Expert Systems with Applications 36 (2009) 6843–6853

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Text feature selection using ant colony optimization

Mehdi Hosseinzadeh Aghdam *, Nasser Ghasem-Aghaee, Mohammad Ehsan Basiri

Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Hezar Jerib Avenue, Esfahan, Iran

ARTICLE INFO

Keywords:
Feature selection
Ant colony optimization
Genetic algorithm
Text categorization

ABSTRACT

Feature selection and feature extraction are the most important steps in classification systems. Feature selection is commonly used to reduce dimensionality of datasets with tens or hundreds of thousands of features which would be impossible to process further. One of the problems in which feature selection is essential is text categorization. A major problem of text categorization is the high dimensionality of the feature space; therefore, feature selection is the most important step in text categorization. At present there are many methods to deal with text feature selection. To improve the performance of text categorization, we present a novel feature selection algorithm that is based on ant colony optimization. Ant colony optimization algorithm is inspired by observation on real ants in their search for the shortest paths to food sources. Proposed algorithm is easily implemented and because of use of a simple classifier in that, its computational complexity is very low. The performance of proposed algorithm is compared to the performance of genetic algorithm, information gain and CHI on the task of feature selection in Reuters-21578 dataset. Simulation results on Reuters-21578 dataset show the superiority of the proposed algorithm.

1. Introduction

Feature selection (FS) is a commonly used step in machine learning, especially when dealing with a high dimensional space of features. The objective of feature selection is to simplify a dataset by reducing its dimensionality and identifying relevant underlying features without sacrificing predictive accuracy. By doing that, it also reduces redundancy in the information provided by the selected features. In real world problems FS is a must due to the abundance of noisy, irrelevant or misleading features. Feature selection is extensive and it spreads throughout many fields, including text categorization, data mining, pattern recognition and signal processing (Jensen, 2005).

Recently, text categorization has become a key technology to deal with and organize a large number of documents. A major problem of text categorization is the high dimensionality of the feature space. Most of these dimensions are not relative to text categorization; even some noise data hurt the performance of the classifier. Hence, we need to select some representative features from the original feature space to reduce the dimensionality of feature space and improve the efficiency and performance of classifier (Shang et al., 2007).

Several approaches are applied to the problem of feature selection in text categorization. Yang and Pedersen (1997) conducted a comparative study on five feature selection criteria for text categorization, including document frequency (DF), information gain (IG), mutual information (MI), a χ^2-test (CHI) and term strength (TS) and found χ^2 statistics and information gain more effective for optimizing classification results, and document frequency a better choice for efficiency and scalability if a small degradation in effectiveness is affordable. In Kim, Howland, and Park (2005) three methods consist of centroid, orthogonal centroid, and LDA/GSVD, which are designed for reducing the dimension of clustered data are used for dimensional reduction in text categorization.

Among too many methods which are proposed for FS, population-based optimization algorithms such as genetic algorithm (GA)-based method and ant colony optimization (ACO)-based method have attracted a lot of attention. These methods attempt to achieve better solutions by application of knowledge from previous iterations.

Genetic algorithms are optimization techniques based on the mechanism of natural selection. They used operations found in natural genetics to guide itself through the paths in the search space (Srinivas & Patnik, 1994). Because of their advantages, recently, GAs have been widely used as a tool for feature selection in data mining (Siedlecki & Sklansky, 1989).

Meta-heuristic optimization algorithm based on behavior of ants was represented in the early 1990s by Dorigo and Caro (1999). ACO is a branch of newly developed form of artificial intelligence called swarm intelligence (SI). Formally, swarm intelligence refers to the problem-solving behavior that emerges from the interaction of cooperative agents, and computational swarm intelligence (CSI) refers to algorithmic models of such behavior. More formally, swarm intelligence is the property of a system

* Corresponding author. Tel.: +98 311 7932671; fax: +98 311 7932670.
E-mail addresses: hosseinzadeh@eng.ui.ac.ir, mehdi_1224@yahoo.com (M.H. Aghdam).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.08.022

whereby the collective behaviors of unsophisticated agents interacting locally with their environment cause coherent functional global patterns to emerge (Engelbrecht, 2005). In groups of insects which live in colonies, such as ants and bees, an individual can only do simple task on its own, while the colony's cooperative work is the main reason determining the intelligent behavior it shows (Liu, Abbass, & McKay, 2004). ACO algorithm is inspired by social behavior of ant colonies. Although they have no sight, ants are capable of finding the shortest route between a food source and their nest by chemical materials called pheromone that they leave when moving (Bonabeau, Dorigo, & Theraulaz, 1999).

ACO algorithm was firstly used for solving traveling salesman problem (TSP) (Dorigo, Maniezzo, & Colorni, 1996) and then has been successfully applied to a large number of difficult problems like the quadratic assignment problem (QAP) (Maniezzo & Colorni, 1999), routing in telecommunication networks, graph coloring problems, scheduling, etc. This method is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal subset every time (Basiri, Ghasem-Aghaee, & Aghdam, 2008; Kanan, Faez, & Aghdam, 2007).

To the best of our knowledge, it is the first paper to apply an ACO-based technique to the problem of feature selection in text categorization. In this paper a new modified ACO-based feature selection algorithm has been introduced. The classifier performance and the length of selected feature subset are adopted as heuristic information for ACO. Thus, proposed algorithm needs no priori knowledge of features. Proposed algorithm is applied to text features of *bag of words* model in which a document is considered as a set of words or phrases (called terms) and each position in the input feature vector corresponds to a given term in original document (Caropreso & Matwin, 2006). Finally the classifier performance and the length of selected feature vector are considered for performance evaluation.

The rest of this paper is organized as follows. Section 2 presents a brief overview of feature selection methods. Ant colony optimization is described in Section 3. Section 4 explains the proposed feature selection algorithm. Genetic algorithms are described in Section 5. Section 6 discusses statistical feature selection methods. Section 7 reports computational experiments. It also includes a brief discussion of the results obtained and finally the conclusion and future works are offered in the last section.

2. Feature selection approaches

Feature selection is included in discrete optimization problems. The whole search space for optimization contains all possible subsets of features, meaning that its size is

$$\sum_{s=0}^{n}\binom{n}{s}=\binom{n}{0}+\binom{n}{1}+\ldots+\binom{n}{n}=2^n, \tag{1}$$

where n is the dimensionality (the number of features) and s is the size of the current feature subset (Mladenić, 2006). Usually FS algorithms involve heuristic or random search strategies in an attempt to avoid this prohibitive complexity. However, the degree of optimality of the final feature subset is often reduced (Jensen, 2005).

Feature selection algorithms can be classified into three categories based on their evaluation procedure (Duda & Hart, 1973). If an algorithm performs FS independent of any learning algorithm (i.e. it is a completely separate preprocessor), then it is included in filter approach category. This approach is mostly includes selecting features based on inter-class separability criterion (Duda & Hart, 1973). If the evaluation procedure is tied to the task (e.g. classification) of the learning algorithm, the FS algorithm is a sort of wrapper approach. This method searches through the feature subset space using the estimated accuracy from an induction algorithm as a measure of subset suitability. If the feature selection and learning algorithm are interleaved then the FS algorithm is a kind of embedded approach (Mladenić, 2006). Although wrappers may produce better results, they are expensive to run and can break down with very large numbers of features. This is due to the use of learning algorithms in the evaluation of subsets, some of which can encounter problems while dealing with large datasets (Forman, 2003; Jensen, 2005).

In the wrapper approach the evaluation function calculates the suitability of a feature subset produced by the generation procedure and it also compares that with the previous best candidate, replacing it if found to be better. A stopping criterion is tested in each of iterations to determine whether or not the FS process should continue.

