Pattern Recognition 48 (2015) 2798–2811

Contents lists available at ScienceDirect

Pattern Recognition

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

Relevance–redundancy feature selection based on ant colony optimization

Sina Tabakhi, Parham Moradi *

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

ARTICLE INFO

Article history:
Received 13 July 2014
Received in revised form
5 February 2015
Accepted 25 March 2015
Available online 8 April 2015

Keywords:
Pattern recognition
Curse of dimensionality
Feature selection
Multivariate technique
Filter model
Ant colony optimization

ABSTRACT

The curse of dimensionality is a well-known problem in pattern recognition in which the number of patterns is smaller than the number of features in the datasets. Often, many of the features are irrelevant and redundant for the classification tasks. Therefore, the feature selection becomes an essential technique to reduce the dimensionality of the datasets. In this paper, unsupervised and multivariate filter-based feature selection methods are proposed by analyzing the relevance and redundancy of features. In the methods, the search space is represented as a graph and then the ant colony optimization is used to rank the features. Furthermore, a novel heuristic information measure is proposed to improve the accuracy of the methods by considering the similarity between subsets of features. The performance of the proposed methods was compared to the well-known univariate and multivariate methods using different classifiers. The results indicated that the proposed methods outperform the existing methods.

1. Introduction

Pattern recognition is a branch of artificial intelligence whose aim is to seek to learn a model with the purpose of automatic classification of new patterns into a number of predefined classes [1]. The rapid development of information technologies in the past several decades has lead to production of datasets with large numbers of features and relatively few patterns. This presents a well-known challenge, called the *curse of dimensionality*, to pattern recognition methods and increases the computational time complexity of building the model. On the other hand, many of the features in the datasets are irrelevant and redundant for the model and may have a negative effect on the prediction accuracy [2–4].

A common way to deal with such problems is the feature selection technique. Feature selection is an important step in data preprocessing for designing many pattern recognition systems, especially in high-dimensional datasets. The goal of the feature selection technique is to seek the relevant features with the most predictive information from the original feature set. This technique reduces the dimensionality of datasets by eliminating many irrelevant and redundant features which improves the performance of the learnt model and avoids overfitting. On the other hand, this reduction helps to speed up the learning process and leads to a simple and understandable predictor model [2,5,6]. Feature selection has been established as an important technique in many practical applications of pattern recognition such as text processing [7,8], face recognition [9,10], bioinformatics [11,12], speaker verification [13], medical diagnosis [14,15], and financial domains [16,17].

The feature selection procedure needs a search strategy to explore the search space and find the optimal subset of features. This strategy requires a measure to evaluate the quality of the feature subsets. Finding the optimal subset requires exhaustive search over all possible combinations of features, meaning that its size is 2^n, where n denotes the number of features. In practical applications, the computational complexity of this approach is impractical even on moderate datasets. Therefore, it has been shown that finding the optimal feature subset is a NP-hard problem [4,18,19]. One approach for dealing with this problem is applying classical search methods such as branch and bound [20] and best first search [21] that avoid exhaustive enumeration of all subsets of features. These methods find the optimal subset, but they rely on the assumption of monotonicity and perform poorly in real-world datasets.

Thus, the other approach is proposed for finding a near-optimal feature subset with less computational effort. This approach seeks to identify and remove irrelevant and redundant features in high-dimensional datasets instead of the exhaustive search over the feature subsets. The feature selection methods in this approach can be classified into four categories including filter, wrapper, embedded, and hybrid models [2,4,6,12,22]. Some of the filter based methods use a specific criterion to evaluate the relevance of features. These kinds of methods which are called the univariate

* Corresponding author. Tel.: +98 8733660073; fax: +98 8733668513.
E-mail addresses: sina.tabakhi@ieee.org (S. Tabakhi), p.moradi@uok.ac.ir (P. Moradi).

http://dx.doi.org/10.1016/j.patcog.2015.03.020
0031-3203/© 2015 Elsevier Ltd. All rights reserved.

filter model can effectively identify and remove the irrelevant features independently of any learning algorithms, but they are unable of removing redundant features. Since the possible dependency between features is disregarded, these methods lead to a weak learning model. On the other hand, some of the filter based methods, called the multivariate filter model, can handle both irrelevant and redundant features which improve the accuracy of the learning model compared to that of the univariate filter based feature selection methods. The search strategy of the multivariate filter model involves only a single iteration and can easily be trapped into local optimum.

The wrapper based feature selection methods apply a learning algorithm to evaluate the quality of feature subsets in the search space iteratively. These methods can effectively identify and remove irrelevant and redundant features. Due to the frequent use of the learning algorithm in the search process, this model requires high computational time, especially for high-dimensional datasets. In the embedded model the feature selection procedure is considered as a part of the process of building the model. Although this model can handle both irrelevant and redundant features, training the learning algorithms with a large number of features will be time-consuming. 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. However, the hybrid model may suffer in terms of accuracy because the filter and wrapper models are considered as two separate steps.

Recently, swarm intelligence based methods have attracted a lot of attention due to their good performance in solving feature selection problems. Among the swarm intelligence based methods, ant colony optimization (ACO) has been successfully used in the feature selection area of research [6,23–25]. 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, similar function to reinforcement learning schema, and a good global and local search capability due to stochastic and greedy components in the algorithm [6,26–30]. Most swarm intelligence-based methods use a learning algorithm in their search strategies to evaluate a feature subset, and they are classified as a type of the wrapper model. Therefore, they suffer the problem of high computational time and inefficiency on the datasets with large number of features.

Since the presentation of a method to handle both irrelevant and redundant features in an acceptable time is an important issue, a major purpose of the current study is to attempt to select a high-quality feature subset within a reasonable time. In this paper, we present novel unsupervised filter based feature selection methods using ACO. They bring together the computational efficiency of the filter model and the acceptable performance of the ACO algorithm. Moreover, the methods use criteria to analyze the relevance and redundancy of the features which are used as prior knowledge in the ACO algorithm to guide the search process. In the proposed methods, each feature is ranked in the iterative improvement process of the ACO algorithm without using any learning algorithms and class labels. Also, we have proposed a new heuristic information measure which considers the similarity between subsets of features to enhance the redundancy reduction process in the proposed methods.

The rest of the paper is organized as follows. Section 2 gives a brief review of previous work. Section 3 presents the proposed feature selection methods based on the ACO. Section 4 reports the experimental results on well-known datasets using different classifiers. Finally, Section 5 presents the conclusion and future work.

2. Review of feature selection algorithms

Feature selection is a fundamental research topic in pattern recognition with a long history since the 1970s, and there have been a number of attempts to review the feature selection methods [2,12,18,31]. In this section, we briefly review various feature selection methods that can be classified into four categories including filter, wrapper, embedded, and hybrid models.

In the filter model, each feature is ranked without considering any learning algorithms based on its discriminating power between different classes. Then a subset of features with the highest ranks is selected [5]. The filter model can broadly be classified into univariate and multivariate approaches [6,12,32]. The univariate filter model uses a specific statistical criterion to evaluate the relevance of each feature individually. To this end, a number of criteria have been proposed in the literature including information gain [33,34], Gini index [33,35], gain ratio [36,37], symmetrical uncertainty [34,38], chi-square test [8], Fisher score [6,39], Laplacian score [40], Relief [41], and term variance [1,6]. The univariate filter model is computationally very efficient due to independence from any learning algorithms. Although this model removes the relevant features, it does not consider the relation between features and cannot identify the redundant features. Moreover, both theoretical and empirical studies show that redundant features also affect the accuracy and computational time of the predictor model and should be removed as well [42].

On the other hand, the multivariate filter model has been developed for solving the problem of ignoring the dependency between features. Minimal-redundancy–maximal-relevance (mRMR) [19] is a well-known multivariate filter-based method which uses an incremental search process to select a subset of features with the highest relevance to the target class based on the mutual information criterion. Moreover, this criterion is used to determine the dependency between pairs of features. Random subspace method (RSM) [32] employed a multivariate search strategy on a randomly selected subset of features to better handle the noise in high dimensional datasets. Mitra et al. [43] presented a two-stage unsupervised feature selection method based on a clustering technique. In the first stage, the original feature set is divided into a number of clusters and then in the second stage, a representative feature is selected from each cluster. Haindl et al. [44] introduced a feature selection method based on mutual correlation to identify the redundancy between features. This method iteratively removes features with the largest average mutual correlations. Fast correlation-based filter (FCBF) [34,45] is an approximation filter-based method which uses the symmetric uncertainty criterion to analyze the relevance and redundancy of the features. In this method a subset of the relevance features is selected and then the final subset is created by identifying and removing the redundant features. Relevance–redundancy feature selection (RRFS) [3] is another multivariate feature selection method based on relevance and redundancy analyses. RRFS starts the selection process with most relevant features based on a given criterion and iteratively adds the next most relevant features to the selected feature subset in a greedy way. Most of the mentioned methods are greedy sequential feature selection ones based on a single iteration search process that can easily be trapped into local optimum [6].

Recently, Tabakhi et al. [6] proposed a multivariate filter method based on the ant colony optimization, called UFSACO. The method is an iterative improvement process where each feature has a chance of being selected in all iterations. The UFSACO performs an explicit redundancy analysis and implicit relevance analysis. However, one of its main limitations is that it cannot determine the relevance of features in datasets without redundancy between features and is thus incapable of eliminating irrelevant features.

In the wrapper model, a given learning algorithm is used to select a subset of features in the search space by maximizing the accuracy of the learning algorithm. In other words, the wrapper model is an iterative search process such that the results of the learning algorithm at each iteration are used to guide the search process [5]. Generally, wrapper-based methods can be classified into greedy and random search approaches [4,12]. The greedy search approach is based on the hill-climbing algorithm in which a single feature is added or removed iteratively in a greedy way. Sequential backward selection and sequential forward selection are two well-known greedy search methods [1,4]. On the other hand, the random search approach applies randomness into its search strategy to explore a large portion of the solution space. Examples of random methods include ant colony optimization (ACO) [24], particle swarm optimization (PSO) [46], genetic algorithm (GA) [47,48], random mutation hill-climbing [49,50], and simulated annealing (SA) [51]. Wrapper-based methods include the interaction with the learning algorithm and thus they outperform filter-based methods in term of prediction accuracy. However, these methods continuously use the learning algorithm in the search process and they are computationally more expensive, especially for high-dimensional datasets.

