Engineering Applications of Artificial Intelligence 32 (2014) 112–123

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

An unsupervised feature selection algorithm based on ant colony optimization

Sina Tabakhi, Parham Moradi *, Fardin Akhlaghian

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

ARTICLE INFO

Article history:
Received 29 August 2013
Received in revised form
2 March 2014
Accepted 11 March 2014
Available online 5 April 2014

Keywords:
Feature selection
Dimensionality reduction
Univariate technique
Multivariate technique
Filter approach
Ant colony optimization

ABSTRACT

Feature selection is a combinatorial optimization problem that selects most relevant features from an original feature set to increase the performance of classification or clustering algorithms. Most feature selection methods are supervised methods and use the class labels as a guide. On the other hand, unsupervised feature selection is a more difficult problem due to the unavailability of class labels. In this paper, we present an unsupervised feature selection method based on ant colony optimization, called UFSACO. The method seeks to find the optimal feature subset through several iterations without using any learning algorithms. Moreover, the feature relevance will be computed based on the similarity between features, which leads to the minimization of the redundancy. Therefore, it can be classified as a filter-based multivariate method. The proposed method has a low computational complexity, thus it can be applied for high dimensional datasets. We compare the performance of UFSACO to 11 well-known univariate and multivariate feature selection methods using different classifiers (support vector machine, decision tree, and naïve Bayes). The experimental results on several frequently used datasets show the efficiency and effectiveness of the UFSACO method as well as improvements over previous related methods.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

The amount of data has been growing rapidly in recent years, and data mining as a computational process involving methods at the intersection of machine learning, statistics, and databases, deals with this huge volume of data, processes and analyzes it (Liu and Yu, 2005). The purpose of data mining is to find knowledge from datasets, which is expressed in a comprehensible structure. A major problem associated with data mining applications such as pattern recognition is the "curse of dimensionality" in which the number of the features is larger than the number of patterns that leads to the large number of classifier parameters (*e.g.*, weights in a neural network). Therefore, the classifier performance may be reduced, and the computational complexity for processing the data will be significantly increased (Theodoridis and Koutroumbas, 2008). Moreover, in the presence of many irrelevant and redundant features, data mining methods tend to fit to the data which decrease its generalization. Consequently, a common way to overcome this problem is reducing dimensionality by removing irrelevant and redundant features and selecting a subset of useful features from the input feature set. Feature selection is one of the important and frequently used techniques in data preprocessing for data mining. It brings the immediate effects for applications such as speeding up a data mining algorithm and improving mining performance (Akadi et al., 2008; Ferreira and Figueiredo, 2012; Lai et al., 2006; Yu and Liu, 2003). Feature selection has been applied to many fields such as text categorization (Chen et al., 2006; Uğuz, 2011; Yang et al., 2011), face recognition (Kanan and Faez, 2008; Yan and Yuan, 2004), cancer classification (Guyon et al., 2002; Yu et al., 2009; Zibakhsh and Abadeh, 2013), finance (Huang and Tsai, 2009; Marinakis et al., 2009), and customer relationship management (Kuri-Morales and Rodríguez-Erazo, 2009).

Feature selection is a process of selecting a subset of features from a larger set of features, which leads to the reduction of the dimensionality of feature space for a successful classification task. The whole search space contains all possible subsets of features, meaning that its size is 2^n, where n is the number of features. Therefore, many problems related to feature selection are shown to be NP-hard. Consequently, finding the optimal feature subset is usually intractable in a reasonable time (Liu and Motoda, 2007; Meiri and Zahavi, 2006; Narendra and Fukunaga, 1977; Peng et al., 2005). To overcome the time complexity problem, there have been proposed approximation algorithms to find a near-optimal feature subset in polynomial time. These algorithms can be classified into four categories including filter, wrapper, embedded, and hybrid approaches (Gheyas and Smith, 2010; Liu and Motoda, 2007; Liu

* Corresponding author. Tel.: +98 8716668513.
E-mail addresses: sina.tabakhi@ieee.org (S. Tabakhi), p.moradi@uok.ac.ir (P. Moradi), f.akhlaghian@uok.ac.ir (F. Akhlaghian).

http://dx.doi.org/10.1016/j.engappai.2014.03.007
0952-1976/© 2014 Elsevier Ltd. All rights reserved.

and Yu, 2005; Martínez Sotoca and Pla, 2010; Saeys et al., 2007). The filter approach relies on statistical properties of the data to decide which features are relevant. This approach reduces the dimensionality of datasets independent of the learning algorithm. The wrapper approach utilizes a given learning algorithm to evaluate the candidate feature subsets. Hence, the feature selection process is wrapped around the learning algorithm. The embedded approach seeks to subsume feature selection as part of the model building process. Therefore, this approach is associated with a specific learning algorithm. On the other hand, the goal of the hybrid approach is to take advantage of both the filter and the wrapper approaches.

All of the feature selection approaches can be applied in the two supervised and unsupervised modes (Guyon and Elisseeff, 2003; He et al., 2005; Liu and Motoda, 2007). In the supervised mode, training patterns are described by a vector of feature values with a class label. The class labels are used to guide the search process for relevant information. On the other hand, the unsupervised mode faces with patterns without class labels. Consequently, feature selection in unsupervised learning is a more difficult issue, due to the unavailability of class labels.

Many approximation methods have been proposed for feature selection based on computational intelligence methods. Due to their acceptable performances, they have attracted a lot of attention. Since most of these methods use the learning algorithm in their processes, they are classified as varieties of the wrapper approach. 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. Swarm intelligence-based methods can be used to find an approximate solution to a hard combinatorial optimization problem. One of the most popular algorithms in the research field of swarm intelligence is ant colony optimization (ACO). ACO uses knowledge from previous iterations to achieve better solutions. Moreover, it can be easily implemented; thus, it is widely used in the feature selection area (Aghdam et al., 2009; Chen et al., 2013; Kanan and Faez, 2008; Li et al., 2013; Nemati and Basiri, 2011; Xiao and Li, 2011).

Since the wrapper approach uses learning algorithms to evaluate the selected feature subsets, it requires a high computational cost for high-dimensional datasets. On the other hand, the filter approach is independent of the learning algorithm, and it is computationally more efficient than the wrapper approach. But, the filter approach has several shortcomings: (1) there are features which do not appear relevant singly, while they will be highly relevant if combined with the other features. Accordingly, considering correlation between features can lead to the improvement of the classification performance (Gheyas and Smith, 2010; Gu et al., 2011; Lai et al., 2006; Leung and Hung, 2010); (2) there are features which may have high-rank values singly, while they are highly correlated with each other and removing one of them can lead to an increase in the classification performance. According to this fact, many studies have shown that removing redundant features can improve the classifier performance (Biesiada and Duch, 2007; Gu et al., 2011; Martínez Sotoca and Pla, 2010; Peng et al., 2005); (3) the filter approach result is based on only one iteration process, and the optimal features set may be obtained in an incremental way, so these methods often converge on a local optimum solution. On the other hand, in some of the incrementally-based filter approaches (Ferreira and Figueiredo, 2012; Peng et al., 2005), the first feature will be selected, based on a specific criterion, and then the next features will be sequentially selected based on the previous selected features.

Providing a good search method for finding a near-optimal feature subset in a huge search space is an important issue in the feature selection process. Moreover, computational time and quality of the selected feature subset are two main issues of the search methods. These issues are in conflict with each other and generally 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 feature selection methods dependent on the learning algorithm (*i.e.*, wrapper, hybrid, and embedded approaches) 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.

In this paper, we will propose a new feature selection method based on ant colony optimization, which aims to achieve a high-quality approximate solution within an acceptable computational time. This method seeks to find the optimal feature subset in an iterative improvement process. In other words, the proposed method consists of a set of agents, called ants, which cooperate with each other through sharing pheromone. Each agent selects a number of features iteratively using heuristic and previous stages information. Due to the iterative and parallel nature of the method, it searches a greater space and selects a feature subset that will be close to the global optimal solution. The proposed method does not need any learning algorithms and class labels to select feature subsets; therefore it can be classified as a filter-based and unsupervised approach. Consequently, it is computationally efficient for high-dimensional datasets. Moreover, each feature is associated with a selection probability, so, any feature can be selected in different steps of the method. Furthermore, the similarity between features will be considered in computation of feature relevance, which leads to the minimization of the redundancy between features.

The rest of this paper is organized as follows. Section 2 presents a brief overview of the previous feature selection algorithms. In Section 3, ant colony optimization is briefly reviewed. Section 4 presents our method, called UFSACO, for feature selection. In Section 5 we compare the experimental results of the proposed method to those of several multivariate and univariate feature selection techniques. Finally, Section 6 presents the conclusion.

