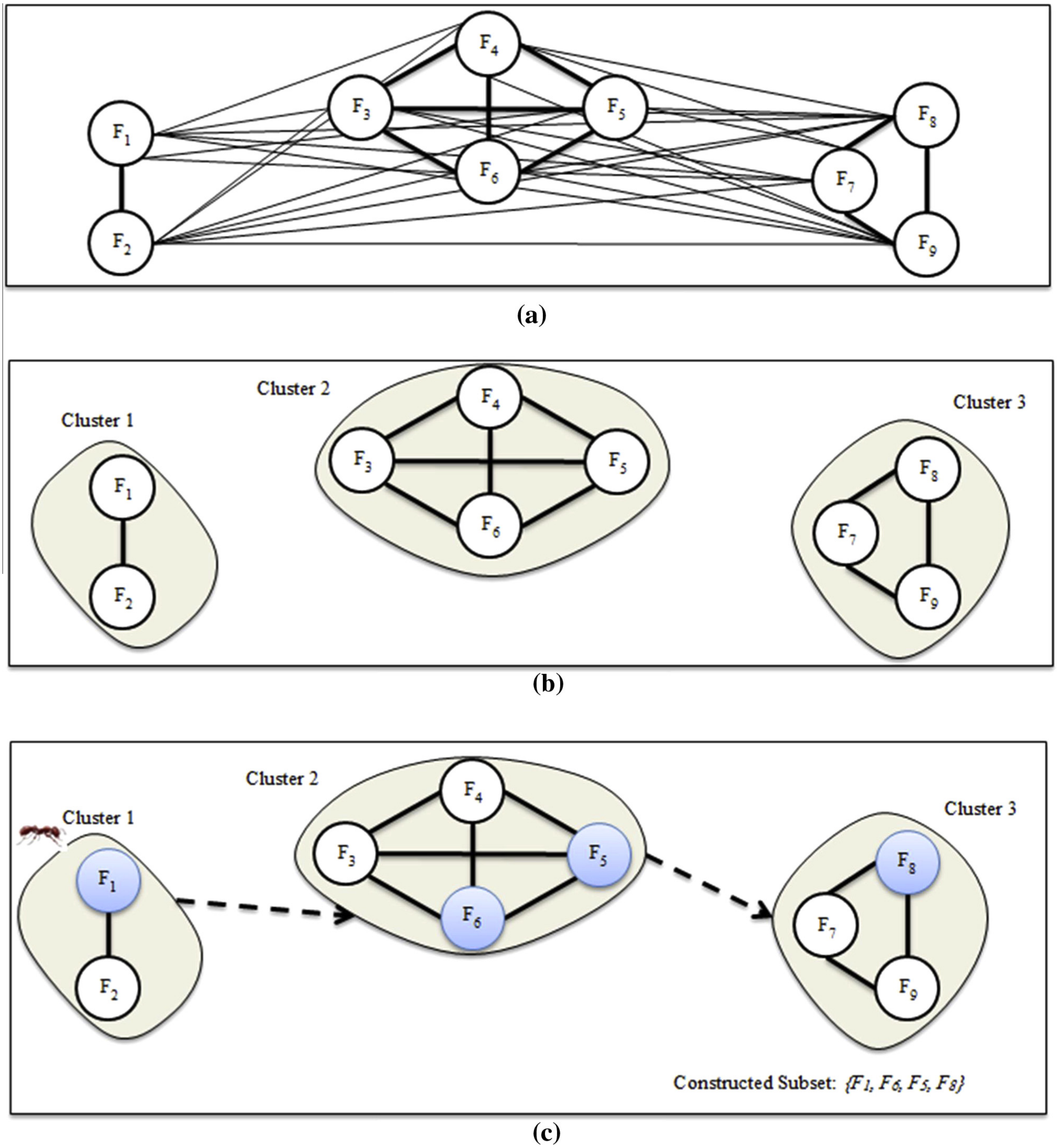


Fig. 1. Illustration of the proposed graph clustering ACO method for feature selection. (a) Graph representation of the original set. (b) Group the features into three clusters. (c) ACO based search for the optimal feature subset.

similarity values, and $\hat{w}_{ij}$ indicates the normalized value of the similarity between features F_i and F_j.

3.2. Feature clustering

The main idea behind feature clustering is to group the original features into several clusters based on the similarity values between features. Therefore, the features in the same cluster are similar to each other. Most of the existing feature clustering methods suffer from some shortcomings [30]. First, indicating the desired number of clusters, the parameter k, has to be specified in advance. Generally defining the proper number of clusters needs exhaustive trial-and-error. Second, the distribution of the data in a cluster is an important factor and the existing methods do not consider the variance of the underlying cluster in similarity computation. Third, all features in a cluster have the same degree of contribution to the resulting extracted feature. To deal with these issues in this paper a community detection method is applied to cluster the features.

Detection of communities in the weighted graph is significant for understanding the graph structures and analysis of the graph properties. The goal of community detection in this study is to cluster the similar highly correlated features into the same community and separate them from the others. In this work, we have used the Louvain community detection algorithm to identify the feature clusters [59]. This algorithm detects the communities in the graph by maximizing a modularity function. This is a simple, efficient and easy-to-implement method for identifying communities in large networks. The computational complexity of the algorithm is $O(n\log n)$, where n is the number of the nodes in the graph, so it can be used to detect communities in very large networks within short computing times. The method detects communities of a network in two steps. In the first step each node is assigned to a community chosen in order to maximize specific network modularity; and the second step simply makes a new network by merging those of the previously found communities. Then the process iterates until a significant improvement of the network modularity is obtained. This method has two advantages. First, its steps are intuitive and easy to implement, and second, the algorithm is extremely fast. It should be noted that in the previous step the problem space was represented by a fully connected graph. Each edge in the graph was associated with a value which denoted the similarity value between every two nodes. Therefore, before using the clustering method, the edges with associated weights lower than the θ parameter will be removed to improve the performance of the clustering method. The θ parameter can be set to any value in the range [0 1], and thus when its value is small (large), more (fewer) edges will be considered in the graph clustering algorithm and the number of obtained clusters will be low (high). Fig. 2 illustrates the feature clustering algorithm for *Sonar* dataset. In Fig. 2(a) the feature space is represented as a weighted complete graph. After removing edge with associated weights lower than the θ parameter the complete graph converted into sparse graph. Fig. 2(b) shows this sparse graph. Finally Fig. 2(c) shows the clustering result for the graph.

3.3. ACO-based search strategy

In this subsection a novel ACO-based search strategy is proposed to select a feature subset from a clustered graph. The algorithm is composed of several iterations. Before the iterations start, the amount of pheromone assigned to each node is initialized to a constant value γ. Also, the discrimination power of feature F_i is evaluated using the Fisher score as follows:

$$F\text{-}Score(F_i) = \frac{\sum_{k=1}^{c} n_i\left(\bar{x}_i^k - \bar{x}_i\right)^2}{\sum_{k=1}^{c} n_i\left(\sigma_i^k\right)^2} \tag{3}$$

where c is the number of classes of the dataset, n_i is the number of samples in class i, $\bar{x}_i$ shows the mean of all the patterns corresponding to feature F_i, and also $\bar{x}_i^k$ and σ_i^k denote the mean and variance of class k corresponding to feature F_i. A larger $F\text{-}Score(F_i)$ value implies that feature F_i has a greater discriminative ability. Similarly to correlation values between features, all *F-Scores* are normalized using softmax scaling method [55]. Then, in each iteration, each ant starts to move from a random cluster and then it selects a feature from a given cluster. In this case the ant decides to choose the next feature from the current cluster or go to the different cluster. To this end a random value is generated and then if the generated value is lower than a predefined ε parameter, the ant chooses the next feature from a different cluster; otherwise the ant will remain in the current cluster. The ant continues to move until all the clusters are selected. In the case of remaining in the current cluster, the ant selects the next feature based on a specified ACO-based probability rule. When all of the ants have finished their traverse on the graph, the quality of each solution (i.e., feature subset) is evaluated by applying a separability index and then the amount of pheromone of each node is updated by applying a specified ACO-based updating rule.

The process is repeated until a given number of iterations; then, the features are sorted based on their pheromone values in decreasing order. Finally, the top $k \times \omega$ features with the highest pheromone values are selected as the final feature subset, where k denotes the number of clusters and ω is a user-specified parameter that controls the size of the final feature subset. The details of our ACO-based search strategy algorithm are provided in Fig. 3.

3.3.1. Probabilistic decision rule

In ACO a probabilistic decision rule, denoting the probability of selection of the nodes, is designed by combining the heuristic desirability and the pheromone density values of the nodes. In the proposed method, each ant traverses the graph using both greedy and probabilistic state transition rules. In the greedy method, the kth ant chooses the next feature F_j applying the following formula:

$$F_j = \underset{F_u \in UF_i^k}{\arg\max}\{[\tau_u]^{\alpha}[\eta(F_u, VF_k)]^{\beta}\}, \quad \text{if } q \leqslant q_0 \tag{4}$$

where UF_i^k is the set of unvisited features by ant k from the current cluster i, τ_u denotes the pheromone intensity value associated with feature F_u, VF_k is denotes the previously selected features (visited features) by ant k, $\eta(F_u, VF^k)$ indicates the heuristic information function, parameters α and β determine the importance of the pheromone versus the heuristic information value, q is a random number in the range [0, 1], and q_0 is a predefined constant parameter $(0 \leqslant q_0 \leqslant 1)$.

In this paper a specific heuristic information function is proposed. According to Eq. (5), features selected by ants are those with both minimum similarity to previously selected features and maximum dependency on the target class. This selection rule results in lower probability for redundant and irrelevant features to be selected. This function is defined as follows:

$$\eta(F_i, VF_k) = \left[F\text{-}Score(F_i) - \frac{1}{|VF_k|}\sum_{F_x \in VF_k} sim(F_i, F_x)\right] \tag{5}$$

where $sim(F_i, F_x)$ denotes the similarity value between feature F_i and features F_x. In the probabilistic method, the kth ant selects the next feature F_j with a probability of $P_k(VF_k, F_j)$ which is calculated as follows:

$$P_k(F_j, VF_k) = \begin{cases} \frac{[\tau_j]^{\alpha}[\eta(F_j, VF_k)]^{\beta}}{\sum_{u \in UF_i^k}[\tau_u]^{\alpha}[\eta(F_u, VF_k)]^{\beta}}, & \text{if } j \in UF_i^k, \quad \text{if } q > q_0 \\ 0, & \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{6}$$

The probabilistic decision rules (i.e., Eqs. (4) and (6)) depend on parameters q and q_0, which aims to provide a proper balance between exploration and exploitation. If $q \leqslant q_0$ the ants select the best feature in the greedy way (i.e., exploitation); otherwise, each feature has a chance of being selected corresponding to its probability value which is computed using equation Eq. (6) (i.e., exploration). It should be noted that the exploration property of the search process prohibits the ants from converging on a common path.