In the embedded model, a given learning algorithm is trained by an original feature set and the obtained results are used to determine the relevance of each feature. In other words, the feature selection process is embedded into the training of the learning algorithm [12,52]. Sugumaran et al. [53] proposed an embedded-based method which used a decision tree (DT) classifier for fault diagnostics of roller bearing. Guyon et al. [15] introduced the idea of using the support vector machine (SVM) for feature ranking. In this approach the relevance of a feature is determined by the weight that the SVM classifier assigns to the feature. In another approach, the naïve Bayes (NB) classifier is used to define the relevance of each feature based on a probability distribution [54]. ElAlami [55] presented an embedded-based method using the artificial neural network (ANN) classifier. The results of the trained ANN are used by GA to find the optimal feature set. Although the embedded model has a lower computational time compared to the wrapper model, it suffers from a time-consumption problem in high-dimensional datasets due to its interaction with a given learning algorithm.

Moreover, the hybrid model was developed to hold the advantages of both the filter and the wrapper models. In this model, the feature selection process is composed of two steps. In the first step, the filter model is applied to identify a relevant feature set and in the second step the final feature subset is selected by applying the wrapper model to the relevant feature set [5]. Leung and Hung [11] presented a hybrid method for gene selection in microarray datasets. In their method, different filter-based methods are used to select the initial subset and then the results of multiple wrapper-based methods using different classifiers are mixed in the final list of features. Unler et al. [5] integrated the mutual information based filter model within the PSO based wrapper model. Other examples of the hybrid model include mutual information and ACO [56], mutual information and GA [57], and information gain and sequential floating search [58]. The computational complexity of the hybrid model is lower than that of the wrapper model because it uses the reduced feature subset in its second step. However, the main idea behind the hybrid model is to use both the filter and the wrapper models in two separate steps that will lead to poor performance [4,6].

3. Proposed methods

This section describes the proposed feature selection methods based on the ant colony optimization (ACO). Section 3.1 describes the relevance–redundancy feature selection methods. Section 3.2 presents a redundancy reduction approach based on the similarities between subsets of features to enhance the accuracy of the methods. Moreover, the computational complexities of the proposed methods are analyzed in Section 3.3.

3.1. Relevance–redundancy feature selection based on ant colony optimization

The proposed methods, called RRFSACO_1 (relevance–redundancy feature selection based on ACO, version1) and RRFSACO_2, select a subset of features using the search strategy of the ACO algorithm. Therefore, in this section the graph representation of the search space, a detailed description of the proposed methods, initial pheromone, state transition and pheromone updating rules are described.

3.1.1. 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 weighted graph. This graph is defined as $G=\langle F, E\rangle$, where $F=\{F_1, F_2,\ldots, F_n\}$ is a set of original features and indicates the nodes in the graph and $E=\{(F_i, F_j): F_i, F_j \in F\}$ denotes the edges of the graph. The weight of each edge $(F_i, F_j)\in E$ is defined by a similarity value between features F_i and F_j as follows:

$$sim(F_i, F_j) = \left|\frac{F_i.F_j}{\|F_i\|\|F_j\|}\right| = \left|\frac{\sum_{k=1}^{p} F_{ik}F_{jk}}{\sqrt{\sum_{k=1}^{p} F_{ik}^{\ 2}}\sqrt{\sum_{k=1}^{p} F_{jk}^{\ 2}}}\right| \tag{1}$$

where p is the number of patterns and F_{ik} indicates the value of feature i for pattern k. According to Eq. (1), when two features are completely similar, their similarity value is equal to 1 and this value for two non-similar features is equal to 0.

Desirability and heuristic information are two basic components of any ACO algorithm in solving the feature selection problem. The desirability, called pheromone, is associated with the graph nodes (*i.e.*, the features) and shows the information collected by ants during the search process. Moreover, the heuristic information represents the prior knowledge about the problem. In this paper, we have used two kinds of heuristic information in the proposed methods. The first heuristic information is simply defined as the inverse of the similarity value between features and the second heuristic information is defined as the relevance of each feature which is associated with the features. To evaluate the relevance of each feature, we have used the term variance [1] as a simplest unsupervised measure. Fig. 1 shows the representation of the search space in the feature selection problem.

3.1.2. Description of the proposed methods

The pseudo-code of the proposed ACO based feature selection method is presented in Fig. 2. The proposed method consists of three major steps including (i) initialization step, (ii) feature probability computation step, and (iii) final feature subset selection step.

In the first step (lines 2–5), the relevance of each feature is evaluated by a given criterion. Then, the similarity values between each pair of features are computed and associated to the graph edges. Finally, the initial pheromone of each feature is calculated. Section 3.1.3 describes the corresponding details.

The second step (lines 6–17) is used to compute the probability of each feature in an iterative process. In this step, the feature counter (FC) array is defined in order to count the number of times that a given feature is selected by the ants. During each iteration, at first, the initial values of FC are set to zero. Then A ants are placed randomly on the different nodes in the graph. Thereafter, each ant selects the next features according to a “*state transition rule*” in an iterative way until a given number of features are selected by the ant. The state transition rule is a function of the desirability and heuristic information to guide the search process of the ants (see Section 3.1.4 for details). Each time a feature F_i is selected by an ant, its corresponding feature counter (*i.e.*, $FC[i]$) is increased. Finally, at the end of each iteration, the pheromone

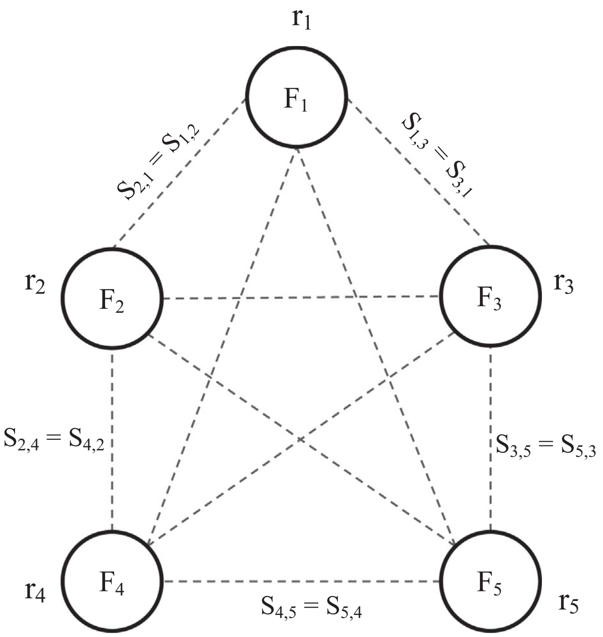

Fig. 1. The graph representation of the search space for the feature selection problem. $S_{i,j}$ is a similarity value between features F_i and F_j and r_i denotes a relevance value of feature F_i.

values of the features are updated according to the "*global pheromone updating rule*". In this rule, a fraction of the pheromone evaporates on all nodes and then the features which are frequently selected by the ants and have obtained higher FC values, will receive greater amounts of pheromone (see Section 3.1.5 for details). The learning procedure is repeated until the maximum number of cycles C is reached.

In the third step (lines 18 and 19), the features are sorted in decreasing order of their pheromone values (*i.e.*, probability values of each feature) and then, the top m features are kept as the final feature subset.

3.1.3. *Initial pheromone*

In the RRFSACO_1 and RRFSACO_2 methods, two different strategies are used for initializing the intensity of pheromones. In the RRFSACO_1 method, the normalized values of the relevance values of the features are used as the initial intensity of pheromones ($i.e., \tau_i(1) = normalize(r_i), \forall\ i = 1 \ldots n$). Moreover, the softmax scaling function [1] is used to normalize the relevance values of the features in the interval [0, 1]. The initialization based on the relevance criterion guides the search process and thereby reduces the search space.

On the other hand, in the RRFSACO_2 method, the initial intensity of pheromone associated with each node is set to a constant value (*i.e.*, $\tau_i(1) = c,\ \forall\ i = 1 \ldots n$).

Algorithm 1. *Relevance-Redundancy Feature Selection based on Ant Colony Optimization (RRFSACO)*

```
Input:  $D: p \times n$ matrix, $n$ dimensional training set with $p$ patterns.
        $m$ ($\le n$): the number of features to keep for final reduced feature set.
        $C$: the maximum number of allowed iterations.
        $A$: define the number of ants.
        $F$: the number of selected features by each ant in each iteration.
Output: $\widetilde{D}: p \times m$ matrix, reduced dimensional training set.

1:  begin algorithm
2:  Compute the relevance $r_i$ of each feature, $\forall\ i = 1 \ldots n$.
3:  Compute the similarity $S_{i,j}$ between features, $\forall\ i,j = 1 \ldots n$.
4:  Set the heuristic information : $\eta_1(F_i) = r_i$ , $\eta_2(F_i, F_j) = \frac{1}{S_{i,j}}, \forall\ i,j = 1 \ldots n$.
5:  Initialize the intensity of pheromone $\tau_i(1)$ associated with the features, $\forall\ i = 1 \ldots n$.
6:  for $t = 1$ to $C$ do
7:      Set the initial features counter $FC[i]$ to zero, $\forall\ i = 1 \ldots n$.
8:      Place the ants randomly on the graph nodes.
9:      for $i = 1$ to $F$ do
10:         for $k = 1$ to $A$ do
11:             Choose the next unvisited feature $f$ according to the state transition rule.
12:             Move the $k$-th ant to the new selected feature $f$.
13:             Increment feature counter associated with feature $f$
14:         end for
15:     end for
16:     Update pheromone according to the pheromone updating rule.
17: end for
18: Sort the features by decreasing order of their pheromones $\tau_i$.
19: Build $\widetilde{D}$ from $D$ by keeping the top $m$ features with highest pheromone.
20: end algorithm
```

Fig. 2. pseudo-code of the proposed ACO based feature selection method.