2. Background and overview of feature selection methods

In this section, we briefly review filter, wrapper, embedded and hybrid feature selection approaches.

2.1. Filter approach

The filter approach evaluates the relevance of features without using learning algorithms. In this approach, intrinsic characteristics of the data are used to evaluate and rank features; then the features with highest rank values will be selected (Unler et al., 2011).

The filter-based feature selection methods can be classified into univariate and multivariate methods (Lai et al., 2006; Saeys et al., 2007; Trevino and Falciani, 2006). In univariate methods, the relevance of a feature is measured individually using an evaluation criterion. There are numerous well-known univariate filter methods, such as information gain (Raileanu and Stoffel, 2004; Yu and Liu, 2003), gain ratio (Mitchell, 1997; Quinlan, 1986), symmetrical uncertainty (Biesiada and Duch, 2007; Yu and Liu, 2003), Gini index (Raileanu and Stoffel, 2004; Shang et al., 2007), Fisher score (Gu et al., 2011; Theodoridis and Koutroumbas, 2008), term variance (Theodoridis and Koutroumbas, 2008), and Laplacian score (He et al., 2005). In univariate methods, the possible dependency between features will be ignored in the feature selection process, while in the

multivariate methods, the dependencies between features will be considered in the evaluation of the relevance of features. Thus, the multivariate methods can be computationally more expensive than the univariate methods, while their performance is better than the univariate methods. There are several multivariate feature selection methods in the literature including minimal-redundancy–maximal-relevance (mRMR) (Peng et al., 2005), mutual correlation (Haindl et al., 2006), random subspace method (RSM) (Lai et al., 2006), and relevance-redundancy feature selection (RRFS) (Ferreira and Figueiredo, 2012). The details of some of the well-known univariate and multivariate methods which are used in the paper's experiments have been described in the following.

Information gain of a feature is usually interpreted as the amount of information provided by the feature to the classification system. More precisely, the information gain of a feature A, relative to a set of patterns S, is defined as

$$IG(S|A) \equiv E(S) - \sum_{v \in Values(A)} P_v\, E(S_v) \tag{1}$$

where $Values(A)$ is the set of all possible values which are taken by the feature A, P_v shows the probability of the patterns S belonging to the value v with respect to the feature A, S_v is a subset of patterns S for which feature A has value v, and $E(S)$ measures the entropy of the set of patterns S.

The entropy of a variable X is defined as follows:

$$E(X) = \sum_{v \in Values(X)} -P_v \log_2(P_v) \tag{2}$$

where P_v is the probability of the patterns S belonging to the value v with respect to the variable X.

Information gain tends to favor features with very large numbers of possible values, but these features have a poor predictive power over unseen patterns. Thus, *gain ratio* and *symmetrical uncertainty* have been introduced to address the problem.

Gain ratio of a feature indicates the broadly and uniformly dividing of the patterns by the feature and is given by

$$\text{Gain ratio}(S,A) \equiv \frac{IG(S|A)}{E(A)} \tag{3}$$

where $IG(S|A)$ shows the information gain of the feature A, and $E(A)$ measures the entropy of the feature A.

Symmetrical uncertainty discourages the selection of features with more values and limits its values in the range of [0,1]. It is defined as follows:

$$SU(S,A) \equiv 2\left[\frac{IG(S|A)}{E(S)+E(A)}\right] \tag{4}$$

where $E(S)$ is the entropy of the set of patterns S. $SU=1$ means that the collection of patterns S and a given feature A are completely correlated, and $SU=0$ indicates that S and A are independent.

Gini index is an impurity split method, and it is suitable for continuous numeric values. The Gini Index of the collection of patterns S is defined as

$$\text{Gini index}(S,A) \equiv Gini(S) - \sum_{v \in Values(A)} P_v Gini(S_v) \tag{5}$$

where

$$Gini(S) = 1 - \sum_{v \in Values(S)} (P_v)^2.$$

Fisher score selects features such that the patterns from the same class are close to each other and the patterns from different classes are far from each other. The Fisher score of the feature A is calculated as

$$F(S,A) = \frac{\sum_{v \in Values(S)} n_v(\overline{A_v} - \overline{A})^2}{\sum_{v \in Values(S)} n_v(\sigma_v(A))^2} \tag{6}$$

where $\overline{A}$ is the mean of all the patterns corresponding to the feature A, n_v is the number of patterns which have class label v, and $\sigma_v(A)$ and $\overline{A_v}$ are the variance and mean of feature A on class v, respectively.

Term variance is the simplest univariate evaluation of the features and indicates that the features with larger values of variance contain valuable information. It is defined as follows:

$$TV(S,A) = \frac{1}{|S|}\sum_{j=1}^{|S|} (A(j) - \overline{A})^2 \tag{7}$$

where $A(j)$ indicates the value of feature A for the pattern j, and $|S|$ is the total number of the patterns.

Laplacian score of a feature evaluates locality preserving power of the feature. It is assumed that if the distances between two patterns are as small as possible, they are related to the same subject. Unlike the many feature selection methods, Laplacian score uses the local structure of data space instead of the global structure.

Minimal-redundancy–maximal-relevance (mRMR) is a solid multivariate filter approach which seeks to select features with the largest dependency on the target class using a specific relevance criterion. Furthermore, it uses a given redundancy criterion to reduce the redundancy between features.

Mutual correlation is a multivariate feature selection method that computes the dependency between two features. In this method, in each iteration, the feature with the highest average correlation value is removed, and the remaining features will be considered as the final subset of features.

Random subspace method (RSM) has been proposed to reduce the computational complexity in determining the relevance features for multivariate methods. This method applies a specific multivariate method to a randomly selected subspace of the original feature space. In order to evaluate a large portion of the features, the selection process is repeated several times, and finally the results are combined as the finally selected features.

Relevance-redundancy feature selection (RRFS) is an efficient feature selection technique based on relevance and redundancy analyses, which can work in both supervised and unsupervised modes. In this method, the first feature will be selected based on a given criterion, and then in each iteration a feature will be selected if its similarity to the last selected feature is smaller than a predefined threshold.

2.2. Wrapper approaches

The wrapper approach uses a learning algorithm for evaluation of a given subset of selected features (Unler et al., 2011). This approach must employ a search method to find an optimal subset of features. Wrapper approaches broadly fall into two categories based on the search strategy: sequential search and random search (Gheyas and Smith, 2010; Liu and Yu, 2005; Saeys et al., 2007).

Sequential search methods add or remove features sequentially, but they have a tendency to become trapped in a local optimum. Examples are sequential backward selection, sequential forward selection, and the floating search method (Theodoridis and Koutroumbas, 2008). On the other hand, random search methods seek to embed randomness into their search procedures to escape local optimum solutions. Examples of these methods include random mutation hill-climbing (Farmer et al., 2004; Skalak, 1994), simulated annealing (Meiri and Zahavi, 2006), genetic algorithm (Sikora and Piramuthu, 2007; Yang and Honavar, 1998), and ant colony optimization (Aghdam et al., 2009).

The wrapper approach often provides better performance for a specific classifier in terms of the classification accuracy, but it has less generalization of the selected features on the other classifiers.

2.3. Embedded approaches

In the embedded approach, the feature selection process can be considered as part of the model construction process, which means that the search for a good subset of features has been performed by a learning algorithm (Saeys et al., 2007).

Support vector machine (SVM), decision tree (DT), and naïve Bayes (NB) are well-known learning algorithms to construct a model in the embedded approach. DT is a classifier in the form of a tree structure which consists of a number of nodes and a number of leaves. Each node represents a selected feature, and each leave shows a class label. The position of a feature in the DT indicates the importance of the feature. Consequently, in the DT-based embedded methods, first of all, the tree will be constructed using a collection of patterns, and then the features which are involved in the classification are selected as a final feature subset (Sugumaran et al., 2007; Tahir et al., 2006). In the SVM-based embedded approach, first of all, a SVM-based classifier is trained using the whole feature set, and then, features with the highest weights are selected as a feature subset (Guyon et al., 2002). On the other hand, NB is an effective classifier which learns from the dataset the conditional probability of each feature given the class label. All features in the NB are conditionally independent of the rest of the features given the class label. The NB-based embedded approach learns a probability distribution for each feature using a given scoring function. Therefore, this concept is used to select features with high probability (Friedman et al., 1997). Moreover, the computational complexity of the embedded approach tends to be between those of the filter and wrapper approaches.