The three mentioned approaches are also classified into five main methods (Mladenić, 2006):

- *Forward selection*: begins with an empty set and features are added to greedily one at a time.
- *Backward elimination*: begins with a feature set containing all features and features are removed from greedily one at a time.
- *Forward stepwise selection*: begins with an empty set and features are either added to or removed from greedily one at a time.
- *Backward stepwise elimination*: begins with a feature set containing all features and features are either added to or removed from greedily one at a time.
- *Random mutation*: begins with a feature set containing randomly selected features. Randomly selected features are added to or removed from one at a time and process stops after a given number of iterations.

Other famous FS approaches are based on the genetic algorithm (GA) (Siedlecki & Sklansky, 1989), Simulated Annealing (SA), particle swarm optimization (PSO) (Wang, Yang, Teng, Xia, & Jensen, 2007) and ant colony optimization (ACO) (Ani, 2005; Jensen, 2005; Kanan et al., 2007; Liu et al., 2004; Zhang & Hu, 2005).

Ani (2005) proposes a subset search procedure based on ACO for speech classification problem. The hybrid of ACO and mutual information has been used for feature selection in the forecaster (Zhang & Hu, 2005). Furthermore, ACO is used for finding rough set reducts (Jensen, 2005) and a new Ant-Miner which used a different pheromone updating strategy has been introduced in Liu et al. (2004). Also, in Kanan et al. (2007) an ACO-based method has been used in the application of face recognition systems and some surveys of feature selection algorithms are given in Bins (2000), Kml and Kittler (1978) and Siedlecki and Sklansky (1988).

3. Ant colony optimization (ACO)

In the early 1990s, ant colony optimization was introduced by Dorigo and colleagues as a novel nature-inspired meta-heuristic for the solution of hard combinatorial optimization (CO) problems. ACO belongs to the class of meta-heuristics, which includes approximate algorithms used to obtain good enough solutions to hard CO problems in a reasonable amount of computation time. The inspiring source of ACO is the foraging behavior of real ants (Dorigo & Blum, 2005).

The first ACO algorithm developed was the ant system (AS) (Dorigo, 1992), and since then several improvement of the AS have been devised (Gambardella & Dorigo, 1995; Gambardella & Dorigo, 1996; Stützle & Hoos, 1997). The ACO algorithm is based on a computational paradigm inspired by real ant colonies and the way they function. The underlying idea was to use several constructive computational agents (simulating real ants). A dynamic memory structure incorporating information on the effectiveness of previous

choices based on the obtained results, guides the construction process of each agent. The behavior of each single agent is therefore inspired by the behavior of real ants (Montemanni, Gambardella, Rizzoli, & Donati, 2002).

The paradigm is based on the observation made by ethologists about the medium used by ants to communicate information regarding shortest paths to food by means of pheromone trails. A moving ant lays some pheromone on the ground, thus making a path by a trail of this substance. While an isolated ant moves practically at random, exploration, an ant encountering a previously laid trail can detect it and decide with high probability to follow it, exploitation, and consequently reinforces the trail with its own pheromone. What emerges is a form of autocatalytic process through which the more the ants follow a trail, the more attractive that trail becomes to be followed. The process is thus characterized by a positive feedback loop, during which the probability of choosing a path increases with the number of ants that previously chose the same path. The mechanism above is the inspiration for the algorithms of the ACO family (Montemanni et al., 2002).

ACO algorithms can be applied to any optimization problem, for which the following problem-dependent aspects can be defined (Bonabeau et al., 1999; Dorigo et al., 1996):

- An appropriate *graph representation* to represent the discrete search space. The graph should accurately represent all states and transitions between states. A solution representation scheme also has to be defined.
- *Heuristic desirability* of links the representation graph.
- An *autocatalytic (positive) feedback process*; that is, a mechanism to update pheromone concentrations such that current successes positively influence feature solution construction.
- A *constraint-satisfaction method* to ensure that only feasible solutions are constructed.
- A solution construction method which defines the way in which solutions are built and a state transition probability.

3.1. Ant colony optimization for feature selection

Feature selection is one of the applications of subset problems. Given a feature set of size n, the FS problem is to find a minimal feature subset of size s ($s < n$) while retaining a suitably high accuracy in representing the original features. Therefore, there is no concept of path. A partial solution does not define any ordering among the components of the solution, and the next component to be selected is not necessarily influenced by the last component added to the partial solution (Blum & Dorigo, 2004; Leguizamon & Michalewicz, 1999). Furthermore, solutions to an FS problem are not necessarily of the same size. To apply an ACO algorithm to solve a feature selection problem, these aspects need to be addressed. The first problem is addressed by redefining the way that the representation graph is used.

3.1.1. Graph representation

The feature selection problem may be reformulated into an ACO-suitable problem. ACO requires a problem to be represented as a graph. Here nodes represent features, with the edges between them denoting the choice of the next feature. The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion. Fig. 1 illustrates this setup. Nodes are fully connected to allow any feature to be selected next. The ant is currently at node f_1 and has a choice of which feature to add next to its path (dotted lines). It chooses feature f_2 next based on the transition rule, then f_3 and then f_4. Upon arrival at f_4, the current subset $\{f_1 f_2, f_3, f_4\}$ is determined to satisfy the traversal

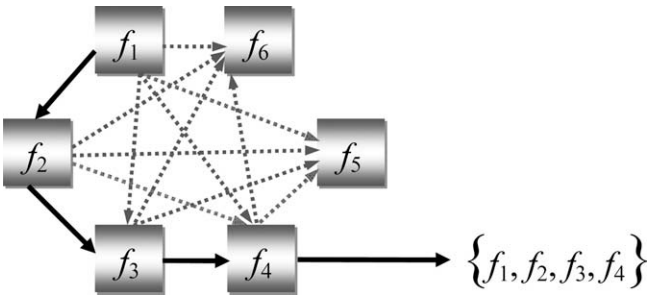

Fig. 1. ACO problem representation for FS.

stopping criterion (e.g. suitably high classification accuracy has been achieved with this subset). The ant terminates its traversal and outputs this feature subset as a candidate for data reduction (Kanan et al., 2007).

On the basis of this reformulation of the graph representation, the transition rules and pheromone update rules of standard ACO algorithms can be applied. In this case, pheromone and heuristic value are not associated with links. Instead, each feature has its own pheromone value and heuristic value.

3.1.2. Heuristic desirability

The basic ingredient of any ACO algorithm is a constructive heuristic for probabilistically constructing solutions. A constructive heuristic assembles solutions as sequences of elements from the finite set of solution components. A solution construction starts with an empty partial solution. Then, at each construction step the current partial solution is extended by adding a feasible solution component from the set of solution components (Dorigo & Blum, 2005). A suitable heuristic desirability of traversing between features could be any subset evaluation function for example, an entropy-based measure (Jensen, 2005) or rough set dependency measure (Pawlak, 1991). In proposed algorithm classifier performance is mentioned as heuristic information for feature selection. The heuristic desirability of traversal and node pheromone levels are combined to form the so-called *probabilistic transition rule*, denoting the probability that ant k will include feature i in its solution at time step t:

$$P_i^k(t) = \begin{cases} \frac{[\tau_i(t)]^\alpha \cdot [\eta_i]^\beta}{\sum_{u \in J^k} [\tau_u(t)]^\alpha \cdot [\eta_u]^\beta} & \text{if } i \in J^k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where J^k is the set of feasible features that can be added to the partial solution; τ_i and η_i are respectively the pheromone value and heuristic desirability associated with feature i. α and β are two parameters that determine the relative importance of the pheromone value and heuristic information.

The transition probability used by ACO is a balance between pheromone intensity (i.e. history of previous successful moves), τ_i, and heuristic information (expressing desirability of the move), η_i. This effectively balances the exploitation–exploration trade-off. The search process favors actions that it has found in the past and which proved to be effective, thereby exploiting knowledge obtained about the search space. On the other hand, in order to discover such actions, the search has to investigate previously unseen actions, thereby exploring the search space. The best balance between exploitation and exploration is achieved through proper selection of the parameters α and β. If $\alpha = 0$, no pheromone information is used, i.e. previous search experience is neglected. The search then degrades to a stochastic greedy search. If $\beta = 0$, the attractiveness (or potential benefit) of moves is neglected.