3.1.4. State transition rule

In the RRFSACO_1 and RRFSACO_2 methods, each ant traverses the graph using both greedy and probabilistic state transition rules. In the greedy way, the kth ant which is placed on feature F_i chooses the next feature F_j by applying the following formula:

$$j = \underset{u \in J_i^k}{\arg\max}\{[\tau_u][\eta_1(F_u)]^{\alpha}[\eta_2(F_i, F_u)]^{\beta}\} \text{ if } q \le q_0 \tag{2}$$

where J_i^k is the unvisited set of features, τ_u denotes the pheromone intensity value associated with feature F_u, $\eta_1(F_u)$ is the relevance value of feature F_u, $\eta_2(F_i, F_u)=1/sim(F_i, F_u)$ indicates the inverse of the similarity value between features F_i and F_u, parameters α and β determine the importance of the pheromone versus two heuristic information values η_1 and η_2, q is a random number in the range [0, 1], and q_0 is a predefined constant parameter ($0 \le q_0 \le 1$).

In the probabilistic way, the kth ant selects the next feature F_j with a probability $P_k(i, j)$ which is calculated as follows:

$$P_k(i,j) = \begin{cases} \frac{[\tau_j][\eta_1(F_j)]^{\alpha}[\eta_2(F_i,F_j)]^{\beta}}{\sum_{u \in J_i^k}[\tau_u][\eta_1(F_u)]^{\alpha}[\eta_2(F_i,F_u)]^{\beta}} & \text{if } j \in J_i^k \\ 0 & \text{otherwise} \end{cases} \quad \text{if } q > q_0 \tag{3}$$

Eqs. (2) and (3) are used by both methods. However, in the RRFSACO_1 parameter α is set to 0 (*i.e.*, $\alpha=0$) and the relevance of the selected features is not considered in the search process of the ants.

3.1.5. Global pheromone updating rule

After all ants finish their traverses, the pheromone values of the features are updated using the following equation:

$$\tau_i(t+1) = (1-\rho)\,\tau_i(t) + \frac{FC[i]}{\sum_{j=1}^{n} FC[j]} \tag{4}$$

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, n is the number of original feature sets, and $FC[i]$ is the counter value of feature F_i.

3.2. A redundancy reduction approach based on similarities between subsets of features

The RRFSACO_1 and RRFSACO_2 methods select the next feature based on its similarity value with that of the previously selected one. This kind of selection strategy leads to poor prediction accuracy in some datasets. For example, consider Fig. 3(a): An ant is currently at feature F_1 in which it has to decide whether to select two features among F_2, F_3, and F_4. Suppose that all features have the same initial pheromones, only the greedy rule (Eq. (2)) is used to select the next feature, and parameter α is equal to 0. Therefore, the ant selects features F_2 and F_3 based on the state transition rule (*i.e.*, Eq. (2)) and finally the ant terminates its traverse. Thus, the selected feature subset is $\{F_1, F_2, F_3\}$. It can be seen from Fig. 3(a) that the similarity value between features F_1 and F_3 is equal to 1 (*i.e.*, completely similar), but this similarity was not considered in the feature selection process. However, these two features are redundant and only one of them should be selected.

Therefore, to overcome this problem, the next feature can be selected based on the lowest average similarity with those of the previously selected features. To understand this search strategy, consider Fig. 3(b): At first, the ant is placed on feature F_1 randomly, then feature F_2 is selected based on the greedy state transition rule (*i.e.*, Eq. (2)). The ant can select features F_3 or F_4, when it is positioned at feature F_2. The average similarities of F_3 and F_4 to the previously selected features F_1 and F_2 are equal to 0.65 and 0.4, respectively. Therefore, feature F_4 has the lowest average similarity value and therefore the final feature subset is $\{F_1, F_2, F_4\}$. To define the average similarity measure, suppose that the kth ant currently selects F_{m-1}^k, the subset of features with m–1 features. The task is to assign the new heuristic information value to the unvisited features as follows:

$$\eta_2(F_j) = \frac{1}{(1/(m-1))\sum_{F_z \in F_{m-1}^k} sim(F_j, F_z)}, \quad F_j \in X - F_{m-1}^k \tag{5}$$

where X denotes the original feature set. Therefore, according to this search strategy, the heuristic information (*i.e.*, η_2) is incrementally updated along the search process in Eqs. (2) and (3). Thus, the IRRFSACO_1 and IRRFSACO_2 methods are the modified versions of RRFSACO_1 and RRFSACO_2, respectively.

3.3. Time complexity analysis

Suppose that n is the number of the original features and p is the number of patterns. In the first step of the RRFSACO_1 and RRFSACO_2 methods (lines 2 and 3), the relevance values of the features are evaluated using the term variance measure, thus the time complexity is $O(np)$. Moreover, the similarity values between each pair of features are computed, so, the time complexity is $O(n^2p)$. Therefore, the overall time complexity of this step is $O(np+n^2p) = O(n^2p)$. Furthermore, in the second step (lines 6–17), A ants start to search the solution space from different points. The search process will be repeated for a number of iterative cycles (*i.e.*, C). Therefore, the time complexity of this part is $O(CAFn)$, where F is the number of the features selected by each ant in each iteration. The time complexity of this part can be reduced to $O(CFn)$, when the ants run in a parallel way. In the third step (lines 18 and 19), all of the features are sorted based on their pheromone values with the time complexity of $O(n \log n)$ and then the m features with highest values are selected as the final subset of features. Consequently, the time complexity of the RRFSACO based methods (versions 1 and 2) is $O(n^2p+CFn+n \log n) = O(n^2p+CFn)$. Generally $F \lll n$ and the time complexity can be reduced to $O(n^2p)$.

On the other hand, in the IRRFSACO based methods (versions 1 and 2), only the second step of the methods was changed. In other words, in the second step of the IRRFSACO_1 and RRFSACO_2 methods, in each iteration of the ant, the similarity value of the next feature to those of the selected ones should be computed, so, the time complexity of this step is $O(CAF^2n)$. When the ants simultaneously start to search the solution space (*i.e.*, in a parallel way), the time complexity of this step can be reduced to $O(CF^2n)$. Consequently, the overall time complexity of the IRRFSACO_1 and IRRFSACO_2 is $O(n^2p+CF^2n+n \log n) = O(n^2p+CF^2n)$. This indicates that when the number of the selected features is much smaller than the number of the original features (*i.e.*,$F \lll n$), especially on high dimensional datasets, the time complexity of IRRFSACO based methods is $O(n^2p)$.

The proposed methods incorporate the iteration improvement process, while filter-based methods use only a single iteration throughout their search processes. Therefore, the time complexity of the proposed methods is a little bit more expensive than those of filter-based methods. On the other hand, the search process of the proposed methods is independent of the learning algorithm. Hence, the time complexity of the proposed methods is much faster than those of wrapper-based methods.

4. Experimental results

In this section, we present the empirical results to compare the performance of the proposed methods with those of well-known feature selection methods based on the filter model using several frequently used datasets.

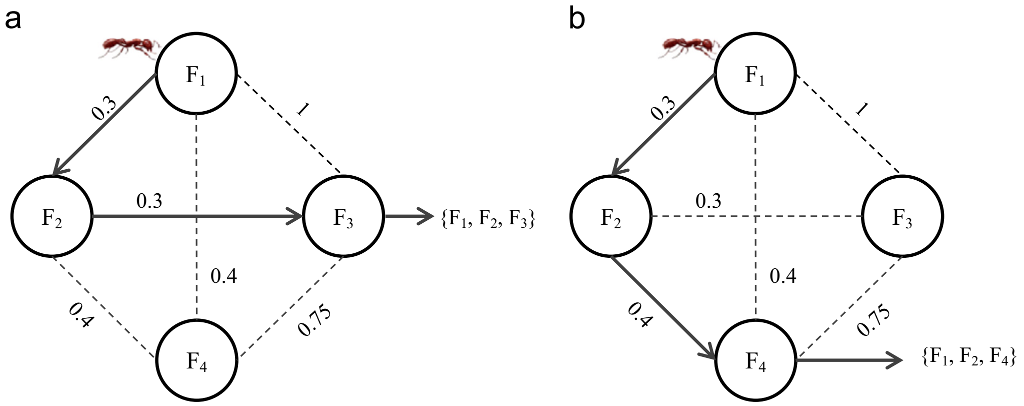

Fig. 3. An example of the redundancy between a subset of features.

4.1. Datasets

The experiments have been performed on several different datasets including *Glass*, *Wine*, *Hepatitis*, *Wisconsin Diagnostic Breast Cancer* (*WDBC*), *Ionosphere*, *Dermatology*, *SpamBase*, *Sonar*, and *Arrhythmia* from UCI machine learning repository [59], *Colon* at the Bioinformatics Research Group from Universidad Pablo de Olavide [60], and *Madelon* and *Arcene* from NIPS2003 feature selection challenge [61]. These datasets are well-known datasets used in the literature [3–5,22,42] and consist of varying numbers of features and patterns in two-class or multiclass classification tasks. Table 1 shows a brief description of the datasets used in the experiments.

The *Hepatitis*, *Dermatology*, and *Arrhythmia* datasets have several missing values in the features. The well-known technique for dealing with missing values is completion of the values by means of the available data of the respective feature [1].

The class labels for the *Madelon* and *Arcene* datasets are not available for the test set. Accordingly, the patterns of the validation set have been used to evaluate the performance of the methods.

Table 1
Characteristics of the datasets used in the experiments.

