Front. Electr. Electron. Eng. China 2009, 4(1): 15–19
DOI 10.1007/s11460-009-0019-9

RESEARCH ARTICLE

Yun ZHANG, Boqin FENG, Shouqiang MA, Lianmeng LIU

Text clustering based on fusion of ant colony and genetic algorithms

Abstract Focusing on the problem that the ant colony algorithm gets into stagnation easily and cannot fully search in solution space, a text clustering approach based on the fusion of the ant colony and genetic algorithms is proposed. The four parameters that influence the performance of the ant colony algorithm are encoded as chromosomes, thereby the fitness function, selection, crossover and mutation operator are designed to find the combination of optimal parameters through a number of iteration, and then it is applied to text clustering. The simulation results show that compared with the classical *k*-means clustering and the basic ant colony clustering algorithm, the proposed algorithm has better performance and the value of *F*-Measure is enhanced by 5.69%, 48.60% and 69.60%, respectively, in 3 test datasets. Therefore, it is more suitable for processing a larger dataset.

Keywords ant colony clustering, genetic algorithm, fusion, text clustering

1 Introduction

Today, text clustering has become an important research topic to solve information overload problems in the field of information processing [1–3].

The ant colony algorithm (ACA) [4] is a novel simulated evolutionary algorithm that shows many promising characteristics [5] such as positive feedback, distributed computing and strong robustness, etc. However, it has shortcomings such as getting into stagnation easily and needing longer computing time. The ant colony algorithm cannot fully search in solution space. Moreover, its parameters are set randomly without theoretical basis. The genetic algorithm (GA) is a global search algorithm, which imitates the mechanism of living creature evolution and natural selection. It has the advantages of not easily falling into local optimum; even in the case where the defined fitness function is non-continuous, irregular and accompanied with noise, it can find the global optimum solution with a high probability. Furthermore, it can be easily combined with other algorithms to achieve better solutions.

In this article, a text clustering algorithm based on the fusion of the ant colony algorithm and the genetic algorithm is proposed. It utilizes the ability of the genetic algorithm on combinatorial optimization problems to get optimal combination of the parameters needed in the ant colony algorithm, and then applies that to a text clustering task.

Translated from *Journal of Xi'an Jiaotong University*, 2007, 41(10): 1146–1150 [译自: 西安交通大学学报]

Yun ZHANG (✉), Boqin FENG, Shouqiang MA, Lianmeng LIU
School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China
E-mail: xjtu.cloud@gmail.com

2 Clustering models and measures

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a dataset into groups (clusters), so that the data in the same group are similar among themselves, and collectively different from the data of other groups. Generally, the distance measure between objects can be used to determine the similarity between them; a formalized description of the clustering model is as follows:

Given a dataset $X = \{x_1, x_2,..., x_n\}$, for $\forall i \in \{1, 2,..., n\}$, $x_i = \{x_{i1}, x_{i2},..., x_{ip}\}$ is an object in X and for $\forall l \in \{1, 2,..., p\}$, x_{il} is an attribute of x_i. According to the inner feature of objects, X can be grouped into clusters $C = \{C_1, C_2,..., C_k\}$, where $\bigcup_{i=1}^{k} C_i = X$, $\forall i, j \in \{1, 2,...,k\}$, $C_i \neq \varnothing$, $C_j \neq \varnothing$, and $C_i \wedge C_j = \varnothing$ $(i \neq j)$. $K = \{X, C\}$ is called a clustering space, C_i is the ith cluster in the clustering space.

There have been several suggestions for a measure of the similarity between two clusterings. Such a measure can be used to compare how well different data clustering

algorithms perform on a set of data. Generally, the sum of the squared error (SSE) of each object to the related cluster center are used as the evaluation function, that is,

$$\text{SSE} = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (d(x_i, c_j))^2, \tag{1}$$

where n_j denotes the number of objects in the jth cluster; $d(x_i, c_j)$ is the distance between object x_i and cluster center c_j. The smaller the SSE value is, the better the clustering results are.

