International Journal of Innovative
Computing, Information and Control
Volume **12**, Number **6**, December **2016**

ICIC International ©2016 ISSN 1349-4198
pp. 1851–1863

A TWO-STAGE IMPROVED ANT COLONY OPTIMIZATION BASED FEATURE SELECTION FOR WEB CLASSIFICATION

Jun Xu and Guangyao Li

College of Electronics and Information Engineering
Tongji University
No. 4800, Caoan Road, Jiading District, Shanghai 201800, P. R. China
cs_xujun@qq.com; lgy@tongji.edu.cn

Received May 2016; revised September 2016

Abstract. *Feature selection is an essential task for web classification, wherein an enormous number of features representing words or terms create fearful challenges to the efficiency and effectiveness of classifiers. It reduces the number of features by removing irrelevant and redundant data. In this paper, a novel two-stage improved ant colony optimization (T-IACO) based feature selection is presented. Irrelevant and redundant features are removed separately in two stages. In the first stage, each feature is ranked by the information gain scores that represent the relevance between features and categories, and we chose the top of the scores that means the irrelevant features are removed. In the second stage, we improve ant colony optimization algorithm to urge ant biasing to remove the redundant features from the reduced features subset that gets from the first stage. In this improved ant colony optimization, ant traversal is not only guided by pheromone concentration and heuristic desirability, but also a relevance function that presents the relevance between the features and the constructed subsets. If the next selection feature is redundant with one of the features that had been selected in the constructed result, the relevance value will be very low to reduce the choosing probability of this feature, and the relevance will be updated after every choice. The performance of the proposed algorithm was evaluated using Naive Bayes classifier and compared with the standard feature selection techniques. Experiments verified that the algorithm provides a better feature subset.*
Keywords: Ant colony algorithm, Feature selection, Web classification, Information gain

1. **Introduction.** Web classification is a key technology to solve the increasingly large web content resources. It has a lot similarity to text classification; for example, the bag of words model is widely used too, but more complicated and diverse. There is a critical problem in the process of web classification, that is, the efficiency of machine learning based classification methods will be greatly reduced because of the high dimension of the feature vector. Most of these features are redundant and irrelevant, and even some noise can degrade the classification performance. Therefore, feature selection (FS) is needed to identify and select a useful subset of features from a larger set of often mutually redundant, irrelevant, features with different associated importance [1].

The objective of feature selection is to identify relevant underlying features and reduce redundancy in the information without sacrificing predictive accuracy. The FS approaches can generally be divided into two groups: filter and wrapper [2]. The filter approach operates independently of learning algorithm. These FS methods are used as a preprocessing of learning algorithm, by setting a feature evaluation method, scoring the features wherein the highly ranked features are selected. Some popular filter methods are document frequency (DF) [3], mutual information (MI) [4], information gain (IG)

[5]. Hall [7] proposed an FS method based on the minimum correlation between features, while maximum relevance between features and class. Dash and Liu [8] proposed a consistency-based algorithm. These methods are widely used because of the high efficiency. The wrapper approach treats the classification performance as the evaluation criterion, and finds the optimum subset which achieves the best performance. Wrapper method can produce better feature sets, but it is significantly inferior to filter method on the time consumption.

It is time exhausted to find an optimal subset when it confronts a large number of features. So the heuristic algorithms have been proposed in recent years, namely: particle swarm optimization (PSO) [9], genetic algorithm (GA) [10], ant colony optimization (ACO) [11]. These algorithms find an approximate optimal feature subset through the constant iteration.

The origin of ant colony optimization is from Dorigo and Blum [12] who proposed an ant system that simulates the ant foraging, initially to solve the traveling salesman problem (TSP). Subsequently, it got rapid development and widely used in a variety of NP hard problems. In recent years the method is applied to feature selection; Ahmed [13] proposed to use a hybrid evaluation measure that is able to estimate the overall performance of subsets as well as the local importance of features in ACO algorithm. Aghdam et al. [6] applied ACO algorithm to text feature selection and took each valid word in the text as a feature to be selected, then structured these features into a graph to find a best subgraph. Kashef and Nezamabadi-Pour [14] proposed an advanced binary ant colony algorithm that each feature node is divided into two sub node express selected or not selected, different from other ACO algorithms, the termination condition of this algorithm is that every feature node is treated and because this, it can find a better subset for classification efficiency. Forsati et al. [11] put forward an improvement that not only considered current iterative information but also the previously traversed edges in the earlier executions to adjust the pheromone values appropriately and prevent premature convergence. Sivagaminathan and Ramakrishnan [15] combined the ant colony algorithm with artificial neural network and yielded a promising result in medical data set. Nemati et al. [16] produced a better performance by making a hybrid of GA and ACO, which used ACO and GA to generate feature subsets in parallel and then evaluated those to find a best subset.

