Swarm Intelligence in Text Document Clustering

Xiaohui Cui
Applied Software Engineering Research Group
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6085
cuix@ornl.gov
Phone: (865)576-9654
FAX: (865)241-0337

Thomas E. Potok
Applied Software Engineering Research Group
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6085
cuix@ornl.gov
Phone: (865)574-0834
FAX: (865)241-0337

Swarm Intelligence in Text Document Clustering

Xiaohui Cui and Thomas E. Potok
Applied Software Engineering Research Group
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6085

ABSTRACT

In this chapter, we introduce three nature inspired swarm intelligence clustering approaches for document clustering analysis. The major challenge of today's information society is being overwhelmed with information on any topic they are searching for. Fast and high-quality document clustering algorithms play an important role in helping users to effectively navigate, summarize, and organize the overwhelmed information. The swarm intelligence clustering algorithms use stochastic and heuristic principles discovered from observing bird flocks, fish schools, and ant food forage. Compared to the traditional clustering algorithms, the swarm algorithms are usually flexible, robust, decentralized, and self-organized. These characters make the swarm algorithms suitable for solving complex problems, such as document clustering.

INTRODUCTION

Text document clustering is a fundamental operation used in unsupervised document organization, automatic topic extraction, and information retrieval. Clustering involves dividing a set of objects into a specified number of clusters (Jain, Murty, & Flynn, 1999). The motivation behind clustering a set of data is to find inherent structure inside the data and to expose this structure as a set of groups. The data objects within each group should exhibit a large degree of similarity while the similarity among different clusters should be minimized (Steinbach, Karypis, & Kumar, 2000; Jain, Murty, & Flynn, 1999; Zhao & Karypis, 2004). The document clustering technology is different from the classifying technology because the task of document clustering is finding natural groups that were previously unknown in the existing document achieves rather than classifying the document into different categories on the basis of pre-defined categories or some externally imposed artificial criteria. Recent studies have shown that partitional clustering algorithms are more suitable for clustering large datasets. The K-means algorithm is the most commonly used partitional clustering algorithm because it can be easily implemented and is the most efficient one in terms of the execution time. The major problem with this algorithm is that its result is sensitive to the selection of the initial partition and may converge to local optima.

To deal with the limitations existing in the traditional partition clustering methods, a number of new document clustering algorithms have been proposed with the inspiration coming from observations of natural processes, which is the area of "Natural Inspired Computing." This area has been further developed into many additional computer science fields, such as swarm intelligence, evolution computing, Neural Network, and self organization map. In recent years, swarm algorithms based on biological models, such as Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995), Ant clustering (Lumer

& Faieta, 1994) and Bird flocking (Reynolds, 1987), have been applied in the field of document clustering (Cui, Gao, & Potok, 2006; Cui & Potok, 2005; Cui, Potok, & Palathingal, 2005; Handl & Meyer, 2002). The swarm intelligence clustering algorithms can achieve better performance in document clustering than K-means clustering algorithms. In this chapter, we describe three swarm intelligence clustering approaches for document clustering analysis.

The remainder of this chapter is organized as follows: Section 2 provides the methods of representing documents in clustering algorithms and computing methods of the similarity between documents. Section 3 provides a general overview of the traditional clustering techniques and introduction of the swarm intelligence, which includes Ant colony algorithm, Particle swarm optimization algorithm and Bird flocking algorithm. Section 4 provides the detailed descriptions about how these swarm intelligence algorithms has been applied for text document clustering. The future trend and conclusion are in Section 5.

PRELIMINARIES

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 x_i corresponds to a single object and is called the feature vector. The feature vector includes proper features to represent the object. The text document objects can be represented with the Vector Space Model (VSM) (Salton, Wong, & Yang, 1975). In this model, the content of a document is formalized as a dot in the multi-dimensional space and represented by a vector d, such as $d=\{w_1, w_2,\ldots..w_n\}$, where $w_i(i = 1,2,\ldots,n)$ is the term weight of the term t_i in one document. The term weight value 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) (Salton & Buckley, 1988). The weight of term i in document j is given in Eq. (1):

$$w_{ji} = tf_{ji} * idf_{ji} = tf_{ji} * \log_2(n / df_{ji}) \qquad (1)$$

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.