2.4. Hybrid approaches

Hybrid approaches seek to select features in two stages: in the first stage, they seek to reduce the original feature set using the filter approach. Then in the second stage, the wrapper approach is applied to select the best subset of features on the reduced feature set. In other words, the aim of the hybrid approach is to use the advantages of both filter and wrapper approaches (Unler et al., 2011). Consequently, the risk of eliminating good features in the hybrid approach is less than that in the filter approach.

Examples of hybrid approaches include ant colony optimization with mutual information (Zhang and Hu, 2005), ant colony optimization and Chi-square statistics with support vector machine (Mesleh and Kanaan, 2008), mutual information and genetic algorithm (Huang et al., 2006), and feature selection based on mutual information and particle swarm optimization with support vector machine (Unler et al., 2011).

3. Ant colony optimization

In this section, we briefly review the ant colony optimization (ACO) algorithm. In the early 1990s, ACO was presented by Dorigo et al. (Dorigo and Gambardella, 1997b) for solving hard combinatorial optimization problems. It is inspired by social behavior of ants while seeking for food. Each ant performs a simple task, but finally a colony's cooperative work can provide models for solving hard combinatorial optimization problems. To communicate with the others, each ant deposits a chemical substance, called pheromone, on the ground where they walk. This substance evaporates over time that decreases the intensity of the pheromone. This process is used to avoid being trapped in a local minimum, to explore new regions of the search space, and to decrease the probability of selecting longer paths. When choosing between two paths, ants prefer in probability to choose a path with more pheromone (higher probability); in other words, more ants pass it on average. In addition, the ants that select the shorter paths to get the food return to the nest faster than other ants. Therefore, shorter paths get a higher amount of pheromone than longer paths. Over time, all ants will be using the shorter paths to find the food. As a result, “evaporation of pheromone” and “probabilistic selection of paths”, allow ants to find the shortest path, and they lead to flexibility for solving combinatorial optimization problems.

The main strengths of the ACO can be expressed as follows (Dorigo and Di Caro, 1999; Dorigo and Gambardella, 1997a, 1997b; Dorigo et al., 1996; Dorigo and Stützle, 2010; Gutjahr, 2007):

- **ACO is a population-based approach**. Unlike traditional optimization methods that start to search from a given point, the ACO starts the search process using a population of the ants, and then a large part of the search space will be simultaneously investigated by the ants. Consequently, the quality of the found solution could be greatly improved, especially for high-dimensional problems.
- **ACO can be classified as a multi-agent system**. This is interesting because the ants cooperate with each other by sharing their knowledge through pheromone trail to solve the problem efficiently.
- **ACO can be implemented in a parallel way**. This is due to the distributed problem solving nature of the ACO and could greatly decrease the computational time.
- **ACO can be interpreted as a reinforcement learning system**. In fact, in the ACO, better solutions get a higher reinforcement. Therefore, the ants will find the better solutions with high probability in the next iterations.
- **ACO uses a distributed long-term memory**. This memory is used to store the knowledge obtained from the ants' previous searches. This leads to a simultaneous exchange of information between the ants. Therefore, each ant can use the information of the other ants to choose the better solution.
- **ACO has good global and local search capabilities**. The stochastic component of the ACO enables an efficient exploration of the search space and hence avoids being trapped in a local minimum, while the greedy component of the ACO has the strong local search ability.

4. Proposed method

In this section, we present a novel unsupervised feature selection method based on ant colony optimization (UFSACO). Furthermore, the redundancy between selected features and the computational complexity of the proposed method have been analyzed in the subsequent sections.

4.1. Unsupervised feature selection based on ant colony optimization

In the proposed method, before the feature selection process starts, the search space must be represented as a suitable form for ACO. Therefore, first of all, the search space is represented as a fully connected undirected weighted graph $G=\langle F, E\rangle$, where $F=\{F_1, F_2,\ldots, F_n\}$ denotes a set of original features in which each feature represents a node in the graph, and $E=\{(F_i, F_j): F_i, F_j \in F\}$ denote the graph edges. The weight of the edge $(F_i, F_j) \in E$ will be set to the similarity value between F_i and F_j. Fig. 1 shows the representation of the feature selection problem.

To compute the similarity between features, we use the absolute value of the cosine similarity between them (Huang, 2008). The cosine similarity between features A and B is calculated

according to the following equation:

$$\mathrm{sim}(A,B)=\left|\frac{\sum_{i=1}^{p}(a_i b_i)}{\left(\sqrt{\sum_{i=1}^{p}a_i^2}\right)\left(\sqrt{\sum_{i=1}^{p}b_i^2}\right)}\right| \tag{8}$$

where A and B show two features with p-dimensional (*i.e.*, p pattern) vectors ($A=\{a_1, a_2,\ldots, a_p\}$ and $B=\{b_1, b_2,\ldots, b_p\}$). According to the equation, it can be seen that the similarity value is between 0 and 1. Moreover, the similarity value of two features which are completely similar will be equal to 1, and this value will be equal to 0 for completely non-similar features. It should be noted that the measure probably works for numerical features.

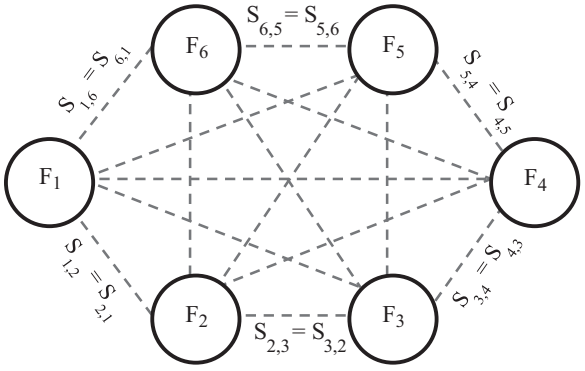

Fig. 1. The representation of the feature selection problem as a graph (The purpose of $S_{i,j}$ is a similarity associated with the edge between features i and j; in other words, $S_{i,j}=sim(F_i, F_j)$).

Furthermore, to use the ACO algorithm in the feature selection problem, "*heuristic information*" and "*desirability*" must be defined as the basic ingredients of any ACO algorithm. In the proposed method, the heuristic information is simply defined as the inverse of the similarity between features. Furthermore, we define a desirability measure τ_i, $\forall i=1\ldots n$, called "*pheromone*", which is associated with the features (*i.e.*, the graph nodes) and will be updated by ants gradually.

The pseudo code of the proposed feature selection method is shown in Fig. 2. The proposed method is composed of several iterations. Before the iterations start, the amount of pheromone assigned to each node is initialized to a constant value c. Then, in each iteration, *NAnt* ants are placed randomly on the different nodes. Continuously, each ant traverses the graph nodes according to a probabilistic "*state transition rule*" in an iterative way until an iteration stopping criterion is satisfied. Stopping criterion is defined as the number of nodes that must be chosen by each ant. The state transition rule seeks to select features with highest pheromone values and lowest similarities to previous selected features. The number of times that a given feature is selected by any ant will be kept in the "*feature counter*" (*FC*) array. Then, at the end of the iteration, the amount of pheromone for each node is updated by applying a "*global updating rule*". The amount of pheromone for each node is computed based on its feature counter value. In fact, the ants tend to give more pheromone to nodes with greater feature counter values. Moreover, a fraction of the pheromone evaporates on all nodes. The process is repeated until a given number of iterations are reached. Then, the features are sorted based on their pheromone values in decreasing order. Finally, the top m features with highest pheromone values are

Algorithm 1. *Unsupervised Feature Selection based on Ant Colony Optimization (UFSACO)*

Input: *X*: $p \times n$ matrix, n dimensional training set with p patterns.
m ($\leq n$): the number of features to keep for final reduced feature set.
NC_{max}: the maximum number of cycles that algorithm repeated.
NAnt: define the number of agents (number of ants).
NF: the number of features selected by each agent in each cycle.
ρ: define the decay rate of the pheromone on each feature.
@sim: function that computes the similarity between features.
Output: $\tilde{X}$: $p \times m$ matrix, reduced dimensional training set.

begin algorithm
Apply @sim to compute the similarity $S_{i,j}$ between features, $\forall$ $i, j= 1...n$.
$\tau_i(1) = c$, $\forall$ $i = 1...n$. /* initial pheromone – c is a constant parameter*/
for $t = 1$ to NC_{max} **do**
  $FC[i] = 0$, $\forall$ $i = 1...n$. /* set the initial features counter to zero */
  Place the agents randomly on the graph nodes.
  for i = 1 to *NF* **do**
    for $k = 1$ to *NAnt* **do**
      Choose the next unvisited feature f according to (9) and (10). /* pseudo-random-proportional rule */
      Move the k-th agent to the new selected feature f.
      $FC[f] = FC[f] + 1$; /* update feature counter associated with feature f*/
    end for
  end for
  $\tau_i(t+1) = (1-\rho)\,\tau_i(t) + \frac{FC[i]}{\sum_{j=1}^{n} FC[j]}$; $\forall$ $i = 1...n$. /* global updating rule */
end for
Sort the features by decreasing order of their pheromones (τ_i).
Build $\tilde{X}$ from X by keeping the top m features with highest pheromone.
end algorithm

Fig. 2. Pseudo code of the proposed feature selection method.

selected as the final feature subset. Note that both heuristic and pheromone information are used by the ants during their traverses to guide the search process.