This paper presented a two-stage feature selection method based on improved ant colony by separately considering the irrelevant features and redundant features. In the first stage, we considered removing the irrelevant features by calculating the information gain value and eliminating features through the information gain value ranked. In the second stage, we focused our work on redundant information. We proposed a new probability transfer formulation in ant colony algorithm, which compelled ants to prefer selecting the features that are having a small correlation with the last round selected feature, to improve the convergence speed of algorithm and remove the redundant features. Our algorithm is totally different from traditional heuristic algorithms because we considered the irrelevant and redundant features individually which led to a better result compared to them. Experimental results show that our algorithm can effectively reduce the number of features with a high classification accuracy.

The rest of this paper is organized as follows. Section 2 describes the ant colony optimization for feature selection. Our proposed algorithm is introduced in Section 3. Section 4 reports the results of our experiments. Finally, Section 5 concludes this paper.

2. **Ant Colony Optimization for Feature Selection.**

Ant colony optimization is a metaheuristic algorithm introduced by Dorigo and Blum [12] who proposed Ant System in

the early 1990s to solve the traveling salesman problem (TSP). In 1999, Corne et al. [17] attributed the various ant system to the framework of ant colony optimization algorithm and standardized description of the algorithm. The algorithm has very strong robustness and can better obtain an approximate global optimal solution, capable of parallel and distributed computing, easy to combine with other algorithms. Therefore, it has been widely used.

The basic idea of ACO algorithm comes from the observation of the foraging behavior of ant. Ants do not have a fully developed visual perception system but they can transmit information between individuals through the pheromone. When ants find food, it will lay some pheromone to mark the path. The pheromone concentration is used to represent the merits of the path, and other ants tend to move in the direction with high pheromone concentration. When there are more and more ants traversal in a path, the probability of ants choosing this path will increase. Because those produce a positive feedback, most of ants will move toward a path to handing food. Ants are routing through the perception of pheromone concentration so as to find the shortest path to handing food; scholars proposed the artificial ants on the basis of their natural counterparts to solve optimization problems.

The similarities between artificial ants and natural ants are: they both lay pheromone on the edges of the graph; the probability to choose the next path is depended on the concentration of pheromone; the pheromone will evaporate with the iteration (time).

The artificial ants also own some distinct features: artificial ants have a certain memory ability, can remember the nodes that have been visited; artificial ants use a certain algorithm to find the next path, rather than blind; artificial ants often use some methods to speed up the convergence, for example, releasing extra pheromone on the global optimal path; artificial ants are usually combined with some local search methods to improve the path quality.

2.1. **Graph representation for feature selection.** Feature selection is one of the applications of the subset selection problem which is included in discrete optimization problems. Those problems need to search all possible features subsets, means the size of solution domain is:

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

where n is the number of features (dimensionality) and k is the size of the current feature set [18]. These NP-hard problems usually involve heuristic and random search strategies to find a near optimal solution.

The feature selection of web classification requires finding a minimal feature subset of size s ($s < n$) in a given feature set of size n while preserving a suitable accuracy. To apply an ACO algorithm to solving this problem, it must be represented as a graph. In addition, feature selection in this problem is no order requirement and the size of the subset does not need to be exactly the same.

In the graphic representation, nodes mean features while the edges between them represent the choice of the next feature. Each value of edge to a specific node is the same, so the choice of the next feature is stored in node but not edges to save storage space. Therefore, feature selection problem is transformed into an ant traversal through the graph where a minimum number of nodes are visited that meets the evaluation function.

2.2. **Probabilistic transition rule.** Because the feature selection does not require the feature to be in order, it is only needed to find a subset of nodes (features). The feature subset is constructed from an empty set, in which the next feature is selected that is

determined by a probability formula, and the feature is added until the evaluation function (stop criterion) is satisfied. The probability that ant k will select node i at the time instance t is given in:

$$P_k^i(t) = \begin{cases} \dfrac{(\tau_i(t))^\alpha \times (\eta_i)^\beta}{\sum_{u \in Allowed(k)} (\tau_u(t))^\alpha \times (\eta_u)^\beta} & i \in Allowed(k) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $Allowed(k)$ is the feasible (have not selected) feature that can be visited by ant; $\tau_i(t)$ and $\eta_i(t)$ are respectively the pheromone value and heuristic desirability associated with feature i; α and β are two constants, respectively, express the weighted value of pheromone and heuristic desirability, and are used to determine the importance of these two variables.

The probability transfer formula achieves the optimal balance between pheromone and desirability heuristic by adjusting α and β. When $a = 0$, the concentration of pheromone is invalid for the feature selection, and the algorithm will degenerate into a greedy algorithm. When $b = 0$, the heuristic desirability is failed, ants will not consider the pros and cons when they select next feature.