The Document Similarity Metrics

The similarity between two documents needs to be measured in the clustering analysis. Over the years, two prominent ways have been proposed to compute the similarity between documents m_p and m_j. The first method is based on Minkowski distances (Salton & Buckley, 1988), given by Eq. (2):

$$D_n(m_p, m_j) = \left(\sum_{i=1}^{d_m} | m_{i,p} - m_{i,j} |^n \right)^{1/n} \qquad (2)$$

For n =2, we obtain the Euclidean distance. In order to manipulate equivalent threshold distances, considering that the distance ranges will vary according to the number of dimensions, this algorithm uses the normalized Euclidean distance as the similarity metric of two documents, m_p and m_j, in the vector space. Eq. (3) represents the distance measurement formula:

$$d(m_p, m_j) = \sqrt{\sum_{k=1}^{d_m} (m_{pk} - m_{jk})^2 / d_m} \qquad (3)$$

where m_p and m_j are two document vectors; d_m denotes the dimension number of the vector space; m_{pk} and m_{jk} stand for the documents m_p and m_j's weight values in dimension k. The other commonly used similarity measure method in the document clustering is the cosine correlation measure (Salton & Buckley, 1988), given by Eq. (4):

$$\cos(m_p, m_j) = \frac{m_p^t m_j}{| m_p || m_j |} \qquad (4)$$

where $m_p^t m_j$ denotes the dot-product of the two document vectors; |.| indicates the length of the vector. Both similarity metrics are widely used in the text document clustering literatures.

Clustering Algorithms

There are two major clustering techniques: "Partitioning" and "Hierarchical" (Jain, Murty, & Flynn, 1999). Most document clustering algorithms can be classified into these two groups. Hierarchical clustering techniques produce a nested sequence of partition, with a single, all-inclusive cluster at the top and a number of clusters of individual points at the bottom. The Hierarchical techniques can be further classified as agglomerative clustering and divisive clustering. Agglomerative clustering starts by placing every object in its own cluster and then merge these clusters into a larger cluster until all the objects are in a single cluster or until certain termination conditions are satisfied. Divisive clustering starts with all objects in one cluster, then divides the cluster into smaller and smaller clusters, until each object forms a cluster on its own or until the desired number of clusters reached and the minimum distance between two closet clusters is above a certain threshold distance. The partitioning clustering method seeks to partition a collection of documents into a set of non-overlapping groups, so as to maximize the evaluation value of clustering. The general criterion for a good partition is that objects of different cluster are "far apart" and objects within the same cluster are "close" to each other. Although the hierarchical clustering technique is often portrayed as a good clustering approach, this technique does not contain any provision for the reallocation of entities, which may have been poorly classified in the early stages of the text analysis

(Jain, Murty, & Flynn, 1999). Moreover, the time complexity of this approach is quadratic (Steinbach, Karypis, & Kumar, 2000).

In recent years, it has been recognized that the partitioning clustering technique is well suited for clustering a large document dataset due to its relatively low computational requirements (Steinbach, Karypis, & Kumar, 2000). The time complexity of the partitioning technique is almost linear, which makes it widely used. The best-known partitioning clustering algorithm is the K-means algorithm and its variants. This algorithm is simple, straightforward and is based on the firm foundation of analysis of variances. The K-means algorithm clusters a group of data vectors into a predefined number of clusters. It starts with a random initial cluster center and keeps reassigning the data objects in the dataset to the cluster centers based on the similarity between the data object and the cluster center. The reassignment procedure will not stop until a convergence criterion is met (e.g., the fixed iteration number or the cluster result does not change after a certain number of iterations).

The K-means algorithm can be summarized as:

(1) Randomly select cluster centroid vectors to set an initial dataset partition.

(2) Assign each document vector to the closest cluster centroids.

(3) Recalculate the cluster centroid vector $\boldsymbol{c_j}$ *using Eq. (5).*

$$c_j = \frac{1}{n_j} \sum_{\forall d_j \in S_j} d_j \qquad (5)$$

where $\boldsymbol{d_j}$ *denotes the document vectors that belong to cluster* $\boldsymbol{S_j}$*;* $\boldsymbol{c_j}$ *stands for the centroid vector;* $\boldsymbol{n_j}$ *is the number of document vectors that belong to cluster* $\boldsymbol{S_j}$*.*

(4) Repeat step 2 and 3 until the convergence is achieved.

The main drawback of the K-means algorithm is that its result is sensitive to the selection of the initial cluster centroids (Zhao & Karypis, 2004). Therefore, the initial selection of the cluster centroids affects the main processing of the K-means and the partition result of the dataset as well. The K-mean cluster method is also sensitive to noise and outlier data points. A small number of such outlier data may cause K-means algorithm converge to the local optima and substantially influence the finial cluster results (Selim & Ismail, 1984).

The processing of K-means is to search the local optimal solution in the vicinity of the initial solution and to refine the partition result. The same initial cluster centroids in a dataset always generate the same cluster results. However, if good initial clustering centroids can be obtained using any other techniques, the K-means would work well in refining the clustering centroids to find the optimal clustering centers (Anderberg, 1973).

