The Journal of
Mathematics and Computer Science

Available online at
http://www.TJMCS.com

The Journal of Mathematics and Computer Science Vol. 4 No.2 (2012) 182 - 196

A Novel Method for Document Clustering using Ant-Fuzzy Algorithm

Javad Rajaie[1],Babak Fakhar[2]

[1] Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
java_raja@yahoo.com,

[2] Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
Fakharbabak@yahoo.com

Received: January 2012, Revised: April 2012
Online Publication: June 2012

Abstract

Availability of large full-text document collection in electronic forms has created a need for tools techniques that assist users in organization. Document clustering is one of the popular methods used for this purpose. Ant-based text clustering is a promising technique that has attracted great research attention. This paper attempts to improve the standard ant-based text-clustering algorithm. The ant behavior model is modified to pursue better algorithmic performance. In this paper, a hybrid approach based on Ant clustering and Fuzzy clustering methods is used. First ant based clustering is used for creating raw and imprecise clusters and then these clusters are refined by means of fuzzy C-Mean (FCM) algorithm. For large datasets these two stages does not suffice and many homogenous small clusters are formed. Thus more iteration of these two stages is usually required and clusters from previous iterations are used as a building block in the following iterations to build finer and larger clusters.
The proposed algorithm is tested with a sample set of documents excerpted from the Reuters-21578 corpus and the experiment results partly indicate that the proposed algorithm perform better than the standard ant-based text-clustering algorithm and the k-means algorithm.

Keywords: Ant colony optimization, Ant-based clustering, text clustering, ant movement strategy.

I. Introduction

Clustering analysis is an important method in data mining. It is a discovery process that groups a set of data such that the intra-cluster similarity is maximized and the inter1-cluster similarity is minimized. Clustering of data in a large dimension space is of a great interest in many data mining applications (P. Berkhin 2002).

Clustering has been widely studied since the early 60's. Some classic approaches include hierarchical algorithms, partitioning method such as K-means, Fuzzy C-means, graph theoretic clustering, neural networks clustering, and statistical mechanics based techniques. Recently, several papers have highlighted the efficiency of stochastic approaches based on ant colonies for data clustering (B.wu, Y. zheng, 2002, Deneubourg 1991, Nicolas 1999).

Whit the abundance of textual documents, e.g. over the World Wide Web and in corporate document management systems, there is an increasing demand for text mining techniques. Consequently, as a noteworthy branch of text mining, text clustering has caught great attention in the last decade; and various data clustering methods have been applied, e.g. agglomerative hierarchical clustering (Ward, J.H. 196) , k-means (Hartigan, J. & Wong, 1979) OPTICS (Ankerst 1999), and genetic algorithm-based clustering (Chiou 2000). However, despite tremendous endeavors, the performance of the existing methods is not often satisfactory in actual applications; more work needs to be done to exploit better text clustering methods.

An early attempt of the ant-based clustering model was given by Deneubourg and Goss et al. In their model, the ants tend to pick up the isolated items (corpses or larvae) and bring them to the positions that already contain items of the same type. A notable effort was made by Lumer and Faieta (1994), which generalized Deneubourg and Goss et al.'s model and applied it to numerical data analysis. Based on these pioneering contributions, more recent endeavors on the ant-based clustering models have also been reported (e.g., Monmarché 1999, Hartigan 1979, and Meyer 2002, Kanade and Hall 2003, Vizine and de Castro et al. 2005); and the proposed ant-based clustering methods have been applied to various areas, such as graph partitioning (Kuntz 1998), intrusion detection (Ramos 2005), and text clustering (Berry 2003). These efforts reveal that ant-based clustering has become an active research field.

The focal point of this paper is to improve the algorithmic performance of ant-based clustering by modifying the ant movement rule. Our observation is that in the majority of the existing ant-based clustering methods, the ant movement is supposed to be completely blind; and this blind walk model could possibly hamper the convergence or at least decrease the efficiency of the algorithm. To overcome this limitation, we try to establish some mechanism to direct a laden ant toward a dense area of items that are in the same type with the item being carried by the current ant, and to direct an unladen ant to a position which contains an item that is dissimilar with the surrounding items, in hope that such modified ant movement rule would boost the algorithm to converge to the appropriate clusters more rapidly.

