# 11

# Clustering Ensemble Using ANT and ART

Yan Yang[1], Mohamed Kamel[2], and Fan Jin[1]

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan, 610031, China yyang@home.swjtu.edu.cn
[2] Pattern Analysis and Machine Intelligence Lab, Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada mkamel@uwaterloo.ca

**Summary.** This chapter presents a clustering ensemble model using ant colony algorithm with validity index and ART neural network. Clusterings are visually formed on the plane by ants walking, picking up or dropping down projected data objects with different probability. The clustering validity index is used to evaluate the performance of algorithm, find the best number of clusters and reduce outliers. ART is employed to combine the clusterings produced by ant colonies with different moving speed. Experiments on artificial and real data sets show that the proposed model has better performance than that of single ant colony clustering algorithm with validity index, the ART algorithm, and the LF algorithm.

## 11.1 Introduction

Ant colony is a kind of social insects, which is capable of selforganization, pheromone communication, distribution, flexibility, and robustness. Researchers have designed a number of successful algorithms such as Ant Colony Optimization and Ant Colony Routing in diverse application fields such as combinatorial optimization, communications networks, and robotics [4]. The ant colony clustering algorithm is inspired by the behavior of ant colonies in clustering their corpses and sorting their larvae. Deneubourg et al. [9] proposed a basic model to explain the clustering behavior. In this model, artificial ants are allowed to randomly move, pick up and drop objects according to the number of similar surrounding objects so as to cluster them. Lumer and Faieta [21] expanded Deneubourg's model to the LF algorithm that is based on a local similarity density in order to make it suitable for data clustering. Ramos and Merelo [24] studied ant-clustering systems with different ant speeds for textual document clustering. Handl and Meyer [15] used inhomogeneous ant populations with "jumps" for document retrieval. Monmarche [22] described an AntClass algorithm in which several items are allowed to be on the same cell corresponding to a cluster. The AntClass algorithm uses stochastic principles of ant colony in conjunction with the deterministic principles of the K-means algorithm. In a similar way, in [30] CSIM algorithm combined CSI model

Y. Yang et al.: *Clustering Ensemble Using ANT and ART*, Studies in Computational Intelligence (SCI) **34**, 243–264 (2006)
www.springerlink.com

(Clustering based on Swarm Intelligence) and K-means algorithm. An ant-based clustering algorithm is aggregated with the fuzzy c-means algorithm in [20].

As the data to be clustered is usually unlabelled, measures that are commonly used for document classification such as the F-measure cannot be used here. Instead we need to use measures that reflect the goodness of the clustering. Cluster validity indices have been proposed in the literature to address this point [26]. Several clustering methods use validity index to find the best number of clusters [14,31,34]. Halkidi et al. [13] proposed multi representative clustering validity index that is suitable for non-spherical cluster shapes.

ART (Adaptive Resonance Theory) neural networks were developed by Grossberg [12] to address the problem of stability-plasticity dilemma. The ART network self-organizes in response to input patterns to form a stable recognition cluster. Models of unsupervised learning include ART1 [5] for binary input patterns, ART2 [6] and ART-2A [7] for analog input patterns, and fuzzy ART [8] for "fuzzy binary" inputs, i.e. analog numbers between 0 and 1. Many variations of the basic unsupervised networks have been adapted for clustering. Tomida et al. [27] applied fuzzy ART as a clustering method for analyzing the time series expression data during sporulation of Saccharomyces cerevisiae. He et al. [17] used fuzzy ART to extract document cluster knowledge from the Web Citation Database to support the retrieval of Web publications. Hussin and Kamel [18] proposed a neural network based document clustering method by using a hierarchically organized network built up from independent SOM (Self-Organizing Map) and ART neural networks.