SWARM INTELLIGENCE

More than 50 years ago, biologists reported that a different type of intelligence form could emerge from some social insects, fish, birds, or mammals (Bonabeau et al., 1998; Bonabeau, Dorigo, & Theraulaz, 1999). Inside an anthill, a termite swarm, a bee colony, a bird flock, a fish school, each individual does not have the requisite neuronal capacity. However, the mere interaction among a great number of individually simple creatures can lead to the emergence of intelligence, which is reactive and adaptable to the environment (Bonabeau, Dorigo, & Theraulaz, 1999). In the insect societies, the whole system is organized in a decentralized model. A large amount of autonomous units with a relatively simple and probabilistic behavior is distributed in the environment. Each unit is provided only with local information. Units do not have any representation or explicit knowledge of the global structure that they are supposed to produce or in which they evolve. They have no plan at all. In other words, the global "task" is not explicitly programmed within individuals, but emerges after the succession of a high number of elementary interactions between individuals, or between individual and environment. This type of collective intelligence model built from multiple simple individual entities has inspired a new discipline in computer science: Swarm Intelligence.

Swarm Intelligence is an artificial intelligence technique involving studies of collective behaviors in decentralized systems. It is the modeling and application of group interactions found in social insects (Dorigo, Bonabeau, & Theraulaz, 2000). Beni and Wang (Wang & Beni, 1989; Wang & Beni, 1990; Wang & Beni, 1988) first introduced the term of Swarm Intelligence in the context of cellular robotic systems. In their experiments, many agents occupy one or two-dimensional environments to generate patterns and to self-organize through interaction with the nearest neighbor. Bonabeau (Bonabeau, Dorigo, & Theraulaz, 1999; Bonabeau et al., 1998) extended the concept of swarm intelligence to any work involved with algorithm design or distributed problem-solving devices. He gave a definition of Swarm Intelligence as “any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies.” This Swarm Intelligence definition is wider and more up-to-date than the original one that only referred to the cellular robotics framework.

Currently, popular research directions in Swarm Intelligence are grounded on the following three research areas: Flocking (Reynolds, 1987), Ant Colony Optimization (ACO) (Bonabeau, Dorigo, & Theraulaz, 1999) and Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995).

Flocking

Flocking model was first proposed by Craig Reynolds (Reynolds, 1987). It is a bio-inspired computational model to simulate the animation of a flock of entities called “boid.” It represents group movement as seen in bird flocks or fish schools in nature. In this model, each boid makes its own decisions on its movement according to a small number of simple rules that react to the neighboring mates in the flock and the environment it can sense. The simple local rules of each boid generate complex global behaviors of the entire flock.

The Flocking model consists of three simple steering rules that need to be executed at each instance over time. Three basic rules include: (1) Separation: Steering to avoid collision with other boids nearby. (2) Alignment: Steering toward the average heading and matching to the velocity of the neighbor flock mates. (3) Cohesion: Steering to the average position of the neighbor flock mates. These three basic rules are sufficient to reproduce the moving behaviors of a single species bird flock on the computer. However, experiments indicate these three rules will eventually result in all boids in the simulation forming a single flock. It cannot reproduce the real phenomena in the nature: the birds or other herd animals not only keep themselves within a flock that is composed of the same species or the same colony creatures, but also keep two or multiple different species or colony flocks separated.

In (Cui, Gao, & Potok, 2006), we propose a new Multiple Specie Flocking (MSF) model to model the multiple specie bird flock behaviors. In the MSF model, in addition to the three basic action rules in the Flocking model, a fourth rule, "feature similarity rule," is added into the basic action rules of each boids to influence the motion of the boids. Based on this rule, the flock boid tries to stay close to other boids that have similar features and stay away from other boids that have dissimilar features. The strength of the attracting force for similarity boid and repulsion force for dissimilarity boid is inversely proportional to the distance between the boids and the similarity value between the boids' features.

Particle Swarm Optimization

Particle Swarm Optimization is a population based stochastic optimization technique that can be used to find an optimal, or near optimal, solution to a numerical and qualitative problem. PSO was originally developed by Eberhart and Kennedy in 1995 (Eberhart & Kennedy, 1995), inspired by the social behavior of flocking birds or schooling fish.
In the PSO algorithm, birds in a flock are symbolically represented as particles. These particles can be considered as simple agents "flying" through a problem space. A problem space in PSO may have as many dimensions as needed to model the problem space. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution's utility.

The velocity and direction of each particle moving along each dimension of the problem space are altered at each generation of movement. It is the particle's personal experience combined with its neighbors' experience that influences the movement of each particle through a problem space. For every generation, the particle's new location is computed by adding the particle's current velocity V-vector to its location X-vector. Mathematically, given a multi-dimensional problem space, the ith particle changes its velocity and location according to the following equations (Clerc, 1999; Clerc & Kennedy, 2002):

$$v_{id} = w * (v_{id} + c_1 * rand_1 * (p_{id} - x_{id}) + c_2 * rand_2 * (p_{gd} - x_{id})) \quad (6)$$

$$x_{id} = x_{id} + v_{id} \quad (7)$$

where p_{id} is the location of the particle where it experiences the best fitness value; p_{gd} is the location of the particle experienced the highest best fitness value in the whole population; x_{id} is the particle current location; c_1 and c_2 are two positive acceleration constants; d is the number of dimensions of the problem space; $rand_1$, $rand_2$ are random values in the range of (0,1); w is called the constriction coefficient (Clerc, 1999) and it is computed by Eq. (8) and Eq. (9):