3.3.2. Pheromone updating rule

At the end of each iteration, when all ants have completed their traverses on the graph, the pheromone level of each feature is updated by applying the following updating rule:

$$\tau_i(t+1) = (1-\rho)\tau_i(t) + \sum_{k=1}^{A}\Delta_i^k(t) \tag{7}$$

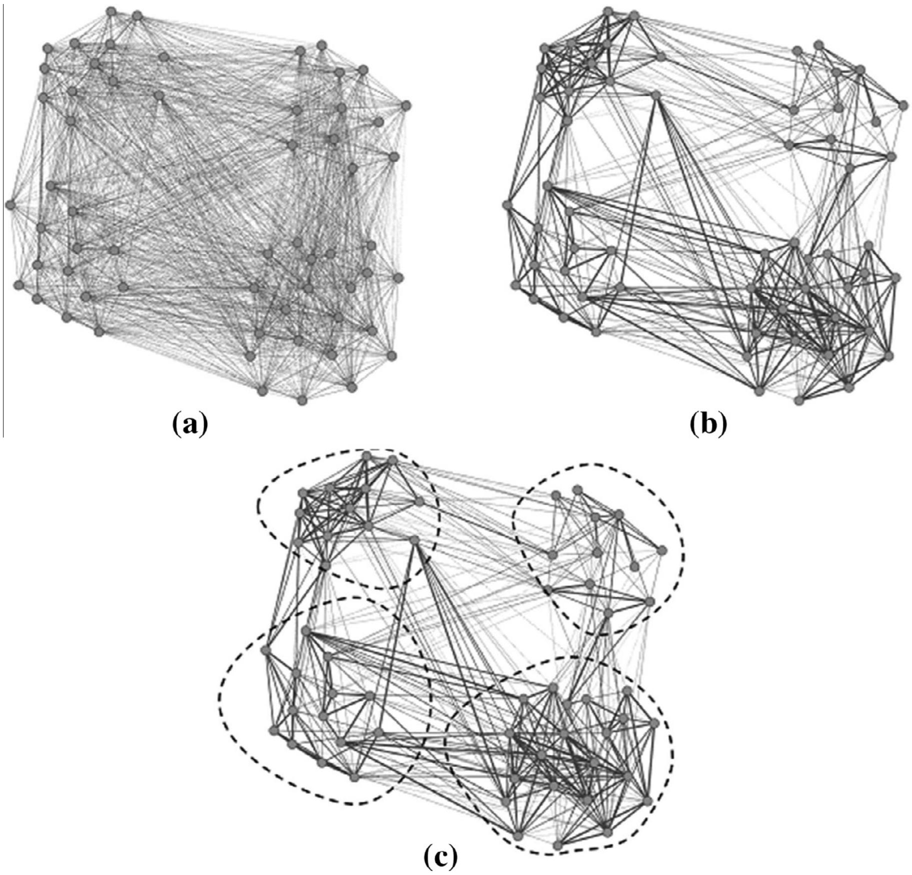


Fig. 2. Illustration of the feature clustering step for *Sonar* dataset. (a) Graph representation of the original set. (b) Remove the edges with associated weights lower that the θ parameter. (c) Apply Louvain community detection algorithm to identify the feature clusters.

where ρ is a pheromone decay parameter, $\tau_i(t)$ and $\tau_i(t+1)$ represent the amounts of pheromone on feature F_i at times t and $t+1$, respectively, A is the number of ants, and $\Delta\tau_i^k(t)$ is the extra pheromone increment to feature F_i by ant k, and is defined as follows:

$$\Delta_i^k(t) = \begin{cases} \gamma(FS^k(t)) & \mathit{if}\ F_i \in FS^k(t) \\ 0 & \mathit{Otherwise} \end{cases} \tag{8}$$

where $FS^k(t)$ is the feature subset founded by ant k at iteration t, and $\gamma(FS^k(t))$ is the evaluation function which measures the quality of solution $FS^k(t)$.

In this paper the quality of each ant's solution is evaluated by means of a specifically devised separability index. The separability index has been derived from the multiple discriminant analysis (MDA) approach. MDA is an extension to c-class problems ($c > 2$) of the Fisher's Linear Discriminant [23], which has been defined for finding the best linear combination of features in case of two class problems. The class separability index is defined by $\gamma(FS) = trace\left(\frac{W^T S_B W}{W^T S_W W}\right)$, where W is the transformation matrix from the original n-dimensional space to the l-dimensional subspace corresponding to the selected subset FS, S_w is the within scatter matrix, S_B is the between scatter matrix which are calculated as:

$$S_w = \sum_{j=1}^{c} \pi_j \Sigma_j \tag{9}$$

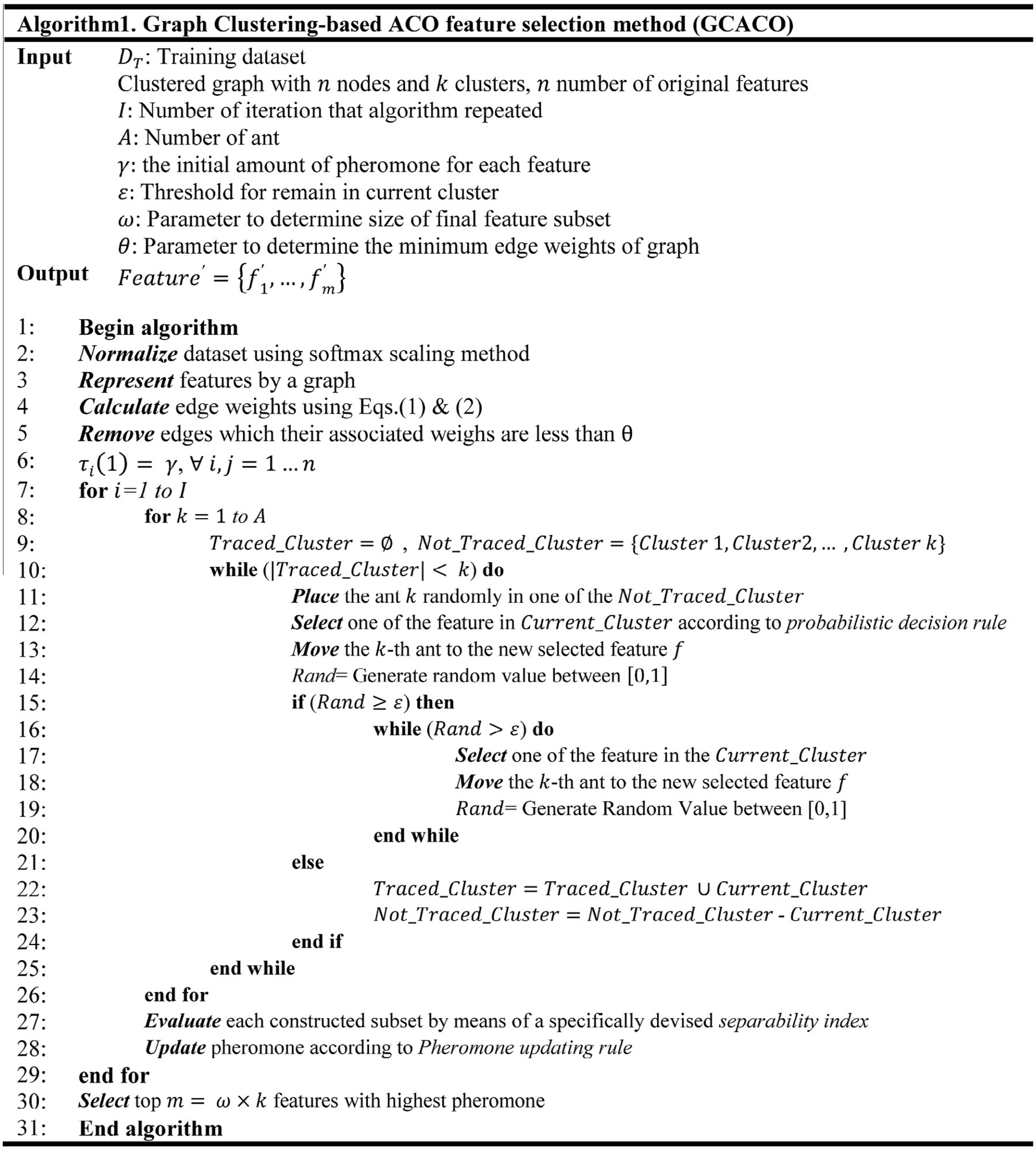
$$S_B = \sum_{j=1}^{c} (\mu_j - M_O)(\mu_j - M_O)^T \tag{10}$$

where π_j is the a priori probability that a pattern belongs to a particular class j, Σ_j is the sample covariance matrix of class j, μ_j is the sample mean vector of that class and M_O is the sample mean vector of the entire data points calculated as $M_o = \sum_{j=1}^{c} \pi_j \mu_j$.

This evaluation function presents two main advantages: (1) This index measures statistical properties of the feature subset and does not depend on any specific learning algorithms and (2) it does not require that the dimensionality of the searched subspace (i.e., the actual number of features to be used) is a priori fixed.