The "*state transition rule*" is designed based on a combination of the heuristic information and the node pheromone values as follows:

when the ant k is located on the feature i, the next feature j can be selected in a greedy way or in a probabilistic way. In the greedy way, the next feature is selected according to the following equation:

$$j = \underset{u \in J_i^k}{\arg\max} \{[\tau_u]\,[\eta(F_i, F_u)]^{\beta}\}, \quad \text{if } q \le q_0 \tag{9}$$

where J_i^k is the unvisited feature set, τ_u is the pheromone assigned to the feature u, $\eta(F_i, F_u) = 1/\mathrm{sim}(F_i, F_u)$ is the inverse of the similarity between features i and u, β is a parameter which is used to control the importance of pheromone *versus* similarity ($\beta > 0$), $q_0 \in [0,1]$ is a constant parameter, and q is a random number in the interval [0,1].

In the probabilistic way, the next feature j will be selected based on the probability $P_k(i, j)$, which is defined as follows:

$$P_k(i,j) = \begin{cases} \dfrac{[\tau_j]\,[\eta(F_i,F_j)]^{\beta}}{\sum_{u \in J_i^k}[\tau_u]\,[\eta(F_i,F_u)]^{\beta}}, & \text{if } j \in J_i^k \quad \text{if } q > q_0 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

State transition rule depends on the parameters q and q_0, which is a trade between *Exploitation* and *Exploration*. If $q \le q_0$, then ants select the best feature in the greedy way (*Exploitation*); otherwise, each feature has a chance of being selected corresponding to its probability value which is computed using (10) (*Exploration*). The aim of the probabilistic way is to avoid being trapped into a local optimum. The combination of both the probabilistic and the greedy ways is the so called "*pseudo-random-proportional rule*".

The "*global updating rule*" is applied to all nodes at the end of each ant's traverse using the following equation:

$$\tau_i(t+1) = (1-\rho)\,\tau_i(t) + \frac{FC[i]}{\sum_{j=1}^{n} FC[j]} \tag{11}$$

where n is the number of original features, $\tau_i(t)$ and $\tau_i(t+1)$ are the amounts of pheromone values of feature i at times t and $t+1$, respectively, ρ is a pheromone evaporation parameter, and $FC[i]$ is the counter corresponding to feature i.

4.2. Justification

In feature selection problems, it is shown that the m best individual features do not generally make the best m features, which is mostly due to the redundancy between features (Martínez Sotoca and Pla, 2010; Peng et al., 2005; Sikora and Piramuthu, 2007). Therefore, researchers have concluded that the redundancy between features must be reduced. Consequently the main goal of the proposed method is to select a subset of features with minimum redundancy between them. To this end, in the proposed method each ant selects the next feature with the lowest similarity to the previous selected feature. Therefore, if a feature is selected by most of the ants, this indicates that the feature has the lowest similarity to the other features. Moreover, the feature receives the greatest amount of pheromone, and the chances of its selection by the other ants will be increased in the next iterations. Finally, the top m selected features have high pheromone values which are obtained using the similarity between features. Thus, we expect that the proposed method selects the best features with minimum redundancy.

4.3. Computational complexity analysis

The proposed method consists of three parts: similarity computation part, probability computation part, and selection part. In the similarity computation part (line 2), the method computes the similarity values between each pair of features. Since the calculation of the similarity values for a pair of features is dependent on the number of patterns p in a dataset, the overall computational complexity of this part will be $O(pn^2)$. In the second part (lines 3–15), the method computes the probability for each feature in an iterative way in which the number of iterations is defined by the maximum number of cycles parameter (NC_{max}). In each iteration, the ants select the next feature according to the *pseudo-random-proportional rule* (Eqs. (9) and (10)). Moreover, the computational complexity of this part is $O(NC_{max}NFNAntn)$. In addition if the ants run in a parallel way, the computational complexity will be reduced to $O(NC_{max}NFn)$. In the selection part of the method (lines 16–17) the features will be sorted based on their probability values, and then the top m high value features will be selected. Therefore, the computational complexity of this part is $O(n\log n)$. Consequently, the final complexity of the method will be $O(pn^2 + NC_{max}NFn + n\log n) = O(pn^2 + NC_{max}NFn)$. When the number of features which are selected by each ant is much smaller than the number of original features ($NF \ll n$), the computational complexity can be reduced to $O(pn^2)$. To reduce the computational complexity of the method for high-dimensional datasets, the similarity between features can be computed when the ant selects the next feature, and the first part of the method can be removed. Thus, the computational complexity can be reduced to $O(2NC_{max}NFpn + n\log n) = O(NC_{max}NFpn)$.

From the complexity analysis of the proposed method, it can be concluded that it is suggested that the similarity values be computed before the feature selection process when the number of features n is small (*i.e.*, $NC_{max}NF > n$); otherwise, these values should be calculated during the feature selection process.

The proposed method does not need any learning algorithms to select feature subsets. Therefore, it is clear that the computational complexity of the proposed method is much lower than those of the wrapper-based methods. On the other hand, due to the iterative nature of the proposed method, its computational complexity is a little bit more expensive than those of the filter-based methods.

5. Experimental results

In this section, the performance of the proposed method has been compared to that of the well-known supervised filter approaches including information gain (IG), gain ratio (GR), symmetrical uncertainty (SU), Gini index (GI), Fisher score (FS), and minimal-redundancy–maximal-relevance (mRMR) and unsupervised filter approaches including term variance (TV), mutual correlation (MC), and random subspace method (RSM). Laplacian score (LS) and relevance-redundancy feature selection (RRFS) methods can be applied in both supervised and unsupervised modes. These methods have been described in Section 2. The comparison experiments have been applied on the several frequently used datasets which have been chosen from the UCI repository (Asuncion and Newman, 2007) and NIPS2003 feature selection challenge (Guyon, 2003). Furthermore, the selected datasets, the parameter settings, the classifiers for evaluation, and the numerical results will be described in the following sections.

5.1. Datasets

We have used several datasets with different properties to assess the proposed method. These datasets include *Wine*, *Hepatitis*,

Table 1
Characteristics of the datasets used in the experiments.

Dataset	Features	Classes	Patterns
Wine	13	3	178
Hepatitis	19	2	155
WDBC	30	2	569
Ionosphere	34	2	351
Dermatology	34	6	366
SpamBase	57	2	4601
Arrhythmia	279	16	452
Madelon	500	2	4400
Arcene	10,000	2	900

Wisconsin Diagnostic Breast Cancer (WDBC), *Ionosphere*, *Dermatology*, *SpamBase* and *Arrhythmia* from UCI repository and *Madelon* and *Arcene* from NIPS2003 feature selection challenge, which have been extensively used in the literature (Ferreira and Figueiredo, 2012; Gheyas and Smith, 2010; Martínez Sotoca and Pla, 2010; Unler et al., 2011). The characteristics of the datasets is shown in Table 1.

The *Wine* dataset has the result of a chemical analysis of wines grown in the same region in Italy. The target class has three states of different cultivars. The *Hepatitis* dataset has two different classes: "Die" and "Live". Moreover, the dataset has several missing values in the features. *WDBC* is a two-class dataset in the medical domain. The dataset features have been computed from a digitized image of a fine needle aspirate of a breast mass. Moreover, the *Ionosphere* dataset contains radar data which have been collected by a system in Goose Bay, Labrador. The *Dermatology* dataset includes the six types of Eryhemato-Squamous diseases. This dataset contains several missing values. The *SpamBase* dataset is a binary classification problem that distinguishes spam from non-spam emails. The aim of the *Arrhythmia* dataset is to classify the presence and absence of cardiac arrhythmia into the 16 different groups. This dataset contains 279 features, 206 of which are linear-valued and the rest are nominal. This dataset also has several missing values in the features.

Finally, the *Madelon* and *Arcene* datasets are binary classification problems. The class labels for the test sets are not available, thus, we use the results of the classifiers' accuracy on the validation sets.

In the experiments, to deal with the missing values in the datasets, the missing values have been provided using the mean of the available data of the respective feature (Theodoridis and Koutroumbas, 2008).