A. Basic concept of ACO

ACO is a meta-heuristic algorithm inspired by the behavior of real ants, and in particular how they forage for food. ACO can be applied to problems that can be described by a graph,

where the solutions to the optimization problem can be expressed in terms of feasible paths on the graph. Among the feasible paths, ACO can be used to find the one with minimum cost. The first member of the ACO algorithm, called ant system (AS), was first applied to the traveling salesman problem (Dorigo et al. 1996, Bonabeau 1999). In ACO, a set of artificial ants was created and they cooperate in finding the solution by exchanging information via pheromone deposited on graph edges. The objective of ACO is to find good solutions for combinational optimization problems. In real world, ants move randomly without using other information initially, and lay some pheromone on the ground. After that, an ant moves originally at random until it encounters a previous trail. This ant will then follow the pheromone trail with high probability, and enhances the trail with its own pheromone. Finally, most ants choose the same path with the greatest amount of pheromone deposit (Bonabeau 1999).

The whole ACO algorithm can be described by taking the traveling salesman problem (TSP) as an example. The TSP is to find a minimal route for a salesperson to take in visiting N cities with each city being visited once Algorithm (1). this problem can be represented by a graph G = (N, E) with N nodes representing the N cities, and E being a set of edges fully connecting the nodes. Let d_{ij} be the length of the edge (i, j) ∈ E, that is the distance between cities i and j, with i, j ∈ N. The TSP can be defined as finding a shortest closed path in G with visiting each node of G exactly once. At each iteration t of ACO, an ant in city i has to choose the next city j to head for from among those cities that it has not yet visited. The probability of picking a certain city j is calculated using the distance between cities i and j, and the amount of pheromone on the edge between these two cities. The probability with which an ant q chooses to go from city i to city j is

$$P_{ij}^{q}(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]\ \left[\eta_{ij}\right]^{\beta}}{\sum_{l \epsilon N_i^q}\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}} & \text{if } j \epsilon N_i^q \\ 0 & \text{otherwise} \end{cases}$$

Where $\tau_{ij}(t)$ is the amount of pheromone trails on edge (i, j) at iteration t, $\eta_{ij} = 1/d_{ij}$ is the heuristic value of moving from city i to city j, N_i^q is the set of neighbors of city i for the q^{th} ant, and parameter b controls the relative weight of pheromone trail and heuristic value. After all ants have completed their tours, the pheromone level is updated by :

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

Where $0 <= \rho < 1$ is the pheromone trail evaporation rate. The update value $\Delta\tau_{ij}$ is related to a quality value F which is used to measure the performance of each ant route. Many updating rules for $\Delta\tau_{ij}$ have been studied (Dorigo and Stu¨tzle 2004)[7]. For example, $\Delta\tau_{ij}$ may be implemented by:

$$\Delta\,\tau_{ij}^{q} = \begin{cases} F = \dfrac{1}{L_{gb}}, & \text{if } (i,j) \epsilon \text{ global} - \text{best} - \text{tour} \\ 0 & \text{otherwise} \end{cases}$$

Where L_{gb} is the length of the global best tour from the beginning of the route. That is, only those edges belonging to the global best tour receive reinforcement.

ALGORITHM I
THE SKELETON OF ACOALGORITHM APPLIED TO THE TSP

```
Procedure ACO algorithm for TSPs
    Set parameters, initialize pheromone trails
    While (termination condition not met) do
        Tour construction
        Pheromone update
    End
End ACO algorithm for TSPs
```

B. Document representation

In most clustering algorithms, the dataset to be clustered is represented as a set of vectors $X = \{x_1,x_2,\ldots,x_n\}$, where the vector xi corresponds to a single document object and is called a ''feature vector'' that contains proper features to represent the object. The text document objects can then be represented using the Vector Space Model (VSM) (B. Everitt, 1980). In this model, the content of a document is formalized as a point in the multi-dimensional space represented by a vector x, such as $x = (w_1,w_2, \ldots , w_n)$, where w_i (i = 1,2,… ,n) is the term weight of the term t_i in one document. The term weight value w_i represents the significance of this term in a document.
To calculate the term weight, the occurrence frequency of the term within a document and in the entire set of documents must be considered. The most widely used weighting scheme combines the Term Frequency with Inverse Document Frequency (TF-IDF) (B. Everitt, 1980). The weight of term *i* in document *j* is given by:

$$W_{ij} = tf_{ij} \times \log (N / df_j)$$

where tf_{ji} is the number of occurrences of term *i* in the document *j*; df_{ji} indicates the term frequency in the collections of documents; and n is the total number of documents in the collection. This weighting scheme discounts the frequent words with little discriminating power. A word with a high frequency within a document and low frequency within the document collection will be assigned a high weight value. Before translating the document collection into TF-IDF VSM, the very common words (e.g. function words: ''a'', ''the'', ''in'', ''to''; pronouns: ''I'', ''he'', ''she'', ''it'') are stripped out completely and different forms of a word are reduced to one canonical form by using Porter's algorithm (M.F. Porter, 1980).

When documents are represented as vectors, as described above, they belong to a very high-dimensional feature space because of one dimension for each unique term in the collection of documents. In order to reduce the dimension of the feature vector, the Document Frequency Thresholding is performed. Some terms whose document frequency

are less than the predetermined threshold or appear in over 90% of the documents are removed. Further, only a small number of *n* terms with the highest weights in each document are chosen as indexing terms. (Hotho, A., Staab, S. & Stumme, G. 2003)

C. The similarity metric

The similarity between two documents needs to be measured in clustering analysis. In order to group similar data objects, proximity metric has to be used to identify objects that are similar. Over the years, two prominent ways have been proposed to compute the similarity between documents x_p and x_j. The first method is based on Minkowski distances, given by

$$D_n(x_p, x_j) = (\sum_{k=1}^{d_x} |x_{k,p} - w_{k,j}|^n)^{1/n}$$

where x_p and x_j are two document vectors; d_x denotes the dimension number of the vector space; $w_{k,p}$ and $m_{k,j}$ stands for the documents x_p and x_j's weight values in dimension k.

D. The Classical Clustering Algorithms

Data clustering is broadly based on two approaches: *hierarchical* and *partitional* Within each of the types, there exists a wealth of subtypes and different algorithms for finding the clusters. In hierarchical clustering, the output is a tree showing a sequence of clustering with each cluster being a partition of the data set (Leung *et al.*, 2000). Hierarchical algorithms can be agglomerative (bottom-up) or divisive (top-down). Agglomerative algorithms begin with each element as a separate cluster and merge them in successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. Hierarchical algorithms have two basic advantages (Friguiand Krishnapuram, 1999). Firstly, the number of classes need not be specified a priori and secondly, they are independent of the initial conditions. However, the main drawback of hierarchical clustering techniques is they are static, i.e. data-points assigned to a cluster can not move to another cluster. In addition to that, they may fail to separate overlapping clusters due to lack of information about the global shape or size of the clusters (Jain 1999).

Partitional clustering algorithm s, on the other hand, attempt to decompose the data set directly into a set of disjoint clusters. They try to optimize certain criteria. The criterion function may emphasize the local structure of the data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically, the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters. The advantages of the hierarchical algorithms are the disadvantages of the partitional algorithms and vice versa.

Clustering can also be performed in two different modes: crisp and fuzzy. In crisp clustering, the clusters are disjoint and non-overlapping in nature. Any pattern may belong to one and only one class in this case. In case of fuzzy clustering, a pattern may belong to all the classes with a certain fuzzy membership grade (Jain 1999).

The most widely used iterative K-means algorithm (MacQueen, 1967) for partitional clustering aims at minimizing the ICS (Intra-Cluster Spread) which for K cluster centers can be defined as:

$$ICS(c_1, c_2, \dots, c_k) = \sum_{i=1}^{k} \sum_{k=1}^{n} \| X_i - m_i \|^2$$

The K-means (or hard C-means) algorithm starts with K cluster-centroids (these centroids are initially selected randomly or derived from some a priori information). Each pattern in the data set is then assigned to the closest cluster-centre. Centroids are updated by using the mean of the associated patterns. The process is repeated until some stopping criterion is met.

In the C-medoids algorithm (Kaufman and Rousseeuw, 1990), on the other hand, each cluster is represented by one of the representative objects in the cluster located near the center. Partitioning around medoids (PAM) (Kaufman and Rousseeuw, 1990) starts from an initial set of medoids, and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering. Although PAM works effectively for small data, it does not scale well for large datasets. Clustering large applications based on randomized search (CLARANS) (Ng and Han, 1994), using randomized sampling, is capable of dealing with the associated scalability issue.