External criteria can also be used in text clustering. F-Measure [6], which combines precision and recall, is an index of system performance in information retrieval field. The precision P and recall R of cluster j and classification i are defined as

$$P = \frac{N_{ij}}{N_i}, \quad R = \frac{N_{ij}}{N_j}, \tag{2}$$

where N_{ij} is the number of objects in cluster j subordinated to classification i; N_i is the number of all objects in classification i; N_j is the number of all objects in cluster j. The formula of F-Measure is as follows:

$$F(i) = \frac{2PR}{P + R}. \tag{3}$$

Thus, the evaluation function of clustering results is defined as

$$F\text{-Measure} = \frac{\sum_i N_i F(i)}{\sum_i N_i}. \tag{4}$$

A greater F-Measure means better clustering results.

3 Fusions of ant colony and genetic algorithms

3.1 Clustering algorithm based on ant-foraging rules

After a long-term observation, biologists found that a moving ant may deposit some pheromone (a particular chemical that ants can smell) on the ground while walking, thus marking the path it follows by a trail of this substance. By sensing pheromone trails foragers can choose the shortest among the available paths from their nest to feeding sources and return. The collective behavior that emerges is a form of autocatalytic behavior where the more the ants are following a trail, the more attractive that trail becomes being followed. The process is thus characterized by a positive feedback loop. The ant colony algorithm is such a heuristic algorithm derived from the study of real ant colonies' behavior.

By applying the ant foraging rules to text clustering, each datum can be viewed as an ant with different attributes, the cluster center can be viewed as food source searched by the ants, and the clustering analysis can be considered as the ant foraging process. In a search iteration, each ant decides the next position by computing the transition probability based on the intensity of the trail and heuristic information such as visibility on each path. At time t, the transition probability from data x_i to cluster center j is defined as

$$p_{ij} = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{l \in \{1,2,\cdots,k\}} [\tau_{il}(t)]^{\alpha}[\eta_{il}(t)]^{\beta}}, \tag{5}$$

where α is the trail heuristic factor that reflects the relative importance of the accumulated intensity of the trail when ants are walking; β is the visibility heuristic factor that reflects the relative importance of visibility when ants are walking; $\tau_{ij}(t)$ is the intensity of the trail on the path from data i to the cluster center j at time t. The initial time has the same intensity of trail on each path, that is, $\tau_{ij}(0) = \tau_0$; $\eta_{ij}(t)$ is the visibility, which is a priori, uncertainty factor used when ants are walking. $\eta_{ij}(t) = 1/d_{ij}$, where

$$d_{ij} = d(x_i, x_j) = \left[\sum_{l=1}^{p} (x_{il} - x_{jl})^2\right]^{1/2}.$$

After each iteration, the cluster center is changed, and the trail intensity of each data to the cluster center is adjusted by the following rules:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t), \tag{6}$$

where ρ is a coefficient such that $1-\rho$ represents the evaporation of the trail;

$$\Delta\tau_{ij}(t) = \frac{Q}{d(x_i, c_j)}, \tag{7}$$

$\Delta\tau_{ij}(t)$ is the quantity per unit of length of the trail substance (pheromone in real ants) laid on the path of data i to the cluster center j between time t and $t+1$. Q is a constant; a greater Q means a faster trail accumulating velocity on the route covered by ants, which affects the convergence rate of the algorithm to a certain extent. Each time an ant transfers from one cluster center to another one will make the clusters' centers changed, which starts the next clustering loop. The loop will be stopped until the clustering results are converged.

3.2 Fusion of ant colony and genetic algorithms

In the ant foraging based clustering algorithm, lots of parameters need to be initialized, which greatly influences the performance of the algorithm. Thus, how to determine an optimal combination of these parameters to achieve an optimal performance of the algorithm is a complex

combination optimization problem, which has no theoretical basis until now. Generally, it is set by experience [4,7,8]. In Ref. [9], the range values of these parameters when using the ant colony algorithm to solve the TSP problem are given as follows: $\alpha \in [0, 5]$, $\beta \in [0, 5]$, $\rho \in [0.1, 0.99]$, $Q \in [10, 10000]$. However, there is no conclusion that the scopes of parameters are also applicable to other applications. We did many experiments on text clustering and the result was not satisfactory.