2.3. **Heuristic desirability.** Heuristic desirability is a quality standard to each edge. A heuristic value, η, act as an attractiveness of the features, improves the ability of exploiting the search space and helps to find the optimum. Ant colony algorithm was first used to solve the travelling salesman problem, using the reciprocal of the distance between the cities (nodes) as the heuristic desirability to guide the choice of ants. In feature selection problems, there is lots of heuristic information, such as document frequency, mutual information, and information chi square test.

2.4. **Pheromone update rule.** After all ants have completed their solutions in each iteration, pheromone update rule on all features is triggered. Ants deposit a quantity of pheromone on each feature that it has selected. And in order to expand the search area and prevent the algorithm converging fast to a local optimal solution, every path pheromone will evaporate in each iteration. In addition, we will strengthen the pheromone concentration in the iterative optimal solution or the global optimal solution with the purpose of speeding up the ants to find the preeminent solutions. Therefore, ant colony algorithm generally abides by the following pheromone update rules:

$$\tau_i(t+1) = (1-\rho)\tau_i(t) + \sum_{k=1}^{n} \Delta\tau_i^k(t) + \Delta\tau_i^g(t) \tag{3}$$

where n is the number of ants at each iteration and $\rho \in (0,1)$ is evaporation rate (the first proposed ant system is usually set to 0.5, maximum and minimum ant system is generally set to 0.02), $\Delta\tau_i^k(t)$ is the deposit pheromone of each ant release in its visited features. $\Delta\tau_i^g(t)$ means augmenting the pheromone concentration in global best feature subset. This helps to the traversal of ants around the optimal solution in the next iteration.

Each ant k deposits pheromone in nodes i at time instance t is generally defined as:

$$\Delta\tau_i^k = \begin{cases} \phi \cdot \gamma\left(S^k(t)\right) + \dfrac{\psi \cdot \left(n - \left|S^k(t)\right|\right)}{n} & \text{if } i \in S^k(t) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $S^k(t)$ is the feature subset found by ant k at iteration t, and $\left|S^k(t)\right|$ means its length. $\gamma\left(S^k(t)\right)$ is the classifier performance with $S^k(t)$. The scores represent the classifier performance and the feature subset length, we tend to deposit more pheromones in the

subset that show a better performance with shorter length. ϕ and ψ are two constants to control the importance of the classifier performance and feature subset length.

3. **Proposed Feature Selection Algorithm.** As mentioned earlier, in this paper we separately consider the irrelevant features and redundant features and propose a two-stage improved ant colony optimization algorithm (T-IACO) for feature selection. The first stage corresponds to eliminate the irrelevant parts. We assign scores to all features based on information gain, and remove those features with low value to produce a reduced subset. The second stage focuses on eliminating redundance in reduced features set. We proposed an improved ant colony algorithm to make ants bias toward choosing the features which have a minimum correlation with already chosen set. Like ACO-based feature selection algorithms, the problem need to be defined as a fully connected graph where nodes represent features, with edges between denoting the choice of the next feature in this stage.

3.1. **Stage 1: Irrelevance removed with information gain.** We use the information gain (IG) method as the first stage to remove the irrelevant features. This value owns a fantastic performance in judging the correlation between features and categories and is highly popular in feature selection community. The IG score for each feature T to the class C is [5]:

$$\begin{aligned} IG(C|T) &= H(C) - H(C|T) \\ &= -\sum_{i=1}^{n} P(C_i)\log_2 P(C_i) + P(t)\sum_{i=1}^{n} P(C_i|t)\log_2 P(C_i|t) \\ &\quad + P(\hat{t})\sum_{i=1}^{n} P\left(C_i|\hat{t}\right)\log_2 P\left(C_i|\hat{t}\right) \end{aligned} \tag{5}$$

$H(C)$ means the entropy of class C, $H(C|T)$ means the conditional entropy of the specified feature T, P represents the corresponding probability. The information gain value of all N features can be calculated in the time complexity of $O(N)$. The features are then sorted in the decreasing order and select these top ranked features to the second stage, or just removed these features with information gain value less than a specific point. So, we get a reduced features set which help the second stage will not time break down and can just focus on redundance.

3.2. **Stage 2: Redundance removed with improved ACO algorithm.** To remove the redundance from the reduced subset generated by the first stage, we model it into an ACO-suitable problem and put forward an improved ACO algorithm (IACO) with a new probability transfer rule to impel ants to choose the features that are independent with the feature set selected already. In our algorithm, ants traversal is not only guided by pheromone concentration and heuristic desirability but also a relevance function R which means relevance between the next chosen features and the already chosen set; the detail process can be described in the flowchart shown in Figure 1.