The fuzzy C-means (FCM) (Kanade 2003) seems to be the most popular algorithm in the field of fuzzy clustering. In the classical FCM algorithm, a *within cluster sum* function J_m is minimized to evolve the proper cluster centers:

$$J_m = \sum_{j=1}^{n} \sum_{i=1}^{c} u_{ij}^m \| X_j - v_i \|^2$$

Where *Vi* is the i-th cluster center, X_j is the j-th d-dimensional data vector and ||.|| is an inner product-induced norm in d dimensions. Given *c* classes, we can determine their cluster centers V_i for i=1 to c by means of the following expression:

$$V_i = \frac{\sum_{j=1}^{n} (u_{ij})^m X_j}{\sum_{j=1}^{n} (u_{ij})^m}$$

Here m (m>1) is any real number that influences the membership grade. Now differentiating the performance criterion with respect to V_i (treating u_{ij} as constants) and with respect to u_{ij} (treating V*i* as constants) and setting them to zero the following relation can be obtained:

$$u_{ik} = \left[\sum_{k=1}^{c} \left(\frac{|X_j - v_i|}{|X - v_i|} \right)^{-\left(\frac{1}{m-1}\right)} \right]^{-1}$$

Several modifications of the classical FCM algorithm can be found in (Hall *et al*., 1999, Gath and Geva, 1989, Bensaid *et al*., 1996, Clark *et al*., 1994, Ahmed *et al*., 2002, Wang *et al*., 2004).

E. ***Standard Ant Colony Based Clustering Algorithms***

Ant colonies provide a means to formulate some powerful nature-inspired heuristics for solving the clustering problems. Among other social movements, researchers have simulated the way, ants work collaboratively in the task of grouping dead bodies so, as to keep the nest clean. It can be observed that, with time the ants tend to cluster all dead bodies in a specific region of the environment, thus forming piles of corpses (Handl 2003).

Larval sorting and corpse cleaning by ant was first modeled by Deneubourg *et al.* for accomplishing certain tasks in robotics (Deneubourg *et al*., 1991). This inspired the Ant-based clustering algorithm (Handl 2003) . Lumer and Faieta modified the algorithm using a dissimilarity-based evaluation of the local density, in order to make it suitable for data clustering (Lumer 1994). This introduced standard Ant Clustering Algorithm (ACA). Many authors (Handl and Meyer, 2002, Ramos *et al*., 2002) proposed a number of modifications to improve the convergence rate and to get optimal number of clusters. Monmarche *et al.* hybridized the Ant-based clustering algorithm with K-means algorithm and compared it to traditional K-means on various data sets, using the classification error for evaluation purposes. However, the results obtained with this method are not applicable to ordinary ant-based clustering since it differs significantly from the latter.

Like a standard ACO, ant-based clustering is a distributed process that employs positive feedback. Ants are modeled by simple agents that randomly move in their environment. The environment is considered to be a low dimensional space, more generally a two-dimensional plane with square grid. Initially, each data object that represents a multi-dimensional pattern is randomly distributed over the 2-D space. Data items that are scattered within this environment can be picked up, transported and dropped by the agents in a probabilistic way. The picking and dropping operation are influenced by the similarity and density of the data items within the ant's local neighborhood. Generally, the size of the neighborhood is 3£3. Probability of picking up data items is more when the object are either isolated or surrounded by dissimilar items. They trend to drop them in the vicinity of similar ones. In this way, a clustering of the elements on the grid is obtained. The ants search for the feature space either through random walk or with jumping using a short term memory. Each ant picks up or drops objects according to the following local probability density measure:

$$f(X_i) = \max\{0, \frac{1}{s^2}\sum_{X_i \in N_{s\times s}(r)}[1 - \frac{d(X_i - X_j)}{\alpha\left(1 + \frac{v-1}{v_{max}}\right)}]\}$$