Thus, we propose a fusion method of the ant colony algorithm and genetic algorithm in this article. Based on the combinational optimization ability of the genetic algorithm, the four parameters used in the ant colony algorithm, that is, α, β, ρ, Q, are encoded as chromosomes in the genetic algorithm. The scope of each chromosome is set large enough. The fitness function and the selection, cross-over and mutation operators are designed. Finally, the optimal combination of the parameters is found after repeated iterations, making the ant colony algorithm achieve optimal performance when solving specific problems. The fitness function should be set combined with the features of the solving problem. In this article, the evaluation function F-Measure is used as fitness function. Each individual chromosome is judged by the quality of clustering results.

The pseudo-code of the ant colony and genetic clustering algorithm (ACGA) is shown as follows:

Step 1 initialize parameters, set the scope of chromosomes.

Step 2 generate the initial populations.

For $i = 1$ to PopSize do

generate population[i].chrom randomly

Step 3 use the ant colony algorithm to obtain the fitness value.

For $i = 1$ to PopSize do

{ //For every individual in the population

(α, β, ρ, Q) = population[i].chrom;

For $j = 1$ to K do //select the initial centers $\{c_1, c_2, ..., c_k\}$

centers[j] = dataset[j];

While($N < \max N_{\mathrm{ACA}}$ || the centers are not changed)

{

for every x_i in X //for each data x_i in dataset X

for $j = 1$ to K do

compute p_{ij} by Eq. (5)

move x_i to C_j, where $p_{ij} = \max\{p_{ij}, 0 < j \leqslant K\}$

for $j = 1$ to K do //update centers

update new centers[j]

for $j = 1$ to TotalNum do //update trail

compute $\tau_{ij}(t+1)$ by Eq. (6)

}

population[i].fitness = F-Measure, compute F-Measure by Eq. (4)

}

Step 4 genetic operations.

Rank all the individuals in the current population by its fitness value, select the new individuals according to the roulette wheel selection, and then generate the new population after the single point crossover and mutation operation.

Step 5 repeat Step 3 and Step 4, until the iteration number is up to $\max N_{\mathrm{GA}}$.

4 Experiments and results

4.1 Dataset

The dataset is extracted from the universal standard text collection, Reuters-21578, in which the documents were collected from Reuters newswire in 1987. 135 different categories have been assigned to 21578 Reuter documents. We selected several categories from it and constructed a dataset; 3 datasets are constructed and the size and the number of topics are different for different datasets. The datasets are described in Table 1. Vector space model is used here; each document is converted from the original format (i.e., strings of characters) to a word vector, $(w_1, w_2, ..., w_n)$, where w_i is the weight of the ith word. Information gain (IG) based approach is used as feature selection method, and the threshold is set as 0.5, which means that the IG value of each word lower than this threshold will be discarded. A popular LTC-weighting method is used. Finally, we can get the word and the dimensions of a word vector in each dataset, which is 145, 155 and 173 respectively. F-Measure and SSE are used to evaluate the performance of k-means algorithm, the ant colony algorithm (ACA) and the ACGA.

4.2 Experimental results

4.2.1 Comparison of clustering results

The predefined cluster number K is set as the number of categories of each dataset, and TotalNum is the size of each dataset. The parameters of the ACA are set as follows: $\alpha = 95$, $\beta = 45$, $\rho = 0.5$, $Q = 400$, $\tau_0 = 20$, and the max iteration times $\max N_{\mathrm{ACA}} = 30$. The parameters of the ACGA are set as follows: the population size PopSize = 30, the crossover

Table 1 Description of text dataset

dataset	number of documents	number of categories	topics
dataset 1	300	4	gnp, gold, jobs, ship
dataset 2	560	6	coffee, crude, grain, interest, money-supply, trade
dataset 3	1240	8	coffee, crude, gold, interest, money-supply, ship, sugar, gnp

probability p_C= 0.8, the mutation probability p_M= 0.8, the max iteration times maxN_{GA}= 50. To verify the sensitivity of the initial cluster centers of various algorithms, two conditions are set to choose the initial cluster centers: initc = 1, that is, poor choice of the initial cluster centers, and thus we will choose the data from the same category as initial centers; initc = 2, that is, better choice of the initial cluster centers, and thus we will choose the data from different categories as initial cluster centers. The experimental results are illustrated in Tables 2, 3 and Figs. 1, 2.