5.2. Parameter settings

The proposed method includes a different number of adjustable parameters. The maximum number of cycles has been set to 50 ($NC_{max}=50$), the initial amount of pheromone for each feature is set to 0.2 ($\tau_i=0.2$), pheromone evaporation coefficient is set to 0.2 ($\rho=0.2$), the exploration/exploitation control parameter is set to 0.7 ($q_0=0.7$), parameter β is set to 1 ($\beta=1$), and finally the number of ants for each dataset is set to the number of its original features ($NAnt=\#features$). But, for the datasets with more than 100 features this parameter is set to 100 ($NAnt=100$).

For the RRFS method we apply several threshold values in the range [0.5, 1), and the number of selections for RSM is set to 50 times.

5.3. Classifiers for evaluation

The feature subset which is obtained using the proposed method is independent of the classifiers. Therefore, we expect the proposed method can improve the accuracy of different classifiers. To this end, in the experiments three different classifiers including *Support Vector Machine* (*SVM*), *Decision Tree* (*DT*), and *Naïve Bayes* (*NB*) have been applied to evaluate the feature selection methods.

SVM (Guyon et al., 2002) is a general learning machine for the classification problem, that was introduced by Vapnik. The goal of SVM is to search for the best hyperplanes that give the maximum possible margin between the datasets.

DT (Quinlan, 1986) is a popular tool for classification. The tree is constructed by training data, and each path from the root to a leaf represents a rule, which provides a classification of the pattern.

NB (Theodoridis and Koutroumbas, 2008) is a very practical learning method. It is based on the simplifying assumption that features are conditionally independent of each other given the target class.

In this work, SMO, J48 (implementation of C4.5 algorithm), and naïve Bayes as WEKA software package (Hall et al.) implementation of SVM, DT, and NB are used, respectively. The kernel used in the SVM is a polykernel. In the multiclass classification, the one-against-rest strategy is adopted. In the pruning phase of DT classifier, the post-pruning method is used. The confidence factor, which is set to 0.25, is used for pruning the tree and the minimum number of instances per leaf is set to 2 in the experiments.

5.4. Results and discussion

All the methods are implemented using Java on an Intel Core-i3 CPU with 4GB of RAM. In the experiments, the classification error rate is used as the performance measure. In all the plots, the *x*-axis denotes the subset of selected features, while the *y*-axis is the average classification error rate over 5 independent runs. In each run, first of all, the datasets were randomly split into a training set (2/3 of the dataset) and a test set and then all methods will be performed on the same train/test partitions.

In the experiments, first of all the performance of the proposed method is evaluated over different classifiers. Tables 2–4 summarize the average classification error rates (in %) together with the average execution times (in s) over 5 independent runs of the UFSACO and unsupervised methods (*i.e.*, RSM, MC, RRFS, TV, and LS) using SVM, DT, and NB classifiers, respectively. It should be noted that the feature selection process in the filter approach is independent of the classifier. Thus, we have reported only the execution time of the feature selection process.

It can be seen from Table 2 results that in most cases the UFSACO obtains the lowest error rate compared to the other methods and it acquires the second lowest error rate only inferior to that of the RRFS method for *SpamBase* and *Arcene* datasets. For example, for *Wine* dataset, UFSACO obtained a 4.92% classification error rate while for RSM, MC, RRFS, TV, and LS this value was reported 18.03%, 10.38%, 6.01%, 10.93%, and 8.74%, correspondingly. Consequently, the average values over all the datasets which have been reported in the last row of Table 2 show that UFSACO with a classification error rate of 19.84% outperforms the other methods in terms of the quality of the found solution. But the execution time of the proposed method is worse than those of the other unsupervised methods, except the LS method.

Table 3 shows that the UFSACO attains the lowest error rate when *Wine*, *Ionosphere*, *Dermatology*, *SpamBase*, and *Arrhythmia* datasets are used. On the other hand, the proposed method acquired the second lowest error rate compared to the other unsupervised methods when applied on *WDBC*, *Madelon*, and *Arcene* datasets. However, the average values over all the datasets show that the overall performance of UFSACO is significantly better than those of the other methods.

Table 4 reported similar results over the NB classifier. For example, the results show that the UFSACO outperforms the other

Table 2
Average classification error rate (in %) together with the average execution time (in s) over 5 runs of the unsupervised feature selection methods considered over different datasets, using SVM classifier. The best result for each dataset is shown in bold face and underlined and the second best is in bold face.

Datasets	# Selected features	UFSACO		RSM		MC		RRFS		TV		LS	
		Error	Time	Error	Time	Error	Time	Error	Time	Error	Time	Error	Time
Wine	5	**4.92**	0.025	18.03	0.007	10.38	0.002	**6.01**	0.001	10.93	0.001	8.74	0.023
Hepatitis	5	**16.85**	0.043	19.06	0.013	**17.27**	0.005	20.56	0.001	20.94	0.001	22.26	0.021
WDBC	5	**9.28**	0.019	16.18	0.008	11.03	0.002	**9.64**	0.001	10.1	0.001	10.2	0.352
Ionosphere	30	**11.39**	0.078	**12.16**	0.005	14.67	0.001	18.89	0.001	13.61	0.001	17.50	0.144
Dermatology	25	**4.72**	0.057	**5.12**	0.008	5.44	0.002	6.56	0.001	8.64	0.001	8.80	0.143
SpamBase	40	**12.21**	0.298	14.53	0.091	13.17	0.057	**11.20**	0.019	12.26	0.011	12.42	95.05
Arrhythmia	20	**40.78**	0.973	43.89	0.137	45.46	0.098	42.47	0.007	**41.43**	0.005	46.49	1.413
Madelon	70	**38.94**	12.73	46.50	4.032	51.50	4.124	–	–	40.67	0.073	**39.17**	272.8
Arcene	20	**39.50**	248.2	44.80	57.89	43.00	72.35	**31.00**	0.132	44.00	0.084	46.00	5.099
Average		**19.84**	29.16	24.47	6.910	23.55	8.516	–	–	**22.51**	0.020	23.51	41.67

Table 3
Average classification error rate (in %) together with the average execution time (in s) over 5 runs of the unsupervised feature selection methods considered over different datasets, using DT classifier. The best result for each dataset is shown in bold face and underlined and the second best is in bold face.

Datasets	# Selected features	UFSACO		RSM		MC		RRFS		TV		LS	
		Error	Time	Error	Time	Error	Time	Error	Time	Error	Time	Error	Time
Wine	5	**4.92**	0.025	13.66	0.007	7.65	0.002	**6.01**	0.001	13.66	0.001	9.84	0.023
Hepatitis	5	21.13	0.043	22.45	0.013	**16.41**	0.005	23.96	0.001	22.83	0.001	**17.16**	0.021
WDBC	5	**8.09**	0.019	13.66	0.008	8.92	0.002	9.02	0.001	**7.94**	0.001	8.14	0.352
Ionosphere	30	**11.39**	0.078	11.66	0.005	11.50	0.001	13.61	0.001	**11.41**	0.001	13.22	0.144
Dermatology	25	**8.16**	0.057	8.40	0.008	8.88	0.002	**8.18**	0.001	10.08	0.001	11.76	0.143
SpamBase	40	**7.43**	0.298	8.24	0.091	8.54	0.057	8.31	0.019	8.03	0.011	**8.01**	95.05
Arrhythmia	20	**40.91**	0.973	**44.29**	0.137	45.44	0.098	53.38	0.007	51.43	0.005	47.01	1.413
Madelon	70	**23.56**	12.73	47.83	4.032	50.00	4.124	–	–	23.83	0.073	**21.83**	272.8
Arcene	20	**32.60**	248.2	46.60	57.89	44.00	72.35	38.00	0.132	**29.00**	0.084	44.00	5.099
Average		**17.58**	29.16	24.09	6.910	22.37	8.516	–	–	**19.80**	0.020	20.11	41.67

feature selection methods on *WDBC*, *Arrhythmia*, *Madelon*, and *Arcene* datasets. Moreover, the UFSACO gets the second best accuracy on the *Dermatology* and *SpamBase* datasets. Finally, it acquires the average classification error rate 23.09% on all the datasets and lies on the first place among the methods.