3.2.1. *Improved probability transfer rules.* In the classical ant colony algorithm, ants only consider the pheromone and heuristic information when they select the next feature. Our algorithm is designed to remove redundant information without considering irrelevant information in this stage. Therefore, in this ACO algorithm, we design a relevance function R to evaluate redundance of every subset and guide ants behavior in their traversal. We

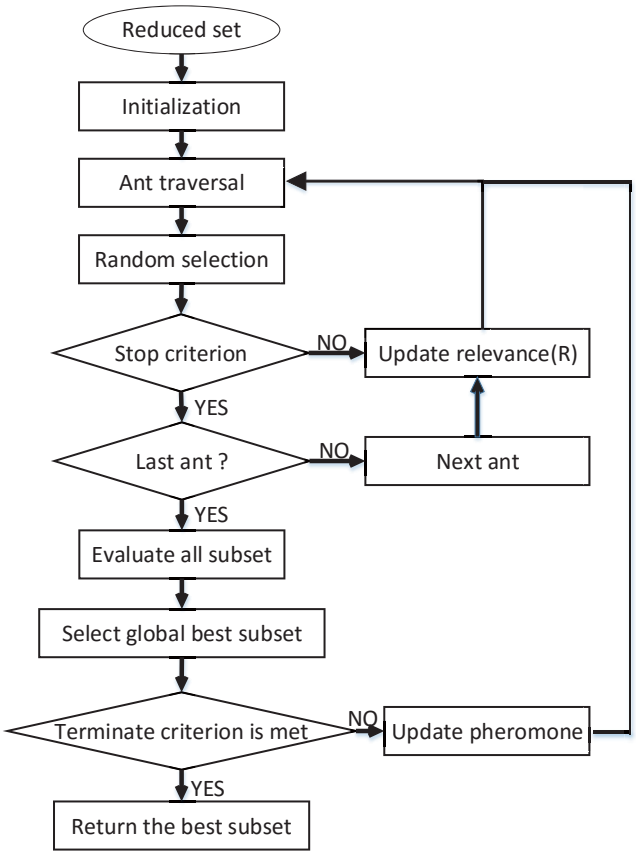

FIGURE 1. The structure of improved ACO algorithm

achieve finding minimum redundance features set by adding R into the probability transfer rule. The rule is:

$$P_k^i(t) = \begin{cases} \dfrac{(\tau_i(t))^\alpha \times (\eta_i)^\beta}{\sum_{u \in J^k} (\tau_u(t))^\alpha \times (\eta_u)^\beta} \times R_k^i(t,m) & i \in J^k \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The probability that ant k will select feature i at the time instance t is introduced in (2). In this paper, the variable of τ_i is set and updated as same as we have discussed in (2) and we try two methods to denote the heuristic desirability that will be introduced as follows. The $R_k^i(t,m)$ is our relevance function defined as:

$$R_k^i(t,m) = \begin{cases} \mu R_k^i(t,m-1)) \Big/ \left(1 + \log_2 \dfrac{P(i,last)}{P(i)P(last)}\right) & \log_2 \dfrac{P(i,last)}{P(i)P(last)} > 0 \\ \mu R_k^i(t,m-1)) & \log_2 \dfrac{P(i,last)}{P(i)P(last)} = 0 \\ \mu R_k^i(t,m-1)) \times \left(1 - \log_2 \dfrac{P(i,last)}{P(i)P(last)}\right) & \log_2 \dfrac{P(i,last)}{P(i)P(last)} < 0 \end{cases} \tag{7}$$

where $R_k^i(t,m)$ means relevance between the feature i (next select feature) and the selected subset in the m-th selection (feature i is the m-th selection of ant and the size of the current subset is $m-1$) for ant k in time instance t. $P(i)$ is the probability of the feature i appearing in category. 'last' is the $(m-1)$-th selected feature, which means the selected feature before selecting feature i. $P(last)$ is its probability. After ant selected, in order

to continuously represent the relevance between feature i and the selected set, $R_k^i(t,m)$ needed to be updated from $R_k^0(t,m)$ to $R_k^n(t,m)$. When ants select a feature, we calculate the relevance between this feature and all others by $\frac{P(i,last)}{P(i)P(last)}$ and update the R for next selection. For each feature, the greater value of $\frac{P(i,last)}{P(i)P(last)}$, the larger relevance between the feature and the last selected counterpart, so we use this to decide to augment or reduce R, and we can obtain the relevance between each feature and the selected subset to help us find the minimum redundance subset.

3.2.2. *Heuristic desirability.* In this algorithm, we try two methods to represent the heuristic desirability including information gain (IG) and Pearson product-moment correlation coefficient (PPMCC).

Information gain has been introduced in (5), this value mirrors the contribution of each feature to the whole system and shows a decent effectiveness in ant traversal.

Pearson product-moment correlation coefficient is the covariance of the two variables divided by the product of their standard deviations, it is a measure of the linear correlation between two variables X and C. This method is able to weigh the correlation between feature X and category C of a given training set. The heuristic desirability of feature X is defined as:

$$\eta_x = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(C_i - \overline{C}\right)}{\sqrt{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2} \sqrt{\sum_{i=1}^{n} \left(C_i - \overline{C}\right)^2}} \tag{8}$$

where X_i and C_i are the value of features and its category, respectively; the variables $\overline{X}$ and $\overline{C}$ represent the mean values of X_i and C_i. If the features X and the category C are completely correlated, then the value would be 1 or -1. And the value would be 0 if they are completely uncorrelated.