In the above expression, $N_{s*s(r)}$ denotes the local area of perception surrounding the site of radius *r*, which the ant occupies in the two-dimensional grid. The threshold αg cales the dissimilarity within each pair of objects, and the moving speed *v* controls the step-size of the ant searching in the space within one time unit. If an ant is not carrying an object and finds an object X*i* in its neighborhood, it picks up this object with a probability that is inversely proportional to the number of similar objects in the neighborhood. It may be expressed as:

$$P_p(X_i) = \left(\frac{k_p}{k_p + f(X_i)}\right)^2$$

If however, the ant is carrying an object x and perceives a neighbor's cell in which there are other objects, then the ant drops of the object it is carrying with a probability that is directly proportional to the object's similarity with the perceived ones. This is given by:

$$P_d(X_i) = \begin{cases} 2.f(X_i) \; if \;\; f(X_i) < k_d \\ 1 \qquad if \;\; f(X_i) \geq k_d \end{cases}$$

The parameters k_p and k_d are the picking and dropping constants (Gath and Geva, 1989) respectively. Function f (X_i) provides an estimate of the density and similarity of elements in the neighborhood of object X*i*. The standard ACA pseudo-code is summarized in Algorithm II.

ALGORITHM II
PROCEDURE ACA

```
1: Place every item Xi on a random cell of the grid;
2: Place every ant k on a random cell of the grid unoccupied by ants;
3: iteration_count = 1;
4:        while iteration_count < maximum_iteration do
5:             for i = 1 to no_ of_ ants do
6:                  if unladen ant and cell occupied by item Xi then
7:                        compute f(Xi) and Pp (Xi);  //Ppick up(Xi)
8:                  else
9:                     if ant carrying item xi and cell empty then
10:                          compute f(Xi) and Pd(Xi);
11:                          drop item Xi with probability Pd(Xi); //Pdrop(Xi)
12:                    end if
13:               end if
14:               move to a randomly selected, neighboring and unoccupied cell ;
15:            end for
16: t = t + 1
17: end while
18: print location of items;
```

Kanade and Hall presented a hybridization of the ant systems with the classical FCM algorithm to determine the number of clusters in a given dataset automatically. In their fuzzy ant algorithm, at first the ant based clustering is used to create raw clusters and then these clusters are refined using the FCM algorithm. Initially the ants move the individual data objects to form heaps. The centroids of these heaps are taken as the initial cluster centers and the FCM algorithm is used to refine these clusters. In the second stage the objects obtained from the FCM algorithm are hardened according to the maximum membership criteria to form new heaps. These new heaps are then sometimes moved and merged by the ants. The final clusters formed are refined by using the FCM algorithm.

A number of modifications have been introduced to the basic ant based clustering scheme that improve the quality of the clustering, the speed of convergence and, in particular, the spatial separation between clusters on the grid, which is essential for the scheme of cluster retrieval.

II. Ant-Fuzzy clustering (Proposed algorithm)

Ant-clustering algorithm can divide data without any information about the number of clusters, but because of accidental nature of this algorithm, more time is needed to achieve the final clusters.

Therefore using definite algorithms such as K-Means and FCM, along with this algorithm, can accelerate the clusters formation process toward the final cluster, and in the meantime improve the quality of clusters produced by ant-clustering algorithm. On the other hand, the two algorithms mentioned above, which are based on partition clustering, are extremely sensitive to the initial division of data and require suitable initial conditions. Such initial condition can be provided through ant-clustering algorithm, which dose not need the knowledge about the number of cluster of clusters and their centers. This is required so that the centers of these raw and imperfect clusters will be rectified through these algorithms.

Ant-fuzzy clustering algorithm is based on combination of two methods of ant-clustering and FCM clustering. But the most important point about FCM algorithm in that unlike (Xia, Wang 2006) which dose not allow any change and reduction of the number of cluster obtained though FCM ant-clustering, in this method after applying FCM algorithm it is possible that the degree of some cluster belonging against all data would come down to zero and these cluster are completely emptied. This means that, these clusters will be wiped of the initial list of clusters. Thus, in additional to the improvement in clusters quality; this algorithm will rectify the number of cluster which is normally more than the member of final clusters.

Considering the number of data and their possible dispersal, usually these for stages do not satisfy the large data collection, and after the end of algorithm, a large number of small and similar heaps still remain. Therefore, more repeats of these two stages are needed. In each stage, fuzzy clusters produced through FCM in previous stage, becomes non-fuzzy by using the most membership degree criterion, so that new heaps with better quality and lesser numbers are formed. These heaps, which in the next stages are considered as one single mass, and can no larger be broken up, could be moved, dropped, or picked by ants. In a situation where clustering masses are the very same clusters of previous stages, ant-clustering algorithm. Thus, this repetitive algorithm can create a sequential clustering from the data and, in each stage combine the previous cluster to construct larges clusters.