$$w = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (8)$$

$$\varphi = c_1 + c_2, \varphi > 4 \quad (9)$$

Ant Colony Optimization

The Ant Colony Optimization is a heuristic algorithm that is inspired from the food foraging behavior of ants. Ant colonies are able to accomplish tasks that would be impossible to be accomplished by a single individual ant. One such task is seeking the shortest path from their nest to the food source. As ants forage they deposit a trail of slowly evaporating pheromone. Ants then use the pheromone as a guide for them to find the route between the nest and the food source if they find one. All foraging ants use the pheromone as a guide regardless of whether the pheromone is deposited by itself or other ants. Pheromones accumulate when multiple ants travel through the same path. The pheromones on the tail evaporate as well. These ants that reach the food first return before the others. The pheromone on their return trail is now stronger than the trails of other ants that have not found food or have longer distances from the food source to nest because the return trail has been traveled twice. This high pheromone volume trail attracts other ants following this trail. The pheromone content on this trail become stronger as the trail is increasing traveled and the other trail's pheromone content will become weaker because fewer ants travel these trails and the pheromone evaporates. Eventually, the trail with the highest content of pheromone and traveled by most of foraging ants will be the shortest trail between the food sources to the nest.

Marco Dorigo introduced the first ACO system in his Ph.D. thesis in 1992. The idea of the ACO algorithm is to mimic the ant's foraging behavior with "simulated ants" walking around the graph searching for the optimal solution. In the ACO algorithm, each path followed by a "simulated ant" represents a candidate solution for a given problem. The simulated ant "deposits" pheromone on the path and the volume of the pheromone is proportional to the quality of the corresponding candidate solution for the target problem. The ants searching for food choose the path(s) with the higher volume of pheromone with a greater probability than the path(s) with a low pheromone volume. Eventually, the searching ants will converge on the path that represent the optimum or near optimum solution for the target problem.

SWARM BASED DOCUMENT CLUSTERING ALGORITHM

Compared to the traditional algorithms, the swarm algorithms usually are flexible, robust, decentralized, and self-organized. These characters make the swarm algorithms potentially suitable for solving dynamic problems, such as document collection clustering. In the following session, we will introduce three nature inspired approaches for document clustering analysis. These clustering algorithms use stochastic and heuristic principles discovered from observing bird flocks, fish schools and ant food forage. Unlike other partition clustering algorithm such as K-means, these nature inspired algorithms either generate more accurate clustering results or does not require initial partitional seeds.

PSO Document Clustering Algorithm

In the past several years, PSO has been proven to be both effective and quick to solve some optimization problems (Kennedy, Eberhart & Shi, 2001). It was successfully applied in many research and application areas. In the document clustering research area, it is possible to view the clustering problem as an optimization problem that locates the optimal centroids of the clusters rather than an optimal partition finding problem. This view offers us a chance to apply PSO optimal algorithm on the clustering solution.

Similar to other partitional clustering algorithms, the objective of the PSO clustering algorithm is to discover the proper centroids of clusters for minimizing the intra-cluster distance as well as maximizing the distance between clusters. The PSO algorithm performs a global searching for solutions whereas most other partitional clustering procedures perform a local searching. In the local searching, the solution obtained usually is located in the vicinity of the solution obtained in the previous step. For example, the K-means clustering algorithm uses the randomly generated seeds as the initial clusters' centroids and refines the position of the centroids at every iteration. The refining process of the K-means algorithm indicates the algorithm only explores the very narrow vicinity surrounding the initial randomly generated centroids.

The whole clustering behavior of the PSO clustering algorithm can be classified into two stages: a global searching stage and a local refining stage. At the initial iterations, based on the PSO algorithm's particle velocity updating equation [Eq. (6)], the particle's initial velocity v_{id}, the two randomly generated values ($rand_1$, $rand_2$) at each generation and the inertia weight factor w provide the necessary diversity to the particle swarm by changing the momentum of particles to avoid the stagnation of particles at the local optima. Multiple particles parallel searching, using multiple different solutions at a time, can explore more areas in the problem space. The initial iterations can be classified as the global searching stage. After several iterations, the particle's velocity will gradually reduce and the particle's explore area will shrink while the particle approaching the optimal solution. The global searching stage gradually changes to the local refining stage. By selecting different parameters in the PSO algorithm, we can control the shift time from the global searching stage to the local refining stage. The later the particle shift from the global searching stage to the local refining stage, the greater the possibility it can find the global optimal solution.