3.2.3. *Terminate criterion.* There are two main criteria to stop our algorithm. The first is to set a fixed classification efficiency; when the classification efficiency achieved or exceeded, stop the iteration. The other is to complete the iteration through the parameter setting, mainly including the number of iterations and ants. In our experiment, we use both of these two methods to check our algorithm.

3.2.4. *Proposed improved ACO algorithm.* The proposed IACO algorithm for removing redundant features can be described by the flowchart shown in Figure 1, which is described as follows.

Step 1: Count $F(x,y)$, the number of arbitrary two features x and y appear together, calculate $P(x,y)$, the probability of the corresponding.

Step 2: Initialize the parameters of IACO, including the numbers iteration and ants *iter* and m, the tunable parameters α, β, ϕ, ψ and ρ, the initial pheromone concentration τ_0, the tunable coefficient μ, the heuristic information, η, which are described in (5) and (8).

Step 3: Construct candidate solutions from an empty set. Select next feature and update relevant function, R, using (7) for each ant until stop criterion is met. The stop criterion is modified by (4), we set a series of ϕ, ψ and choose a minimum score as stop criterion.

Step 4: Score every selected subset based on (4).

Step 5: Using (3) to evaporate pheromone in every feature, update it in all traversed features and enhance it in global best solution.

Step 6: Go back to Step 3 until terminate criterion is met, and then return the global best solution.

4. **Experiment Results.** A series of experiments was conducted to illustrate and validate the effectiveness of the proposed algorithm. All experiments were performed on a laptop with Core(TM)2 Duo 2.60GHz CPU and implemented using python. The first dataset, Chinese web pages, consists of four categories news more than forty thousand web pages in total, which are finance, war, mobile and sports, crawling from www.163.com by web spider. We transfer this dataset to two categories prove meliority of T-IACO in binary classification problem. The financial category pages are defined as the category to be classified, and the other three categories are mixed as the other category. The second dataset, English texts from 20 newsgroup benchmark dataset, was downloaded from the UCI Machine Learning Repository[1] to demonstrate the effectiveness of the proposed algorithm in multi classification problem.

To validate the results obtained by the proposed algorithm, we compared with information gain (IG) that widespread used, ant colony optimization (ACO) and EACO [19] which are reported to be a very strong algorithm in feature selection, ACO based on reduced subset by our first stage (T-ACO). All experiments are judged by the evaluation function defined as follows using Naive Bayes classifier.

4.1. **Evaluation methodology.** The performance of the proposed algorithm was evaluated with the well known metrics precision, recall, accuracy, F-measure and feature-reduction. Precision for a class is the number of true positives (the number of items correctly labelled as belonging to the positive class) divided by the total number of elements labelled as belonging to the positive class as in (9). Recall is defined as the number of true positives divided by the total number of elements that actually belong to the positive class as in (10). Let TC, TN, FC and FN define as follows:

TC – the number of C pages classified into C
FC – the number of *non-C* pages classified into C
FN – the number of C pages classified into *non-C*
TN – the number of *non-C* pages classified into *non-C*

$$Precision = TC/(TC + FC) \tag{9}$$

$$Recall = TC/(TC + FN) \tag{10}$$

F-measure combines the precision and recall to measure the classifier's performance. It is defined as the harmonic mean of precision (P) and recall (R) as:

$$F = 2PR/(P + R) \tag{11}$$

Classification accuracy (ACC) is the percentage of pages rightly classified:

$$ACC = \frac{Number\ of\ pages\ correctly\ classified}{Total\ number\ of\ the\ test\ pages} \times 100 \tag{12}$$

Another evaluation criterion is used for comparing the rate of feature reduction

$$FR = \frac{n - r}{n} \tag{13}$$

where n is the total number of features and r is the number of selected features. FR means the average feature reduction. The more features are reduced, the more it is close to 1, and the classifier performance is better.

In this paper, we calculate the mean of the scores computed for each individual category to get the average values. It weighs equally in all the classes irrespective of the number

[1]http://archive.ics.uci.edu/ml/datasets.html

of documents present in it. The average precision and recall are defined as:

$$\bar{P} = \frac{\sum_{i=1}^{|C|} P^i}{|C|} \tag{14}$$

$$\bar{R} = \frac{\sum_{i=1}^{|C|} R^i}{|C|} \tag{15}$$

4.2. **Parameter setting.** The parameters of ant colony algorithm mainly include: the number of ants, the number of iterations, evaporation coefficient, adjust coefficient. It is hard to find a set of parameters that are used for all the data and methods. Generally, increasing the number of ants can improve the global search capability and the stability of the algorithm while weakening the positive feedback of the information and augmenting the execution time. The number of iterations is a balance of running time and subset performance; when the number of iterations reaches a threshold value, the optimal subset will almost no longer change. Evaporation coefficient is directly related to the global searching ability and the convergence rate of the ant colony algorithm. When this parameter is large, the randomness is weakened while the convergence is accelerated, and vice versa. The adjust coefficients are used to change the randomness, adjust the convergence rate, and prevent the algorithm into a local optimum.