Dataset	Features	Classes	Patterns
Glass	9	6	214
Wine	13	3	178
Hepatitis	19	2	155
WDBC	30	2	569
Ionosphere	34	2	351
Dermatology	34	6	366
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

4.2. Experimental setting

The performance of the proposed methods is compared with those of 12 well-known and state-of-the-art univariate and multivariate feature selection methods based on the filter model. The univariate methods include information gain (IG) [33], gain ratio (GR) [36,37], symmetrical uncertainty (SU) [34,38], Gini index (GI) [33,35], Fisher score (FS) [39], term variance (TV) [1], and Laplacian score (LS) [40]. Moreover, the multivariate feature selection methods include unsupervised feature selection based on ACO (UFSACO) [6], minimal-redundancy–maximal-relevance (mRMR) [19], mutual correlation (MC) [44], random subspace method (RSM) [32], and relevance–redundancy feature selection (RRFS) [3].

In the proposed methods, there are different adjustable parameters that need to be set. The maximum number of iterations is set to $C=50$, the pheromone decay rate is set to $\rho=0.2$, the constant parameter q_0 in the state transition rule (Eqs. (2) and (3)) is set to 0.7, parameter β is set to 1, and finally the number of ants is equal to the number of the original features in each dataset ($A=\#features$). But parameter A is set to 100 in the datasets with more than 100 features ($A=100$). Moreover, in the RRFSACO_2 and IRRFSACO_2 methods the initial intensity of the pheromone for each feature is set to 0.2 ($\tau_i(1)=0.2$) and parameter α is set to 1. Furthermore, in the RRFSACO_1 and IRRFSACO_1 methods parameter α is set to 0.

For the remaining methods, there are some parameters that need to be set. The maximum allowed similarity between features in the RRFS method is set to several values in the interval $[0.5, 1)$ as recommended in the original paper [3]. To make a fair comparison, the number of selections parameter in the RSM method is set to 50 times.

The search strategy of the proposed methods is independent of any classifiers and thus we expect that the proposed methods attain acceptable performance on the different classifiers. To this end, three widely used classifiers including support vector machine (SVM) [15], decision tree (DT) [37], and naïve Bayes (NB) [1] were selected to evaluate the feature selection methods. The SMO, J48, and Naïve Bayes are used as the SVM, DT, and NB classifiers, respectively, implemented in the WEKA data mining software package[62]. The SVM classifier uses a polykernel as its kernel and applies the one-against-rest technique in multiclass problems. The DT classifier adopts the post-pruning algorithm in the pruning phase and the confidence factor for pruning is set to 0.25.

The average classification error rate over 5 different runs with random train/test partition of the datasets was used as a performance measure in the experiments. In each run, a given dataset was randomly divided into a training set ($\frac{2}{3}$ of the dataset) and a test set. All the experiments were carried out on an Intel Core-i3 CPU with 4 GB RAM, using the Java implementation.

4.3. Results and discussion

In this section, we present the comprehensive study to evaluate the performance of the proposed methods. In the first set of experiments, the RRFSACO_1 and RRFSACO_2 methods are compared with the mentioned feature selection methods in terms of execution time and classification error rate. In the second set of experiments, we explore the effect of using the redundancy reduction approach in the IRRFSACO_1 and IRRFSACO_2 methods.

4.3.1. Comparison between RRFSACO based methods and other feature selection methods

Tables 2–4 report the average classification error rates (over 5 independent runs) of the proposed methods (*i.e.*, RRFSACO_1 and RRFSACO_2) compared to those of the unsupervised feature selection methods including UFSACO, RSM, MC, RRFS, TV, and LS by applying SVM, DT, and NB classifiers, respectively. The number of selected features are chosen based on the best performance obtained by the proposed methods.

From Table 2 it can be observed that the proposed methods get the lowest classification error rates on most of the datasets, all except the *SpamBase* dataset. For example, for the *Dermatology* dataset, RRFSACO_1 and RRFSACO_2 got 22.96% and 26.00% classification error rates, respectively, while for the UFSACO,

Table 2
Average classification error rates (in %) over 5 runs of SVM classifier using the unsupervised feature selection methods considered on different datasets. The best result for each dataset is indicated in bold face and underlined and the second best is in bold face.

Datasets	#Selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
Glass	5	**50.41**	48.21	51.78	54.52	51.50	54.79	54.24	52.60
Wine	3	11.47	**12.46**	12.78	27.87	12.57	16.72	39.34	15.85
Hepatitis	8	17.54	**19.24**	20.56	22.07	23.02	19.62	21.89	19.62
WDBC	5	**9.41**	9.28	9.28	16.18	11.03	9.64	10.10	10.20
Ionosphere	30	**11.66**	14.83	11.39	12.16	14.67	18.89	13.61	17.50
Dermatology	5	22.96	**26.00**	39.44	48.72	50.08	35.60	32.56	51.20
SpamBase	50	11.91	11.34	11.94	11.48	10.43	10.19	**10.07**	10.01
Sonar	20	31.54	**28.16**	31.83	27.04	30.70	29.29	37.74	43.94
Arrhythmia	40	**35.71**	34.93	40.78	46.36	45.71	42.20	36.49	42.99
Madelon	40	39.33	**39.55**	**39.55**	47.17	48.67	–	39.33	39.83
Colon	20	12.72	**15.45**	21.81	24.54	38.18	24.54	21.81	33.63
Arcene	60	37.00	**34.50**	36.00	45.00	43.00	27.00	44.00	45.00
Average		**24.31**	**24.50**	27.26	31.93	31.63	–	30.10	31.86

Table 3
Average classification error rates (in %) over 5 runs of DT classifier using the unsupervised feature selection methods considered on different datasets. The best result for each dataset is indicated in bold face and underlined and the second best is in bold face.

Datasets	#Selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
Glass	5	36.43	34.24	42.46	35.61	41.09	**34.79**	38.08	35.88
Wine	3	15.08	13.11	16.39	30.60	**13.66**	**13.66**	32.24	14.08
Hepatitis	8	21.32	**20.94**	23.01	20.97	19.43	22.07	23.58	20.95
WDBC	5	8.76	8.24	**8.09**	13.66	8.92	9.02	7.94	8.14
Ionosphere	30	9.67	**11.16**	11.39	11.66	11.50	13.61	11.41	13.22
Dermatology	5	**28.08**	27.76	40.64	49.52	52.00	34.80	31.76	50.24
SpamBase	50	7.51	8.34	7.86	8.20	8.24	8.04	**7.58**	8.09
Sonar	20	34.36	30.70	33.52	35.77	**31.26**	34.36	36.05	38.59
Arrhythmia	40	**37.01**	36.75	43.24	46.62	45.71	47.40	45.97	43.25
Madelon	40	22.75	21.17	**20.67**	51.33	49.00	–	22.17	20.00
Colon	20	23.63	27.27	**24.54**	28.18	33.63	34.54	31.81	39.09
Arcene	60	24.50	**29.50**	30.80	47.80	44.00	39.00	32.00	44.00
Average		22.42	**22.43**	25.22	31.66	29.87	–	26.72	27.96

Table 4
Average classification error rates (in %) over 5 runs of NB classifier using the unsupervised feature selection methods considered on different datasets. The best result for each dataset is indicated in bold face and underlined and the second best is in bold face.

Datasets	#selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
Glass	5	47.94	**51.50**	54.24	59.72	53.42	54.24	60.54	52.87
Wine	3	15.70	16.03	16.39	22.41	**10.93**	10.49	25.68	10.49
Hepatitis	8	21.51	20.00	20.37	17.73	19.81	**18.36**	23.77	19.43
WDBC	5	**8.71**	9.28	7.58	13.35	9.07	9.95	9.69	9.48
Ionosphere	30	19.33	18.50	19.44	**16.16**	16.00	22.22	20.83	23.33
Dermatology	5	**28.64**	26.48	32.56	46.48	46.72	32.56	30.64	48.40
SpamBase	50	**20.74**	20.97	20.13	20.98	21.00	21.15	21.51	20.76
Sonar	20	**35.49**	**35.49**	36.33	35.77	30.14	36.05	41.40	39.15
Arrhythmia	40	67.91	56.75	56.62	**48.70**	46.75	50.23	51.30	50.39
Madelon	40	**39.75**	39.17	40.50	48.50	48.67	–	40.50	40.83
Colon	20	**18.18**	15.45	28.18	26.36	31.81	32.72	41.81	47.27
Arcene	60	**37.00**	**37.00**	**33.60**	48.40	44.00	39.00	**37.00**	53.00
Average		**30.08**	28.89	30.50	33.71	31.53	–	33.72	34.62

RSM, MC, RRFS, TV, and LS methods this value was recorded 39.44%, 48.72%, 50.08%, 35.60%, 32.56%, and 51.20%, accordingly. Moreover, the average values over all of the datasets, reported in the last row of Table 2 show that the proposed methods outperform all the other methods in terms of classification accuracy.

The results of Table 3 show that the RRFSACO_1 method is superior to all other feature selection methods in terms of classification error rate over the *Ionosphere*, *SpamBase*, *Colon*, and *Arcene* datasets. Moreover, the performance of the RRFSACO_2 is better than those of the other methods on the *Glass*, *Wine*, *Dermatology, Sonar,* and *Arrhythmia* datasets, and it acquires the second lowest error rate on the *Hepatitis*, *Ionosphere*, and *Arcene* datasets. The last row of Table 3 shows that the RRFSACO_1 is the best method in terms of average classification error rate improvement over all of the datasets using the DT classifier. In other words, the RRFSACO_1 outperforms UFSACO by 2.8%, RSM by 9.24%, MC by 7.45%, TV by 4.3%, and LS by 5.54%. Furthermore, the RRFSACO_2 method attains the second lowest average value over all the datasets compared to those of the unsupervised methods.