Additionally, the performance of the proposed method has been compared to those of the unsupervised feature selection methods on the *Arcene* dataset. Table 5 reports the average classification error rate over 5 independent runs for RSM, MC, RRFS, TV, LS, and UFSACO methods using NB and DT classifiers. The reported results for the NB classifier show that in most cases, the UFSACO method has the best performance compared to the other methods. Note that the worst classification error rate of the UFSACO was 41.20%, while for RSM, MC, RRFS, TV, and LS the worst classification error rates were 48.40%, 46%, 44%, 43%, and 54%, respectively. Moreover, the average classification error rate of the proposed method was 34.92%, which indicates that the UFSACO method is superior among the unsupervised feature selection methods. It is clear from DT classifier results that the UFSACO method performed better than the other methods when the number of selected features was 40 or 60. Moreover, the proposed method obtained the second best result when the number of selected features was 20 or 100. Therefore, the average classification error rate of the proposed method was 33.88% which shows that the overall performance of the UFSACO method is much better than those of the others, except for the TV, the performance of which is slightly higher (*i.e.*, less than 0.1%).

Additionally, we have compared the proposed method to unsupervised feature selection methods. Figs. 3 and 4 plot the classification error rate (average over 5 independent runs) curves of SVM and DT classifiers on *Wine* and *Arrhythmia* datasets, respectively.

Fig. 3(a) indicates that the UFSACO is superior to the other methods applied on the SVM classifier when the number of selected features is less than 8. For example, when 3 features are selected, the classification error rate for UFSACO is around 11%, while this error rate for LS, MC, RSM, TV, and RRFS is around 16%, 13%, 28%, 39%, and 17%, respectively. In addition, Fig. 3(b) shows that the overall performance of UFSACO is better than those of LS, MC, RSM and TV methods and comparable with RRFS method.

Fig. 4(a) illustrates that when the number of selected features is less than 20, the performance of the proposed method is better than the performances of all methods. Moreover, the performance of UFSACO method is better than the performances of the unsupervised multivariate methods (*i.e.*, MC, RSM, and RRFS) when the number of features is less than 40. The results in Fig. 4(b) demonstrate that the UFSACO is significantly superior to all of the other methods when the number of selected features is less than 40. Especially, when 20 features were selected, the classification error rates were around 47%, 46%, 44%, 52%, 53%, and 41% for LS, MC, RSM, TV, RRFS, and UFSACO, respectively.

Furthermore, the proposed method has been compared to supervised feature selection methods. Tables 6 and 7 compare the average classification error rate over 5 independent runs of UFSACO to those of supervised feature selection methods including IG, GR, GI, FS, SU, LS, mRMR, and RRFS on *Madelon* dataset using SVM and DT classifiers.

The results of Table 6 demonstrate that the proposed method got the best result when the number of selected features is 10, 70, and 100, and for the other cases, the proposed method's classification error rate is slightly greater than the best obtained results (*i.e.*, less than 2%). Therefore, the average classification error rate of the UFSACO was reported 39.86%, and it attains the second best result among the supervised methods. UFSACO method achieved the

Table 4
Average classification error rate (in %) together with the average execution time (in s) over 5 runs of the unsupervised feature selection methods considered over different datasets, using NB classifier. The best result for each dataset is shown in bold face and underlined and the second best is in bold face.

Datasets	# Selected features	UFSACO		RSM		MC		RRFS		TV		LS	
		Error	Time	Error	Time	Error	Time	Error	Time	Error	Time	Error	Time
Wine	5	9.83	0.025	19.67	0.007	7.11	0.002	**4.26**	0.001	6.56	0.001	**6.23**	0.023
Hepatitis	5	20.94	0.043	**17.36**	0.013	**17.55**	0.005	22.83	0.001	20.00	0.001	20.94	0.021
WDBC	5	**7.58**	0.019	13.35	0.008	**9.07**	0.002	9.95	0.001	9.69	0.001	9.48	0.352
Ionosphere	30	19.44	0.078	**16.16**	0.005	**16.00**	0.001	22.22	0.001	20.83	0.001	23.33	0.144
Dermatology	25	**6.08**	0.057	**5.28**	0.008	**6.08**	0.002	10.56	0.001	8.00	0.001	8.80	0.143
SpamBase	40	**20.77**	0.298	26.89	0.091	26.22	0.057	**19.64**	0.019	21.14	0.011	21.24	95.05
Arrhythmia	20	**42.72**	0.973	**44.14**	0.137	44.94	0.098	58.44	0.007	51.04	0.005	45.71	1.413
Madelon	70	**39.28**	12.73	44.83	4.032	52.33	4.124	–	–	**39.33**	0.073	40.50	272.8
Arcene	20	**41.20**	248.2	44.40	57.89	44.00	72.35	44.00	0.132	**43.00**	0.084	54.00	5.099
Average		**23.09**	29.16	25.79	6.910	24.81	8.516	–	–	**24.40**	0.020	25.58	41.67

Table 5
Classification error rate (average over 5 runs, in %) of unsupervised feature selection methods considered over Arcene dataset using NB and DT classifiers. Std. is the standard deviation of the classification error rates. The best result for each number of features is shown in bold face.

#selected features	NB classifier						DT classifier					
	UFSACO	RSM	MC	RRFS	TV	LS	UFSACO	RSM	MC	RRFS	TV	LS
20	**41.20**	44.40	44.00	44.00	43.00	54.00	32.60	46.60	44.00	38.00	**29.00**	44.00
40	**32.40**	46.00	44.00	37.00	43.00	52.00	**33.00**	48.00	44.00	35.00	44.00	44.00
60	**33.60**	48.40	44.00	39.00	37.00	53.00	**30.80**	47.80	44.00	39.00	32.00	44.00
80	34.80	48.40	44.00	**28.00**	34.00	52.00	38.00	45.80	44.00	**31.00**	32.00	44.00
100	32.60	47.60	46.00	**28.00**	34.00	48.00	35.00	48.80	44.00	**32.00**	**32.00**	44.00
Average	**34.92**	46.96	44.40	35.20	38.20	51.80	33.88	47.40	44.00	35.00	**33.80**	44.00
Std.	3.64	1.73	**0.89**	7.05	4.55	2.28	2.74	1.19	**0.00**	3.54	5.85	**0.00**

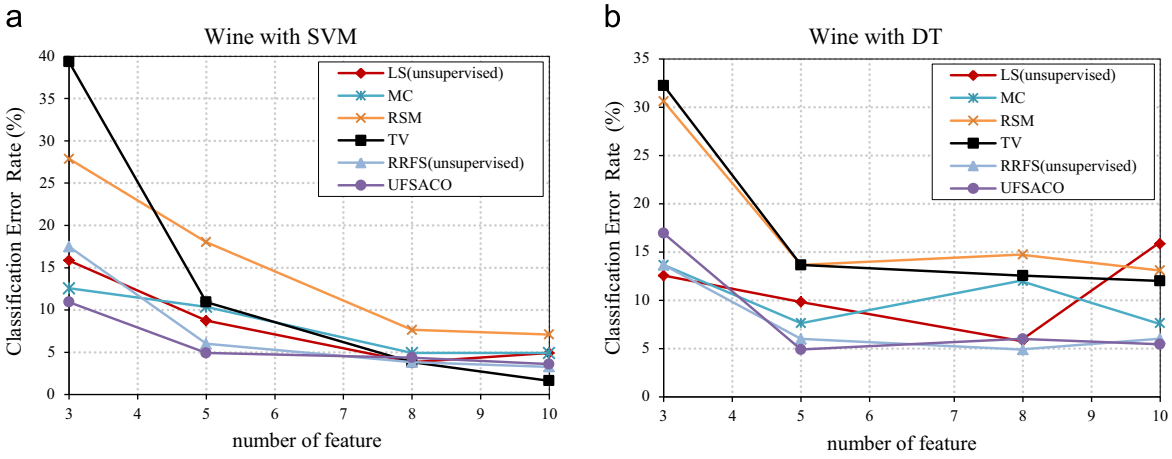

Fig. 3. Classification error rate (average over 5 runs), with respect to the number of selected features with unsupervised methods on: (a) Wine dataset with support vector machine and (b) Wine dataset with decision tree.

lowest classification error rate (*i.e.*, 38.17%) when the number of selected features was 10. Note that due to the high similarity between features on the *Madelon* dataset, RRFS method could not select more than 10 features.

From Table 7 results, it can be seen that UFSACO obtained the second best result when the number of selected features was 10, 40, or 70. Therefore, the proposed method achieved the average classification error rate 23.05% and lay on the second place among the methods. Consequently, it is clear that the overall performance of the UFSACO method is comparable with those of the well-known supervised feature selection methods.