Below, is a brief explanation of ant-fuzzy algorithm clustering combination? In the beginning, this algorithm acts upon incoming data, and in the final repeat, data final clustering method is determined the non-fuzzy heaps.

ALGORITHM III
MOVMENT RULE OF A LADEN ANT

1- Perform an-clustering algorithm.
2- Calculate the heap centers obtained from pervious stage, and perform FCM algorithm on these heaps and their centers.
3- Using the most belonging degree measure for fuzzy cluster obtained from previous stage, make the new heaps non-fuzzy.
4- If more stages and needed, repeat 3-1 whilst considering each heaps as a single mass.

III. Evaluation Experiments

In this section the test results of the proposed algorithm is reported. Although our current test is still primitive, with a relatively small-sized dataset being used, the positive results have partially indicated the benefits of the proposed algorithm. In the current test, 50 documents are arbitrarily excerpted from the Routers-21578 corpus, which is one of the most-widely adopted benchmarking datasets in the text mining field (Lewis 2006). The selected documents cover 3 topics of "gas", "gold" and "livestock", with each topic containing 10 documents. The keywords (concepts) to represent these documents are extracted by using the text mining tool, *TextAnalyst*™ (Megaputer 2006). Furthermore, WordNet® is used as the base ontology to analyze the semantic similarity between concepts, as well as between documents.

With these resources and tools, we test the performance of our revised algorithm, comparing with the standard ant-based text clustering algorithm.

IV. Performance Analysis of the Modified Ant Algorithm

We test the performance of the proposed clustering algorithm with the aforementioned dataset. The basic parameters of the algorithm are set as shown in Table I.

TABLE I
PARAMETER SETTING FOE THE ANT ALGORITHM PARAMETER VALUE

Grid Size	**15*15**
Count of Ants	12
k_p	Initial picking threshold 0.35
k_d	Initial dropping threshold 0.35
R	Initial Moore Neighborhood 3

With the above setting, the proposed algorithm is tested.
To numerically evaluate the performance of the proposed algorithm, we furthermore calculate the F-measure, the precision and recall of the clustering result. We define the precision of the cluster "*j*" in the type "*i*" (predefined for the purpose of testing) as:

$$precision(i,j) = \frac{N_{ij}}{N_j}$$

And the recall as:

$$recall(i,j) = \frac{N_{ij}}{N_i}$$

Where, N_{ij} refers to the number of documents in the cluster *j* that belongs to the type *i*; N_i refers the number of documents in the type *i*; and N_j is the number of documents in the cluster *j*. For calculating the F-measure for class *i* and cluster *j*, we use by:

$$F(i,j) = 2\frac{Recall(i,j) \times precision(i,j)}{Recall(i,j) + precision(i,j)}$$

In final the measure of F for all clusters with *n* data calculates by :

$$F = \sum_i \frac{n_i}{n} max_j (F(i,j))$$

With these two definitions, the precision and recall of the different categories of documents in our experiment can be calculated.

Actually, we do the experiment on the same dataset multiple times, Obtaining somewhat different clustering results every time (as the proposed clustering algorithm is in nature a nondeterministic approach). The average recall and precision of the repeated experiments are shown in Table II.

TABLE II
RECLALL AND PRECISION OF THE TEST EXAMPLE

Type	Recall	Precision
Livestock	0.99198	0.623336
gold	0.89000	0.82143
gas	0.81429	0.85000
Average	0.89875	0.764922

The recall and the precision of the proposed algorithm are then compared with those of the standard ant clustering algorithm. As we suggest modifying the standard ant-based text clustering .The recall and precision of the each method are shown in Table III and IV, respectively.