3.4. Complexity analysis

In the first step of the proposed method to represent the graph it is needed to compute the similarity values between each two features, so the time complexity of this step is $O(n^2 p)$ where n is the number of the original features and p denotes the number of patterns. Moreover, in the second step, the Louvain community detection algorithm is used to group the features into several clusters and the time complexity of this algorithm is $O(n \log n)$. Furthermore, in the third step, first of all, the relevance values of the features are evaluated using the Fisher score measure while the time complexity is $O(ncp)$, where c is the number of classes. Then a specific ant colony based search strategy is used to select the final features. Based on this strategy each ant starts to search the solution space from different points. The search process will be repeated for a number of iterative cycles (i.e., I). Thus, the time complexity of this part is $O(IAkf_k)$, where A is the number of the ants, k is the number of the clusters and f_k denotes the average number of features in each cluster which approximates n/k, so this complexity can be represented by $O(IAn)$. In addition if the ants run



```
Algorithm1. Graph Clustering-based ACO feature selection method (GCACO)
Input   $D_T$: Training dataset
        Clustered graph with $n$ nodes and $k$ clusters, $n$ number of original features
        $I$: Number of iteration that algorithm repeated
        $A$: Number of ant
        $\gamma$: the initial amount of pheromone for each feature
        $\varepsilon$: Threshold for remain in current cluster
        $\omega$: Parameter to determine size of final feature subset
        $\theta$: Parameter to determine the minimum edge weights of graph
Output  $Feature' = \{f'_1, \dots, f'_m\}$

Begin algorithm
Normalize dataset using softmax scaling method
Represent features by a graph
Calculate edge weights using Eqs.(1) & (2)
Remove edges which their associated weighs are less than θ
$\tau_i(1) = \gamma, \forall\, i, j = 1 \dots n$
for $i$=1 to $I$
    for $k = 1$ to $A$
        $Traced\_Cluster = \emptyset$ , $Not\_Traced\_Cluster = \{Cluster\ 1, Cluster 2, \dots , Cluster\ k\}$
        while ($|Traced\_Cluster| < k$) do
            Place the ant $k$ randomly in one of the $Not\_Traced\_Cluster$
            Select one of the feature in $Current\_Cluster$ according to probabilistic decision rule
            Move the $k$-th ant to the new selected feature $f$
            Rand= Generate random value between [0,1]
            if ($Rand \geq \varepsilon$) then
                while ($Rand > \varepsilon$) do
                    Select one of the feature in the $Current\_Cluster$
                    Move the $k$-th ant to the new selected feature $f$
                    $Rand$= Generate Random Value between [0,1]
                end while
            else
                $Traced\_Cluster = Traced\_Cluster \cup Current\_Cluster$
                $Not\_Traced\_Cluster = Not\_Traced\_Cluster - Current\_Cluster$
            end if
        end while
    end for
    Evaluate each constructed subset by means of a specifically devised separability index
    Update pheromone according to Pheromone updating rule
end for
Select top $m = \omega \times k$ features with highest pheromone
End algorithm
```

Fig. 3. Pseudo code of the proposed feature selection method.

in a parallel way, the computational complexity will be reduced to $O(In)$. Moreover, at the end of each iteration in the third step, the quality of each constricted subset is evaluated by means of a separability index. The time complexity of the separability index is $O(s^2np)$, where s is the cardinality of the selected subset. When the ACO algorithm is over, all of the features are sorted based on their pheromone values with the time complexity of $O(nlogn)$ and then the m features with highest values are selected as the final subset of features. Therefore, the time complexity of the third step is $O(ncp + In + s^2np + n\text{log}n)$. Consequently, the final time complexity of the GCACO method is $O(n^2p + n\log n + ncp + In + s^2np + n\log n)$. When the number of features which are selected by each ant (i.e., s) is much smaller than the number of original features ($s^2 \ll n$), the computational complexity of the proposed method can be reduced to $O(n^2p + In)$.

4. Experimental results

In order to evaluate the proposed method several experiments were conducted in terms of the classification accuracy, the number of selected features and the execution time to obtain the final feature subset. Generally, the classification accuracy is used as a measure to evaluate the performance of the feature selection algorithms. This is due to the fact that the relevant features are usually not known in advance, and we cannot directly evaluate how good a feature selection algorithm is by the features selected. The classification accuracy is defined as the proportion of the total number of predictions that were correct. Moreover, for each dataset, the classification accuracy is obtained over ten independent runs to achieve relatively accurate and stable estimations. In each run, first of all, the normalized datasets were randomly split into a training set (2/3 of dataset) and a test set (1/3 of dataset). The training set was used to select the final feature subset while the test set was used to evaluate the selected features using a learning model. Then to attain fair results, all the methods will be performed on the same train/test partitions. According to the randomness of the datasets and also the randomness in the proposed method, we reported both the average and the standard deviation of the classification accuracy. The experiments have been run on a machine with a 3.2 GHz CPU and 2 GB of RAM. Moreover, the proposed method was compared to the well-known and state of the art filter-based feature selection methods which are listed below:

Laplacian Score (L-Score) [64] is a feature selection method that has been designed for serving both supervised and unsupervised learning. The basic idea behind L-Score is to evaluate the features according to their locality preserving power.

Fisher Score (F-Score) [41] is a supervised feature selection algorithm that seeks features with the best discrimination abilities. Moreover, similarly to the other filter-based methods, the F-Score selects some top high-ranked features that have maximum locality preserving power computed in terms of the Laplacian score.

Relevance–redundancy feature selection (RRFS) [4] is an efficient feature selection technique based on relevance and relevance/redundancy analyses, which uses a specific criterion to choose an adequate number of features. RRFS has log-linear time complexity, with respect to the number of features.

Minimal-redundancy–maximal-relevance (mRMR) [22] is a solid multivariate filter approach which returns a feature subset with features that are mutually far away from each other (minimizing the redundancy) as well as highly correlated with the classification label (maximizing the relevance).

ReliefF [49] is an extension of the Relief algorithm, which applies a feature weighting scheme and searches for several nearest neighbors. A key idea behind the original ReliefF algorithm is to estimate the quality of attributes according to their ability to distinguish between instances that are close to each other.

Unsupervised feature selection method based on ant colony optimization (UFSACO) [54] is an ACO-based method, proposed to find an optimal solution to the feature selection problem, which tries to minimize the redundancy of the selected features without using any learning models. The UFSACO can be classified as a filter-based multivariate method.

In addition, the proposed method was compared to state of the art wrapper-based feature selection methods which are listed below:

Hybrid genetic algorithm for feature selection (HGAFS) [38] is a method that integrates two quite different new techniques in genetic algorithm for feature selection problem. This method limits the number of selected features and the local search operation.

Ant colony optimization algorithm for feature selection (ACOFS) [37] is a hybrid ant colony optimization algorithm for feature selection. ACOFS uses a hybrid search technique that combines the advantages of wrapper and filter approaches.

Particle swarm optimization for feature selection (PSOFS) [6] is PSO-based feature selection method to selecting a smaller number of features and achieving similar or even better classification performance than using all features.

Moreover, the detailed descriptions of the datasets, employed classifiers, user-specified parameters, experimental results, and sensitivity analysis are discussed in the following subsections.

4.1. Datasets

In this paper, several datasets with different properties were used in the experiments to show the effectiveness of the proposed method. These datasets include *Wine*, *Hepatitis*, *WDBC*, *Ionosphere*, *Spambase*, *Sonar*, *Arrhythmia, Madelon*, *Colon*, and *Arcene*. The basic characteristics of these ten datasets are summarized in Table 2. The detailed descriptions of these datasets except *Madelon* and *Arcene* datasets are available in the University of California Irvine machine learning repository [5]. Moreover, *Madelon* and *Arcene* datasets are collected from NIPS2003 feature selection challenge, available at its website[1] Some of these datasets contain attributes with missing values; therefore, to deal with these types of values in the experiments, each missing value was replaced with the mean of the available data on the respective feature [55]. Moreover, in many practical situations a designer is confronted with features whose values lie within different ranges. Thus, the features associated with large range values dominate those associated with small range values. To overcome this problem, a nonlinear normalization method called softmax scaling [55] is used to scale the datasets.

4.2. Used classifiers

To show the generality of the proposed method, the classification prediction capability of the selected features was tested using several well-known classical classifiers, i.e., *Support Vector Machine* (*SVM*), *Decision Tree* (*DT*), *Naïve Bayes* (*NB*), *k-Nearest Neighbor* (*kNN*) and *Random Forest* (*RF*). These classifiers are the most influential algorithms that have been widely used in the data mining community. Moreover, the Weka (Waikato Environment for Knowledge Analysis) software Hall et al. [20], which is a collection of machine learning algorithms for data mining tasks, was used as a workbench in the experiments to evaluate the selected feature. In this work, SMO, J48 (implementation of the C4.5 algorithm), Naïve Bayes, IBk and RandomForest as WEKA implementation of SVM, DT, NB, kNN and RF were used, respectively. Furthermore, the parameters of the mentioned classifiers for each experiment were set to the default values of the Weka software.

4.3. User-specified parameters

There are several user-specified parameters used in the methods of the experiments and thus their corresponding values should be determined by the user. Note that some of these parameters are not specific to the proposed method, and are generally required for most of the ACO-based feature selection methods. These parameters were chosen after a number of preliminary runs, and were not meant to be optimal. Since α and β parameters in the ACO-based methods determine respectively the relative importance of the pheromone and the heuristic information, proper selection of these parameters is needed to achieve an effective balance between exploitation and exploration. Table 3 shows the parameters and their corresponding values.

4.4. Results

In this subsection, we present the experimental results in terms of the classification accuracy, the number of selected features and the execution time. Also, for the purpose of exploring the statistical

[1] http://www.nipsfsc.ecs.soton.ac.uk/.

Table 2
Characteristics of the used datasets.

Dataset	Features	Classes	Patterns
Wine	13	3	178
Hepatitis	19	2	155
WDBC	30	2	569
Ionosphere	34	2	351
Spambase	57	2	4601
Sonar	60	2	208
Arrhythmia	279	16	452
Madelon	500	2	4400
Colon	2000	2	62
Arcene	10,000	2	900

Table 3
Common parameters for all datasets.

Parameter	Notation	Method	Value
Maximum number of cycles	I	GCACO, UFSACO,ACOFS, HGAFS,PSOFS	50
Ant number	A	GCACO, UFSACO, ACOFS	100
The initial amount of pheromone	γ	GCACO, UFSACO,ACOFS	0.2
Pheromone evaporation coefficient	ρ	GCACO	0.9
Pheromone evaporation coefficient	ρ	UFSACO	0.2
The exploration/exploitation parameter	q_0	GCACO, UFSACO	0.7
Relative importance of the pheromone value	α	GCACO, UFSACO	1
Relative importance of the heuristic information	β	GCACO, UFSACO	1
Threshold for remain in current cluster	ε	GCACO	0.4

significance of the results, we performed a nonparametric Friedman test [18] to statistically compare different methods on multiple datasets.

4.4.1. Classification accuracy

In the experiments, first of all the performance of the proposed method was evaluated over different classifiers. Tables 4–8 show the average classification accuracy (in %) of the proposed method (i.e., GCACO) with ten independent runs over the SVM, DT, NB, kNN and RF classifiers, respectively. The best result for each dataset is shown in bold face and the numbers in the parentheses show the rank of the algorithms. Moreover, the results are compared with those of the filter methods including L-Score, F-Score, RRFS, mRMR, ReliefF and UFSACO.