3.1.3. Pheromone update rule

After all ants have completed their solutions, pheromone evaporation on all nodes is triggered, and then according to Eq. (3) each ant k deposits a quantity of pheromone, $\Delta\tau_i^k(t)$, on each node that it has used

$$\Delta\tau_i^k(t) = \begin{cases} \phi \cdot \gamma(S^k(t)) + \frac{\varphi \cdot (n - |S^k(t)|)}{n} & \text{if } i \in S^k(t) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $S^k(t)$ is the feature subset found by ant k at iteration t, and $|S^k(t)|$ is its length. The pheromone is updated according to both the measure of the classifier performance, $\gamma(S^k(t))$, and feature subset length. ϕ and φ are two parameters that control the relative weight of classifier performance and feature subset length, $\phi \in [0,1]$ and $\varphi = 1 - \phi$. This formula means that the classifier performance and feature subset length have different significance for feature selection task. In our experiment we assume that classifier performance is more important than subset length, so they were set as $\phi = 0.8$, $\varphi = 0.2$.

In practice, the addition of new pheromone by ants and pheromone evaporation are implemented by the following rule applied to all the nodes:

$$\tau_i(t+1) = (1-\rho)\tau_i(t) + \sum_{k=1}^{m} \Delta\tau_i^k(t) + \Delta\tau_i^g(t), \quad (4)$$

where m is the number of ants at each iteration and $\rho \in (0,1)$ is the pheromone trail decay coefficient. The main role of pheromone evaporation is to avoid stagnation, that is, the situation in which all ants constructing the same solution. g indicates the best ant at each iteration. All ants can update the pheromone according to Eq. (4) and the best ant deposits additional pheromone on nodes of the best solution. This leads to the exploration of ants around the optimal solution in next iterations.

3.1.4. Solution construction

The overall process of ACO feature selection can be seen in Fig. 2. The process begins by generating a number of ants which are then placed randomly on the graph i.e. each ant starts with one random feature. Alternatively, the number of ants to place on the graph may be set equal to the number of features within the data; each ant starts path construction at a different feature. From these initial positions, they traverse nodes probabilistically until a traversal stopping criterion is satisfied. The resulting subsets are gathered and then evaluated. If an optimal subset has been found or the algorithm has executed a certain number of times, then the process halts and outputs the best feature subset encountered. If none of these conditions hold, then the pheromone is updated, a new set of ants are created and the process iterates once more.

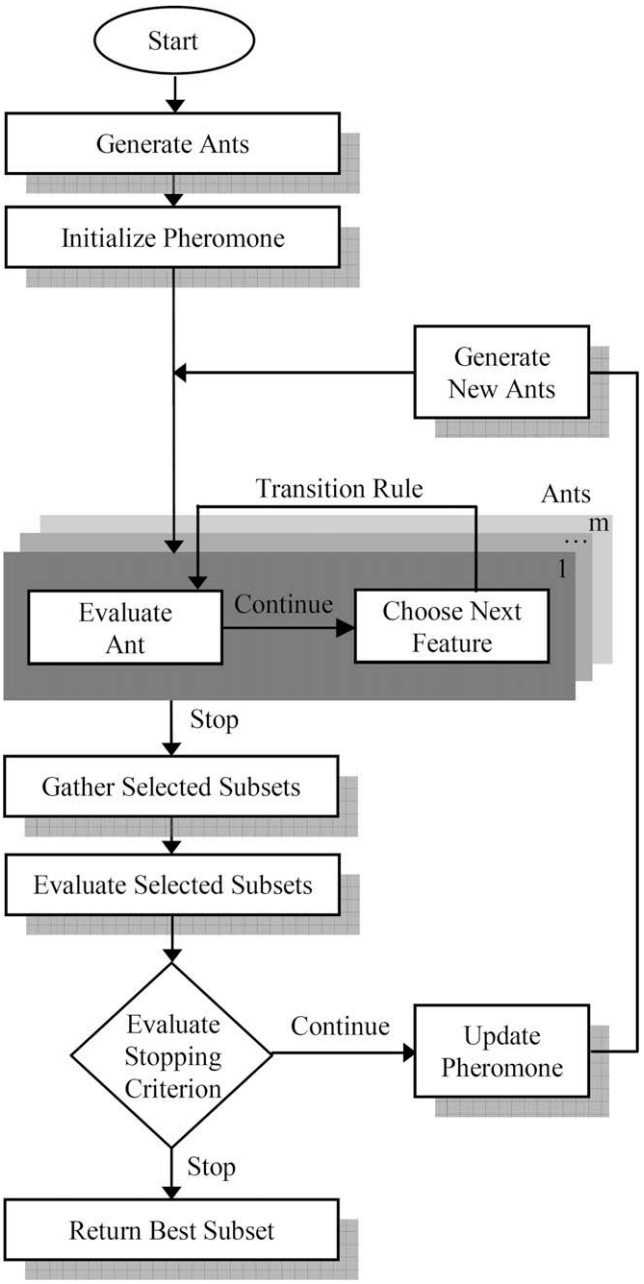

Fig. 2. ACO-based feature selection algorithm.

4. Proposed feature selection algorithm

Typically a text categorization system consists of several essential parts including feature extraction and feature selection. After preprocessing of text documents, feature extraction is used to transform the input text document into a feature set (feature vector). Feature selection is applied to the feature set to reduce the dimensionality of it. This process is shown in Fig. 3. ACO is used to explore the space of all subsets of given feature set. The performance of selected feature subsets is measured by invoking an evaluation function with the corresponding reduced feature space and measuring the specified classification result. The best feature subset found is then output as the recommended set of features to be used in the actual design of the classification system.

The main steps of proposed feature selection algorithm are as follows:

1. Initialization
   - Determine the population of ants.
   - Set the intensity of pheromone trial associated with any feature.
   - Determine the maximum of allowed iterations.

2. Solution generation and evaluation of ants.
   - Assign any ant randomly to one feature and visiting features, each ant builds solutions completely. In this step, the evaluation criterion is mean square error (MSE) of the classifier. If an ant is not able to decrease the MSE of the classifier in ten successive steps, it will finish its work and exit.

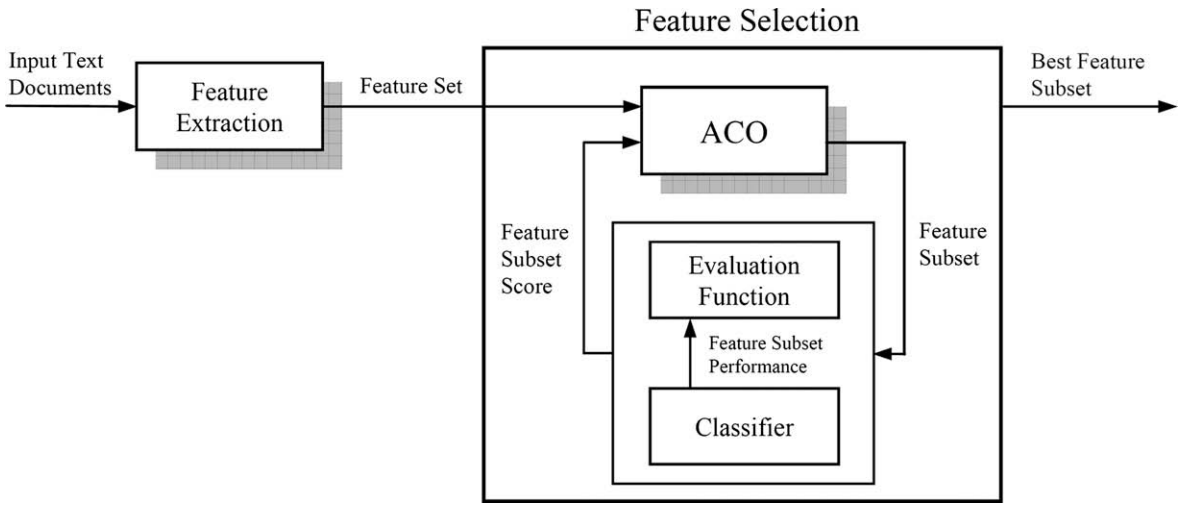

Fig. 3. Block diagram of proposed feature selection algorithm.