Clustering ensembles have emerged as a powerful method for improving the quality and robustness of the clusterings. However, finding a combination of multiple clusterings is a more difficult and challenging task than combination of supervised classifications. Without the labeled pattern, there is no explicit correspondence between cluster labels in different partitions of an ensemble. Another intractable label correspondence problem results from different partitions containing different numbers of clusters. Recently a number of approaches have been applied to the combination of clusterings, namely the consensus function, which creates the combined clustering [29]. A co-association matrix was introduced for finding a combined clustering in [11]. Co-association values represent the strength of association between objects appearing in the same cluster. The combined clustering comes from the co-association matrix by applying a voting-type algorithm. Strehl and Ghosh [25] represented the clusters as hyperedges on a graph whose vertices correspond to the objects to be clustered, and developed three hypergraph algorithms: CSPA, HGPA, and MCLA for finding consensus clustering. Topchy et al. [28] proposed new consensus function based on mutual information approach. They employed combination of so-called weak clustering algorithm related to intra-class variance criteria. Recent approaches to combine cluster ensembles based on graph and information theoretic methods appear in [1] and [2]. An approach based on re-labeling each bootstrap partition using a single reference partition is presented in [10].

Neural network ensemble is a learning paradigm where several neural networks are jointly used to solve a problem. In [32], multistage ensemble neural network

model was used to combine classifier ensemble results. Ensemble of SOM neural networks has been also used for image segmentation where the pixels in an image are clustered according to color and spatial features with different SOM neural networks, and the clustering results are combined as the image segmentation [19].

In this chapter, an ensemble model, i.e. combination of ant colony clustering algorithms with validity index (called ACC-VI) and ART network, is applied to clustering. Clusterings are visually formed on the plane by ants walking, picking up or dropping down projected data objects with different probability. The clustering validity index is used to evaluate the performance of the algorithm, find the best number of clusters and reduce outliers. ART network is employed to combine the clusterings. Experiments on artificial and real data sets show that the proposed model has better performance than that of the individual ACC-VI algorithm, the ART-2A algorithm, and the LF algorithm.

The rest of the chapter is organized as follows. Section 2 introduces the ACC-VI algorithm. Section 3 describes the ART algorithm. Section 4 presents the clustering ensemble model. Section 5 reports the results of the experiments conducted to evaluate the performance of the proposed model. Finally, Section 6 offers a conclusion of the chapter.

## 11.2 Ant Colony Clustering Algorithm with Validity Index (ACC-VI)

### 11.2.1 Ant Colony Clustering Algorithm

The ant colony clustering algorithm is based on the basic LF model and its added feature proposed by Lumer and Faieta [21] and the ant-based clustering algorithm by Yang and Kamel [33]. First, data objects are randomly projected onto a plane. Second, each ant chooses the object at random, and picks up or moves or drops down the object according to picking-up or dropping probability with respect to the similarity of current object within the local region by probability conversion function. Finally, clusters are collected from the plane.

Let us assume that an ant is located at site $\gamma$ at time $t$, and finds an object $o_i$ at that site. The local density of objects similar to type $o_i$ at the site $\gamma$ is given by

$$f(o_i) = \max\{0, \frac{1}{s^2} \sum_{o_j \varepsilon Neigh_{s\times s}(\gamma)} [1 - \frac{d(o_i, o_j)}{\alpha(1 + ((v-1)/v_{max}))}]\} \tag{11.1}$$

where $f(o_i)$is a measure of the average similarity of object $o_i$ with the other objects $o_j$ present in its neighborhood. $Neigh_{s\times s}(\gamma)$ denotes the local region.It is usually a square of $s \times s$ sites surrounding site $\gamma$. $d(o_i, o_j)$ is the distance between two objects $o_i$ and $o_j$ in the space of attributes.The Cosine distance is computed as

$$d(o_i, o_j) = 1 - sim(o_i, o_j) \tag{11.2}$$

where $sim(o_i,o_j)$ reflects the similarity metric between two objects. It measures the cosine of the angle between two objects (their dot product divided by their magnitudes)

$$sim(o_i,o_j) = \frac{\sum_{k=1}^{q}(o_{ik}.o_{jk})}{\sqrt{\sum_{k=1}^{q}(o_{ik})^2 \cdot \sum_{k=1}^{q}(o_{jk})^2}} \tag{11.3}$$

where $q$ is the number of attributes. As the objects become more similar, the Cosine similarity $sim(o_i,o_j)$ approaches 1 and their Cosine distance approaches 0.