For the proposed algorithm, the reduced features subset are set to 500 in the first stage. In the second stage, the maximum iterations and total number of ants are set to 200 and 50 in the first experiment while set to 150 and 500 in second experiment. The evaporation coefficient, ρ is 0.2. The initial pheromone intensity (τ_0 is set to 5). The α and β are determined the importance of pheromone and the heuristic information; we set α to 0.2 and adjust β in the range of $[0, 1]$ and set it to 0.8 finally.

4.3. **Results and analysis.** Our first experiment is based on Chinese web pages. Figure 2 and Figure 3 display the better performance with F-measure of our algorithm when fixing the size of features number in the first dataset, where T-ACO means doing classic ACO algorithm based on the reduced feature set produced by our first stage. Comparing T-IACO to T-ACO, we can find that T-IACO is more suitable and excellent for a reduced set. As T-IACO and T-ACO are based on a processed set, they are easy to get a better result at the beginning of iteration compared with ACO. Figure 4 shows the relationship between the accuracy and the number of features under different algorithms. It proved

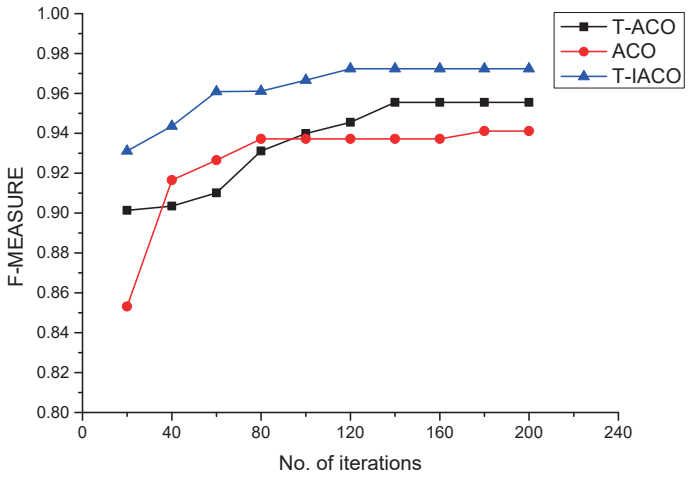

FIGURE 2. The F-measure performance with 30 features subset

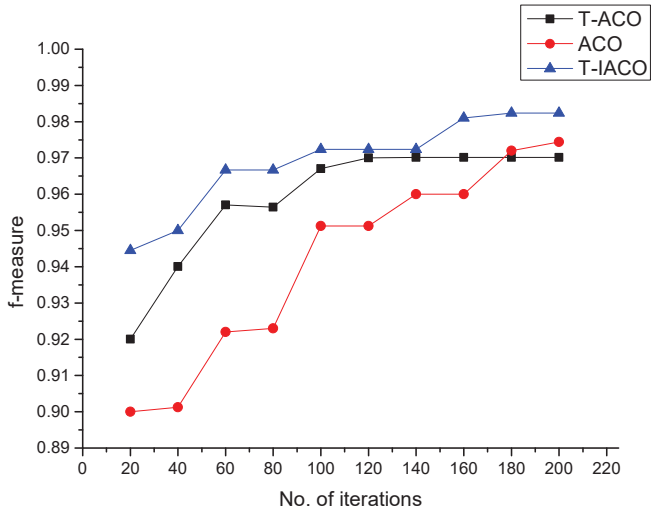

FIGURE 3. The F-measure performance with 60 features subset

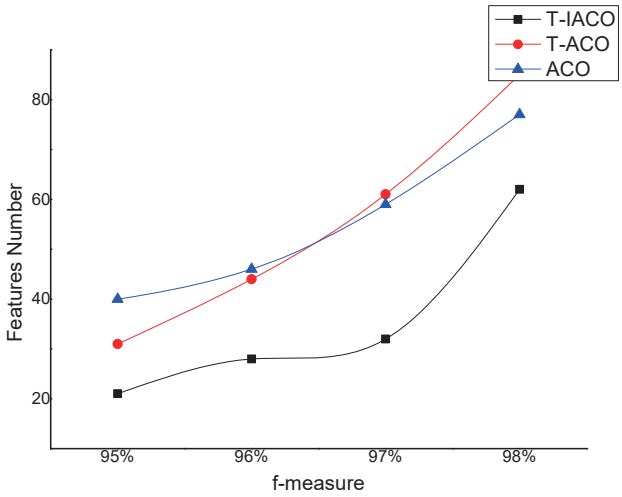

FIGURE 4. The relationship between F-measure and the size of set

that T-IACO can get a better feature reduction and the second stage IACO is obviously enhanced than utilizing classic ACO in this stage.