Table 4 reports the average classification error rate of the NB classifier over 5 independent runs. It can be seen from Table 4 that the RRFSACO_1 acquires the second lowest error rate on *WDBC*, *Dermatology*, *SpamBase*, *Sonar*, *Madelon*, *Colon*, and *Arcene* datasets and it is superior to the RSM, MC, RRFS, TV, and LS methods. On the other hand, RRFSACO_2 is superior to the unsupervised feature selection methods when *Dermatology*, *Madelon*, and *Colon* datasets are used. Furthermore, the average values of the RRFSACO_1 and RRFSACO_2 methods over all the datasets were 30.08% and 28.89%, respectively, and they acquire the second and the first lowest average classification error rates, correspondingly.

Furthermore, the runtimes of the feature selection methods are evaluated over all of the datasets reported in Table 1. Table 5 records the average execution times (in seconds) taken to the feature selection process by the RRFSACO_1, RRFSACO_2, and the unsupervised feature selection methods. From the results it can be observed that the proposed methods performed faster compared to the LS method. Also, the execution time of the proposed methods is comparable to that of the UFSACO method. On the other hand, the RRFS methods get the lowest execution times over all the datasets.

From Tables 2–5, it can be concluded that although the runtimes of the proposed methods were not faster than those of

Table 5
Execution times (in seconds) for the unsupervised feature selection methods.

Datasets	#selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
Glass	5	0.0030	0.0070	0.0040	0.0050	0.0020	0.0010	0.0010	0.0130
Wine	3	0.0082	0.0050	0.0410	0.0072	0.0032	0.0004	0.0001	0.0244
Hepatitis	8	0.0158	0.0178	0.0250	0.0138	0.0038	0.0008	0.0004	0.0222
WDBC	5	0.0220	0.0196	0.0256	0.0094	0.0018	0.0008	0.0006	0.3570
Ionosphere	30	0.0578	0.0602	0.0784	0.0062	0.0010	0.0012	0.0004	0.1308
Dermatology	5	0.0332	0.0190	0.0280	0.0168	0.0014	0.0016	0.0014	0.1358
SpamBase	50	0.3553	0.3707	0.3493	0.0827	0.0530	0.0120	0.0100	90.8950
Sonar	20	0.1232	0.1234	0.1006	0.0096	0.0020	0.0012	0.0004	0.0844
Arrhythmia	40	1.6557	1.7183	1.6023	0.1280	0.0953	0.0083	0.0047	1.3210
Madelon	40	12.3150	14.0860	11.3510	6.0220	4.4293	0.0733	0.0760	271.54
Colon	20	8.1367	8.6420	6.484	0.6756	0.6320	0.0050	0.0090	0.1826
Arcene	60	246.8900	249.8800	229.9500	56.2150	67.7100	0.0800	0.1185	5.0625
Average		22.4680	22.9120	20.8360	5.2659	6.0779	0.0154	0.0185	30.8140

Table 6
Average classification error rates (in %) over 5 runs of the unsupervised feature selection methods on different number of selected features of the Arrhythmia dataset using SVM classifier. Std. is the standard deviation of the classification error rates. The best result for each number of features is indicated in bold face and underlined and the second best is in bold face.

#selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
10	34.80	**39.22**	43.90	45.06	43.51	50.52	46.75	50.13
20	**40.12**	39.99	40.78	43.89	45.46	42.47	41.43	46.49
30	40.25	35.84	43.43	43.33	49.22	43.89	**40.00**	41.95
40	**35.71**	34.93	40.78	46.36	45.71	42.20	36.49	42.99
50	40.25	**36.23**	43.12	45.45	49.22	37.79	35.97	42.21
Average	**38.23**	37.24	42.40	44.82	46.62	43.37	40.13	44.75
Std.	2.73	2.22	**1.51**	1.22	2.52	4.6	4.36	3.51

Table 7
Average classification error rates (in %) over 5 runs of the unsupervised feature selection methods on different number of selected features of the Arrhythmia dataset using DT classifier. Std. is the standard deviation of the classification error rates. The best result for each number of features is indicated in bold face and underlined and the second best is in bold face.

#selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
10	37.66	**42.98**	43.77	44.93	43.51	52.47	53.77	50.13
20	**40.51**	39.35	40.91	44.29	45.44	53.38	51.43	47.01
30	38.18	**40.51**	42.85	43.37	49.22	51.95	50.26	43.12
40	**37.01**	36.75	43.24	46.62	45.71	47.40	45.97	43.25
50	**39.48**	37.14	47.01	45.71	49.87	45.45	46.62	42.59
Average	38.57	**39.35**	43.56	44.98	46.75	50.13	49.61	45.22
Std.	**1.41**	2.56	2.21	1.25	2.7	3.49	3.29	3.26

the unsupervised methods, their performances in terms of classification error rate were superior to those of the unsupervised feature selection methods over different classifiers. Therefore, the proposed methods perform the trade-off between classification accuracy and execution time.

Additionally, Tables 6–11 show the results of the comparison between the performance of the proposed methods and those of unsupervised feature selection methods over different numbers of selected features using the *Arrhythmia*, *Colon*, and *Arcene* datasets.

Table 6 compares the classification error rates of the proposed methods with those of the unsupervised feature selection methods when the SVM classifier was used on the *Arrhythmia* dataset. It can be seen from the results that when the number of selected features is 20, 30, and 40, the RRFSACO_2 outperforms the other

Table 8
Average classification error rates (in %) over 5 runs of the unsupervised feature selection methods on different number of selected features of the Colon dataset using SVM classifier. Std. is the standard deviation of the classification error rates. The best result for each number of features is indicated in bold face and underlined and the second best is in bold face.

#selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
20	12.72	**15.45**	21.81	24.54	38.18	24.54	21.81	33.63
40	13.63	**13.64**	15.45	18.18	29.09	18.18	26.36	40.00
60	**14.54**	10.91	22.72	24.54	19.09	18.18	20.00	35.45
80	23.63	**15.45**	9.09	19.09	19.09	18.18	17.27	30.91
100	**16.36**	**14.54**	18.18	**16.36**	19.09	24.54	17.27	30.90
Average	**16.18**	14.00	17.45	20.54	24.91	20.72	20.54	34.18
Std.	4.38	1.88	5.50	3.78	8.59	**3.48**	3.78	3.78

Table 9
Average classification error rates (in %) over 5 runs of the unsupervised feature selection methods on different number of selected features of the Colon dataset using NB classifier. Std. is the standard deviation of the classification error rates. The best result for each number of features is indicated in bold face and underlined and the second best is in bold face.

#selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
20	**18.18**	15.45	28.18	26.36	31.81	32.72	41.81	47.27
40	**19.09**	17.27	26.36	22.72	22.72	30.90	38.18	39.09
60	16.36	**13.63**	12.72	29.99	39.08	25.45	35.45	50.00
80	17.27	23.63	**19.09**	30.90	25.45	**19.09**	29.08	38.18
100	**20.90**	27.27	26.36	30.90	18.18	27.27	24.54	50.90
Average	18.36	**19.45**	22.54	28.17	27.45	27.09	33.81	45.09
Std.	1.75	5.77	6.51	**3.58**	8.16	5.31	6.97	6.05

Table 10
Average classification error rates (in %) over 5 runs of the unsupervised feature selection methods on different number of selected features of the Arcene dataset using DT classifier. Std. is the standard deviation of the classification error rates. The best result for each number of features is indicated in bold face and underlined and the second best is in bold face.

#selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
20	44.50	44.00	**32.60**	46.60	44.00	38.00	29.00	44.00
40	36.50	28.99	**33.00**	48.00	44.00	35.00	44.00	44.00
60	24.50	**29.50**	30.80	47.80	44.00	39.00	32.00	44.00
80	**31.00**	29.50	38.00	45.80	44.00	**31.00**	32.00	44.00
100	31.00	**31.50**	35.00	48.80	44.00	32.00	32.00	44.00
Average	**33.50**	32.70	33.88	47.40	44.00	35.00	33.80	44.00
Std.	7.47	6.39	2.74	**1.19**	0.00	3.54	5.85	0.00

Table 11
Average classification error rates (in %) over 5 runs of the unsupervised feature selection methods on different number of selected features of the Arcene dataset using NB classifier. Std. is the standard deviation of the classification error rates. The best result for each number of features is indicated in bold face and underlined and the second best is in bold face.

#selected features	RRFSACO_1	RRFSACO_2	UFSACO	RSM	MC	RRFS	TV	LS
20	**43.00**	44.00	41.20	44.40	44.00	44.00	**43.00**	54.00
40	43.00	41.00	32.40	46.00	44.00	**37.00**	43.00	52.00
60	**37.00**	**37.00**	33.60	48.40	44.00	39.00	**37.00**	53.00
80	34.00	**33.00**	34.80	48.40	44.00	28.00	34.00	52.00
100	36.00	37.00	**32.60**	47.60	46.00	28.00	34.00	48.00
Average	38.60	38.40	34.92	46.96	44.40	**35.20**	38.20	51.80
Std.	4.16	4.22	3.64	**1.73**	0.89	7.05	4.55	2.28

feature selection methods in terms of classification error rate and gets the second lowest error rate for the other cases. Also, in most cases, the RRFSACO_1 has better performance compared to the unsupervised feature selection methods, except for the RRFSACO_2 method. Furthermore, the obtained average values of the RRFSACO_1 and RRFSACO_2 methods over different numbers of selected features were 38.23% and 37.24% which indicates that the overall performance of the proposed methods is much better than those of the unsupervised feature selection methods.