3. Evaluation of the selected subsets.
   - Sort selected subsets according to classifier performance and their length. Then, select the best subset.

4. Check the stop criterion.
   - Exit, if the number of iterations is more than the maximum allowed iteration, otherwise continue.

5. Pheromone updating.
   - Decrease pheromone concentrations of nodes then, all ants deposit the quantity of pheromone on graph. Finally, allow the best ant to deposit additional pheromone on nodes.

6. Generation of new ants.
   - In this step previous ants are removed and new ants are generated.

7. Go to 2 and continue.

The time complexity of proposed algorithm is $O(Imn)$, where I is the number of iterations, m the number of ants, and n the number of original features. This can be seen from Fig. 2. In the worst case, each ant selects all the features. As the heuristic is evaluated after each feature is added to the candidate subset, this will result in n evaluations per ant. After the first iteration in this algorithm, mn evaluations will have been performed. After I iterations, the heuristic will be evaluated Imn times.

5. Genetic algorithm (GA)

The GAs are stochastic global search methods that mimic the metaphor of natural biological evolution (Srinivas & Patnik, 1994). These algorithms are general purpose optimization algorithms with a probabilistic component that provide a means to search poorly understood, irregular spaces. Instead of working with a single point, GAs work with a population of points. Each *point* is a vector in hyperspace representing one potential (or candidate) solution to the optimization problem. A population is, thus, just an ensemble or set of hyperspace vectors. Each vector is called a chromosome in the population. The number of elements in each vector (chromosome) depends on the number of parameters in the optimization problem and the way to represent the problem. How to represent the problem as a string of elements is one of the critical factors in successfully applying a GA (or other evolutionary algorithm) to a problem. A typical series of operations carried out when implementing a GA paradigm is

- Initialize the population;
- Calculate fitness for each chromosome in population;
- Reproduce selected chromosomes to form a new population;
- Perform crossover and mutation on the population;
- Loop to second step until some condition is met.

Initialization of the population is commonly done by seeding the population with random values. The fitness value is proportional to the performance measurement of the function being optimized. The calculation of fitness values is conceptually simple. It can, however, be quite complex to implement in a way that optimizes the efficiency of the GAs search of the problem space. It is this fitness that guides the search of the problem space.

After fitness calculation, the next step is reproduction. Reproduction comprises forming a new population, usually with the same total number of chromosomes, by selecting from members of the current population using a stochastic process that is weighted by each of their fitness values. The higher the fitness, the more likely it is that the chromosome will be selected for the new generation. One commonly used way is a *roulette wheel* procedure that assigns a portion of a roulette wheel to each population member where the size of the portion is proportional to the fitness value. This procedure is often combined with the elitist strategy, which ensures that the chromosome with the highest fitness is always copied into the next generation. The next operation is called crossover. To many evolutionary computation practitioners, crossover is what distinguishes a GA from other evolutionary computation paradigms. Crossover is the process of exchanging portions of the strings of two "parent" chromosomes. An overall probability is assigned to the crossover process, which is the probability that given two parents, the crossover process will occur. This probability is often in the range of 0.65–0.80. The final operation in the typical GA procedure is mutation. Mutation consists of changing an element's value at random, often with a constant probability for each element in the population. The probability of mutation can vary widely according to the application and the preference of the person exercising the GA. However, values of between 0.001 and 0.01 are not unusual for mutation probability.

5.1. Genetic algorithm for feature selection

Several approaches exist for using GAs for feature subset selection. The two main methods that have been widely used in the past

are as follow. First is due to (Siedlecki & Sklansky, 1989), of finding an optimal binary vector in which each bit corresponds to a feature (binary vector optimization (BVO) method). A '1' or '0' suggests that the feature is selected or dropped, respectively. The aim is to find the binary vector with the smallest number of 1's such that the classifier performance is maximized. This criterion is often modified to reduce the dimensionality of the feature vector at the same time (Yang & Honavar, 1998). The second and more refined technique uses an m-ary vector to assign weights to features instead of abruptly dropping or including them as in the binary case (Punch, Goodman, Pei, Hovland, & Enbody, 1993). This gives a better search resolution in the multidimensional space (Raymer, Punch, Goodman, Kuhn, & Jain, 2000).

6. Statistical approaches

Most of text feature selection methods are based on statistical theory and machine learning. Some well-known methods are information gain, CHI, term frequency, expected cross entropy, the weight of evidence of text, odds ratio and mutual information (Yang & Pedersen, 1997). Yang and Pedersen (1997) reported information gain and CHI performed best in their multi-class benchmarks. Therefore, in our experiment we use these two methods amongst statistical methods for comparison with proposed algorithm.

6.1. Information gain (IG)

Information gain is frequently employed as a term goodness criterion in the field of machine learning (Mitchell, 1996; Yang & Pedersen, 1997). It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. The information gain of term t is

$$\begin{aligned}\mathrm{Gain}(t) = &-\sum_{i=1}^{|C|} P(c_i)\cdot\log P(c_i)\\ &+P(t)\cdot\sum_{i=1}^{|C|} P(c_i|t)\cdot\log P(c_i|t)\\ &+P(\bar{t})\cdot\sum_{i=1}^{|C|} P(c_i|\bar{t})\cdot\log P(c_i|\bar{t})\end{aligned} \tag{5}$$

where c_i denotes ith category, $P(c_i)$ is the probability of the ith category, $P(t)$ is the probability that term t occurred, $P(c_i|\ t)$ is the conditional probability of the ith category given that term t occurred, $P(\bar{t})$ is the probability that term t did not occur and $P(c_i|\bar{t})$ is the conditional probability of the ith category given that term t did not occur.

6.2. χ^2 Statistic (CHI)

The χ^2 statistic measures the lack of independence between t and c and can be compared to the χ^2 distribution with one degree of freedom to judge extremeness. Using the two-way contingency table of a term t and a category c, where A is the number of times t and c co-occur, B is the number of time the t occurs without c, C is the number of times c occurs without t, D is the number of times neither c nor t occurs, and N is the total number of documents, the term-goodness measure is defined in Eq. (6) (Yang & Pedersen, 1997)

$$\mathrm{CHI}(t,c) = \frac{N\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)}. \tag{6}$$

7. Experimental results

A series of experiments was conducted to show the utility of proposed feature selection algorithm. All experiments have been run on a machine with 3.0 GHz CPU and 512 MB of RAM. We implement proposed ACO algorithm and other three feature selection algorithms in Matlab R2006a. The operating system was Windows XP Professional. The following sections describe Reuters-21578 dataset and implementation results.

7.1. Dataset

There are some publicly available standard datasets that can be used as test collections for text categorization. The most widely used is the Reuters collection, consisting of stories from Reuter's news agency which classified under categories related to economics. The Reuters collection accounts for most of the experimental works in text categorization so far.

We used Reuters-21578, the newer version of the corpus. In Reuters-21578 dataset, we adopt the top ten classes; 5785 documents in training set and 2299 documents in test set. The distribution of the class is unbalance. The maximum class has 2721 documents, occupying 47.04% of training set. The minimum class has 72 documents, occupying 1.24% of training set. Table 1 shows the ten most frequent categories along with the number of training and test examples in each.