Table 4 compares the classification accuracy of the proposed method with those of the other filter based methods over SVM classifier. From the results it can be observed that in most cases the GCACO obtained the highest classification accuracy compared to those of filter-based methods. For example, for the *Colon* dataset, GCACO obtained a 81.42% classification accuracy while for L-Score, F-Score, RRFS, mRMR, ReliefF and UFSACO this value was reported 67.14%, 69.52%, 71.91%, 73.34%, 70.48% and 72.86%, correspondingly. Moreover, compared to the original dataset (i.e., that all features), the average classification accuracy over all datasets using the SVM classifier improved by 5.17 (i.e., 82.26–77.09), 0.64 (i.e., 77.73–77.09), 0.51 (i.e., 77.60–77.09) and 1.56 (i.e., 78.65–77.09) percentage points, where GCACO, F-Score, RRFS, and UFSACO were respectively used for selecting the final feature subset. Unfortunately, in this case the average classification accuracy of L-Score, mRMR and ReliefF decreased by 1.38, 0.82 and 2.15 percentage points, respectively. Furthermore, the results show that the GCACO obtained 82.26 average classification accuracy and achieved the first rank with a margin of 3.61% compared to the UFSACO which obtained the second best average classification accuracy (i.e., 81.23).

Furthermore, Table 5 reports the results over DT classifier. It can be seen from Table 5 results that in most cases the GCACO obtained the highest classification accuracy compared to those of filter-based methods and acquired the second place only for the *WDBC* dataset. For example, for the *Colon* dataset, GCACO obtained an 80.00% (5.38) classification accuracy (standard deviation) while for L-Score, F-Score, RRFS, mRMR, ReliefF and UFSACO this value was reported 66.67% (5.03), 69.05% (6.05), 73.33% (5.11), 71.43% (4.49), 69.53% (6.05) and 71.91% (6.52), respectively. Moreover, Tables 6–8 reported similar results for the NB, kNN and RF classifiers, respectively. It can be seen from the results that the proposed methods achieved the best classification accuracy compared to the other methods. Consequently, it can be concluded from Tables 4–8 results that the proposed method obtained the best results for the datasets with large numbers of features. For example, from Table 8 results we can see that the differences between the obtained classification accuracy of the proposed method and that of the second best ones were reported 5.13 (i.e., 71.45–66.32) and 4.42 (i.e. 82.28–77.86) for the *Arcene* (10,000) and *Colon* (2000) datasets (number of features), respectively. Moreover, for the proposed method, the best result was reported for the NB classifier. The NB classifier supposes that the features are conditionally independent from each other. On the other hand, in univariate methods (i.e., L-Score, F-Score and ReliefF), each feature is considered separately, thereby ignoring feature dependencies. While in these datasets, there are redundancies between features and thus, univariate methods will simply fail in the case of NB classifier. However, the proposed GCACO method considers the dependency of the selected features and selects a subset of features with minimum redundancy between them, thus in this case the best results were reported for the NB classifier.

For the purpose of exploring the relationship between the feature selection methods and the classifiers, i.e., which method is more suitable for which classifier, seven feature selection methods is ranked according to their obtained classification accuracy for a given dataset and for a specific classifier as reported in Tables 4–8. To this end, the rank values of the feature selection methods for each dataset are provided in these tables. Moreover, the average ranks of each method over all the datasets are reported. The results show that the proposed GCACO method achieved the best rank among the other methods for all of the employed classifiers. For example the GCACO obtained the average 1.1 rank for the DT classifier while this value was reported 6.3, 4.1, 3.8, 3.7, 5.6 and 3.2 for L-Score, F-Score, RRFS, mRMR, ReliefF and UFSACO, respectively.

4.4.2. Number of selected features

Table 9 shows the number of selected features of the different feature selection methods over ten independent runs. From the results it can be observed that, generally all the feature selection methods achieve significant reduction of dimensionality by selecting only a small portion of original features. Moreover, the results show that the GCACO, on average, selected the lowest number of features. For example from the results it can be seen that the proposed method selected the average number of 23.1 features, while, the other feature selection methods, on average selected 28.5 features.

Moreover, several experiments were conducted to compare the accuracy of the proposed method with the other feature selection methods based on the different numbers of selected features. Figs. 4 and 5 plot the classification accuracy (average over 10 independent runs) curves of SVM and DT classifiers on *Arrhythmia* and *Colon* datasets, respectively. In these plots, the x-

Table 4
Average classification accuracy (Acc) and standard deviation (Std) over ten independent runs using SVM classifier. The best result for each dataset between all feature selection methods is shown in bold face and the numbers in the parentheses show the rank of the algorithms.

Dataset		Feature selection method							
		GCACO	L-Score	F-Score	RRFS	mRMR	ReliefF	UFSACO	All features
Wine	Acc (%)	94.09 (3)	89.01 (6)	92.29 (4)	**96.55 (1)**	80.48 (7)	91.63 (5)	94.26 (2)	96.88
	Std	2.58	2.68	3.54	1.80	2.11	2.37	3.01	0.93
Hepatitis	Acc (%)	**84.52 (1)**	80.93 (7)	82.63 (4)	82.44 (5)	83.38 (2)	83.01 (3)	82.25 (6)	75.27
	Std	1.90	2.25	3.18	2.81	3.18	3.20	1.31	6.38
WDBC	Acc (%)	94.14 (2)	91.08 (7)	91.54 (6)	91.85 (4)	**94.24 (1)**	91.75 (5)	92.06 (3)	95.65
	Std	1.36	0.63	0.81	0.88	2.32	1.20	0.77	1.65
Ionosphere	Acc (%)	**90.41 (1)**	86.13 (6)	87.39 (5)	87.81 (4)	88.90 (2)	85.96 (7)	87.92 (3)	86.46
	Std	1.90	1.44	1.85	1.44	1.71	1.58	0.76	1.39
Spambase	Acc (%)	**88.38 (1)**	83.96 (7)	86.55 (5)	87.71 (3)	87.51 (4)	86.14 (6)	87.92 (2)	88.81
	Std	1.33	2.10	1.60	1.29	1.21	1.60	0.76	0.51
Sonar	Acc (%)	**82.38 (1)**	71.26 (6)	72.67 (4)	73.23 (3)	72.53 (5)	70.98 (7)	76.75 (2)	75.63
	Std	1.51	3.46	4.26	3.63	3.40	3.64	4.65	3.32
Arrhythmia	Acc (%)	**60.51 (1)**	53.56 (6)	54.28 (5)	58.37 (3)	58.50 (2)	53.11 (7)	55.70 (4)	56.03
	Std	5.42	1.44	1.50	4.87	4.80	1.49	3.63	2.98
Madelon	Acc (%)	**78.42** (**1**)	74.59 (3)	76.83 (2)	63.34 (5)	60.29 (6)	58.23 (7)	71.67 (4)	56.38
	Std	2.34	1.94	1.68	2.57	2.64	1.72	3.12	2.01
Colon	Acc (%)	**81.42 (1)**	67.14 (7)	69.52 (6)	71.91 (4)	73.34 (2)	70.48 (5)	72.86 (3)	65.23
	Std	3.51	4.74	5.59	5.24	4.01	4.92	6.36	5.05
Arcene	Acc (%)	**68.38** (**1**)	59.45 (6)	63.67 (3)	62.85 (5)	63.56 (4)	58.19 (7)	65.14 (2)	74.56
	Std	1.79	2.34	2.13	1.78	2.64	2.43	2.06	1.46
Average	Acc (%)	**82.26 (1.3)**	75.71 (6)	77.73 (4.3)	77.60 (3.6)	76.27 (4)	74.94 (5.8)	78.65 (3.1)	77.09
	Std	2.36	2.30	2.61	2.63	2.80	2.41	2.64	2.56

Table 5
Average classification accuracy (Acc) and standard deviation (Std) over ten independent runs using DT classifier. The best result for each dataset is shown in bold face and the numbers in the parentheses show the rank of the algorithms.

Dataset		Feature selection method							
		GCACO	L-Score	F-Score	RRFS	mRMR	ReliefF	UFSACO	All features
Wine	Acc (%)	**93.76 (1)**	88.52 (6)	91.47 (4)	91.96 (3)	80.81 (7)	90.65 (5)	92.94 (2)	90.81
	Std	2.87	2.89	4.14	4.85	2.05	2.89	3.09	4.24
Hepatitis	Acc (%)	**84.33 (1)**	81.69 (6)	83.01 (7)	81.12 (3)	83.38 (2)	82.63 (4)	82.07 (5)	77.73
	Std	2.18	2.52	3.20	1.98	3.18	3.18	0.99	5.24
WDBC	Acc (%)	92.27 (2)	91.34 (7)	91.44 (6)	92.06 (3)	**93.98 (1)**	91.80 (5)	91.96 (4)	90.41
	Std	0.62	0.54	0.69	0.94	2.56	1.42	0.74	2.45
Ionosphere	Acc (%)	**89.74 (1)**	85.37 (7)	87.64 (3)	87.55 (4)	89.23 (2)	85.71 (6)	86.88 (5)	86.71
	Std	2.01	1.38	2.05	1.57	1.17	1.68	1.68	1.30
Spambase	Acc (%)	**89.21 (1)**	85.32 (7)	86.86 (5)	86.97 (4)	87.20 (3)	85.81 (6)	88.01 (2)	88.93
	Std	0.98	2.08	1.43	1.87	1.57	1.60	0.78	0.57
Sonar	Acc (%)	**79.57 (1)**	72.25 (6)	73.65 (4)	73.94 (3)	73.23 (5)	70.70 (7)	76.61 (2)	74.22
	Std	3.99	4.45	4.09	3.70	3.63	3.80	4.56	3.11
Arrhythmia	Acc (%)	**60.38 (1)**	53.30 (6)	54.47 (5)	57.07 (3)	58.37 (2)	52.78 (7)	55.57 (4)	55.02
	Std	5.54	1.57	1.57	4.41	4.88	1.33	3.62	2.52
Madelon	Acc (%)	**82.73 (1)**	77.64 (5)	80.38 (2)	75.69 (6)	77.96 (4)	73.43 (7)	79.96 (3)	58.62
	Std	2.38	1.69	2.14	2.13	1.64	1.84	1.95	1.75
Colon	Acc (%)	**80.00 (1)**	66.67 (7)	69.05 (6)	73.33 (2)	71.43 (4)	69.53 (5)	71.91 (3)	63.33
	Std	5.38	5.03	6.05	5.11	4.49	6.05	6.52	3.22
Arcene	Acc (%)	**67.24 (1)**	56.67 (7)	60.95 (5)	62.87 (3)	62.17 (4)	57.25 (6)	64.78 (2)	70.79
	Std	2.32	2.14	2.56	1.45	1.94	1.84	1.46	2.87
Average	Acc (%)	**81.92 (1.1)**	75.87 (6.3)	77.89 (4.1)	78.25 (3.8)	77.77 (3.7)	76.02 (5.6)	79.06 (3.2)	75.65
	Std	2.82	2.42	2.79	2.80	2.71	2.56	2.53	2.72

axis denotes the subset of selected features, while the *y*-axis is the average classification accuracy. Fig. 4(a) shows that the GCACO is superior to the other methods applied on the SVM classifier when the number of features is less than 40. Moreover it can be seen from the results that for higher numbers of features the proposed method obtained the second place after the F-Score method. In addition, Fig. 4(b) represents similar results when the GCACO was applied on the DT classifier.