Furthermore, Table 7 shows similar results when the methods are applied on the *Arrhythmia* dataset using the DT classifier. It can be seen from the results that the proposed methods attained significantly lower classification error rates than the other feature selection methods. Moreover, the worst classification error rates of the RRFSACO_1 and RRFSACO_2 were 40.51% and 42.98%, correspondingly, while for the UFSACO, RSM, MC, RRFS, TV, and LS methods the worst classification error rates were reported 47.01%, 46.62%, 49.87%, 53.38%, 53.77%, and 50.13%, respectively. On the other hand, the average classification error rates over different numbers of selected features show that the RRFSACO_1 outperforms UFSACO by 4.99%, RSM by 6.41%, MC by 8.18%, RRFS by 11.56%, TV by 11.04%, and LS by 6.65%. Additionally, the RRFSACO_2 method acquires the second lowest error rate and is only inferior to the RRFSACO_1 method.

Table 8 shows the classification error rates of the proposed methods and unsupervised feature selection methods for SVM classifier on the *Colon* dataset. It can be observed from Table 8 that RRFSACO_1 acquired the lowest classification error rates when 20 and 40 features were selected and it obtained the second lowest error rates when the number of selected features was 60 and 100. On the other hand, RRFSACO_2 got the best result when the number of selected features was 60 and 100, and for the other cases, the second best result is attained. Consequently, the average values reported in Table 8 show that RRFSACO_1 and RRFSACO_2 with classification error rates 16.18% and 14%, respectively, outperforms the other feature selection methods.

Moreover, Table 9 presents similar results when the methods are applied on the *Colon* dataset using the NB classifier. The reported results show that the classification error rate of RRFSACO_1 is lower than those of the other methods when 80 features were selected and it got the second lowest error rates when the number of selected features was 20, 40, and 100. Additionally, the results show that the performance of the RRFSACO_2 method is superior to the other methods when 20 and 40 features were selected. For example, when 20 features are selected, RRFSACO_2 outperforms UFSACO by 12.73%, RSM by 10.91%, MC by 16.36%, RRFS by 17.27%, TV by 26.36%, and LS by 31.82%. Finally, the RRFSACO_1 and RRFSACO_2 methods achieved the average classification error rates 18.36% and 19.45%, correspondingly, over different numbers of features and laid on the first and second places among the mentioned feature selection methods.

Table 10 demonstrates the comparison results when the DT classifier was used on the *Arcene* dataset. The results show that the RRFSACO_1 method was superior to all the other methods when the number of selected features was 60 and 100. Also, the RRFSACO_2 achieved the lowest error rate when the number of selected features was 40 and 80, and it got the second lowest error rate when the number of selected features was 60 and 100. For example, when 60 features were selected, RRFSACO_1 performed better than UFSACO by 6.3%, RSM by 23.3%, MC by 19.5%, RRFS by 14.5%, TV by 7.5%, and LS by 19.5%. Finally, it can be concluded that the RRFSACO_2 and RRFSACO_1 lay on the first and second places, respectively, among the unsupervised feature selection methods in terms of average classification error rate.

Furthermore, Table 11 shows similar results when the NB classifier is used on the *Arcene* dataset. The results show that when the number of selected features was 20 and 60, the RRFSACO_1 achieved the lowest error rates, expected than the UFSACO method. Furthermore, the RRFSACO_2 attained the lowest error rates when the number of selected features was 60 and 80, expected than the UFSACO and the RRFS methods. It is clear that the overall performance of the proposed methods is better than those of the RSM, MC, and LS methods and their performances are comparable with that of the TV method.

Moreover, the proposed methods have been compared to the supervised multivariate feature selection methods. Table 12 illustrates the average classification error rates of the RRFSACO_1, RRFSACO_2, mRMR, and RRFS methods by applying SVM, DT, and NB classifiers. From the results it is clear that RRFSACO_1 is superior to all other methods when SVM classifier has been applied on *Hepatitis*, *Dermatology*, *Madelon*, and *Colon* datasets. Also, RRFSACO_2 acquired the lowest error rates on *Glass*, *Wine*, *WDBC*, *SpamBase*, *Sonar*, and *Arrhythmia* datasets when SVM classifier was used. Additionally, it can be seen that the classification error rates of RRFSACO_1 are better than those of the other methods when DT classifier was used on *Ionosphere*, *Sonar*, and *Arcene* datasets. On the other hand, RRFSACO_2 acquired the lowest error rates for DT classifier on *Wine*, *WDBC*, *Arrhythmia*, and *Colon* datasets. Moreover, Table 12 shows that similar results have been reported when NB classifier is used. For example, when RRFSACO_1 is applied on *Glass*, *Wine*, *WDBC*, and *Madelon* datasets, the lowest classification error rates on NB classifier is obtained. Therefore, it can be concluded that the overall performance of the proposed methods is superior to those of the supervised multivariate methods over different datasets, especially when SVM and DT classifiers are used.

Table 12
Classification error rate (average over 5 runs, in %), with respect to the number of selected features by proposed methods and supervised multivariate methods for different datasets, using SVM, DT, and NB classifiers. The best result for each dataset is indicated in bold face.

Datasets	#selected features	SVM Classifier				DT Classifier				NB Classifier			
		RRFSACO_1	RRFSACO_2	mRMR	RRFS	RRFSACO_1	RRFSACO_2	mRMR	RRFS	RRFSACO_1	RRFSACO_2	mRMR	RRFS
Glass	5	50.41	**48.21**	55.89	52.87	36.43	34.24	**31.77**	35.34	**47.94**	51.50	55.34	51.50
Wine	10	4.59	**3.28**	5.90	**3.28**	9.18	**7.87**	9.83	10.82	**1.97**	3.93	3.28	4.59
Hepatitis	8	**17.54**	19.24	21.13	19.24	21.32	20.94	21.69	**20.19**	21.51	20.00	20.75	**17.92**
WDBC	25	3.35	**2.37**	2.42	–	6.34	**6.08**	7.16	–	**5.05**	7.42	5.88	–
Ionosphere	30	11.66	14.83	**11.50**	13.16	**9.67**	11.16	11.17	14.00	19.33	18.50	20.16	**18.16**
Dermatology	5	**22.96**	26.00	24.40	29.92	28.08	27.76	**23.28**	26.08	28.64	26.48	**22.00**	28.96
SpamBase	50	11.91	**11.34**	12.20	11.35	7.51	8.34	**7.16**	7.33	20.74	20.97	**19.68**	20.21
Sonar	25	30.42	**27.04**	32.11	29.57	**30.42**	36.62	34.92	32.11	38.3	38.02	38.59	**34.36**
Arrhythmia	20	40.12	**39.99**	40.52	44.67	40.51	**39.35**	43.64	40.77	65.71	70.51	**51.43**	51.55
Madelon	10	**38.50**	38.92	43.17	39.17	23.75	22.66	46.67	**19.33**	**37.83**	38.75	44.50	39.00
Colon	40	**13.63**	13.64	25.45	17.27	24.54	**21.81**	29.99	23.63	19.09	17.27	49.09	**13.63**
Arcene	60	37.00	34.50	–	**22.00**	**24.50**	29.50	–	26.00	37.00	37.00	–	**29.00**

Table 13
Average classification error rates (in %) over 5 runs, with respect to different number of selected features by proposed methods and supervised feature selection methods over Madelon datasets, using SVM, DT, and NB classifiers. The best result for each number of features is indicated in bold face and underlined and the second best is in bold face.

Classifiers	SVM Classifier			DT Classifier			NB Classifier		
#Selected features	10	40	150	10	40	150	10	40	150
RRFSACO_1	38.50	39.33	**41.67**	23.75	22.75	23.92	37.83	39.75	40.08
RRFSACO_2	38.92	39.55	42.92	**22.66**	**21.17**	23.08	38.75	39.17	38.58
IG	38.67	**39.17**	42.22	24.17	23.67	24.83	**38.33**	39.83	**38.66**
GR	38.67	38.94	41.50	24.17	23.83	**21.83**	**38.33**	**39.50**	39.67
GI	38.67	**39.17**	42.72	24.17	23.67	25.00	**38.33**	39.83	39.33
FS	**38.61**	40.67	42.22	24.67	25.17	28.17	39.67	40.00	41.33
SU	38.67	38.94	42.11	24.17	23.83	21.67	**38.33**	**39.50**	40.00
LS	38.67	39.33	43.17	23.50	19.67	23.17	44.50	51.33	43.50
mRMR	43.17	54.00	43.00	46.67	50.67	44.17	38.50	51.33	40.33
RRFS	39.17	–	–	19.33	–	–	39.00	–	–