7.2. Feature extraction

Text documents cannot be directly interpreted by a classifier. Because of this, an indexing procedure that maps a text document into a compact representation of its content needs to be uniformly applied to documents. As showed in Eq. (7) a text d_j is usually represented as a vector of term weights

$$d_j = \{w_{1j}, w_{2j}, \ldots, w_{|T|j}\} \tag{7}$$

where T is the set of terms (features) that occur at least once in at least one document of training set, and $0 \leqslant w_{kj} \leqslant 1$ represents, how much term t_k contributes to the semantics of document d_j.

Typically each position in the input feature vector corresponds to a given word or phrase. This representation often called *bag of words* model. Weights are determined using normalized *tfidf* function (Salton et al., 1987), defined as

$$w_{kj} = \frac{tfidf(t_k,d_j)}{\sqrt{\sum_{s=1}^{|T|}(tfidf(t_s,d_j))^2}} \tag{8}$$

where

$$tfidf(t_k,d_j) = \#(t_k,d_j)\cdot\log\frac{|Tr|}{\#_{Tr}(t_k)}. \tag{9}$$

where $\#(t_k,d_j)$ is the number of occurrence of t_k in d_j, Tr is the training set and $|Tr|$ is its length. $\#_{Tr}(t_k)$ denote the number of documents in Tr in which t_k occurs.

7.3. Performance measure

Usually precision (π) and recall (ρ) are used for performance measurement. They showed in the following equations:

Table 1
Number of training/test documents

Category name	Number of train	Number of test
Acquisition	1484	664
Corn	170	53
Crude	288	126
Earn	2721	1052
Grain	72	32
Interest	165	74
Money-fx	313	106
Ship	122	42
Trade	297	99
Wheat	153	51

$$\pi_i = \frac{TP_i}{TP_i + FP_i} \tag{10}$$

$$\rho_i = \frac{TP_i}{TP_i + FN_i} \tag{11}$$

where TP_i is the number of test documents correctly classified under ith category (c_i), FP_i is the number of test documents incorrectly classified under c_i, and FN_i is the number of test documents incorrectly classified under other categories these probabilities may be estimated in terms of the contingency table for c_i on a given test set which is shown in Table 2. Another commonly used measure in TC is F1 measure that is defined in Eq. (12) (Rijsbergen, 1979)

$$F1 = \frac{2 \times \pi \times \rho}{(\pi + \rho)} \tag{12}$$

When dealing with multiple classes there are two possible ways of averaging above measures, namely, macro average and micro average. The macro average weights equally all the classes, regardless of how many documents belong to it. The micro average weights equally all the documents, thus favoring the performance on common classes. The *global* contingency table which is shown in Table 3 is thus obtained by summing over category-specific contingency tables; Eqs. (13)–(16) show micro averaging and macro averaging on precision and recall

$$\pi^{\mu} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)} \tag{13}$$

$$\rho^{\mu} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)} \tag{14}$$

$$\pi^{M} = \frac{\sum_{i=1}^{|C|} \pi_i}{|C|} \tag{15}$$

$$\rho^{M} = \frac{\sum_{i=1}^{|C|} \rho_i}{|C|} \tag{16}$$

where μ denotes micro averaging and M denotes macro averaging.

7.4. Results

To show the utility of proposed ACO-based algorithm we compare proposed algorithm with genetic algorithm, information gain and CHI. Various values were tested for the parameters of proposed algorithm. The results show that the highest performance is achieved by setting the parameters to values shown in Table 4. These values were empirically determined in our preliminary experiments; but we make no claim that these are optimal values. Parameter optimization is a topic for future research.

Analyzing the precision and recall shown in Table 5, we see that on average, the ACO-based algorithm obtained a higher accuracy value than the genetic algorithm, information gain and CHI.

To graphically illustrate the progress of the ant colony as it searches for optimal solutions, we take percent features as the horizontal coordinate and the F1 measure as the vertical coordinate. This should illustrate the process of improvement of the best ant as the number of features increase. Figs. 4 and 5 show the micro-averaged and macro-averaged F1 measure for each of the feature selection algorithms as we change the number of selected features.

Table 2
The contingency table for category c_i

Category c_i		Expert judgments	
		Yes	No
Classifier	Yes	TP_i	FP_i
Judgments	No	FN_i	TN_i

Table 3
The global contingency table

Category set $C = \{c_1, \ldots, c_{-C-}\}$		Expert judgments	
		Yes	No
Classifier	Yes	$TP = \sum_{i=1}^{|C|} TP_i$	$FP = \sum_{i=1}^{|C|} FP_i$
Judgments	No	$FN = \sum_{i=1}^{|C|} FN_i$	$TN = \sum_{i=1}^{|C|} TN_i$

The results show that as the percentage of selected features exceeds 12% in micro-F1 and macro-F1 measures, the ACO-based algorithm outperforms genetic algorithm, information gain and CHI.

Table 6 describes micro-F1 and macro-F1 for four feature selection algorithms. From this table, we can see that the best categorization performance is achieved with GA and ACO. To further highlight the search process, we graph percent selected features of every ant's current iteration (horizontal coordinate) against classifier performance (vertical coordinate). Each point in the figure is an ant. The process of the ant colony searching for optimal solutions for Reuters-21578 dataset is given in Figs. 6a–6k.

From the results and figures, we can see that, compared with GA, ACO is quicker in locating the optimal solution. In general, it can find the optimal solution within tens of iterations. If exhaustive search is used to find the optimal feature subset in the Reuters-21578 dataset, there will be tens of billions of candidate subsets, which is impossible to execute. But with ACO, at the 100nd iteration the optimal solution is found.

ACO has powerful exploration ability; it is a gradual searching process that approaches optimal solutions. The running time of ACO is affected more by the problem dimension (feature numbers), and the size of data. For some datasets with more features, after finding a sub-optimal solution, the GA cannot find a better one. However, ACO can search in the feature space until the optimal solution is found. The GA is affected greatly by the number of features.

ACO comprises a very simple concept, and the ideas can be implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed. This optimization technique does not suffer, however, from some of the difficulties of GAs; interaction in the colony enhances rather than detracts from progress toward the solution. Further, an ant colony system has memory, which the genetic algorithm does not have. Changes in genetic populations result in the destruction of previous knowledge of the problem. In ant colony optimization, ants that past optima are tugged to return towards them; knowledge of good solutions is retained by all ants.

8. Conclusion

This paper discusses the shortcomings of statistical approaches to feature selection. These techniques often fail to find optimal reductions, as no perfect heuristic can guarantee optimality. On the other hand, complete searches are not feasible for even medium sized datasets. So, stochastic approaches provide a promising feature selection mechanism.

We propose a new optimal feature selection technique based on ant colony optimization (ACO). We compare its performance with the other feature selection methods in text categorization. ACO has the ability to converge quickly; it has a strong search capability in the problem space and can efficiently find minimal feature subset. Experimental results demonstrate competitive performance.