Moreover, Fig. 5(a) illustrates that the performance of the proposed method is superior to the performances of all methods for different numbers of selected features when the SVM classifier and *Colon* dataset are used in the experiments. The results in Fig. 5(b) report similar results and demonstrate that the GCACO is significantly superior to all of the other methods. Especially, when 40 features were selected, the classification error rates were around 66%, 69%, 71%, 71%, and 80% for L-Score, F-Score, mRMR, UFSACO and GCACO, respectively.

4.4.3. Compare with wrapper based methods

The performance of the proposed method has been compared to those of the wrapper-based feature selection methods including; HGAFS [38], ACOFS [37] and PSOFS [65] on the different datasets. Table 10 reports the average classification accuracy over ten independent runs for HGAFS, ACOFS, PSOFS and GCACO methods using SVM and NB classifiers. It can be seen from the results that the proposed method obtained the highest classification accuracy while it was applied on the Wine, Hepatitis, Ionosphere, Spambase and Madelon datasets for the NB classifier. Moreover

Table 6
Average classification accuracy (Acc) and standard deviation (Std) over ten independent runs using NB classifier. The best result for each dataset is shown in bold face and the numbers in the parentheses show the rank of the algorithms.

Dataset		Feature selection method							
		GCACO	L-Score	F-Score	RRFS	mRMR	ReliefF	UFSACO	All features
Wine	Acc (%)	**95.73 (1)**	89.17 (6)	90.98 (4)	95.08 (2)	82.78 (7)	89.99 (5)	94.42 (3)	91.80
	Std	2.76	2.58	4.03	5.24	2.22	3.32	2.70	4.37
Hepatitis	Acc (%)	**84.71 (1)**	81.12 (7)	82.82 (3)	83.01 (2)	82.06 (6)	82.44 (4)	82.25 (5)	76.41
	Std	2.26	2.35	3.13	3.44	2.98	2.36	0.97	5.91
WDBC	Acc (%)	**93.67 (1)**	91.54 (7)	91.80 (5)	92.84 (3)	93.57 (2)	91.75 (6)	92.16 (4)	90.41
	Std	1.50	0.34	0.93	2.23	2.37	1.62	0.62	2.45
Ionosphere	Acc (%)	**90.24 (1)**	82.68 (7)	86.97 (5)	89.07 (2)	88.73 (3)	85.96 (6)	87.13 (4)	86.55
	Std	3.38	4.67	1.98	0.88	1.38	1.58	1.37	1.18
Spambase	Acc (%)	**88.22 (1)**	81.96 (6)	86.62 (2)	83.04 (5)	80.50 (7)	85.50 (4)	86.48 (3)	83.05
	Std	0.52	4.30	1.41	5.85	6.16	1.55	3.52	5.80
Sonar	Acc (%)	**77.60 (1)**	71.26 (7)	75.34 (2)	73.23 (5)	73.94 (4)	71.68 (6)	75.06 (3)	75.49
	Std	5.32	3.46	3.40	3.63	3.71	4.17	5.10	3.19
Arrhythmia	Acc (%)	**60.38 (1)**	53.89 (5)	59.34 (2)	54.86 (4)	52.78 (6)	52.46 (7)	55.44 (3)	54.28
	Std	5.54	1.62	4.81	2.92	4.23	0.79	3.66	1.19
Madelon	Acc (%)	**79.32 (1)**	71.45 (3)	73.45 (2)	60.45 (6)	61.45 (5)	58.74 (7)	68.28 (4)	59.27
	Std	2.14	2.04	2.43	1.96	2.83	2.67	2.59	2.27
Colon	Acc (%)	**79.04 (1)**	63.32 (7)	69.05 (4)	71.43 (3)	70.00 (6)	65.23 (5)	72.39 (2)	65.73
	Std	2.45	2.29	5.15	4.49	4.52	5.05	6.26	4.39
Arcene	Acc (%)	**68.94 (1)**	59.35 (6)	62.45 (4)	61.32 (5)	64.72 (3)	58.39 (7)	66.56 (2)	72.36
	Std	2.17	1.35	1.67	1.29	1.93	2.13	1.67	1.82
Average	Acc (%)	**81.78 (1)**	74.57 (6.1)	77.88 (3.3)	76.43 (3.7)	75.05 (4.9)	74.21 (5.7)	78.01 (3.3)	75.53
	Std	2.80	2.50	2.89	3.19	3.23	2.52	2.84	3.25

Table 7
Average classification accuracy (Acc) and standard deviation (Std) over ten independent runs using kNN classifier. The best result for each dataset is shown in bold face and the numbers in the parentheses show the rank of the algorithms.

Dataset		Feature selection method							
		GCACO	L-Score	F-Score	RRFS	mRMR	ReliefF	UFSACO	All features
Wine	Acc (%)	93.11 (3)	89.50 (6)	91.63 (4)	**94.42 (1)**	81.63 (7)	90.81 (5)	93.93 (2)	96.71
	Std	3.16	3.01	3.58	1.58	2.76	3.01	2.89	0.77
Hepatitis	Acc (%)	**85.84 (1)**	81.88 (7)	82.44 (6)	83.01 (5)	83.76 (4)	84.32 (3)	84.90 (2)	77.73
	Std	1.33	1.82	3.21	3.44	2.83	3.67	1.54	5.24
WDBC	Acc (%)	92.27 (2)	90.87 (7)	91.44 (5)	92.11 (3)	**92.58 (1)**	91.23 (6)	91.85 (4)	94.09
	Std	0.62	0.65	0.69	0.68	1.67	1.07	0.69	2.50
Ionosphere	Acc (%)	**89.40 (1)**	85.87 (6)	86.97 (4)	87.89 (3)	88.90 (2)	85.29 (7)	86.46 (5)	85.71
	Std	2.17	1.23	1.86	1.54	0.86	1.49	1.60	1.48
Spambase	Acc (%)	**88.94 (1)**	83.09 (7)	86.41 (2)	85.40 (4)	84.45 (6)	85.62 (3)	85.16 (5)	88.62
	Std	1.06	1.66	1.57	1.73	3.81	1.66	1.49	0.41
Sonar	Acc (%)	**80.41 (1)**	71.68 (7)	73.37 (6)	74.64 (3)	73.94 (5)	74.36 (4)	77.46 (2)	76.33
	Std	2.14	4.17	4.42	3.63	3.71	4.08	4.09	3.16
Arrhythmia	Acc (%)	**60.25 (1)**	53.11 (6)	54.60 (5)	57.07 (3)	55.96 (4)	52.91 (7)	57.91 (2)	56.78
	Std	5.69	1.25	1.45	4.41	3.58	1.50	5.38	3.31
Madelon	Acc (%)	**74.82 (1)**	67.58 (6)	70.48 (2)	68.69 (5)	70.47 (3)	60.58 (7)	69.48 (4)	60.63
	Std	1.79	1.78	2.64	2.39	1.85	2.64	2.17	1.79
Colon	Acc (%)	**80.47 (1)**	68.10 (6)	70.01 (5)	72.86 (3)	71.90 (7)	71.44 (4)	75.24 (2)	63.85
	Std	4.72	4.53	5.27	5.04	4.73	3.89	6.26	4.60
Arcene	Acc (%)	66.14 (2)	56.34 (7)	63.52 (3)	61.85 (5)	62.17 (4)	57.61 (6)	**66.78 (1)**	72.82
	Std	1.32	1.82	2.64	1.46	2.73	2.05	2.31	1.27
Average	Acc (%)	**81.16 (1.4)**	74.80 (6.5)	77.08 (4.2)	77.79 (3.5)	76.57 (4.3)	75.41 (5.2)	78.91 (2.9)	77.32
	Std	2.40	2.19	2.73	2.59	2.85	2.50	2.84	2.45

the GCACO method acquired the second best results for WDBC and Arcene datasets. While for the other cases the wrapper based methods achieved the best results compared to the proposed method. Consequently, it can be concluded from the reported results that the overall performance of the GCACO method is comparable with those of the state of the art wrapper feature selection methods.

4.4.4. Computational complexity and execution time comparison

The aim of this section is to compare the computational complexity and execution time of the aforementioned methods. Table 11 shows the computational complexity of filter and wrapper based feature selection methods. Moreover, descriptions of notations are also provided in the table. It should be noted that wrapper based feature selection methods require a classifier (i.e. a learning model) to evaluate the selected feature subset in each iteration. Up to now, these methods have been employed different classifiers such as DT, NN, kNN, SVM and RF. For example the HGAFS a specific NN classifier to calculate the fitness value of the particles. The computational complexities of these classifiers are different from each other. Therefore into report the computation complexity of the wrapper based methods (i.e. ACOFS, HGAFS and PSOFS) we referred to $O(classifiere)$ in their corresponding computational complexity formula.

Furthermore, several experiments were performed in order to compare the execution time of the feature selection methods. In these experiments, the average execution times (in milliseconds) of filter and wrapper based methods are respectively reported in

Table 8
Average classification accuracy (Acc) and standard deviation (Std) over ten independent runs using RF classifier. The best result for each dataset is shown in bold face and the numbers in the parentheses show the rank of the algorithms.