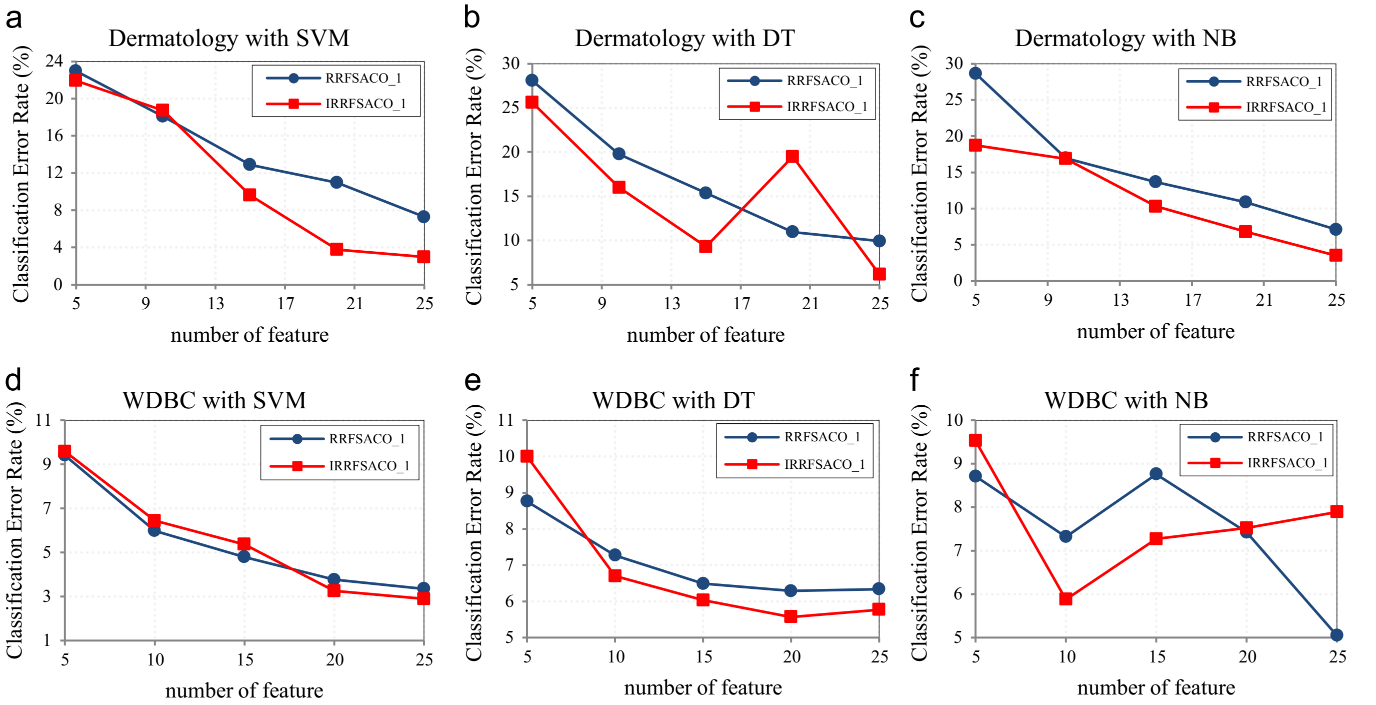

Fig. 4. Classification error rates (average over 5 different runs) of the IRRFSACO_1 and RRFSACO_1 methods using: (a) SVM classifier on Dermatology, (b) DT classifier on Dermatology, (c) NB classifier on Dermatology, (d) SVM classifier on WDBC, (e) DT classifier on WDBC, and (f) NB classifier on WDBC.

Table 13 compares the performance of the RRFSACO_1 and RRFSACO_2 methods with those of the supervised feature selection methods including IG, GR, GI, FS, SU, LS, mRMR, and RRFS on the *Madelon* dataset using the SVM, DT, and NB classifiers. Note that in this table, the notation '–' means that the feature selection method was not able to select the predefined number of features due to high similarity between features. It can be seen from the results that the RRFSACO_1 outperforms the other methods using the SVM classifier when 10 features are selected and gets the lowest classification error rate, expect than GR, when the number of selected features is 150. Moreover, the classification error rates of the RRFSACO_2 over SVM are slightly greater than those of the best methods. The proposed methods are superior to the IG, GI, FS, and mRMR methods using the DT classifier. Also, the proposed methods got the lowest classification error rates compared to the other methods, except than the LS and RRFS, when the number of selected features was 10 and 40. Furthermore, it can be seen from the results that the classification error rate of the RRFSACO_1 using the NB classifier is superior to those based on the other methods when the number of selected features is 10. Moreover, the results show that in this case, RRFSACO_2 outperforms the other feature selection methods when 40 and 150 features are selected. Therefore, it can be concluded that the overall performance of the proposed methods using the different classifiers (*i.e.*, SVM, DT, and NB) is comparable to those of the supervised feature selection methods, especially when the NB and SVM classifiers are used.

4.3.2. Comparison between IRRFSACO and RRFSACO

In this section, the results of the comparison between the RRFSACO_1, RRFSACO_2, IRRFSACO_1, and IRRFSACO_2 methods are presented. Figs. 4 and 5 demonstrate the classification error rates of the SVM, DT, and NB classifiers on the *Dermatology* and

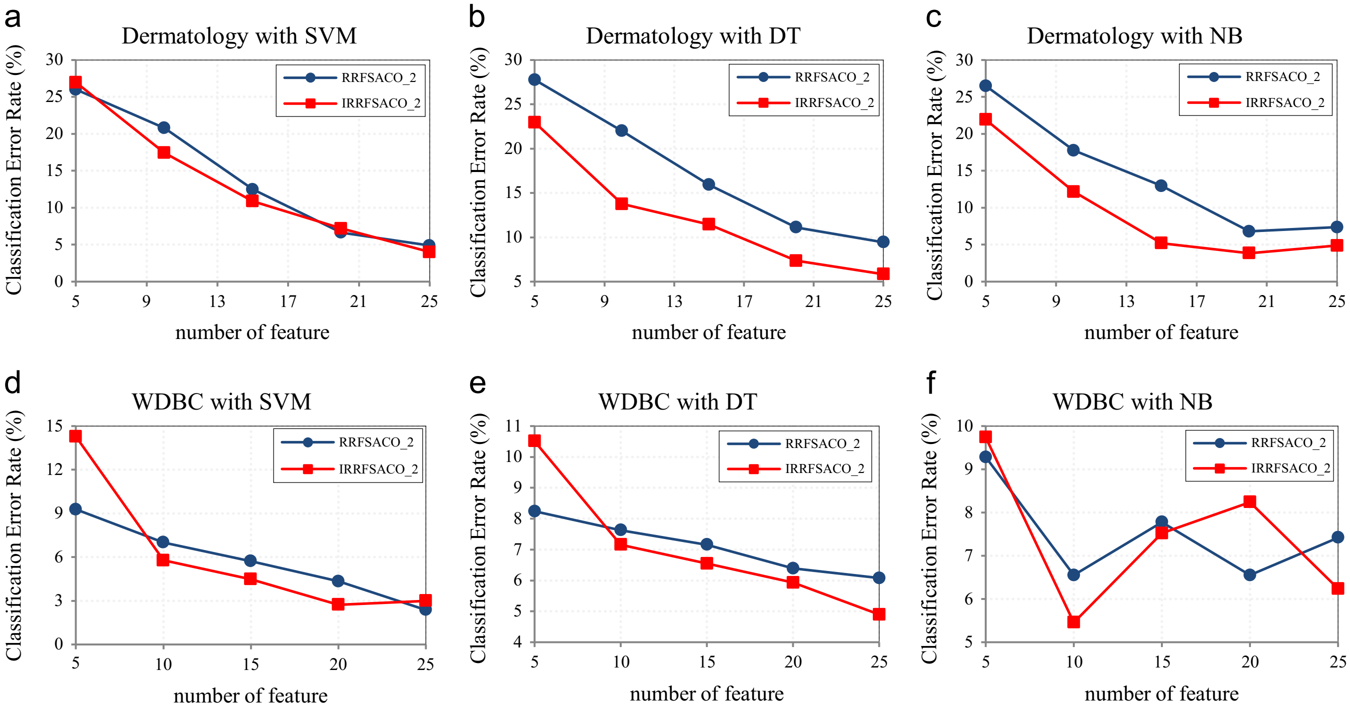

Fig. 5. Classification error rates (average over 5 different runs) of the IRRFSACO_2 and RRFSACO_2 methods using: (a) SVM classifier on Dermatology, (b) DT classifier on Dermatology, (c) NB classifier on Dermatology, (d) SVM classifier on WDBC, (e) DT classifier on WDBC, and (f) NB classifier on WDBC.

WDBC datasets. In these figures, the *x*-axis shows the number of selected features, whereas the *y*-axis denotes the average classification error rate.

Fig. 4 shows the comparison results between IRRFSACO_1 and RRFSACO_1. As seen in Fig. 4(a), IRRFSACO_1 achieved significantly lower error rates compared to the RRFSACO_1 for almost all the numbers of selected features using the SVM classifier. For example, when the number of selected features was 20, the classification error rate of IRRFSACO_1 was around 4%, while this value for the RRFSACO_1 was around 11%. Moreover, Fig. 4(b) illustrates that IRRFSACO_1 attained lower classification error rates compared to RRFSACO_1 for different numbers of features, except when the number of selected features was 20. From Fig. 4(c) it can be observed that the overall performance of the IRRFSACO_1 is superior to that of the RRFSACO_1 method for different numbers of selected features when the NB classifier is used on *Dermatology* dataset. For example, when 5 features were selected, the classification error rate of IRRFSACO_1 was around 19%, while this value for the RRFSACO_1 was around 29%. Moreover, Fig. 4(d) shows that when the number of selected features is larger than 20, the performance of IRRFSACO_1 method is better than RRFSACO_1 method. Additionally, Fig. 4(e) indicates that the classification error rate curve of IRRFSACO_1 on *WDBC* dataset is lower than RRFSACO_1 when the number of selected features is larger than 10. As illustrated in Fig. 4(f) IRRFSACO_1 got lower classification error rates when the number of features is relatively small, within the range between 10 and 20.

Furthermore, Fig. 5 shows the comparison results between IRRFSACO_2 and RRFSACO_2. Fig. 5(a) shows that the classification error rate curve of the IRRFSACO_2 was superior to that of the RRFSACO_2 when the number of selected features was 10, 15, and 25, and it attained a little greater error rate than the RRFSACO_2 (*i.e.*, less than 1%) in the other cases. Moreover, Fig. 5(b) indicates that the performance of IRRFSACO_2 is much better than that of the RRFSACO_2 for all the cases on the *Dermatology* dataset. In other words, when the number of selected features was 5, 10, 15, 20, and 25, IRRFSACO_2 got 22.96%, 13.76%, 11.44%, 7.36%, and 5.84% classification error rates, respectively, while in this case the classification error rates of the RRFSACO_2 were 27.76%, 22%, 15.92%, 11.12%, and 9.44%, respectively. It is clear from the results that when the number of features is increased, the classification error rate of the proposed methods is decreased uniformly. Moreover, the results of Fig. 5(c) show that the IRRFSACO_2 outperformed the RRFSACO_2 by 4.56%, 5.6%, 7.76%, 2.96%, and 2.48% when the number of selected features were 5, 10, 15, 20, and 25, correspondingly. Fig. 5(d) demonstrates that IRRFSACO_2 is superior to the RRFSACO_2 when 10, 15, and 20 features were selected. In addition, Fig. 5(e) shows that the performance of the IRRFSACO_2 is better than the RRFSACO_2 over different subsets of features. As illustrated in Fig. 5(f) IRRFSACO_2 gets lower classification error rates when the numbers of features were 10, 15, and 25.