As shown in formula (1), $\alpha$ is a factor that defines the scale of similarity between objects. Too Large values of $\alpha$ will result in making the similarity between the objects larger and forces objects to lay in the same clusters. When $\alpha$ is too small, the similarity will decrease and may in the extreme result in too many separate clusters. On the other hand, parameter $\alpha$ adjusts the cluster number and the speed of convergence. The bigger $\alpha$ is, the smaller the cluster number and the faster the algorithm converges.

In formula (1), the parameter $v$ denotes the speed of the ants. Fast mov-ing ants form clusters roughly on large scales, while slow ants group objects at smaller scales by placing objects with more accuracy. Three types of speed in different ant colonies are considered [33].

— *v is a constant.*All ants move with the same speed at any time;
— *v is random.*The speed of each ant is distributed randomly in $[1, v_{max}]$, where $v_{max}$ is the maximum speed;
— *v is randomly decreasing.*The speed term starts with large value (form-ing clusters), and then the value of the speed gradually decreases in a random manner (helping ants to cluster more accurately).

The picking-up and dropping probabilities both are a function of $f(o_i)$ that converts the average similarity of a data object into the probability of picking-up or dropping for an ant. The converted approaches are based on: the smaller the similarity of a data object is (i.e. there aren't many objects that belong to the same cluster in its neighborhood), the higher the picking-up probability is and the lower the dropping probability is; on the other hand, the larger the similarity is, the lower the picking-up probability is (i.e. objects are unlikely to be removed from dense clusters) and the higher the dropping probability is. The sigmoid function is used as probability conversion function in our algorithm [33]. Only one parameter needs to be adjusted in the calculation.

The picking-up probability $P_p$ for a randomly moving ant that is currently not carrying an object to pick up an object is given by

$$P_p = 1 - sigmoid(f(o_i)) \tag{11.4}$$

The dropping probability $P_d$ for a randomly moving loaded ant to deposit an object is given by

$$P_d = sigmoid(f(o_i)) \tag{11.5}$$

where

$$sigmoid(x) = \frac{1}{1+e^{-\beta x}} \tag{11.6}$$

and β is a constant that can speed up the algorithm convergence if it is in-creased. Selecting larger values for β can help ants to drop faster the outliers at the later stages of algorithm [33].

### 11.2.2 Clustering Validity Index

Halkidi et al. [13] proposed multi representative clustering validity index that is based on cluster compactness and cluster separation. A clustering of data set into $c$ clusters can be represented as $D = \{U_1, U_2 ..., U_c\}$,where $U_i = \{u_{i1}, u_{i2}, ...u_{ir_i}\}$ is the set of representative points of cluster $i, r_i$ is the number of representative point of the $i$th cluster.

The standard deviation of the ith cluster is defined as [13]

$$stdev(U_i) = \sqrt{\frac{1}{n_i - 1} \sum_{k=1}^{n_i} d^2(x_k, m_i)} \tag{11.7}$$

where $n_i$ is the number of data in the $i$th cluster, $d$ is the distance between $x_k$ and $m_i$. $x_i$ is the data belonging to the $i$th cluster, and $m_i$ is the mean of the $i$th cluster.

Intra-cluster density is defined as the average density within clusters, that is, the number of points that belong to the neighborhood of representative points of the clusters [13]. A bigger Intra-cluster density value indicates a more compacted cluster. It is defined by

$$Intra\_den(c) = \frac{1}{c} \sum_{i=1}^{c} \frac{1}{r_i} \sum_{j=1}^{r_i} \frac{density(u_{ij})}{stdev(U_i)}, c > 1 \tag{11.8}$$

The term $density(u_{ij})$ is defined by

$$density(u_{ij}) = \sum_{l=1}^{n_i} f(x_l, u_{ij}) \tag{11.9}$$

where $x_l$ belongs to the $i$th cluster, $u_{ij}$ is the $j$th representative point of $i$th cluster, $n_i$ is the number of the $i$th cluster, and $f(x_l, u_{ij})$ is defined by

$$f(x_l, u_{ij}) = \begin{cases} 1 & , \quad d(x_l, u_{ij}) \leq stdev(U_i) \\ 0 & , \quad otherwise \end{cases} \tag{11.10}$$