Dataset		Feature selection method							
		GCACO	L-Score	F-Score	RRFS	mRMR	ReliefF	UFSACO	All features
Wine	Acc (%)	**96.19 (1)**	88.02 (6)	93.32 (3)	91.55 (4)	82.34 (7)	90.63 (5)	95.32 (2)	96.74
	Std	2.32	2.18	2.87	1.89	2.78	2.24	2.48	1.62
Hepatitis	Acc (%)	83.32 (2)	81.25 (6)	82.74 (3)	81.34 (5)	**83.81 (1)**	80.18 (7)	81.82 (4)	76.67
	Std	1.39	2.63	2.61	2.92	2.16	2.73	1.63	3.24
WDBC	Acc (%)	**95.36 (1)**	90.18 (6)	92.68 (3)	91.57 (5)	94.42 (2)	89.32 (7)	92.53 (4)	94.42
	Std	1.71	0.82	0.98	1.73	2.74	1.46	1.68	1.81
Ionosphere	Acc (%)	**90.76 (1)**	85.53 (6)	86.34 (5)	87.93 (2)	87.07 (4)	84.49 (7)	88.16 (2)	87.45
	Std	1.56	1.67	1.84	1.69	1.38	1.49	0.76	1.86
Spambase	Acc (%)	**89.19 (1)**	81.69 (7)	85.46 (5)	86.38 (4)	87.82 (2)	84.39 (6)	87.45 (3)	88.24
	Std	1.68	2.63	1.84	1.92	1.94	2.42	0.83	1.09
Sonar	Acc (%)	**81.43 (1)**	72.35 (5)	73.64 (4)	72.19 (6)	76.53 (2)	71.88 (7)	75.68 (3)	74.13
	Std	1.67	2.67	3.59	3.72	3.81	3.47	3.82	3.67
Arrhythmia	Acc (%)	**61.58 (1)**	51.45 (6)	52.46 (5)	57.59 (3)	60.75 (2)	51.07 (7)	56.19 (4)	55.18
	Std	4.45	1.32	1.87	3.79	3.61	1.86	3.72	2.63
Madelon	Acc (%)	76.36 (2)	75.19 (3)	**77.93 (1)**	72.86 (5)	69.29 (6)	63.84 (7)	73.94 (6)	57.18
	Std	2.74	1.80	2.84	2.97	2.25	1.08	3.73	2.15
Colon	Acc (%)	**82.28 (1)**	69.46 (7)	70.71 (6)	72.89 (4)	71.34 (5)	73.86 (3)	77.86 (2)	69.38
	Std	3.93	4.72	4.47	4.68	3.89	3.16	4.59	4.37
Arcene	Acc (%)	**71.45 (1)**	60.16 (6)	63.45 (5)	64.19 (4)	64.36 (3)	58.48 (7)	66.32 (2)	75.66
	Std	1.68	2.96	2.68	1.82	2.49	2.37	2.13	1.48
Average	Acc (%)	**82.79 (1.2)**	75.52 (5.8)	77.87 (4)	77.84 (4.3)	77.77 (3.4)	74.81 (6.3)	79.52 (3)	77.50
	Std	2.31	2.34	2.55	2.71	2.70	2.22	2.53	2.39

Table 9
Number of selected features of feature selection methods over ten independent runs.

Dataset	Feature selection method						
	GCACO	L-Score	F-Score	RRFS	mRMR	ReliefF	UFSACO
Wine	6	5	5	5	5	5	5
Hepatitis	6	5	5	5	5	5	5
WDBC	6	5	5	5	5	5	5
Ionosphere	15	20	20	20	20	20	20
Spambase	24	30	30	30	30	30	30
Sonar	24	30	30	30	30	30	30
Arrhythmia	20	20	20	20	20	20	20
Madelon	40	60	60	60	60	60	60
Colon	40	50	50	50	50	50	50
Arcene	50	60	60	60	60	60	60
Average	23.1	28.5	28.5	28.5	28.5	28.5	28.5

Tables 12 and 13. Form Table 12 results, it can be seen that the univariate feature selection methods (i.e., F-Score, and ReliefF) are much faster than the multivariate feature selection methods (i.e., mRMR, UFSACO and GCACO). This is due to the fact that in the univariate methods, the possible dependency between features is ignored in the feature selection process; thus, these methods can be computationally less expensive than the multivariate methods. Moreover, the reported results show that the proposed method is faster than other ACO-based feature selection method (i.e., UFSACO).

Additionally, the execution time of the proposed method has been also compared to those of the wrapper-based feature selection methods on the different datasets and the obtained results are reported in Table 13. The reported results show that in most cases the execution time of the proposed method is lower than the wrapper based methods. For example it can be seen from the results that the GCACO selected the final subset of features for Spambase dataset after 6846 ms. While in this case the HGAFS, ACOFS and HGAFS selected the final subset after 98,328, 87,193 and 531,439 ms respectively. In this case the results show the proposed method is nearly 88 times faster than the PSOFS method. While the results of Table 10 show that the accuracy of PSOFS method is only 0.5 times higher than GCACO method. Moreover, the results show that for very large datasets such as *Arcene* the execution time of the HGAFS and ACOFS methods are lower than the proposed method. This is due to the fact that these methods are executed in two steps. In the first step they applied a filter based method (such as Information gain or Gini-index) to rank the features independently and then a small size subset (i.e. a

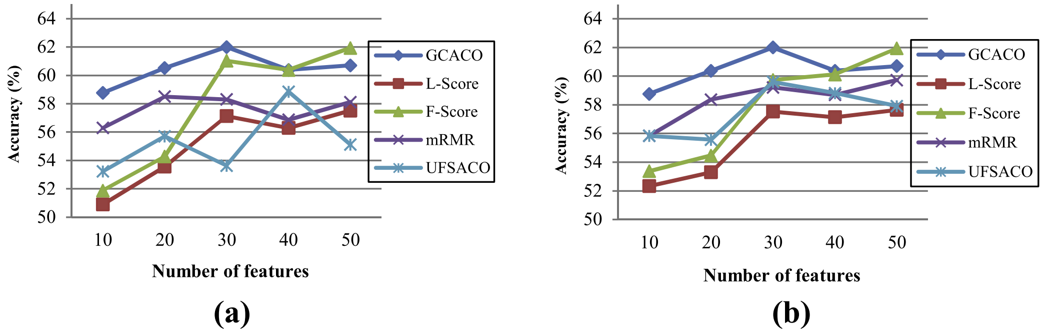


Fig. 4. Average classification accuracy (in %) over ten independent runs, with respect to the number of selected features with different methods on: (a) Arrhythmia dataset with SVM. (b) Arrhythmia dataset with DT.

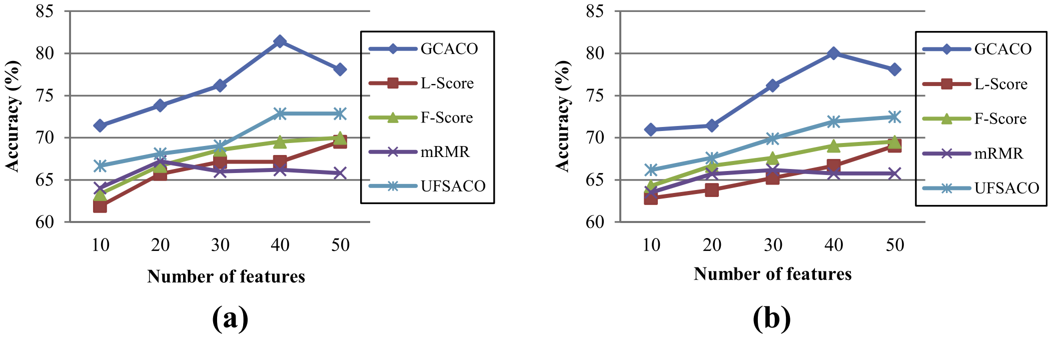


Fig. 5. Average classification accuracy (in %) over ten independent runs, with respect to the number of selected features with different methods on: (a) Colon dataset with SVM. (b) Colon dataset with DT.

Table 10
Classification accuracy of proposed method compared with wrapper-based feature selection methods using SVM and NB classifiers. The best result for each dataset is shown in bold face and underlined and the second best is in bold face.

Dataset		Feature selection method			
		GCACO	HGAFS	ACOFS	PSOFS
Wine	SVM	94.09	92.78	93.44	**94.42**
	NB	**95.73**	94.09	91.14	92.78
Hepatitis	SVM	**84.52**	81.50	82.63	83.95
	NB	**84.71**	80.18	82.51	84.17
WDBC	SVM	94.14	93.15	92.76	**94.81**
	NB	93.67	91.60	92.95	**94.69**
Ionosphere	SVM	90.41	87.97	89.49	**90.75**
	NB	**90.24**	86.13	88.65	88.06
Spambase	SVM	88.38	85.04	**87.30**	88.69
	NB	**88.22**	82.11	84.45	85.58
Sonar	SVM	**82.38**	76.33	79.29	78.72
	NB	77.60	**80.70**	79.99	80.13
Arrhythmia	SVM	60.51	60.06	61.68	**62.52**
	NB	60.38	61.42	**63.11**	62.55
Madelon	SVM	78.42	76.89	77.84	**80.06**
	NB	**79.32**	77.35	77.12	79.17
Colon	SVM	81.42	83.81	83.80	**85.23**
	NB	79.04	79.52	82.37	**83.33**
Arcene	SVM	68.38	64.57	66.12	**71.42**
	NB	66.53	62.74	63.82	**69.78**
Average	SVM	82.26	80.21	81.43	**83.05**
	NB	81.54	79.58	80.61	**82.02**

subset with only 100 top ranked features) are selected to be incorporated a wrapper based method in the second step. While in this case the proposed method is applied over all of the original features. On the other hand Table 10 results show that in this case the classification accuracy of the proposed method is much higher than the HGAFS and ACOFS methods.

Furthermore, from the reported results on classification accuracy (i.e. Tables 4–8 and Table 10) and also the execution time (i.e. Tables 12 and 13) of the filter and wrapper based methods, it can be concluded that the computational complexity and the quality of the selected feature subset are two main goals of the search methods. These goals are generally in conflict with each other and improving one of them causes the others to worsen. In other words, the filter-based feature selection methods have paid much attention to the computational time, while the wrapper feature selection methods usually consider the quality of the selected features. Therefore, a trade-off between these two issues has become an important and necessary goal to providing a good search method.

4.4.5. Statistical test results

In the experiments, a specific significance test known as the Friedman test [18] was employed to further compare the mentioned feature selection methods on multiple datasets, with different classifier. The Friedman test can be used to compare k methods over N datasets by ranking each method on each dataset separately. The method that obtains the best performance gets the rank of 1, the second best ranks 2, and so on. We have used the SPSS statistics acquired by IBM [40] for this purpose. The reported results show that the Friedman test reported a p-value of 0.000008 for the classification accuracy values of the SVM classifier in Table 4; since this value is below 0.05, we can claim the statistical significance of the proposed method results. Moreover, the Friedman test reported 0, 0, 0.000003 and 0.000001 p-values for the other classifiers including DT, NB, kNN and RF, respectively. Since, these classifiers have p-values below 0.05 making these results statistically significant. Also, the Friedman test is also performed for the reported results of the wrapper based methods (i.e. Table 10). In this case the statistical test reported the p-values of 0.000026 and 0. 085801 for SVM and NB classifiers, respectively. In SVM classifier, the obtained p-value lower than 0.05, thus, it can be concluded that the achieved classification accuracy results over the SVM classifier are statistically significant. In contrast in the case of the NB classifier, the corresponding results are not statistically significant and it can be inferred that none of the feature selection methods performed consistently better than the others.

Furthermore, additional statistical tests are also conducted for other measures such as number of features and also the execution time. The conducted Friedman test on the reported execution time results (i.e. Table 12) achieved a p-value of 6.4036E−8 $\ll 0.05$ and thus these results are statistically significant. The performed Friedman statistical test on the results of Table 9, reported a p-value of 0.423, thus it cannot be concluded the statistically significant of the results.