In a recent study (Cui, Potok, & Palathingal, 2005), a PSO document clustering algorithm for clustering document collection was proposed. In the PSO document clustering algorithm, the multi-dimensional document vector space is modeled as a problem space. Each term in the document dataset represents one dimension of the problem space. Each document vector can be represented as a dot in the problem space. The whole document dataset can be represented as a multiple dimension space with a large number of dots in the space. One particle in the swarm represents one possible solution for clustering the document collection. Therefore, a swarm represents a number of clustering solution candidates for the document collection. Each particle maintains a matrix $X_i = (C_1, C_2, \ldots, C_i, .., C_k)$, where C_i represents the ith cluster centroid vector and k is the number of clusters. According to its own experience and the knowledge of its neighbors, the particle can adjust the centroid vector's position in the vector space at each generation.

The PSO clustering algorithm can be summarized as:

(1) At the initial stage, each particle randomly chooses k different document vectors from the document collection as the initial cluster centroid vectors.

(2) For each particle:

(a) Assign each document vector in the document set to the closest centroid vector.

(b) Calculate the fitness value based on Eq. (7).

(c) Using the velocity and particle position to update Eq. (6a) and (6b) and to generate the next solutions.

(3) Repeat step (2) until one of the following termination conditions is satisfied.

(a) The maximum number of iterations is exceeded

or

(b) The average change in centroid vectors is less than a predefined value.

Contrary to the local searching in the K-means algorithm, the PSO clustering algorithm performs a global search in the entire solution space. Utilizing the PSO algorithm's optimal ability, if given enough time, the PSO clustering algorithm could generate more compact clustering results from the document datasets than the traditional K-means clustering algorithm. However, in order to cluster the large document datasets, PSO requires much more iteration (generally more than 500 iterations) to converge to the optima than the K-mean algorithm does. Although the PSO algorithm is inherently parallel and can be implemented using parallel hardware, such as a computer cluster, the computation requirement for clustering extremely huge document datasets is still high. In terms of execution time, the K-means algorithm is the most efficient one for a large dataset (Al-Sultan & Khan, 1996). The K-means algorithm tends to converge faster than the PSO, but it usually only finds the local maximum. Therefore, it is a dilemma

regarding choosing the algorithm for clustering a large document dataset. Based on this reason, a hybrid PSO+K-means document clustering algorithm was proposed and studied.

Hybrid PSO+K-Means Clustering Algorithm

As we mentioned in the previous session, the K-means algorithm result is sensitive to the selection of the initial cluster centroids and the outlier data points. A small number of such outlier data may cause K-means algorithm converge to the local optima and substantially influence the finial cluster results (Selim & Ismail, 1984). The initial selection of the cluster centroids decides the main processing of K-means and the partition result of the dataset as well. The main processing of K-means is to search the local optimal solution in the vicinity of the initial solution and to refine the partition results. If the right initial clustering centroids can be obtained using any of the other techniques, the K-means would work well in refining the clustering centroids to find the optimal clustering centers (Anderberg, 1973). It is necessary to employ some other global optimal searching algorithm for generating the initial cluster centroids. The PSO algorithm can be used to generate good initial cluster centroids for the K-means. In this section, we present a hybrid PSO+K-means document clustering algorithm that performs a faster document clustering and can avoid being trapped in a local optimal solution.

The hybrid PSO+K-means algorithm includes two modules, the PSO module and the K-means module. The global searching stage and local refine stage are accomplished by these two modules, respectively. In the initial stage, the PSO module is executed for a short period (50 to 100 iterations) to discover the vicinity of the optimal solution by a global search and at the same time to avoid consuming high computation. The result from the PSO module is used as the initial seed for the K-means module. The K-means algorithm is applied to refine and generate the final results. The whole approach can be summarized as:

(1)Starting the PSO clustering process until the maximum number of iterations is exceeded

(2) Inheriting clustering result from PSO as the initial centroid vectors of K-means module.

(3)Starting K-means process until maximum number of iterations is reached.

In the PSO+K-means algorithm, the ability of the global searching of the PSO algorithm and the fast convergence of the K-means algorithm are combined. The PSO algorithm is used at the initial stage to help discovering the vicinity of the optimal solution by a global search. The result from PSO is used as the initial seed of the K-means algorithm, which is applied for refining and generating the final result.

Some researchers (Omran, Engelbrecht, & Salman, 2005; Merwe & Engelbrecht, 2003) proposed using the K-means+PSO hybrid algorithm for clustering the low dimension datasets. They argue that the K-means algorithm tends to converge faster than other clustering algorithms, but usually with less accuracy. The performance of the clustering algorithm can be improved by seeding the initial swarm with the result of the K-means

algorithm. To compare the performance of different kinds of hybrid algorithms, the K-means+PSO, the PSO+K-means and the K-means+PSO+K-means algorithms were applied on document datasets, respectively (Cui & Potok, 2005). The results from the three different hybrid K-means algorithms indicate the PSO+K-means hybrid algorithm generate the most accurate clustering result comparing to other two hybrid clustering algorithms. The experimental results (Cui & Potok, 2005) illustrates that the sequence of the hybrid PSO K-means algorithms is very important. Performing the K-means clustering in advance of the PSO clustering module will reduce the global searching ability of the PSO algorithm and reduce the whole algorithm's performance.