Furthermore, the proposed method has been compared to supervised multivariate feature selection methods. Table 8 shows the classification error rates of the UFSACO, mRMR, and RRFS methods using SVM, NB, and DT classifiers. From the results of Table 8, it is clear that UFSACO performs better than the other supervised methods when SVM classifier has been applied on *Hepatitis* (5 selected features), *Ionosphere* (30 selected features), and *Madelon* (40 and 70 selected features) datasets. Moreover, the proposed method achieved a lower error rate compared to the mRMR over *Wine*, *Dermatology*, and *SpamBase* datasets. In addition, UFSACO acquires the lowest error rates on *Wine* (8 selected features), *Arrhythmia* (10 and 20 selected features) and *Madelon* (40 and 70 selected features) datasets when NB classifier was used. It can be seen from Table 8 that similar results have been reported when DT classifier is used. For example, the accuracy of the proposed method was significantly superior to the other supervised methods when applied on *Wine* (5 and 8 selected features), *Ionosphere* (25 selected features), *Dermatology* (15 selected features), *Madelon* (40 and 70 selected features), and *Arcene*

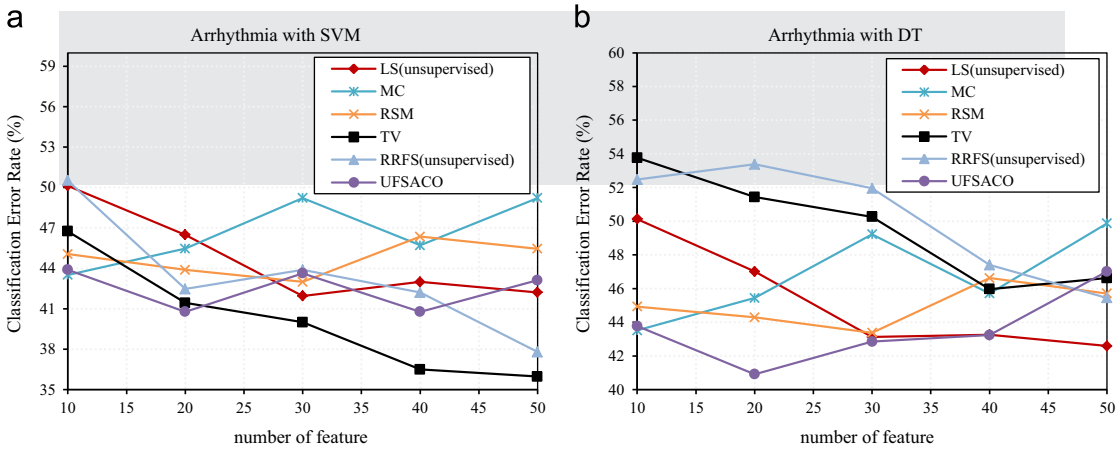

Fig. 4. Classification error rate (average over 5 runs), with respect to the number of selected features with unsupervised methods on: (a) Arrhythmia dataset with support vector machine and (b) Arrhythmia dataset with decision tree.

Table 6
Classification error rate (average over 5 runs, in %) of supervised feature selection methods considered over Madelon dataset using SVM classifier. Std. is the standard deviation of the classification error rates. The best result for each number of features is shown in bold face and underlined and the second best is in bold face.

# Selected features	UFSACO	IG	GR	GI	FS	SU	LS	mRMR	RRFS
10	**38.17**	38.67	38.67	38.67	**38.61**	38.67	38.67	43.17	39.17
20	39.67	38.83	**38.67**	**38.39**	39.33	**38.67**	39.17	44.00	–
40	39.55	**39.17**	**38.94**	**39.17**	40.67	**38.94**	39.33	54.00	–
70	**38.94**	**39.50**	39.78	41.67	43.33	41.61	40.33	50.83	–
100	**39.67**	42.28	41.22	42.33	42.33	41.78	**41.17**	48.17	–
150	43.16	42.22	**41.50**	42.72	42.22	**42.11**	43.17	43.00	–
Average	**39.86**	40.11	**39.80**	40.49	41.08	40.30	40.31	47.19	–
Std.	1.72	1.68	**1.28**	1.96	1.86	1.69	**1.66**	4.57	–

Table 7
Classification error rate (average over 5 runs, in %) of supervised feature selection methods considered over Madelon dataset using DT classifier. Std. is the standard deviation of the classification error rates. The best result for each number of features is shown in bold face and underlined and the second best is in bold face.

# Selected features	UFSACO	IG	GR	GI	FS	SU	LS	mRMR	RRFS
10	**21.89**	24.17	24.17	24.17	24.67	24.17	23.50	46.67	**19.33**
20	20.89	**19.50**	21.00	21.67	24.33	21.00	**16.17**	45.50	–
40	**20.67**	23.67	23.83	23.67	25.17	23.83	**19.67**	50.67	–
70	**23.56**	26.67	25.67	25.00	26.50	25.33	**20.83**	48.67	–
100	25.28	24.83	**23.83**	24.17	27.67	24.33	**21.83**	52.67	–
150	26.00	24.83	**21.83**	25.00	28.17	**21.67**	23.17	44.17	–
Average	**23.05**	23.94	23.39	23.95	26.08	23.39	**20.86**	48.06	–
Std.	2.26	2.40	1.69	**1.23**	**1.61**	1.68	2.71	3.23	–

(20 selected features) datasets. Consequently, we can conclude that the overall performance of the proposed method is comparable to those of the supervised multivariate feature selection methods over different classifiers, especially when *Wine*, *Ionosphere*, *SpamBase*, *Arrhythmia*, and *Madelon* datasets are used.

Among the results of the performed experiments, the following interesting points deserve attention:

- We have argued in this paper that the trade-offs between the execution time and the quality of the found solution should be considered in development of feature selection methods. It can be concluded from Tables 2–4 results that the proposed method achieves better qualitative results by spending a little more time compared to the other unsupervised methods.
- From Tables 2–4, it can be concluded that the overall performance of the proposed method is much better than those of the mentioned unsupervised methods (*i.e.*, RSM, MC, RRFS, TV, and LS) over different classifiers and datasets. Moreover, the UFSACO method outperforms unsupervised methods for different numbers of selected features (Table 5, Figs. 3 and 4).
- The performance of the proposed method is always superior to those of all the unsupervised univariate methods (*i.e.*, TV and LS) when using NB classifiers (Tables 4 and 5). That is because in these methods, the possible dependency between features will be ignored in the feature selection process. On the other hand, NB classifiers assume that features are conditionally independent from each other. Based on this assumption, univariate methods should obtain high accuracies, while in the real datasets, there are redundancies between features and thus, univariate methods will simply fail in the case of NB classifier. On the other hand, UFSACO method selects a subset of features with minimum redundancy between them (as mentioned in Section 4.2), so it is greatly suitable for NB classifier.
- The proposed method is an unsupervised method and does not need class labels in its search process. It has been shown through experiments that the proposed method is superior to the well-known supervised univariate feature selection methods (*i.e.*, IG, GR, GI, FS, SU, and LS). Furthermore, the results of the proposed method are comparable to those of supervised multivariate feature selection methods (*i.e.*, mRMR and RRFS). This is demonstrated through extensive experiments on different datasets using different classifiers (Tables 6–8).

6. Conclusion

In this paper, a new method based on ant colony optimization, called UFSACO, was proposed for finding an optimal solution to the feature selection problem. Ant colony optimization is a distributed method in which a set of agents cooperate to find a good solution. The proposed method is a multivariate approach in which possible dependencies between features are considered to reduce the redundancy among the selected features.

To evaluate the effect of UFSACO, we used three classifiers: support vector machine (SVM), decision tree (DT), and naïve Bayes (NB) on several standard datasets. Moreover, the proposed method has been compared to 11 well-known univariate and multivariate feature selection algorithms including information gain (IG), gain ratio (GR), symmetrical uncertainty (SU), Gini index (GI), Fisher score (FS), term variance (TV), Laplacian score (LS), minimal-redundancy–maximal-

Table 8
Classification error rate (average over 5 runs, in %), with respect to the number of selected features by proposed method and supervised multivariate methods for different datasets, using SVM, NB, and DT classifiers. The best result for each classifier is shown in bold face.