More experimentation and further investigation into this technique may be required. The pheromone trail decay coefficient (ρ) and pheromone amount ($\Delta\tau_i^k(t)$) have an important impact on

Table 4
GA and ACO parameters settings

	Population	Iteration	Crossover probability	Mutation probability	Initial pheromone	α	β	ρ
GA	50	100	0.7	0.005	–	–	–	–
ACO	50	100	–	–	1	1	0.1	0.2

Table 5
The performance (precision and recall) of information gain, CHI, GA and ACO on Reuters-21578 dataset

Category name	IG		CHI		GA		ACO	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Acquisition	91.7466	71.988	89.5575	76.2048	92.4528	81.1747	90.1325	92.1687
Corn	87.8049	67.9245	87.8049	67.9245	90.6977	73.5849	90.9091	75.4717
Crude	84.9057	71.4286	83.0189	69.8413	78.3217	88.8889	82.963	88.8889
Earn	84.3803	94.4867	84.9829	94.6768	90.1961	96.1977	98.2018	93.4411
Grain	63.6364	65.625	60.6061	62.5	69.697	71.875	66.6667	75
Interest	45.7627	36.4865	54	36.4865	60	36.4865	49.3506	51.3514
Money-fx	51.7241	56.6038	61.2245	56.6038	67.8899	69.8113	68.5185	69.8113
Ship	33.8235	54.7419	37.5	57.1429	57.1429	66.6667	59.2593	76.1905
Trade	68.7023	90.9091	73.9837	91.9192	72.3577	89.899	77.5862	90.9091
Wheat	91.3043	82.3529	89.3617	82.3529	87.7551	84.3137	87.7551	84.3137
Average	70.3791	69.2547	72.204	69.5653	76.6511	75.8898	77.1343	79.7546

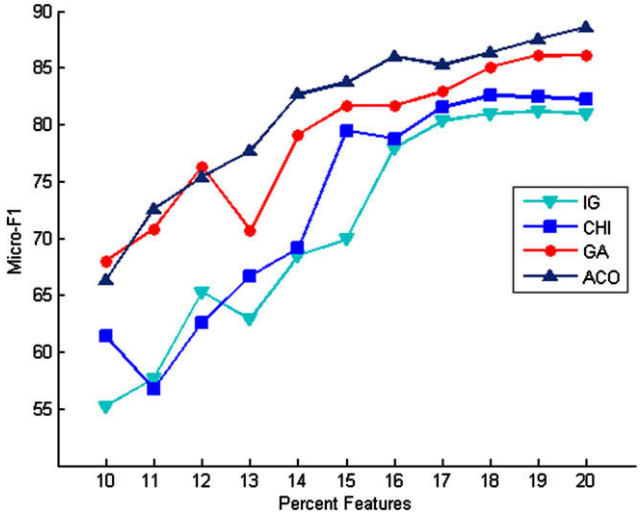

Fig. 4. Comparison of micro-F1 of four algorithms.

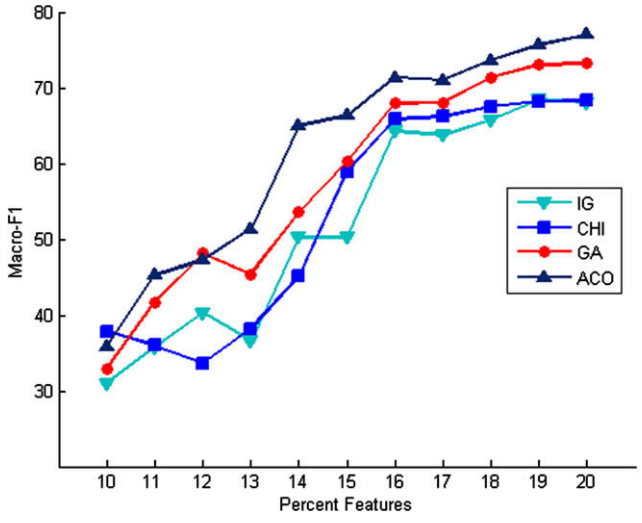

Fig. 5. Comparison of macro-F1 of four algorithms.

Table 6
Micro-F1 and macro-F1 of four algorithms

Feature selection algorithms	Macro-F1	Micro-F1
IG	69.8124	80.9482
CHI	70.8601	82.2097
GA	76.2685	86.3854
ACO	78.4224	89.0822

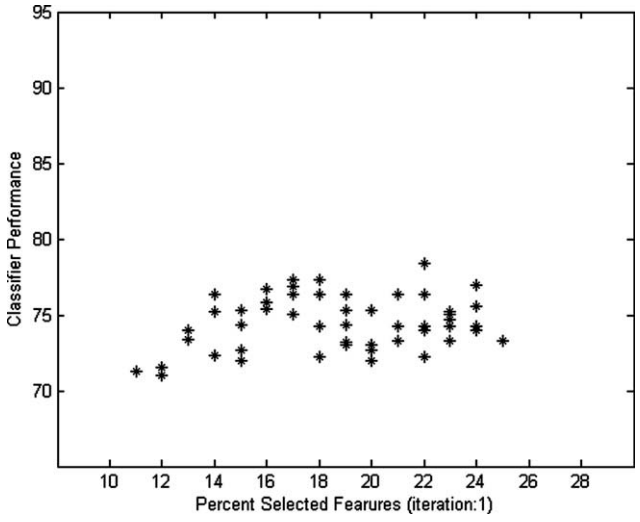

Fig. 6a. Iteration 1 of ACO on dataset Reuters-21578.

the performance of ACO. The selection of the parameters may be problem-dependent. The deposited pheromone, $\Delta\tau_i^k(t)$, calculated using Eq. (3), expresses the quality of the corresponding solution. ρ simulates the pheromone evaporation. Evaporation becomes more important for more complex problems. If $\rho = 0$, i.e. no evaporation, the algorithm does not converge. If pheromone evaporates too much (a large ρ is used), the algorithm often converged to sub-optimal solutions for complex problem. In many practical problems, it is difficult to select the best ρ without trial-and-error. α and β are also key factors in ACO for feature selection. For large

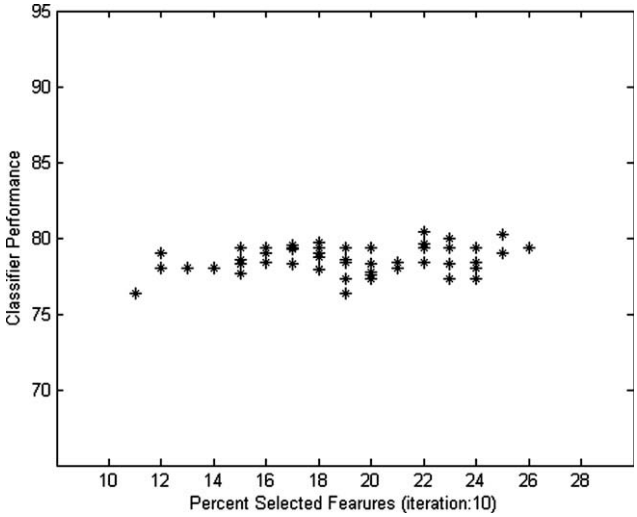

Fig. 6b. Iteration 10 of ACO on dataset Reuters-21578.

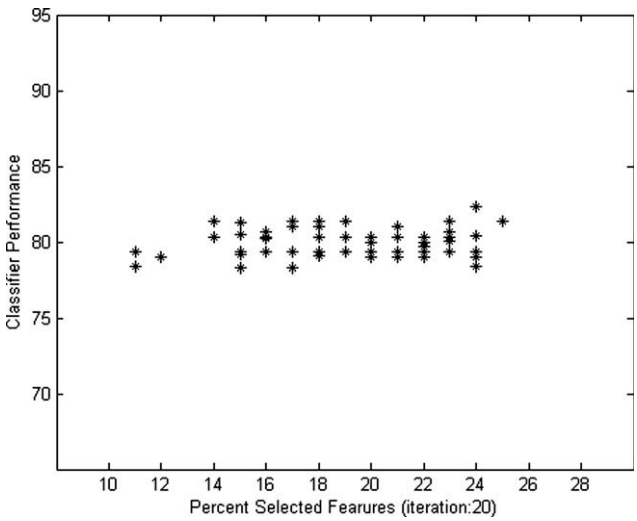

Fig. 6c. Iteration 20 of ACO on dataset Reuters-21578.