Ant Clustering Algorithm

The ant clustering algorithm is one type of bottom-up approach for solving the data clustering and classification problems. Ant Colony Optimization is not just a data clustering application of the Ant Colony Optimization Algorithm, it is another bio-inspired algorithm derived from several ant species' natural behaviors. This algorithm and ant colony optimization algorithm are commonly be considered belong to the same ant swarm algorithm category.

The ant clustering algorithm inspired from entomologists' observation that some species of ants have the capability to sort large corpses into clusters. Deneubourg and his colleagues (Deneubourg et al., 1991) proposed a "Basic Model" to explain the ants' behavior of piling corpses and eggs. In their study, a population of ant-like agents randomly moved in a 2 dimensional grid. Each agent only follows one simple rule: moving randomly in the grid and establishing a probability of picking up the data object it meets if it is free of load, or establishing a probability of dropping down the data object if it is loading the data object. After several iterations, a clustering result emerges from the collective activities of these agents. Lumer, Faieta and other researchers (Lumer & Faieta, 1994) extended this "Basic Model" and introduced the standard Ant Clustering Algorithm for multi-dimensional data clustering analysis. In this standard ant clustering algorithm, the input data set composed of ***N d-***dimensional vector objects are spread randomly over a bi-dimensional grid of size ***m*n*** and each grid cell can only be occupied by a single data object at any time. A colony of artificial ants is allowed to randomly move on the grid for picking up, carrying, and dropping off data objects based on some probabilistic rules. The ants can perceive the neighborhood in the environment. Ants perform two probabilistic actions: (1) Picking: if an ant finds a data object in its neighborhood and it is not holding any object, it may pick up the data object; the probability of the ant making a decision to pick up the object can be calculated by Eq.(11); (2) Dropping: An ant carrying a data object can drop the object on a free grid cell at any stage; the probability of the ant making a decision to drop its holding object can be calculated by Eq. (12).

$$P_{pickup}(i) = \left(\frac{k_p}{k_p + f(i)} \right)^2 \qquad (11)$$

$$P_{drop}(i) = \left(\frac{f(i)}{k_d + f(i)} \right)^2 \qquad (12)$$

where $\boldsymbol{k_d}$ and $\boldsymbol{k_p}$ are adjustable parameters and ***f(i)*** is the measure of the average swarm similarity of object ***i*** with other objects in the neighborhood of ***i***. ***f(i)*** can be calculated by Eq. (13)

$$. \ f(i) = \max\left(0, \frac{1}{|s|} \sum_{j \in Neigh(r)} \left(1 - \frac{d(i,j)}{\alpha} \right) \right) \qquad (13)$$

where ***Neigh(r)*** denotes all data objects in the object ***i***'s neighborhood area within a radius ***r***. ***d(i,j)*** denotes the distance of data object ***i*** and ***j***. $\boldsymbol{\alpha}$ is a parameter that defines the scale of dissimilarity. It is a key parameter that directly affects the convergence of the algorithm and the clustering number of the result. The algorithm pseudo code is presented in the list as follows:

```
Randomly deploy every data object i on an un-occupied cell of the grid
Randomly place every artificial ant k on an un-occupied cell of the grid
LOOP
        FOR every ant k
                Move to a randomly selected neighbor un-occupied cell
                If object i in the ant's neighborhood cell
                If ant k doesn't carry any data object
                        Copute f(i) and Pp(i)
                        Pickup item i with probability Pp(i)
                If ant k carrying data object j AND there are at least one un-occupied cell
                in the neighborhood
                        Compute f(j) and Pd(j)
                        Drop object j at the nearest un0occupied cell with probability Pd(j)
        END
END
```

Ant clustering algorithm is a partitioning algorithm that does not require a prior knowledge of the probable number of clusters or the initial partition. In 2002, Handl and Meyer employed the Ant Clustering Algorithm for their visual document searching system that targets for classifying online documents by contents-similarity. The algorithm generates a clustering of a given set of data through the embedding of the high-dimensional data items on a two-dimensional grid for easy clustering result retrieval and visualization. Wu (Wu & Shi, 2001) combined Ant clustering algorithm with the clustering algorithm (K-means) to achieve more accurate clustering results. According to the reported experiment results of Wu (Wu & Shi, 2001) and Handl (Handl & Meyer, 2002), the ant clustering algorithms for document clustering can generate much better clustering results than that from the K-means algorithm. However, in Ant clustering algorithm, the clustered data objects themselves do not have any mobility. The data objects' movements have to be implemented through the movements of a small number

of ant agents, which slow down the clustering speed. Since each ant agent that carries an isolated data object does not communicate with any other ant agents, it does not know the best location to drop the data object. The ant agent has to move or jump randomly in the grid space until it finds a place that satisfies its object dropping criteria, which usually consumes a large amount of computation time. Our experiments show that the ant clustering algorithm needs more iterations to generate an acceptable clustering result.