As it can be seen from Tables 2 and 3, the clustering performance of the ACGA and the ACA is better than the *k*-means algorithm. The *F*-Measure of the ACGA is improved by 5.69%, 48.60% and 69.60% compared with the *k*-means algorithm when poor choice of the initial cluster centers is selected. It also indicates that the ACGA is more suitable for dealing with larger dataset than the *k*-means algorithm. In addition, comparing the results obtained from different initial cluster centers selection methods, we can see that the ACA is sensitive to the initial cluster centers, which is the same as the *k*-means algorithm. If the initial cluster centers are inappropriately selected, it is difficult to obtain good clustering results. The fusion algorithm of the ant colony and genetic algorithms reduces the sensitivity to improper selection of the initial cluster centers and can get better clustering results in any case.

Also, it can be seen from Fig. 3 that, though the *k*-means clustering algorithm can get a small SSE value, in most cases, the ACGA can get a smaller SSE value than the *k*-means as well as the optimum *F*-Measure value. This further proves that the ACGA is an effective clustering method.

It must be pointed out that, the ACGA has a lower efficiency than the ACA and the *k*-means algorithm because it needs to execute ACA more than one time. However, this shortcoming is acceptable compared with the advantages brought by the ACGA algorithm, such as reducing the sensitivity to the initial cluster centers, automatically finding the global optimum combination of the parameters in the ACA, and improving the clustering results, etc.

Table 2 *F*-Measure value of 3 clustering algorithms on 3 datasets

algorithm	*F*-Measure					
	dataset 1		dataset 2		dataset 3	
	initc = 1	initc = 2	initc = 1	initc = 2	initc = 1	initc = 2
k-means	0.772438	0.963841	0.433219	0.87998	0.42797	0.857516
ACA	0.772438	0.963841	0.457711	0.880174	0.671905	0.857479
ACGA	0.816356	0.963841	0.643776	0.906593	0.725849	0.865642

Table 3 SSE value of 3 clustering algorithms on 3 datasets

algorithm	SSE					
	dataset 1		dataset 2		dataset 3	
	initc = 1	initc = 2	initc = 1	initc = 2	initc = 1	initc = 2
k-means	118.632	105.434	239.56	184.583	436.445	321.072
ACA	118.632	105.434	290.777	184.621	370.97	321.07
ACGA	117.44	105.434	215.216	199.757	389.805	321.104

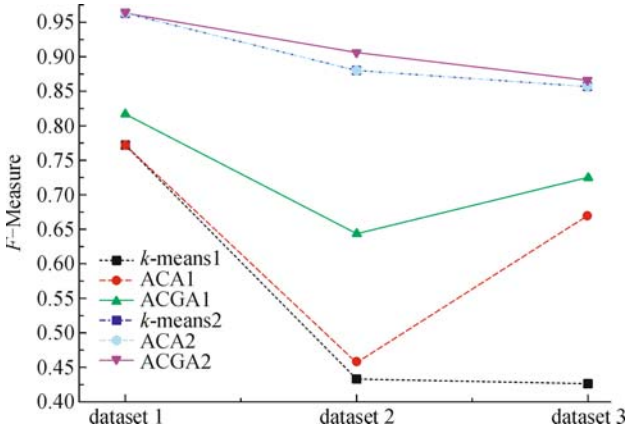

Fig. 1 *F*-Measure of different clustering algorithms

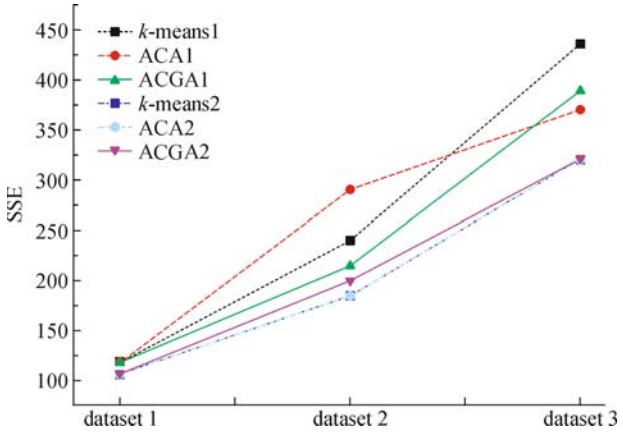

Fig. 2 SSE of different clustering algorithms

Table 4 Optimal *F*-measure values obtained by ACGA (initc = 1)