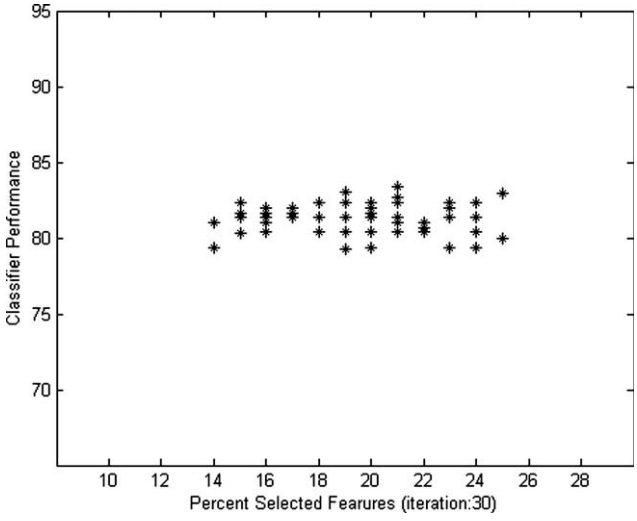

Fig. 6d. Iteration 30 of ACO on dataset Reuters-21578.

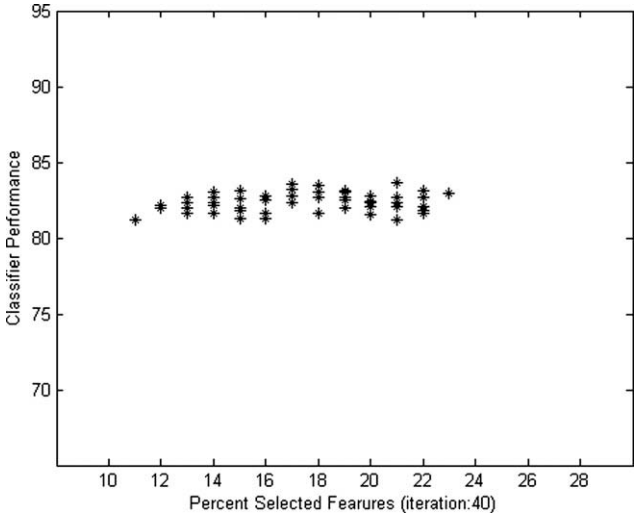

Fig. 6e. Iteration 40 of ACO on dataset Reuters-21578.

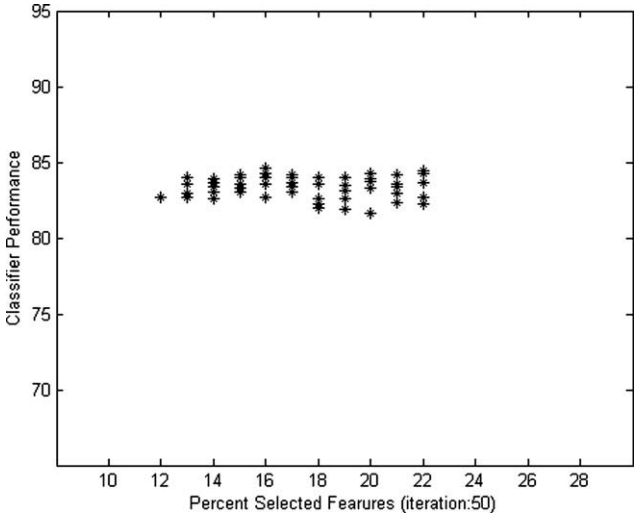

Fig. 6f. Iteration 50 of ACO on dataset Reuters-21578.

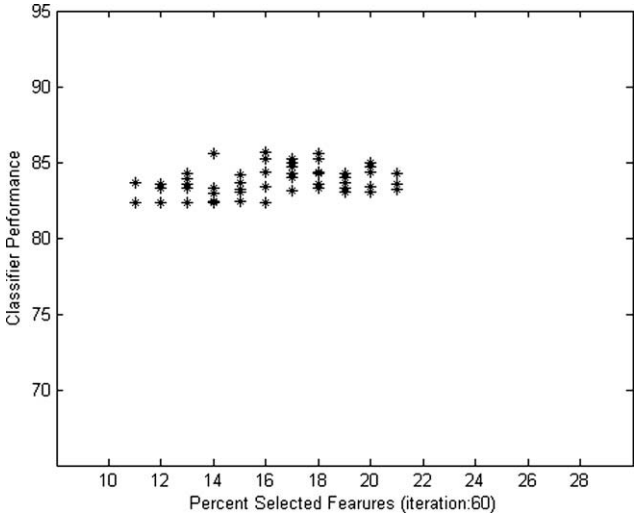

Fig. 6g. Iteration 60 of ACO on dataset Reuters-21578.

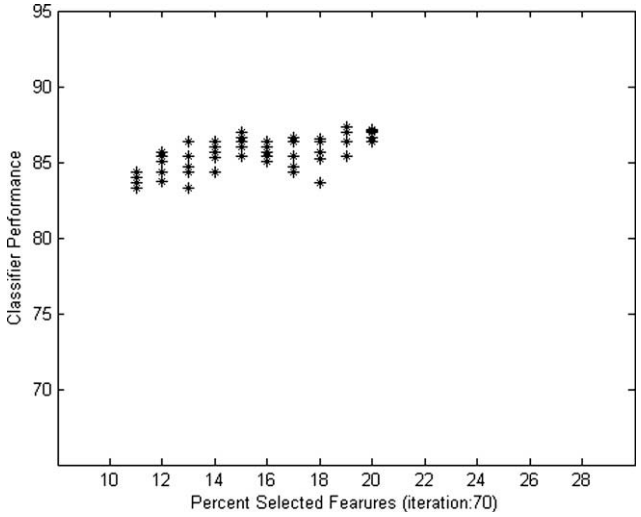

Fig. 6h. Iteration 70 of ACO on dataset Reuters-21578.

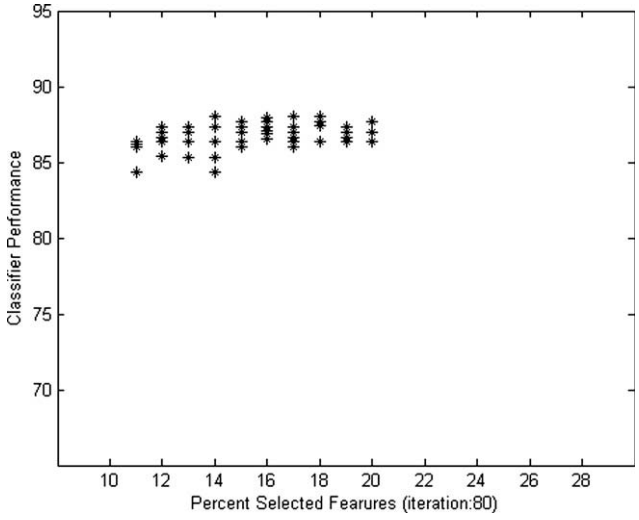

Fig. 6i. Iteration 80 of ACO on dataset Reuters-21578.

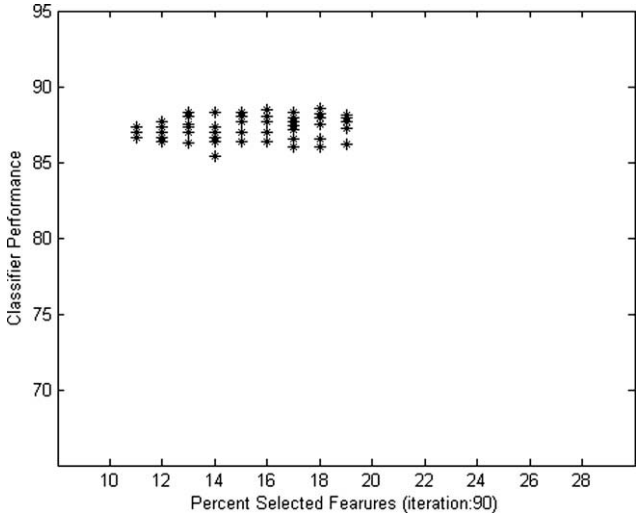

Fig. 6j. Iteration 90 of ACO on dataset Reuters-21578.