Flocking Based Clustering Algorithm

Inspired by the bird's ability of maintaining a flock as well as separating different species or colony flocks, Cui and colleague (Cui, Gao, & Potok, 2006) proposed a new Multiple Species Flocking (MSF) model for the multiple species bird flock behaviors. In the MSF model, in addition to the three basic action rules in the Flocking model, a fourth rule, the "feature similarity rule," is added into the basic action rules of each boid to influence the motion of the boids. Based on this rule, the flock boid tries to stay close to other boids that have similar features and stay away from other boids that have dissimilar features. The strength of the attracting force for similar boids and repulsion force for dissimilar boids is inversely proportional to the distance between the boids and the similarity value between the boids' features.

One application of the MSF model is document clustering (Cui & Potok, 2006). The MSF clustering algorithm uses a simple and heuristic way to cluster document datasets. In the MSF clustering algorithm, each document is projected as a boid in a two- dimensional virtual space. The document is represented as the feature of the boid. The boids that share similar document features (similar as the bird's species in nature) will automatically group together and became a boid flock. Other boids that have different document features will stay away from this flock. After several iterations, the simple local rules followed by each boid results in generating complex global behaviors of the entire document flock, and eventually a document clustering result is emerged.

This approach uses stochastic and heuristic principles discovered from observing bird flocks or fish schools. Unlike other partition-clustering algorithms, such as K-means, the Flocking based algorithm does not require initial partition seeds. The algorithm generates a clustering of a given set of data through the embedding of the high-dimensional data items on a two-dimensional grid for easy clustering result retrieval and visualization. The simple local rules followed by each flock boid results in generating complex global behaviors of the entire document flock, which eventually result in a clustering of the documents. The efficiency of the algorithm is evaluated with both a synthetic dataset and a real document collection that includes 100 news articles collected from the Internet (Cui, Gao, & Potok, 2006; Cui & Potok, 2006). The synthetic dataset consists of four data types, each including 200 two dimensional (x, y) data objects. x and y are distributed according to the Normal distribution. This is the same dataset that has been used by Lumer and Faieta for their Ant clustering algorithm (Lumer & Faieta, 1994). There are many studies in the document clustering (Ramos & Merelo, 2002; Handl & Meyer, 2002) to use this synthetic dataset as a performance evaluation benchmark. In the real document collection dataset, a document collection that contains 100 news articles

was used. These articles are collected from the Internet at different time stages and have been categorized by human experts and manually clustered into 12 categories. Results have shown that the Flocking-clustering algorithm achieves better performance for real document clustering than the K-means and the Ant clustering algorithm.

FUTURE TRENDS

Intelligence analysts are currently overwhelmed with the tremendous amount of information streams generated everyday. There is a lack of comprehensive tools that can real-time analyze the information streams. Document clustering analysis plays an important role in improving the accuracy of information retrieval. However, most clustering technologies can only be applied for analyzing the static document collection because they normally require a large amount of computation resource and a long time to get accurate results. It is very difficult to cluster dynamic changed text information streams on an individual computer.

The swarm clustering algorithms described in this chapter can continually refine the clustering result and quickly react to the change of individual data. This character of the swarm clustering algorithm enables it to be suitable for clustering dynamic changed document information, such as the text information stream. However, the computational requirement for real-time clustering a large amount of text collection is high. In the information society of today, the tremendous amounts of text information are continuously accumulated. Inevitably, the swarm intelligence clustering algorithm approach of using a single processor machine to cluster the dynamic text stream requires a large amount of memory and a faster execution CPU. Since the decentralized character of the swarm intelligence algorithms, a distributed approach is a very natural way to improve the clustering speed of the algorithms. The future work will be developing a distributed swarm approach for the text information stream clustering and analyzing.

ACKNOWLEDGMENTS

Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725.
Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