Datasets	# Selected features	SVM classifier			NB classifier			DT classifier		
		UFSACO	mRMR	RRFS	UFSACO	mRMR	RRFS	UFSACO	mRMR	RRFS
Wine	5	4.92	21.63	**3.93**	9.83	12.78	**4.59**	**4.92**	14.75	8.20
	8	4.37	10.16	**3.61**	**5.25**	11.14	5.57	**6.01**	12.13	9.18
Hepatitis	5	**16.85**	19.81	20.38	20.94	19.62	**18.11**	21.13	**17.73**	21.32
	8	20.56	21.13	**19.24**	20.37	20.75	**17.92**	23.01	21.69	**20.19**
WDBC	5	9.28	5.46	**5.21**	7.58	**6.44**	8.09	8.09	**5.98**	9.33
	15	5.05	**2.94**	5.57	6.80	**6.49**	7.52	7.06	**6.34**	8.61
Ionosphere	25	13.05	**11.00**	12.16	16.39	**16.16**	18.16	**9.17**	11.50	10.00
	30	**11.39**	11.50	13.16	19.44	20.16	**18.16**	11.39	**11.17**	14.00
Dermatology	15	12.32	13.60	**9.76**	12.56	**9.44**	9.60	**11.44**	17.92	15.20
	25	4.72	4.80	**3.92**	6.08	**4.16**	4.88	8.16	11.44	**5.84**
SpamBase	30	15.60	16.06	**11.33**	24.67	27.52	**18.08**	8.55	9.58	**7.43**
	40	12.21	14.05	**10.41**	20.77	24.13	**20.27**	7.43	8.33	**7.31**
Arrhythmia	10	43.90	**42.86**	47.79	**45.97**	50.78	52.59	43.77	**41.69**	46.88
	20	40.78	**40.52**	44.67	**42.72**	51.43	51.55	40.91	43.64	**40.77**
Madelon	40	**39.55**	54.00	–	**40.50**	51.33	–	**20.67**	50.67	–
	70	**38.94**	50.83	–	**39.28**	49.00	–	**23.56**	48.67	–
Arcene	20	39.50	–	**26.00**	41.20	–	**31.00**	**32.60**	–	35.00
	60	36.00	–	**22.00**	33.60	–	**29.00**	30.80	–	**26.00**

relevance (mRMR), mutual correlation (MC), random subspace method (RSM), and relevance-redundancy feature selection (RRFS). Our comprehensive experiments on different datasets demonstrate that the proposed method can effectively reduce the redundancy between selected features. Moreover, the experimental results show that in most cases the UFSACO significantly outperforms the unsupervised methods and is comparable with the supervised methods in terms of the classification error rate and number of selected features. We also showed that our UFSACO method can be classifier independent.

In the future work, we will use our method in supervised selection procedures with a new similarity measure. Another extension would be to develop a new updating rule in ACO algorithm to improve the efficiency of the feature selection process.

References

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

Akadi, A.E., Ouardighi, A.E., Aboutajdine, D., 2008. A powerful feature selection approach based on mutual information. Int. J. Comput. Sci. Netw. Secur. 8, 116–121.

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

Biesiada, J., Duch, W., 2007. Feature Selection for High-Dimensional Data: A Pearson Redundancy Based Filter, Computer Recognition Systems 2. Springer, Berlin Heidelberg, pp. 242–249.

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

Chen, C.-M., Lee, H.-M., Tan, C.-C., 2006. An intelligent web-page classifier with fair feature-subset selection. Eng. Appl. Artif. Intell. 19, 967–978.

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

Dorigo, M., Gambardella, L.M., 1997a. Ant colonies for the travelling salesman problem. Biosystems 43, 73–81.

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

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

Dorigo, M., Stützle, T., 2010. Ant Colony Optimization: Overview and Recent Advances, Handbook of Metaheuristics. Springer, US, pp. 227–263.

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

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

Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. Mach. Learn. 29, 131–163.

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

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

Gutjahr, W., 2007. Mathematical runtime analysis of ACO algorithms: survey on an emerging issue. Swarm Intell. 1, 59–79.

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

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

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

Haindl, M., Somol, P., Ververidis, D., Kotropoulos, C., 2006. Feature Selection Based on Mutual Correlation, Pattern Recognition, Image Analysis and Applications. Springer, Berlin Heidelberg, pp. 569–577.

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

He, X., Cai, D., Niyogi, P., 2005. Laplacian score for feature selection. Adv. Neural Inf. Process. Syst. 18, 507–514.

Huang, A., 2008. Similarity measures for text document clustering. In: Proceedings of the Sixth New Zealand Computer Science Research Student Conference, pp. 49–56.

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

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

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

Kuri-Morales, A., Rodríguez-Erazo, F., 2009. A search space reduction methodology for data mining in large databases. Eng. Appl. Artif. Intell. 22, 57–65.

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

Leung, Y., Hung, Y., 2010. A multiple-filter–multiple-wrapper approach to gene selection and microarray data classification. IEEE/ACM Trans. Comput. Biol. Bioinforma. 7, 108–117.

Li, Y., Wang, G., Chen, H., Shi, L., Qin, L., 2013. An ant colony optimization based dimension reduction method for high-dimensional datasets. J. Bion. Eng. 10, 231–241.

Liu, H., Motoda, H., 2007. Computational Methods of Feature Selection. Chapman and Hall, London.

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

Marinakis, Y., Marinaki, M., Doumpos, M., Zopounidis, C., 2009. Ant colony and particle swarm optimization for financial classification problems. Expert Syst. Appl. 36, 10604–10611.
Martínez Sotoca, J., Pla, F., 2010. Supervised feature selection by clustering using conditional mutual information-based distances. Pattern Recognit. 43, 2068–2081.
Meiri, R., Zahavi, J., 2006. Using simulated annealing to optimize the feature selection problem in marketing applications. Eur. J. Oper. Res. 171, 842–858.
Mesleh, A.M.D., Kanaan, G., 2008. Support vector machine text classification system: using ant colony optimization based feature subset selection. In: Proceedings of the International Conference on Computer Engineering & Systems, pp. 143–148.
Mitchell, T.M., 1997. Machine Learning. McGraw-Hill, New York.
Narendra, P.M., Fukunaga, K., 1977. A branch and bound algorithm for feature subset selection. IEEE Trans. Comput. 26, 917–922.
Nemati, S., Basiri, M.E., 2011. Text-independent speaker verification using ant colony optimization-based selected features. Expert Syst. Appl. 38, 620–630.
Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1226–1238.
Quinlan, J.R., 1986. Induction of decision trees. Mach. Learn. 1, 81–106.
Raileanu, L.E., Stoffel, K., 2004. Theoretical comparison between the Gini index and information gain criteria. Ann. Math. Artif. Intell. 41, 77–93.
Saeys, Y., Inza, I., Larrañaga, P., 2007. A review of feature selection techniques in bioinformatics. Bioinformatics 23, 2507–2517.
Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., Wang, Z., 2007. A novel feature selection algorithm for text categorization. Expert Syst. Appl. 33, 1–5.
Sikora, R., Piramuthu, S., 2007. Framework for efficient feature selection in genetic algorithm based data mining. Eur. J. Oper. Res. 180, 723–737.
Skalak, D.B., 1994. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: Proceedings of the 11th International Conference on Machine Learning pp. 293–301.
Sugumaran, V., Muralidharan, V., Ramachandran, K.I., 2007. Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. Mech. Syst. Signal Process. 21, 930–942.
Tahir, N.M., Hussain, A., Samad, S.A., Ishak, K.A., Halim, R.A., 2006. Feature selection for classification using decision tree. In: Proceedings of the Fourth Student Conference on Research and Development, pp. 99–102.
Theodoridis, S., Koutroumbas, K., 2008. Pattern Recognition. Academic Press, Oxford.
Trevino, V., Falciani, F., 2006. GALGO: an R package for multivariate variable selection using genetic algorithms. Bioinformatics 22, 1154–1156.
Uğuz, H., 2011. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowl.-Based Syst. 24, 1024–1032.
Unler, A., Murat, A., Chinnam, R.B., 2011. mr2PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. Inf. Sci. 181, 4625–4641.
Xiao, J., Li, L., 2011. A hybrid ant colony optimization for continuous domains. Expert Syst. Appl. 38, 11072–11077.
Yan, Z., Yuan, C., 2004. Ant Colony Optimization for Feature Selection in Face Recognition, Biometric Authentication. Springer, Berlin Heidelberg, pp. 221–226
Yang, J., Honavar, V., 1998. Feature subset selection using a genetic algorithm. IEEE Intell. Syst. Appl. 13, 44–49.
Yang, J., Liu, Y., Liu, Z., Zhu, X., Zhang, X., 2011. A new feature selection algorithm based on binomial hypothesis testing for spam filtering. Knowl.-Based Syst. 24, 904–914.
Yu, H., Gu, G., Liu, H., Shen, J., Zhao, J., 2009. A modified ant colony optimization algorithm for tumor marker gene selection. Genomics, Proteomics Bioinforma. 7, 200–208.
Yu, L., Liu, H., 2003. Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the 20th International Conference on Machine Learning, pp. 856–863.
Zhang, C.-K., Hu, H., 2005. 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, pp. 1728–1732.
Zibakhsh, A., Abadeh, M.S., 2013. Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function. Eng. Appl. Artif. Intell. 26, 1274–1281.