Inter-cluster density is defined as the density between clusters [13]. For well-separated clusters, it will be significantly low. It is defined by

$$Inter\_den(c) = \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j \neq i}}^{c} \frac{d(close\_rep(i), close\_rep(j))}{stdev(U_i) + stdev(U_j)} density(z_{ij}), c > 1 \tag{11.11}$$

where $close\_rep(i)$ and $close\_rep(j)$ are the closest pair of representatives of the $i$th and $j$th clusters, $z_{ij}$ is the middle point between the pair points $close\_rep(i)$ and $close\_rep(j)$. The term $density(z_{ij})$ is defined by

$$density(z_{ij}) = \frac{1}{n_i + n_j} \sum_{l=1}^{n_i+n_j} f(x_l, z_{ij}) \tag{11.12}$$

where $x_l$ belongs to the $i$th and $j$th clusters, $n_i$ is the number of the $i$th cluster, $n_j$ is the number of the $j$th cluster, and $f(x_l, z_{ij})$ is defined by

$$f(x_l, z_{ij}) = \begin{cases} 1, & d(x_l, z_{ij}) \leq (stdev(U_i) + stdev(U_j))/2 \\ 0, & otherwise \end{cases} \tag{11.13}$$

Clusters' separation measures separation of clusters. It contains the distances between the closest clusters and the Inter-cluster density, and is defined as follows [13]

$$Sep(c) = \sum_{i=1}^{c} \sum_{\substack{j \\ j \neq i}}^{c} \frac{d(close\_rep(i), close\_rep(j))}{1 + Inter\_den(c)}, c > 1 \tag{11.14}$$

Then the validity index *CDbw*, which is called "Composing Density Between and Within clusters" [13], is defined as

$$CDbw(c) = Intra\_den(c) \cdot Sep(c), c > 1. \tag{11.15}$$

### 11.2.3 ACC-VI Algorithm

A good clustering algorithm produces partitions of the data such that the Intra-cluster density is significantly high, the Inter-cluster density and Clusters' separation are significantly low, and the validity index *CDbw* has a maximum, which corresponds to natural number of clusters. In [13], experiments showed that *CDbw* can be used to find the optimal number of clusters at the maximum value. So we use *CDbw* not only to evaluate the clustering algorithm, but also to find the best number of clusters.

In the ant colony clustering algorithm, the outliers with dissimilarity to all other neighborhood are dropped alone. The local clustering validity index is taken into account to reduce outliers in our algorithm. The process is described below. First, try to drop each outlier into each cluster, recalculate the new local *CDbw*, and compare to the old value for each cluster. Then, move the outlier to the cluster at the highest difference.

A pseudo code of ACC-VI algorithm is listed in Table 1. Essentially, the algorithm works as follows. Firstly, the data objects are projected onto a plane, that is, a pair of coordinates is given to each object randomly. Each ant is marked as unloaded and chooses an object at random initially. Secondly, the similarity $f(o_i)$ for each ant walking randomly is computed by formula (1). In the first case, each ant is unloaded, that is ants are not holding any objects. The picking-up probability $P_p$ is calculated by formula (4). If $P_p$ is greater than a random probability and an

object is not picked up by the other ants simultaneously, the ant picks up this object, moves it to a new position, and marks itself as loaded. On the other hand, if $P_p$ is less than a random probability, the ant does not pick up this object and re-selects another object randomly. In the second case, the ant is loaded, i.e. holding an object. The dropping probability $P_d$ is calculated by formula (5). If $P_d$ is greater than a random probability, the ant drops the object, marks itself as unloaded, and re-selects a new object randomly. Otherwise, the ant continues moving the object to a new position. The third step is to collect the clustering results on the plane. Whether crowded or isolated for an object can be determined by the number of its neighbor. If an object is isolated, that is the number of its neighbor is less than a given constant, the object is labeled as an outlier. On the other hand, if the object is in a crowd, that is the number of its neighbor is more than the given constant, it is given a labeling number denoting a cluster and is given same number recursively to those objects who are the neighbors of this object within a local region. At the fourth step, the validity index *CDbw* is calculated so as to find the optimal number of clusters. Finally, try to drop outlier at the cluster with the highest *CDbw* difference.