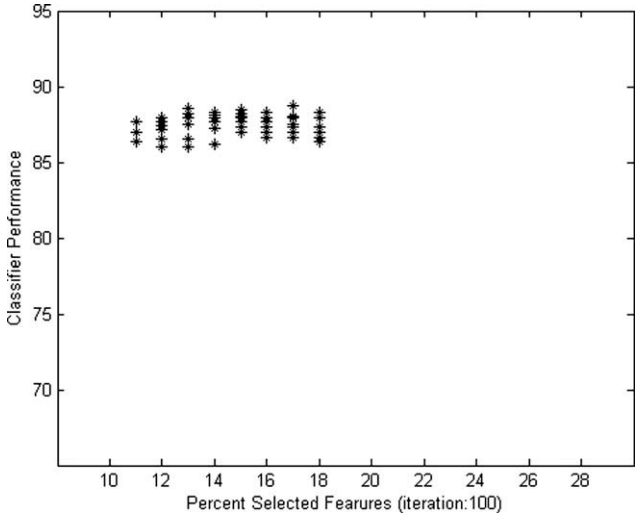

Fig. 6k. Iteration 100 of ACO on dataset Reuters-21578.

data sets, to speed up the computations of reducts, parallel algorithm may be employed (Susmaga, 1998; Susmaga, 2004).

In this paper, we apply ACO to find a minimal feature subset and, like classical genetic algorithm. In the proposed algorithm, the classifier performance and the length of selected feature subset are adopted as heuristic information. So, we can select the optimal feature subset without the prior knowledge of features. To show the utility of proposed algorithm and to compare it with information gain and CHI a set of experiments was carried out on Reuters-21578 dataset. The computational results indicate that proposed algorithm outperforms information gain and CHI methods since it achieved better performance with the lower number of features. To show the effectiveness of proposed algorithm, we use a simple classifier (nearest neighbor classifier) in that which can affect the categorization performance.

For future work, the authors intend to investigate the performance of proposed feature selection algorithm by taking advantage of using more complex classifiers in that. Another research direction will involve experiments with other kinds of datasets. Finally, the authors plan to combine proposed feature selection algorithm with other population-based feature selection algorithms.

Acknowledgement

The authors wish to thank the Office of Graduate studies of the University of Isfahan for their support.

References

Ani, A. A. (2005). Ant colony optimization for feature subset selection. *Transaction on Engineering, Computing and Technology, 4*, 35–38.

Basiri, M. E., Ghasem-Aghaee, N., & Aghdam, M. H. (2008). Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins. In *Proceedings of 6th European conference on evolutionary computation, machine learning and data mining in bioinformatics, LNCS* (Vol. 4973, pp. 12–23), Berlin, Heidelberg: Springer-Verlag.

Bins, J. (2000). *Feature selection from huge feature sets in the context of computer vision*. Ph.D. dissertation, Department Computer Science, Colorado State University.

Blum, C., & Dorigo, M. (2004). The hyper-cube framework for ant colony optimization. *IEEE Transaction on Systems, Man, and Cybernetics – Part B, 34*(2), 1161–1172.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. New York: Oxford University Press.

Caropreso, M. F., & Matwin, S. (2006). *Beyond the bag of words: A text representation for sentence selection* (pp. 324–335). Berlin: Springer.

Dorigo, M., & Caro, G. D. (1999). Ant colony optimization: A new meta-heuristic. In *Proceedings of IEEE Congress on Evolutionary Computing*.

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 243–278.

Dorigo, M. (1992). *Optimization, learning and natural algorithms*. Ph.D. dissertation, Dipartimento di Electtronica, Politecnico di Milano. Italy.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transaction on Systems, Man, and Cybernetics-Part B, 26*(1), 29–41.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. Chichester: John Wiley & Sons.

Engelbrecht, A. P. (2005). *Fundamentals of computational swarm intelligence*. London: John Wiley & Sons.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 1289–1305.

Gambardella, L. M., & Dorigo, M. (1995). Ant-Q: A reinforcement learning approach to the TSP. In *Proceedings of the 12th international conference on machine learning* (pp. 252–260).

Gambardella, L. M., & Dorigo, M. (1996). Solving symmetric and asymmetric TSPs by ant colonies. In *Proceedings of IEEE international conference on evolutionary computation* (pp. 622–627).

Jensen, R. (2005). *Combining rough and fuzzy sets for feature selection*. Ph.D. dissertation, School of Information, Edinburgh University.

Kanan, H. R., Faez, K., & Aghdam, M. H. (2007). Face recognition system using ant colony optimization-based selected features. In *Proceedings of the first IEEE symposium on computational intelligence in security and defense applications* (pp. 57–62), USA.

Kim, H., Howland, P., & Park, H. (2005). Dimension reduction in text classification with support vector machines. *Journal of Machine Learning Research, 6*, 37–53.

Kml, L., & Kittler, J. (1978). Feature set search algorithms. In Chen, C. H. (Ed.), *Proceedings of Pattern Recognition and Signal Processing, Sijhoff and Noordhoff*.

Leguizamon, G., & Michalewicz, Z. (1999). A new version of ant system for subset problems. In *Proceedings of IEEE Congress on Evolutionary Computation* (pp. 1465).

Liu, B., Abbass, H. A., & McKay, B. (2004). Classification rule discovery with ant colony optimization. *IEEE Computational Intelligence Bulletin, 3*(1), 31–35.

Maniezzo, V., & Colorni, A. (1999). The ant system applied to the quadratic assignment problem. *IEEE Transaction on Knowledge and Data Engineering, 11*(5), 769–778.

Mitchell, T. (1996). *Machine learning*. McCraw Hill.

Mladenić, D. (2006). Feature selection for dimensionality reduction. Subspace, latent structure and feature selection, statistical and optimization, perspectives workshop, SLSFS 2005, Bohinj, Slovenia, Lecture Notes in Computer Science. Springer (Vol. 3940, pp. 84–102).

Montemanni, R., Gambardella, L. M., Rizzoli, A. E., & Donati, A. V. (2002). *A new algorithm for a dynamic vehicle routing problem based on ant colony system*. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, Technical Report IDSIA-23-02.

Pawlak, Z. (1991). *Rough sets: Theoretical aspects of reasoning about data*. Dordrecht: Kluwer Academic Publishing.

Punch, W. F., Goodman, E. D., Pei, L. C. S. M., Hovland, P., & Enbody, R. (1993). Further research on feature selection and classification using genetic algorithms. In *Proceedings International Conference on Genetic Algorithms* (pp. 557–564).

Raymer, M., Punch, W., Goodman, E., Kuhn, L., & Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computing, 4*, 164–171.

Rijsbergen, C. J. van. (1979). *Information retrieval* (2nd ed.). London, UK: Butterworth.

Salton, G., & Buckley, C. (1987). *Term-weighting approaches in automatic text retrieval*. Cornell University Ithaca, NY, USA, Technical Report TR87-881.

Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., & Wang, Z. (2007). A novel feature selection algorithm for text categorization. *Expert Systems with Applications, 33*(1), 1–5.

Siedlecki, W., & Sklansky, J. (1988). On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence, 2*(2), 197–220.

Siedlecki, W., & Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters, 10*(5), 335–347.

Srinivas, M., & Patnik, L. M. (1994). *Genetic algorithms: A survey*. Los lamitos: IEEE Computer Society Press.

Stützle, T., & Hoos, H. H. (1997). MAX-MIN ant system and local search for the traveling salesman problem. In *Proceedings of IEEE international conference on evolutionary computation* (pp. 309–314).

Susmaga, R. (1998). Parallel computation of reducts. In *Proceedings of the first international conference on rough sets and current trends in computing* (pp. 450–457). London: Springer.

Susmaga, R. (2004). Tree-like parallelization of reduct and construct computation. *Proceedings of the international conference on rough sets and current trends in computing*. Springer.

The reuters-21578 text categorization test collection. <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>.

Wang, X., Yang, J., Teng, X., Xia, W., & Jensen, R. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters, 28*(4), 459–471.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the 14th international conference on machine learning* (pp. 412–420).

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems, 13*, 44–49.

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).