Table 1 and Table 2 show the best result with criterion function (4), where ϕ and ψ are 0.9 and 0.1, n is 200, $\gamma\left(s^k(t)\right)$ return F-measure as classification performance. We can conclude from the tables that proposed T-IACO can obtain a better feature reduction with a good classification accuracy, compared to other algorithms.

Our second experiment is aimed to compare with EACO. Table 3 compared results of different algorithms in the second dataset which use 90% the documents with overfitting for testing. The max iterations of these evolutionary algorithms were set to 150. We can obviously find that our algorithm has obvious advantages in recall, although the performance of our precision is inferior. Table 4 shows the average precision, recall and F-measure using (15).

Figure 5 shows the time consumption to execute the algorithm also increases with the number of features selected with one iteration. The slope of each line reflects the time

TABLE 1. Classification performance with Chinese web pages using IG as heuristic desirability

Algorithm	IG	ACO	T-ACO	T-IACO	ACO	T-ACO	T-IACO
Iterations		40	40	40	80	80	80
Size	178	101	69	58	138	65	62
FR	11%	49.5%	65.5%	71%	31%	67.5%	69%
Precision	0.8515	0.9545	0.9468	0.9773	0.9782	0.9674	0.9778
Recall	0.9663	0.9438	0.9820	0.9663	0.9920	0.9920	0.9887
F-measure	0.9053	0.9491	0.9641	0.9717	0.9849	0.9896	0.9832
ACC	0.8152	0.9577	0.9724	0.9824	0.9889	0.9894	0.9894

TABLE 2. Classification performance with Chinese web pages using PPMCC as heuristic desirability

Algorithm	IG	ACO	T-ACO	T-IACO	ACO	T-ACO	T-IACO
Iterations		40	40	40	80	80	80
Size	178	123	59	48	128	71	42
FR	11%	38.5%	71.5%	76%	36%	64.5%	69%
Precision	0.8815	0.9506	0.9568	0.9773	0.9660	0.9680	0.9788
Recall	0.9463	0.9338	0.9860	0.9663	0.9820	0.9920	0.9860
F-measure	0.9129	0.9421	0.9641	0.9717	0.9739	0.9798	0.9824
ACC	0.8152	0.9477	0.9524	0.9844	0.9782	0.9824	0.9866

TABLE 3. Performance of ACO, EACO, T-IACO

Categories	ACO		EACO		T-IACO	
	Precision	Recall	Precision	Recall	Precision	Recall
alt.atheism	0.7892	0.8171	0.7927	0.8257	0.7892	0.8403
comp.graphics	0.7363	0.9257	0.7416	0.9291	0.7251	0.9564
rec.autos	0.6721	0.9127	0.6932	0.9291	0.7012	0.9311
rec.sport.hockey	0.9375	0.8279	0.9675	0.8347	0.9532	0.8521
sci.crypt	0.9642	0.6792	0.9527	0.6598	0.9310	0.7012
talk.politics.mideast	0.9531	0.6712	0.9614	0.6924	0.9324	0.7112

TABLE 4. Comparison of macro averaged precision and recall for second dataset

FS algorithm	Precision	recall	F-measure
IG	0.8304	0.7992	0.8145
ACO	0.8421	0.8056	0.8234
EACO	0.8515	0.8118	0.8312
T-IACO	0.8304	0.8306	0.8304

required to select one more feature; it is obvious that the time taken in our algorithm is far less to EACO, while about a third to ACO because we use a reduced set.

5. **Conclusion.** Feature selection is a significant task which can crucially affect the performance of web classification. Conventionally, the web or text classification community either just reduces irrelevant data by employing feature ranking metrics or merely uses feature subset selection algorithm that always time exhausted. In this paper, we present

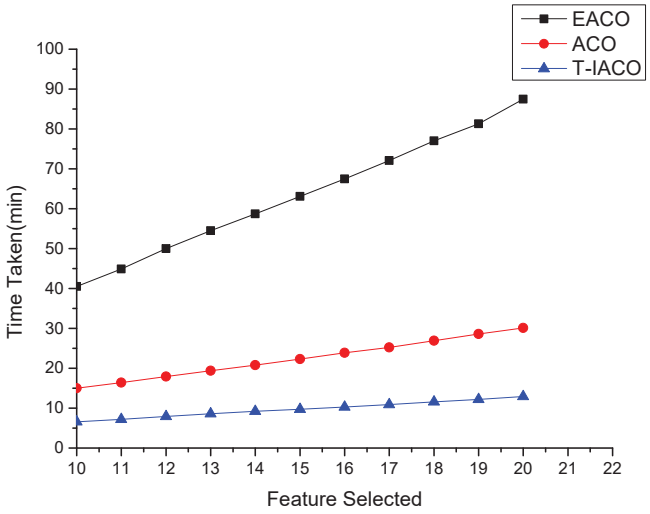

FIGURE 5. The comparison of time consumption

a two-stage improved ant colony optimization based feature selection, which separately considers the irrelevance and redundance, not only can effectively find an admirable feature subset, but also has a strong search capability with reasonable time. We remove the irrelevant feature in first stage by information gain, and achieve reduced redundance by improving the ACO algorithm to urge ants to choose the features that have minimum relevance with already selected subset. This algorithm is compared with some classic and powerful algorithms, including IG, ACO and EACO.