## 11.3 ART Algorithm

ART (Adaptive Resonance Theory) models are neural networks that develop stable recognition codes by self-organization in response to arbitrary sequences of input patterns. They are capable of solving well-known dilemma, stability-plasticity. How can a learning system be designed to remain plastic or adaptive in response to significant events and yet remain stable in response to irrelevant events? That means new clusters can be formed when the environment does not match any of the stored pattern, but the environment cannot change stored pattern.

A typical ART network consists of three layers: input layer (F0), com-parison layer (F1) and competitive layer (F2) with $N, N$ and $M$ neurons, respectively (see Fig. 11.1). The input layer F0 receives and stores the input patterns. Neurons in the input layer F0 and comparison layer F1 are one-to-one connected. F1 combines input signals from F0 and F2 layer to measure similarity between an input signal and the weight vector for the specific cluster unit. The competitive layer F2 stores the prototypes of input clusters. The cluster unit with the largest activation becomes the candidate to learn the input pattern (winner-take-all). There are two sets of connections, top-down and bottom-up, between each unit in F1 and each cluster unit in F2. Interactions between F1 and F2 are controlled by the orienting subsystem using a vigilance threshold ρ. The learning process of the network can be described as follows (refer to ART1 [3, 5]).

For a non-zero binary input pattern $x$ ($x_j$ {0, 1}, $j$=1, 2, .... , $N$), the network attempts to classify it into one of its existing clusters based on its similarity to the stored prototype of each cluster node. More precisely, for each node $i$ in the F2 layer, the bottom-up activation $T_i$ is calculated, which can be expressed as

$$T_i = \frac{|w_i \bigcap x|}{\mu + |w_i|} \qquad i = 1, ..., M \tag{11.16}$$

**Table 11.1.** Algorithm 1

| ACC-VI algorithm |
|---|
| Step 0. Initialize the number of ants: *ant_number*, maximum number of iteration:*Mn*, side length of local region: *s*, maximum speed of ants moving: $v_{max}$ , and other parameters: $\alpha, \beta$. |
| Step 1. Project the data objects on a plane, i.e. give a pair of coordinate $(x, y)$ to each object randomly. Each ant that is currently unloaded chooses an object at random. |
| Step 2. For $i = 1, 2, \cdots, Mn$<br>for $j = 1, 2, \cdots$, *ant_number*<br>2.1 Compute the similarity of an object within a local region by formula (1), where $v$ is chosen as three kinds of speed : constant, random, and randomly decreasing for different colony;<br>2.2 If the ant is unloaded, compute picking-up probability $P_p$ by formula (4). If $P_p$ is greater than a random probability, and this object is not picked up by the other ants simultaneously, then the ant picks up the object, labels itself as loaded, and moves the object to a new position; else the ant does not pick up this, object and reselect another object randomly;<br>2.3 If the ant is loaded, compute dropping probability $P_d$ by formula (5). If $P_d$ is greater than a random probability, then the ant drops the object, labels itself as unloaded, and reselects a new object randomly; else the ant continues moving the object to a new position. |
| Step 3. For $i = 1, 2, \ldots, N$ // for all data objects<br>3.1 If an object is isolated, or the number of its neighbor is less than a given constant, then label it as an outlier;<br>3.2 Else give this object a cluster sequence number, and recursively label the same sequence number to those objects who is the neighbors of this object within local region, then obtain the number of clusters $c$. |
| Step 4. For $i = 1, 2, \ldots, c$// for $c$ clusters<br>4.1 Compute the mean of the cluster, and find four representative points by scanning the cluster on the plane from different direction of $x$-axis and $y$-axis;<br>4.2 Compute the validity index *CDbw* by formula (15) as the foundation in finding the optimal number of clusters. |
| Step 5. For $i = 1, 2, \cdots, c$//for $c$ clusters<br>5.1 Try to drop outlier into cluster, recalculate the new *CDbw*, and compare to the old value for each cluster;<br>5.2 Move the outlier to the cluster with the highest difference. |