	F-Measure		
	dataset 1	dataset 2	dataset 3
the scope of parameters in Ref. [9]	0.507291	0.331047	0.521478
the scope of parameters in this paper	0.816356	0.643776	0.725849

4.2.2 Influence of different scopes of parameters on ant colony algorithm

When the ACA is used to solve classical TSP problems, the scope of different parameters are set as follows [9]: $0 \leqslant \alpha \leqslant 5$, $0 \leqslant \beta \leqslant 5$, $0.1 \leqslant \rho \leqslant 0.99$, $10 \leqslant Q \leqslant 10000$. The scope of different parameters in the ACGA are set as follows: $0 \leqslant \alpha \leqslant 100$, $0 \leqslant \beta \leqslant 100$, $0 \leqslant \rho \leqslant 1$, $10 \leqslant Q \leqslant 1000$. The other parameters are set as the same in Sect. 4.2.1. Testing them on three datasets separately, the optimal *F*-Measure values found by the genetic algorithm in 50 generations are illustrated in Table 4.

It can be seen from Table 4 that the scopes of parameters used when ACA solves TSP problems are not suitable for solving all the other problems, such as the text clustering problem. Compared with the ACA, the algorithm proposed in this article can better address the setting values of the multi-parameter problem.

5 Conclusions

The genetic algorithm (GA) is a self-adapted probability search method used to solve optimization problems, which has been applied widely in science and engineering. The ant colony algorithm simulates the social behavior of ant colonies, and has become a hotspot in recent years. The proposed ant colony–genetic fusion clustering algorithm in this article combines the genetic algorithm with the ant colony algorithm, using the ability of the genetic algorithm to solve combinational optimization problems to determine the optimal value of multiple parameters in the ant colony algorithm, and then is applied to the text clustering task to achieve better results.

Future research will be focused on further improving the efficiency of the ant colony–genetic fusion algorithm, and trying to study the setting of various parameters theoretically in the ant colony algorithm.

Acknowledgements This work was supported by the Hi-Tech Research and Development Program of China (No. 2006AA01Z210).

References

1. Liu Y C, Wang X L, Xu Z M, Guan Y. A survey of document clustering. Journal of Chinese Information Processing, 2006, 20(3): 55–62 (in Chinese)
2. Sasaki M, Shinnou H. Spam detection using text clustering. In: Proceedings of the 2005 International Conference on Cyberworlds (CW'05), Singapore. 2005, 316–319
3. He F, Ding X Q. Combining text clustering and retrieval for corpus adaptation. Proceedings of SPIE. 2007, 6500: 65000P1–7
4. Dorigo M, Blum C. Ant colony optimization theory: a survey. Theoretical Computer Science, 2005, 344(2–3): 243–278
5. Zhu X L, Li J Z. An ant colony system-based optimization scheme of data mining. In: Proceedings of the 6th International Conference on Intelligent Systems Design and Applications (ISDA'06), Jinan, Shandong, China. 2006, 400–403
6. van Rijsbergen C J. Information Retrieval. 2nd ed. London: Butterworths, 1979
7. Wu C M, Chen Z, Jiang M. The research on initialization of ants system and configuration of parameters for different TSP problems in ant algorithm. Acta Electronica Sinica, 2006, 34(8): 1530–1533 (in Chinese)
8. Huang Y Q, Liang C Y, Zhang X D. Parameter establishment of an ant system based on uniform design. Control and Decision, 2006, 21 (1): 93–96 (in Chinese)
9. Duan H B. Ant Algorithm－Theory and Its Applications. Beijing: Science Press, 2005 (in Chinese)