It can be concluded from Figs. 4 and 5 that the proposed redundancy reduction approach used in the IRRFSACO_1 and IRRFSACO_2 methods leads to enhance the efficiency and improve the performance of the RRFSACO based methods (*i.e.*, RRFSACO_1 and RRFSACO_2) in terms of the classification error rate using the different classifiers (*i.e.*, SVM, DT, and NB) on some of the datasets.

5. Conclusion

In this paper, novel unsupervised feature selection methods were proposed based on the ACO algorithm by analyzing the relevance and redundancy of features. The proposed methods combined the efficiency of the filter model with the advantages of the ACO algorithm. Moreover, a heuristic information measure was proposed to consider the dependencies between subsets of features which enhanced the quality of the found solution.

The performances of the proposed methods were compared to those of the well-known and state-of-the-art univariate feature selection methods including information gain, gain ratio, symmetrical uncertainty, Gini index, Fisher score, term variance, and Laplacian score and multivariate feature selection methods including UFSACO, mRMR, MC, RSM, and RRFS in terms of execution time and classification error rate of the support vector machine, decision tree, and naïve

Bayes classifier. The experimental results performed on the low and high dimensional datasets indicated that the proposed RRFSACO based methods effectively removed the irrelevant and redundant features. The joint use of the filter model and ACO algorithm in the proposed methods lead to classification results superior to those of the unsupervised feature selection methods and comparable with those of the supervised feature selection methods. Moreover, the results over the *Dermatology* and *WDBC* datasets show that the heuristic information which is applied in the search process of the IRRFSACO based methods can be able to improve the classification accuracy of the RRFSACO based methods in some cases. Future work will address the development of new heuristic information measures to improve the efficiencies of the proposed methods. Also, we will develop a new state transition rule to control the randomness in the ACO algorithm.

Appendix A. Supplementary Information

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.patcog.2015.03.020.

References

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

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

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

[4] I.A. Gheyas, L.S. Smith, Feature subset selection in large dimensionality domains, Pattern Recognit. 43 (2010) 5–13.

[5] 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, Inf. Sci. 181 (2011) 4625–4641.

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

[7] 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.

[8] J. Yang, Y. Liu, Z. Liu, X. Zhu, X. Zhang, A new feature selection algorithm based on binomial hypothesis testing for spam filtering, Knowl.-Based Syst. 24 (2011) 904–914.

[9] 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.

[10] Z. Yan, C. Yuan, Ant colony optimization for feature selection in face recognition, in: David Zhang, Anil K. Jain (Eds.) Biometric Authentication, Springer, Berlin, Heidelberg, 2004, pp. 221–226.

[11] Y. Leung, Y. Hung, A Multiple-Filter-Multiple-Wrapper, Approach to gene selection and microarray data classification, IEEE/ACM Trans. Comput. Biol. Bioinform. 7 (2010) 108–117.

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

[13] S. Nemati, M.E. Basiri, Text-independent speaker verification using ant colony optimization-based selected features, Expert Syst. Appl. 38 (2011) 620–630.

[14] H. Yu, G. Gu, H. Liu, J. Shen, J. Zhao, A modified ant colony optimization algorithm for tumor marker gene selection, Genomics Proteomics Bioinform. 7 (2009) 200–208.

[15] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.

[16] C.-L. Huang, C.-Y. Tsai, A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting, Expert Syst. Appl. 36 (2009) 1529–1539.

[17] Y. Marinakis, M. Marinaki, M. Doumpos, C. Zopounidis, Ant colony and particle swarm optimization for financial classification problems, Expert Syst. Appl. 36 (2009) 10604–10611.

[18] H. Liu, H. Motoda, Computational Methods of Feature Selection, Chapman & Hall/CRC, London, 2007.

[19] H. Peng, F. 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.

[20] P.M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. Comput. 26 (1977) 917–922.

[21] L. Xu, P. Yan, T. Chang, Best first strategy for feature selection, in: Proceedings of the Ninth International Conference on Pattern Recognition, 1988, pp. 706–708.

[22] J. Martínez Sotoca, F. Pla, Supervised feature selection by clustering using conditional mutual information-based distances, Pattern Recognit. 43 (2010) 2068–2081.

[23] S. Nemati, M.E. Basiri, N. Ghasem-Aghaee, M.H. Aghdam, A novel ACO–GA hybrid algorithm for feature selection in protein function prediction, Expert Syst. Appl. 36 (2009) 12086–12094.

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

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

[26] M. Dorigo, T. Stützle, Ant colony optimization: overview and recent advances, in: Handbook of Metaheuristics, Springer, US, 2010, pp. 227–263.

[27] M. Dorigo, G. Di Caro, Ant colony optimization: a new meta-heuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999, pp. 1470–1477.

[28] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1997) 53–66.

[29] M. Dorigo, L.M. Gambardella, Ant colonies for the travelling salesman problem, Biosystems 43 (1997) 73–81.

[30] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern.-Part B: Cybern. 26 (1996) 29–41.

[31] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[32] C. Lai, M.J.T. Reinders, L. Wessels, Random subspace method for multivariate feature selection, Pattern Recognit. Lett. 27 (2006) 1067–1076.

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

[34] L. 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.

[35] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, Z. Wang, A novel feature selection algorithm for text categorization, Expert Syst. Appl. 33 (2007) 1–5.

[36] T.M. Mitchell, Machine Learning, McGraw-Hill, Inc., 1997.

[37] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.

[38] J. Biesiada, W. Duch, Feature selection for high-dimensional data: a Pearson redundancy based filter, in: Computer Recognition Systems, vol. 2, Springer, Berlin, Heidelberg, 2007, pp. 242–249.

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

[40] X. He, D. Cai, P. Niyogi, Laplacian Score for feature selection, Adv. Neural Inf. Process. Syst. 18 (2005).

[41] K. Kira, L.A. Rendell, The feature selection problem: traditional methods and a new algorithm, in: Proceedings of the Tenth National Conference on Artificial intelligence, AAAI Press, San Jose, CA, 1992, pp. 129–134.

[42] G. Wang, Q. Song, B. Xu, Y. Zhou, Selecting feature subset for high dimensional data via the propositional FOIL rules, Pattern Recognit. 46 (2013) 199–214.

[43] P. Mitra, C.A. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 301–312.

[44] M. Haindl, P. Somol, D. Ververidis, C. Kotropoulos, Feature selection based on mutual correlation, in: Pattern Recognition, Image Analysis and Applications, Springer, Berlin, Heidelberg, 2006, pp. 569–577.

[45] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.

[46] A. Unler, A. Murat, A discrete particle swarm optimization method for feature selection in binary classification problems, Eur. J. Oper. Res. 206 (2010) 528–539.

[47] R. Sikora, S. Piramuthu, Framework for efficient feature selection in genetic algorithm based data mining, Eur. J. Oper. Res. 180 (2007) 723–737.

[48] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intell. Syst. Their Appl. 13 (1998) 44–49.

[49] M.E. Farmer, S. Bapna, A.K. Jain, Large scale feature selection using modified random mutation hill climbing, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 287–290.

[50] D.B. Skalak, Prototype and feature selection by sampling and random mutation hill climbing algorithms, in: Proceedings of the 11th International Conference on Machine Learning 1994, pp. 293–301.

[51] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, Eur. J. Oper. Res. 171 (2006) 842–858.

[52] H. Chouaib, O.R. Terrades, S. Tabbone, F. Cloppet, N. Vincent, Feature selection combining genetic algorithm and Adaboost classifiers, in: 19th International Conference on Pattern Recognition, 2008, ICPR 2008, 2008, pp. 1–4.

[53] V. Sugumaran, V. Muralidharan, K.I. Ramachandran, Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing, Mech. Syst. Signal Process. 21 (2007) 930–942.

[54] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (1997) 131–163.

[55] M.E. ElAlami, A filter model for feature subset selection based on genetic algorithm, Knowl.-Based Syst. 22 (2009) 356–362.

[56] C.-K. Zhang, H. Hu, Feature selection using the hybrid of ant colony optimization and mutual information for the forecaster, in: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 2005, pp. 1728–1732.

[57] J. Huang, Y. Cai, X. Xu, A wrapper for feature selection based on mutual information, in: Proceedings of the 18th International Conference on Pattern Recognition, 2006, pp. 618–621.

[58] H.-H. Hsu, C.-W. Hsieh, M.-D. Lu, A hybrid feature selection mechanism, in: Proceedings of the Eighth International Conference on Intelligent Systems Design and Applications, 2008, pp. 271–276.

[59] A. Asuncion, D. Newman, UCI repository of machine learning datasets, Available from: ⟨http://archive.ics.uci.edu/ml/datasets.html⟩, 2007.

[60] A. Statnikov, C.F. Aliferis, I. Tsamardinos, Gems: Gene Expression Model Selector, Available from: ⟨http://www.gems-system.org/⟩, 2005.

[61] I. Guyon, NIPS feature selection challenge, Available from: ⟨http://www.nipsfsc.ecs.soton.ac.uk/datasets⟩, 2003.

[62] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten, The WEKA data mining software, Available from: ⟨http://www.cs.waikato.ac.nz/ml/weka⟩.

Sina Tabakhi received the B.S. degree with honors in information technology from Azad University, Sanandaj branch, Iran, in 2011, and the M.S. degree with honors in computer engineering from University of Kurdistan, Iran, 2013. His research interests include pattern recognition, machine learning, feature selection, and data mining.

Parham Moradi received his Ph.D. degree in computer science from Amirkabir University of Technology, Iran, in 2011. Currently he is working as an assistant professor in the Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran. His research focuses on feature selection, recommender systems, reinforcement learning and social network analysis.