In order to evaluate the performance of the proposed algorithm, experiments were performed using two different categories datasets; one is crawled by web spider in a Chinese website while the other is from 20 newsgroup benchmark. The experimental results provide obvious evidences to confirm our algorithm made a huge improvement. In first experiment, our T-IACO can produce a much higher performance especially in small size set. In second experiment, we get a high performance with time more appropriate. ACO based algorithm is vulnerable to be influenced by the parameters, so the investigation on the parameters and testing the T-IACO are areas of future research; meanwhile, heuristic desirability will be another one for the same reason.

REFERENCES

[1] S. M. Vieira, J. Sousa and T. A. Runkler, Two cooperative ant colonies for feature selection using fuzzy models, *Expert Systems with Applications*, vol.37, no.4, pp.2714-2723, 2010.

[2] R. Kohavi and G. H. John, Wrappers for feature subset selection, *Artificial Intelligence*, vol.97, no.1, pp.273-324, 1997.

[3] Y. Yang and J. O. Pedersen, A comparative study on feature selection in text categorization, *ICML*, vol.97, pp.412-420, 1997.

[4] W. Qian and W. Shu, Mutual information criterion for feature selection from incomplete data, *Neurocomputing*, vol.168, pp.210-229, 2015.

[5] C. Shang, M. Li, S. Feng, Q. Jiang and J. Fan, Feature selection via maximizing global information gain for text classification, *Knowledge-Based Systems*, vol.54, pp.298-309, 2013.

[6] M. H. Aghdam, N. Ghasem-Aghaee and M. E. Basiri, Text feature selection using ant colony optimization, *Expert Systems with Applications*, vol.36, no.3, pp.6843-6853, 2009.

[7] M. A. Hall, *Correlation-Based Feature Selection for Machine Learning*, The University of Waikato, 1999.

[8] M. Dash and H. Liu, Consistency-based search in feature selection, *Artificial Intelligence*, vol.151, no.1, pp.155-176, 2003.

[9] L.-Y. Chuang, S.-W. Tsai and C.-H. Yang, Improved binary particle swarm optimization using catfish effect for feature selection, *Expert Systems with Applications*, vol.38, no.10, pp.12699-12707, 2011.
[10] W. Siedlecki and J. Sklansky, A note on genetic algorithms for large-scale feature selection, *Pattern Recognition Letters*, vol.10, no.5, pp.335-347, 1989.
[11] R. Forsati, A. Moayedikia, R. Jensen, M. Shamsfard and M. R. Meybodi, Enriched ant colony optimization and its application in feature selection, *Neurocomputing*, vol.142, pp.354-371, 2014.
[12] M. Dorigo and C. Blum, Ant colony optimization theory: A survey, *Theoretical Computer Science*, vol.334, no.2, pp.243-278, 2005.
[13] A.-A. Ahmed, Feature subset selection using ant colony optimization, *International Journal of Computational Intelligence*, vol.37, no.4, pp.214-223, 2005.
[14] S. Kashef and H. Nezamabadi-Pour, An advanced ACO algorithm for feature subset selection, *Neurocomputing*, vol.147, pp.271-279, 2015.
[15] R. K. Sivagaminathan and S. Ramakrishnan, A hybrid approach for feature subset selection using neural networks and ant colony optimization, *Expert Systems with Applications*, vol.33, no.2, pp.49-60, 2007.
[16] S. Nemati, M. E. Basiri, N. Ghasem-Aghaee and M. H. Aghdam, A novel ACO-GA hybrid algorithm for feature selection in protein function prediction, *Expert Systems with Applications*, vol.36, no.10, pp.12086-12094, 2009.
[17] D. Corne, M. Dorigo and F. Glover, The ant colony optimization meta-heuristic, *New Ideas in Optimization*, McGraw Hill, New York, 1999.
[18] D. Mladenić, Feature selection for dimensionality reduction, *Expert Systems with Applications*, 2006.
[19] M. J. Meena, K. R. Chandran, A. Karthik and A. V. Samuel, An enhanced ACO algorithm to select features for text categorization and its parallelization, *Expert Systems with Applications*, vol.39, no.5, pp.5861-5871, 2012.