TABLE III
RECALL COMPERARION OF THE TWO METHODS

Type	Standard Ant Clustering	New Ant Clustering
Gold	0.36667	0.76667
Livestock	0.30000	0.93333
Gas	0.26667	0.96667
Average	0.311113	0.88889

TABLE IV
PRECISION COMPARION OF THE TWO METHEODS

Type	Standard Ant Clustering	New Ant Clustering
Gold	0.39000	0.43000
Livestock	0.32667	0.32000
Gas	0.44000	0.72667
Average	0.385557	0.49222

The contribution of the modified ant clustering process performs is mainly on the improvement of the clustering precision, whereas the effect of the modified ant clustering process on the recall is not so significant. This result basically fits our anticipation when designing the algorithm. The main purpose of the modification of the ant clustering process is to increase the convergence speed of the algorithm.

Our experiments show that the modified ant clustering process has advantage on the algorithmic efficiency over the standard ant-clustering process. Using our ant-clustering algorithm, good clusters are formed and stabilized after about 5,000 ant steps, whilst the standard ant clustering algorithm usually reaches the convergence after 20,000 ant steps.

We also compare the proposed algorithm with the k-means method, which can be regarded as today's benchmarking clustering technique. The recall and precision comparisons between our algorithm and the k-means algorithm are illustrated in Table V.

The results in TableV show that the clustering precision of the proposed algorithm is basically at the same level as that of the k-means algorithm; however, in terms of the recall, the proposed ant clustering algorithm performs apparently better than the k-means algorithm.

that the proposed algorithm, at least in our test case, has better performance than the standard ant clustering algorithm and the k-means algorithm.

TABLE V
COMPARION WITH THE K-MEANS METHOD

	Recall		**Precision**	
	New_Ant_Clustering	K-means	New_Ant_Clustering	K-means
Gold	0.71429	0.42000	0.75000	0.67920
Livestock	1.0000	0.54000	0.52336	0.56500
Gas	0.71429	0.42000	0.75000	0.67920
Average	0.809527	0.46	0.674453	0.641133

V. Conclusions and Future Works

In this paper, we present a new ant-fuzzy clustering algorithm, and apply it to the field of text clustering. The proposed algorithm is a revised version of an algorithm the authors proposed earlier (Xia, Wang 2006), trying to increase its scalability to cater for larger datasets. the methodology is provided with compound clustering on the basis of ant

clustering and fuzzy clustering for document clustering .The combination of each one may lead to remove the deficiencies of each method and finally will enjoy the related advantages of two methods. The algorithm of ant clustering doesn't require having primary appropriate condition and information about the numbers of clusters. Regarding its randomized nature, ant clustering requires more temporal duration to have access to final clusters. In contrast, partitioning algorithms to say FCM are very sensitive to primary conditions. Such conditions may be achieved by ant clustering algorithm. FCM algorithm may be used to accelerate achieving process of final cluster and better qualification of clusters. The efficiency of the given methodology was evaluated by using such two criteria as the number of gained clusters and F-measure. More reiteration of these two stages and the development of a clustering algorithm, as it was regarded as unique object in the next achieved heaps from earlier stages, will contribute us to a hierarchical clustering.

Our experiments on the proposed algorithm show that on one hand, the proposed algorithm efficiently converges to reasonably good clusters, comparing with the standard ant-based text clustering algorithm; on the other hand, the recall and precision of the clustering results of the proposed algorithm are higher than the standard ant-based text clustering algorithm and the k-means algorithm. These results partly indicate that it may be worthwhile to give further investigations on the proposed algorithm.

The algorithm of ant clustering used in this research has many regulating parameters which have been gained empirically. Some given amounts vary greatly with the amounts used in main research based on method (Xia, Wang 2006). The results of the related algorithm are very sensitive to the decision made on K_p , K_d , R and the amounts of these three parameters should be regulated on the basis of the type of data and statistical characteristics. The future research field should focus on automatic regulation of these parameters by using more advanced types of ant clustering algorithm.

Acknowledgements

The author needs to appreciate Islamic Azad University, Mahshahr Branch for supporting of this research.