REFERENCES

Al-Sultan, K., & Khan, M. M. (1996). Computational experience on four algorithms for the hard clustering problem. *Pattern Recognition Letters*, 17(3), 295-308.
Anderberg, M. R. (1973). *Cluster Analysis for Applications*. New York: Academic Press, Inc.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence from natural to artificial systems*. New York: Oxford University Press.
Bonabeau, E., Henaux, F., Guerin, S., Snyers, D., Kuntz, P., & Theraulaz, G. (1998). Routing in telecommunications networks with ant-like agents. In (pp. 60).
Clerc, M. (1999). *The swarm and the queen: towards a deterministic and adaptive particle swarm optimization.* Paper presented at the Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA.
Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation,* 6(1), 58-73.
Cui, X., Gao, J., & Potok, T. E. (2006). A Flocking Based Algorithm for Document Clustering Analysis. *Journal of System Architecture*(Special issue on Nature Inspired Applied Systems).
Cui, X., & Potok, T. E. (2005). Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm. *Journal of Computer Sciences,* Special Issue, 27-33.
Cui, X., & Potok, T. E. (2006). *A distributed flocking approach for information stream clustering analysis,* Las Vegas, NV, United States.
Cui, X., Potok, T. E., & Palathingal, P. (2005, 2005). *Document clustering using particle swarm optimization.* Paper presented at the 2005 IEEE Swarm Intelligence Symposium, 8-10 June 2005, Pasadena, CA, USA.
Deneubourg, J. L., Goss, S., SendovaFranks, N., Detrain, C., & Chretien, L. (1991, 1991). *The dynamics of collective sorting robot-like ants and ant-like robots.* Paper presented at the Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats, Cambridge, MA, USA.
Dorigo, M., Bonabeau, E., & Theraulaz, G. (2000). Ant algorithms and stigmergy. *Future Generation Computer Systems,* 16(8), 851-871.
Eberhart, R., & Kennedy, J. (1995). *A new optimizer using particle swarm theory.* Paper presented at the Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan.
Handl, J., & Meyer, B. (2002). *Improved ant-based clustering and sorting in a document retrieval interface,* Granada, Spain.
Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys,* 31(3), 264-323.
Kennedy, J. F., Eberhart, R. C., Shi, Y., & NetLibrary, I. (2001). *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers.
Lumer, E. D., & Faieta, B. (1994, 1994). *Diversity and adaptation in populations of clustering ants.* Paper presented at the Proceedings of 3rd International Conference on Simulation of Adaptive Behaviour, 8-12 Aug. 1994, Brighton, UK.
Omran, M. G., Engelbrecht, A. P., & Salman, A. (2005). A color image quantization algorithm based on Particle Swarm Optimization. *Informatica (Ljubljana),* 29(3), 261-269.
Ramos, V., & Merelo, J. (2002, feb, 2002). *Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning.* Paper presented at the 1st Spanish Conference on Evolutionary and Bio-Inspired Algorithms, Merida, Spain.
Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics (ACM),* 21(4), 25-34.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513-523.
Salton, G., Wong, A., & Yang, C. S. (1975). Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11), 613-620.
Selim, S. Z., & Ismail, M. A. (1984). K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1), 81-87.
Steinbach, M., Karypis, G., & Kumar, V. (2000, 2000). *A comparison of document clustering techniques.* Paper presented at the KDD Workshop on Text Mining.
van der Merwe, D. W., & Engelbrecht, A. P. (2003, 2003). *Data clustering using particle swarm optimization.* Paper presented at the 2003 Congress on Evolutionary Computation, 8-12 Dec. 2003, Canberra, ACT, Australia.
Wang, J., & Beni, G. (1988). *Pattern generation in cellular robotic systems*, Arlington, VA, USA.
Wang, J., & Beni, G. (1989). *Cellular robotic system with stationary robots and its application to manufacturing lattices*, Albany, NY, USA.
Wang, J., & Beni, G. (1990). *Distributed computing problems in cellular robotic systems*, Ibaraki, Japan.
Wu, b., & Shi, Z. (2001, 2001). *A clustering algorithm based on swarm intelligence.* Paper presented at the 2001 International Conferences on Info-tech and Info-net. Proceedings, 29 Oct.-1 Nov. 2001, Beijing, China.
Zhao, Y., & Karypis, G. (2004). Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3), 311-331.

KEY TERMS AND THEIR DEFINITIONS

Document Clustering:
Document Clustering is the process dividing a set of document collections into different number of groups based on Document contents-similarity.

Swarm Intelligence:
Swarm intelligence is an emerging field of biologically-inspired artificial intelligence based on the collective behavior model of social insect colonies and other animal societies.

Vector Space Model:
The vector space model (VSM) is an algebraic model used for information filtering and information retrieval. It represents natural language documents in a formal manner by the use of vectors in a multi-dimensional space. It was used for the first time by the SMART Information Retrieval System. (http://en.wikipedia.org)

Hierarchical Clustering:
The hierarchical clustering techniques produce a nested sequence of partition, with a single, all-inclusive cluster at the top and single clusters of individual points at the bottom.

Partitioning Clustering:

The partitioning clustering method seeks to partition a collection of documents into a set of non-overlapping groups, so as to maximize the evaluation value of clustering.

Flocking Model:
The Flocking Model is a bio-inspired computational model for simulating the animation of a flock of entities. The Flocking model was first proposed by Craig Reynolds in 1987.

Particle Swarm Optimization:
The Particle Swarm Optimization (PSO) is a population based stochastic optimization technique that can be used to find an optimal, or near optimal, solution to a numerical and qualitative problem. PSO was originally developed by Eberhart and Kennedy in 1995, inspired by the social behavior of flocking birds or a school of fish.

Ant Colony Optimization:
The Ant Colony Optimization (ACO) is a heuristic algorithm that is inspired from the food foraging behavior of ants. Dorigo introduced the first ACO system in his Ph.D. thesis in 1992.