4.4.6. Sensitivity analysis of the parameters

Like many other feature selection methods, the proposed method requires the ω and θ parameters. The ω is a user-specified parameter that controls the size of the final feature subset. Accurate setting of the ω parameter substantially influences the results of the GCACO method. This parameter can be set to any value in the range $[1..n]$ and still $\omega \times k$ should be smaller than the number of original features (i.e., $\omega \times k \leqslant n$). If its corresponding value is set to a high value, thus too many features will be selected and natural patterns in the data will be blurred by noise and the redundant features can be selected with a high probability. On the other hand when the parameter is set to a small value, too few features are chosen and thus, there will not be enough information for the classification task. Moreover, in order to explore the proper determination of the ω parameter values, several experiments were performed to reveal how the classification accuracy is changing with different values of the parameter.

Table 11
Comparison of computational complexity of different feature selection methods.

Method	Type	Computational complexity	Description of notations
L-Score [64]	Filter Univariate Unsupervised	$O(p^2n)$	p: number of patterns (or number of instances)
F-Score [41]	Filter Univariate Supervised	$O(ncp)$	n: number of original features c: number of classes
mRMR [22]	Filter Multivariate Supervised	$O(nmp)$	m: number of selected features ($m \ll n$)
ReliefF [49]	Filter Univariate Supervised	$O(Ipn)$	I: maximum number of iterations
UFSACO [54]	Filter Multivariate Unsupervised	$O(n^2p + Imn)$	k: number of particles (PSO and GA), number of ants (ACO)
PSOFS[65]	Wrapper Supervised	$O(IknO(\text{classifier}))$	$O(\text{classifier})$: the complexity of executing a classifier for wrapper based methods. Usually kNN classifier is used to evaluate a particle. The complexity of k-NN is $O(nm + nlogn)$
HGAFS [38]	Wrapper Supervised	$O(n^2p + Ink + IkO(\text{classifier}))$	
ACOFS [37]	Wrapper Supervised	$O(n^2p + Imnk + IkO(\text{classifier}))$	
GCACO (Proposed method)	Filter Multivariate Unsupervised	$O(n^2p + In)$	

Table 12
Average execution time (in ms) of proposed method compared with filter feature selection methods over ten independent runs.

Dataset	Feature selection method						
	GCACO	L-Score	F-Score	RRFS	mRMR	ReliefF	UFSACO
Wine	882	111	12	4	1187	79	1044
Hepatitis	995	82	23	18	2348	161	1548
WDBC	1814	1863	29	26	3161	1464	2275
Ionosphere	1738	897	38	35	2595	963	2392
Spambase	6846	132,678	167	158	14,968	128,052	7279
Sonar	2653	457	48	39	3362	769	3981
Arrhythmia	15,157	5354	56	48	4879	3657	20,190
Madelon	307,493	312,423	569	3247	5759	184,734	351,823
Colon	102,384	1457	92	83	11,776	2389	131,476
Arcene	326,881	12,681	218	13,632	15,482	264,671	425,634
Average	76684.3	46800.3	125.2	1729	6551.7	58693.9	94764.2

Table 13
Average execution time (in ms) of proposed method compared with wrapper feature selection methods over ten independent runs.

Dataset	Feature selection method			
	GCACO	HGAFS	ACOFS	PSOFS
Wine	882	753	639	1250
Hepatitis	995	3367	2596	3742
WDBC	1814	9354	6327	8926
Ionosphere	1738	14,172	12,387	14,731
Spambase	6846	98,328	87,193	531,439
Sonar	2653	42,108	37,481	45,983
Arrhythmia	15,157	67,912	63,471	648,238
Madelon	307,493	174,921	151,870	25,486,251
Colon	102,384	27,812	24,971	672,827
Arcene	326,881	89,381	78,509	67,245,317
Average	76684.3	52810.8	46544.4	9465870.4

Fig. 6(a) and (b) shows the ω parameter sensitivity analysis for SVM and DT classifiers respectively. The results compare the classifier accuracy on the *Hepatitis*, *WDBC*, *Ionosphere*, *Spambase* and *Sonar* datasets for different values of ω in the range of [2..6]. The obtained results indicated that in most cases, when the parameter was set to 4 (i.e., $\omega = 4$), the proposed method obtained the best results.

Furthermore, in the second step (i.e., feature clustering) the edges with associated weights lower that the θ parameter are removed. The result of the feature clustering algorithm depends on the value of θ. This parameter can be set to any value in the range of [0 1]. When this value is set to a small value, more edges are considered in the graph clustering algorithm and the number of obtained clusters is low. On the other hand when the parameter is set to a high value, the graph clustering algorithm identifies more clusters. In this study several experiments were also performed to analyze the θ parameter. Table 14 shows the average SVM classification accuracy results of over 10 independent runs for different values of the θ parameter for the *Hepatitis*, *WDBC*, *Ionosphere* and *Sonar* datasets. Moreover, the number of obtained clusters is also reported in Table 14. The obtained results indicated that in most cases when the parameter was set to 0.6, the proposed method obtained the best results. It can be concluded from the results that when the parameter was set to a small value, the graph clustering

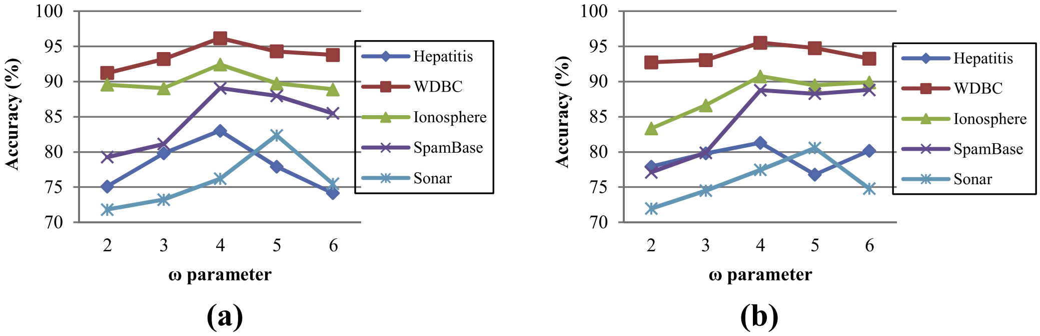


Fig. 6. Average classification accuracy (in %) over ten independent runs, with different ω values on Hepatitis, WDBC, Ionosphere and Sonar datasets with: (a) SVM classifier and (b) DT classifier.

Table 14
Average classification accuracy for different values of the θ parameter over ten independent runs. The best average classification accuracy is marked in boldface.

Dataset	θ	#Obtained clusters	Acc (in %)
Hepatitis	0.2	2	83.01
	0.4	3	84.71
	0.6	4	**88.10**
	0.7	6	79.05
WDBC	0.2	2	91.23
	0.4	3	**95.90**
	0.6	6	95.17
	0.7	7	95.38
Ionosphere	0.2	2	81.67
	0.4	3	86.71
	0.6	4	**88.98**
	0.7	6	85.20
Sonar	0.2	2	67.60
	0.4	3	76.05
	0.6	4	**85.34**
	0.8	5	83.65

algorithm identified a lower number of clusters. Therefore in this case the proposed method selects a smaller number of features and thus most representative features cannot be selected, which reduces the classifier accuracy.

4.5. Discussion

This subsection briefly explains why the performance of the GCACO is better than those of the other feature selection methods. There are three differences that might contribute to better performance of GCACO compared to the other methods.

1. Irrelevant features, along with redundant features, severely affect the accuracy of the learning algorithm [8,10,34]. Thus, feature selection should be able to identify and remove as many of the irrelevant and redundant features as possible. Of the many feature selection methods, some can effectively remove irrelevant features but fail to handle the redundant features. In the univariate methods (i.e., L-Score, F-Score and ReliefF) the relevance of a feature is measured individually and the possible dependency between features is ignored in the feature selection process; thus, these methods cannot remove the redundant features precisely. On the other hand, some of the multivariate feature selection methods only eliminate the redundant features. For example, the main goal of the UFSACO method is to select a subset of features with minimum redundancy and there is no guarantee to select the optimal feature set. This is due to the fact that the selected features may not constitute the best representative set although they are highly dissimilar to each other. To this end, we have developed a novel method which can efficiently and effectively deal with both irrelevant and redundant features. In the proposed method each ant selects the features with minimum similarity with those of the previously selected ones while it maximizes the dependency on the target class. By applying this kind of selection rules, the redundant and the irrelevant features have a lower probability to be selected.
2. One of the main shortcomings of existing univariate filter-based feature selection methods is ranking the features independently without considering their dependency on the other features. In this case some of the features with lower ranks may have strong discriminatory power while they are considered with the other features in a group. To overcome this problem, we focus on exploring a new framework to retain the useful structure among features as many as possible. In the proposed method, based on the probabilistic decision rule, to avoid being trapped into a local optimum, each feature has a chance of being selected corresponding to its probability value which is computed using Eq. (6). Also, the proposed updating rule takes into account all of the features based on their quality in the constructed feature subset. Pheromone updating rule is intended to allocate a greater amount of pheromone to better solution. Thus, individually weak and collectively strong features increase their selection probability.
3. The GCACO method integrates the graph clustering method with the search process of the ACO algorithm. Using the feature clustering method improves the performance of the proposed method in several aspects. First, the time complexity is reduced compared to those of the other ACO-based feature selection methods. This is due to the fact that the ant does not need to traverse a complete graph; thus, the probability computation in a clustered graph is reduced compared to that of the completed graph. Second, the search process is guided in such a way that at least one feature is selected per cluster, along the search process of each ant, and also relatively fewer correlated features are injected in a high proportion with respect to more correlated features to the consecutive iteration.

5. Conclusion

Feature selection plays an important role in the world of machine learning and more specifically in the classification task. Moreover, the computational cost is reduced and on the other hand, the model is constructed from the simplified data and this improves the general abilities of classifiers. In this paper, a novel feature selection method has been developed by integrating the concept of graph clustering with the search process of the ant colony optimization. The proposed method works in three steps: in the first step, the problem space is represented as a graph by considering the entire feature set as the vertex set and having feature similarity as the corresponding edge weight. In the second step, features are divided into several clusters by employing a community detection method. Finally, in the third step, we use a novel ACO-based feature selection algorithm that has been developed by utilizing the feature clusters. The proposed method can deal with both irrelevant and redundant features. This is because of the fact that each ant in the clustered graph tries to search for the features with minimum similarity and it maximizes the dependency on the target class. The proposed method has been compared to the six well-known and state-of-the-art filter-based feature selection methods including *L-Score*, *F-Score*, *RRFS*, *mRMR*, *ReliefF* and *UFSACO* and three state of the art wrapper based methods including *HGAFS, ACOFS and PSOFS* from the three different aspects of classification accuracy, size of subset selected features and execution time. The reported results show that in most cases the proposed method obtained the best classification accuracy. Furthermore, the results indicate that the execution time of the proposed method is comparable to those of feature selection methods. Moreover, the performed statistical test over three different measures including classification accuracy, number of selected features and execution time, show that the obtained results are statistically significant.