Reference

[1] Ankerst, M., Breunig, M., Kriegel, H.P. & Sander, J. (1999). OPTICS: Ordering points to identify clustering structure. In: Proceedings of the ACM SIGMOD Conference, pp. 49-60

[2] B.wu,Y.zheng,S.liu and Z.shi, SIM:A Document Clustering Algorithm Based on Swarm Intelligence. IEEE World Congress on Computational Intelligence,Hawaiian,PP.477-482.2002

[3] Berry, M. (ed.) (2003). Survey of Text Mining: Clustering, Classification, and Retrieval. Springer, New York

[4] Bonabeau, E., Dorigo, M. & Theraulaz, G. (1999). Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York

[5] Chiou, Y-C. & Lan, L.W. (2000). Genetic clustering algorithms. European Journal of *A Modified Ant-Based Text Clustering Algorithm with Semantic Similarity Measure* 490 JOURNAL OF SYSTEMS SCIENCE AND SYSTEMS ENGINEERING Operational Research, 135: 413-427

[6] Deneubourg J L , Goss S , Frank N , Sendova-hanks A ,Detrain C ,Chrerien L. The dynamics of collective sorting: robot-like ants and ant-like robots. In : Proceedings of the 1st International Conference on Simulation of Adaptive Behavior : From Animals to Animats, MIT Press/Bradford Books, Cambridge, MA ,1991. 356–363
[7] Handl, J., Knowles, J. & Dorigo, M.(2003). On the performance of ant-based clustering. In: Proceedings of the Third International Conference on Hybrid Intelligent Systems. pp. 204-213, IOS Press
[8] Hartigan, J. & Wong, M. (1979). Algorithm AS136: A k-means clustering algorithm. Applied Statistics, 28: 100-108
[9] Hotho, A., Staab, S. & Stumme, G. (2003).Wordnet improves text document clustering. In: Proceedings of the Semantic Web Workshop at SIGIR-2003, 26th Annual International ACM SIGIR Conference, Toronto, Canada. July 28-August 1, 2003
[10] Jain, A.K., Murty, M.N. & Flynn, P.J. (1999). Data clustering: a review. ACM Computing Surveys, 31(3): 264-323
[11] Kanade, P. & Hall, L.O. (2003). Fuzzy ants as a clustering concept. In: Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 227-232
[12] Kuntz, P., Snyers, D. & Layzell, P. (1998). A stochastic heuristic for visualising graph clusters in a bi-dimensional space prior to partitioning. Journal of Heuristics, 5: 327-351
[13] Labroché, N. Monmarché, N. & Venturini, G. (2002). A new clustering algorithm based on the chemical recognition system of ants. In: Proceedings of the 2002 European Conference on Artificial Intelligence, pp. 345-349
[14] Lewis, D. (2006). Reuters-21578 text categorization test collection. Available via: http://www.daviddlewis.com/resources/test collections/reuters21578. Cited Nov. 10, 2006.
[15] Lumer, E. & Faieta, B. (1994). Diversity and adaption in populations of clustering ants. In: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour, MIT Press, Cambridge, MA
[16] Megaputer Intelligence Inc. (2006). Online introduction to TextAnalystTM. Available via:
[17] http://www.megaputer.com/products/, Cited Nov. 12, 2006
[18] M.F. Porter, An algorithm for suffix stripping, Program 14 (3) (1980) 130–137. K. Cios, W. Pedrycs, R. Swiniarski, Data Mining—Methods for Knowledge Discovery, Kluwer Academic Publishers, 1998.
[19] Nicolas c, Mohamed Slimane, Gilles Venturini. AntClass: discovery of clusters in numeric data by an hybridization of an ant colony with the Kmeans algorithm, Internal Repport No 213, E3i, January 1999
[20] P.Berkhin. Survey of Clustering Data Mining Techniques. Accrue Software Research Paper.2002.
[21] Ramos, V. & Abraham, A. (2005). ANTIDS: self organized ant based clustering model for intrusion detection system. In: Proceedings of The Fourth IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST'05), pp. 977-986, Springer-Verlag, Berlin
[22] Salton, G., Wong, A. & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11): 613-620

[23] Ward, J.H. (1963). Hierarchical grouping to optimize an objective function. Journal *A Modified Ant-Based Text Clustering Algorithm with Semantic Similarity Measure* 492 JOURNAL OF SYSTEMS SCIENCE AND SYSTEMS ENGINEERING of the American Statistical Association, 58: 236-244

[24] Xia, H., Wang, S. & Yoshida, T. (2006). Toward a revised ant-based text clustering algorithm. In: Proceedings of 7th International Symposium on Knowledge and Systems Sciences, pp. 159-166, Global-Link Publisher, Hong Kong B. Everitt, Cluster Analysis, second ed., Halsted Press, New York, 1980.