References

[1] M.H. Aghdam, N. Ghasem-Aghaee, M.E. Basiri, Text feature selection using ant colony optimization, Expert Syst. Appl. 36 (2009) 6843–6853.

[2] H. Ahn, K.-J. Kim, Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach, Appl. Soft Comput. 9 (2009) 599–607.

[3] A. Al-Ani, A. Alsukker, R.N. Khushaba, Feature subset selection using differential evolution and a wheel based search strategy, Swarm Evolut. Comput. 9 (2013) 15–26.

[4] Artur J. Ferreira, M.A.T. Figueiredo, An unsupervised approach to feature discretization and selection, Pattern Recogn. 45 (2012) 3048–3060.

[5] A. Asuncion, D. Newman, UCI Repository of Machine Learning Datasets, 2007. <http://archive.ics.uci.edu/ml/datasets.html>.

[6] Bing Xue, Mengjie Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: a multi-objective approach, IEEE Trans. Cybernet. 43 (2013) 1656–1671.

[7] B. Bonev, F. Escolano, D. Giorgi, S. Biasotti, Information-theoretic selection of high-dimensional spectral features for structural recognition, Comput. Vis. Image Underst. 117 (2013) 214–228.

[8] C. De Stefano, F. Fontanella, C. Marrocco, A.S.d. Freca, A GA-based feature selection approach with an application to handwritten character recognition, Pattern Recogn. Lett. 35 (2014) 130–141.

[9] J.M. Cadenas, M.C. Garrido, R. Martínez, Feature subset selection filter–wrapper based on low quality data, Expert Syst. Appl. 40 (2013) 6241–6252.

[10] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28.

[11] B. Chen, L. Chen, Y. Chen, Efficient ant colony optimization for image feature selection, Signal Process. 93 (2013) 1566–1576.

[12] Y. Chen, D. Miao, R. Wang, A rough set approach to feature selection based on ant colony optimization, Pattern Recogn. Lett. 31 (2010) 226–233.

[13] Chih-Fong Tsai, William Eberle, Chi-Yuan Chu, Genetic algorithms in feature and instance selection, Knowl.-Based Syst. 39 (2013) 240–247.

[14] Chuen-Horng Lin, Huan-Yu Chen, Y.-S. Wua, Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection, Expert Syst. Appl. 41 (2014) 6611–6621.

[15] S.F. da Silva, M.X. Ribeiro, J.d.E.S. Batista Neto, C. Traina-Jr, A.J.M. Traina, Improving the ranking quality of medical image retrieval using a genetic feature selection method, Decis. Support Syst. 51 (2011) 810–820.

[16] K.A. Dowsland, J.M. Thompson, An improved ant colony optimisation heuristic for graph colouring, Discr. Appl. Math. 156 (2008) 313–324.

[17] Forsati, Moayedikia, Jensen, Shamsfard, Meybodi, Enriched ant colony optimization and its application in feature selection, Neurocomputing 142 (2014) 354–371.

[18] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92.

[19] I. Guyon, A.e. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. (2003) 1157–1182.

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten, The WEKA Data Mining Software. <http://www.cs.waikato.ac.nz/ml/weka>.

[21] X. Han, X. Chang, L. Quan, X. Xiong, J. Li, Z. Zhang, Y. Liu, Feature subset selection by gravitational search algorithm optimization, Inform. Sci. 281 (2014) 128–146.

[22] Hanchuan Peng, Fuhui Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1226–1238.

[23] P.E. Hart, R.O. Duda, D.G. Stork, Pattern Classification, John Wiley & Sons Inc., 2001.

[24] S.-Y. Ho, C.-C. Liu, S. Liu, Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm, Pattern Recogn. Lett. 23 (2002) 1495–1503.

[25] C.-L. Huang, J.-F. Dun, A distributed PSO–SVM hybrid system with feature selection and parameter optimization, Appl. Soft Comput. 8 (2008) 1381–1391.

[26] H. Huang, H.B. Xie, J.Y. Guo, H.J. Chen, Ant colony optimization-based feature selection method for surface electromyography signals classification, Comput. Biol. Med. 42 (2012) 30–38.

[27] H.H. Inbarani, A.T. Azar, G. Jothi, Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis, Comput. Methods Prog. Biomed. 113 (2014) 175–185.

[28] J. Kennedy, R. Eberhart, Particle swarm optimization, in: The Proceedings of the 1995 IEEE International Conference on Neural Network, 1995, pp. 1942–1948.

[29] Jiansheng Wu, Zusong Lu, L. Jin, A novel hybrid genetic algorithm and simulated annealing for feature selection and kernel optimization in support vector regression, in: IEEE 13th International Conference on Information Reuse and Integration (IRI), 2012, 2012, pp. 401–406.

[30] Jung-Yi Jiang, Ren-Jia Liou, S.-J. Lee, A fuzzy self-constructing feature clustering algorithm for text classification, IEEE Trans. Knowl. Data Eng. 23 (2011) 335–349.

[31] H.R. Kanan, K. Faez, An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system, Appl. Math. Comput. 205 (2008) 716–725.

[32] L. Ke, Z. Feng, Z. Ren, An efficient ant colony optimization approach to attribute reduction in rough set theory, Pattern Recogn. Lett. 29 (2008) 1351–1357.

[33] Lei Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 856–863.

[34] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (2005) 491–502.

[35] M. Dorigo, G.D. Caro, Ant colony optimization: a new meta-heuristic, in: Proceeding of the Congress on Evolutionary Computing, 1999.

[36] Mauricio Schiezaro, H. Pedrini, Data feature selection based on artificial bee colony algorithm, EURASIP J. Image Video Process. 2013 (2013) 47.

[37] Md. Monirul Kabir, Md. Shahjahan, K. Murase, A new hybrid ant colony optimization algorithm for feature selection, Expert Syst. Appl. 39 (2012) 3747–3763.

[38] Md. Monirul Kabir, Md. Shahjahan, K. Murase, A new local search based hybrid genetic algorithm for feature selection, Neurocomputing 74 (2011) 2914–2928.

[39] T.M. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.

[40] N.H. Nie, C.H. Hull, J.G. Jenkins, K. Steinbrenner, D.H. Bent, Statistical Package for the Social Sciences, McGraw Hill, New York, NY, 1975.

[41] Quanquan Gu, Zhenhui Li, J. Han, Generalized Fisher score for feature selection, in: Proceedings of the International Conference on Uncertainty in Artificial Intelligence, 2011.

[42] L.E. Raileanu, K. Stoffel, Theoretical comparison between the Gini index and information gain criteria, Ann. Math. Artif. Intell. 41 (2004) 77–93.

[43] J.F. Ramirez-Cruz, O. Fuentes, V. Alarcon-Aquino, L. Garcia-Banuelos, Instance selection and feature weighting using evolutionary algorithms, in: 15th International Conference on Computing, 2006, CIC '06, Publishing, 2006, pp. 73–79.

[44] Rana Forsati, Alireza Moayedikia, A. Keikha, A novel approach for feature selection based on the bee colony optimization, Int. J. Comput. Appl. (2012) 43.

[45] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, A simultaneous feature adaptation and feature selection method for content-based image retrieval systems, Knowl.-Based Syst. 39 (2013) 85–94.

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

[47] M. Reed, A. Yiannakou, R. Evering, An ant colony algorithm for the multi-compartment vehicle routing problem, Appl. Soft Comput. 15 (2014) 169–176.

[48] K.R. Robbins, W. Zhang, J.K. Bertrand, The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification, J. Math. Med. Biol. (2008) 1–14.

[49] M. Robnik-Sikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (2003) 23–69.

[50] F. Ros, S. Guillaume, M. Pintore, J. Chrétien, Hybrid genetic algorithm for dual selection, Pattern Anal. Appl. 11 (2008) 179–198.

[51] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (2007) 2507–2517.

[52] R.K. Sivagaminathan, S. Ramakrishnan, A hybrid approach for feature subset selection using neural networks and ant colony optimization, Expert Syst. Appl. 33 (2007) 49–60.

[53] Stjepan. Oreski, G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment, Expert Syst. Appl. 41 (2014) 2052–2064.

[54] S. Tabakhi, P. Moradi, F. Akhlaghian, An unsupervised feature selection algorithm based on ant colony optimization, Eng. Appl. Artif. Intell. 32 (2014) 112–123.

[55] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, Oxford, 2008.

[56] C.-F. Tsai, W. Eberle, C.-Y. Chu, Genetic algorithms in feature and instance selection, Knowl.-Based Syst. 39 (2013) 240–247.

[57] H. Uğuz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowl.-Based Syst. 24 (2011) 1024–1032.

[58] A. Unler, A. Murat, R.B. Chinnam, Mr2PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification, Inform. Sci. 181 (2011) 4625–4641.

[59] V. Blondel, J.G.R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech: Theory Exp. 10008 (2008) 1–12.

[60] S.M. Vieira, J.M.C. Sousa, T.A. Runkler, Two cooperative ant colonies for feature selection using fuzzy models, Expert Syst. Appl. 37 (2010) 2714–2723.

[61] S. Wang, W. Pedrycz, Q. Zhu, W. Zhu, Unsupervised feature selection via maximum projection and minimum redundancy, Knowl.-Based Syst. 75 (2015) 19–29.

[62] Y. Wang, Y. Liu, L. Feng, X. Zhu, Novel feature selection method based on harmony search for email classification, Knowl.-Based Syst. 73 (2015) 311–323.

[63] Wenzhu Yang, Daoliang Li, L. Zhu, An improved genetic algorithm for optimal feature subset selection from multi-character feature set, Expert Syst. Appl. 38 (2011) 2733–2740.

[64] Xiaofei He, Deng Cai, P. Niyogi1, Laplacian score for feature selection, Adv. Neural Inform. Process. Syst. 18 (2005) 507–514.

[65] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2014) 261–276.

[66] Yi-Leh Wu, Cheng-Yuan Tang, Maw-Kae Hor, P.-F. Wub, Feature selection using genetic algorithm and cluster validation, Expert Syst. Appl. 38 (2011) 2727–2732.

[67] Ying Li, Gang Wang, Huiling Chen, Lian Shi, Lei Qin, An ant colony optimization based dimension reduction method for high-dimensional datasets, J. Bionic Eng. 10 (2013) 231–241.

[68] D. Zhao, L. Luo, K. Zhang, An improved ant colony optimization for the communication network routing problem, Math. Comput. Model. 52 (2010) 1976–1981.