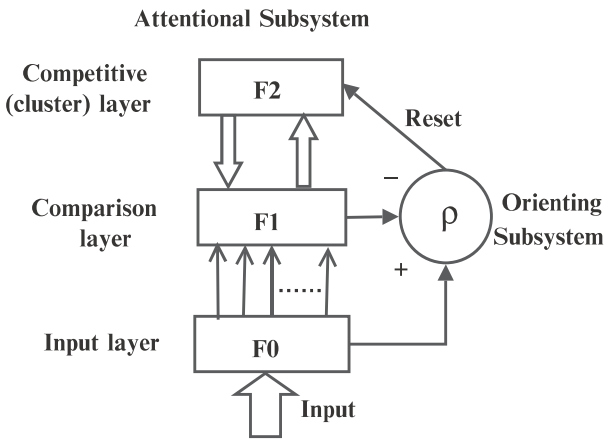


**Fig. 11.1.** The ART network architecture

where $|.|$ is the norm operator ($|x| = \sum_j^N x_j$), $w_i$ is the binary weight vector of cluster $i$, in which case the bottom-up and top-down weights are identical for simplicity [5], and $\mu > 0$ is the choice parameter. Then the $F2$ node $I$ that has the highest bottom-up activation, *i.e.* $T_I = \max\{T_i | i = 1,...,M\}$, is selected winner-take-all. The weight vector of the winning node $(w_I)$ will then be compared to the current input at the comparison layer. If they are similar enough, *i.e.* they satisfy the

$$\frac{|w_I \bigcap x|}{|x|} \geq \rho \qquad i = 1...,M \tag{11.17}$$

matching condition, where $\rho$ is a system parameter called vigilance $(0 < \rho \leq 1)$, $F2$ node $I$ will capture the current input and the network learns by modifying $w_I$:

$$w_I^{new} = \eta(w_I^{old} \bigcap x) + (1 - \eta) w_I^{old} \tag{11.18}$$

where $\eta$ is the learning rate $(0 < \eta \leq 1)$. All other weights in the network remain unchanged.

If, however, the stored prototype $w_I$ does not match the input sufficiently, i.e. formula (11.17) is not met, the winning F2 node will be reset (by activating the reset signal in Fig. 11.1) for the period of presentation of the current input. Then another F2 node (or cluster) is selected with the highest $T_i$, whose prototype will be matched against the input, and so on. This "hypothesis-testing" cycle is repeated until the network either finds a stored cluster whose prototype matches the input well enough, or inserts the input prototype into F2 as a new reference cluster. Insertion of a new cluster is normally done by creating an all-ones new node in F2 as the winning node $w_I$ and temporarily set the learning rate to 1.0, then learning takes place according

to formula (18). It is important to note that once a cluster is found, the comparison layer F1 holds $|w_I \bigcap x|$ until the current input is removed.

The number of clusters can be controlled by setting $\rho$. The higher vigilance value $\rho$, the lager number of more specific clusters will be created. At the extreme, $\rho = 1$, the network will create a new cluster for every unique input.

ART is a family of different neural architectures. Except ART1 basic architecture, ART2 [6] is a class of architectures categorizing arbitrary sequences of analog input patterns. ART-2A [7] simplifies the learning process by using the dot product as similarity measure. A pseudo code of the ART-2A learning process is summarized in Table 11.2 [16].

**Table 11.2.** Algorithm 2

| ART-2A learning process | |
|---|---|
| Step 0. | Initialize vigilance parameter $\rho(0 < \rho \leq 1)$; learning rate $\eta(0 < \eta \leq 1)$. |
| Step 1. | While stopping condition is false, do Steps 2-10. |
| Step 2. | For each training input do Steps 3-9. |
| Step 3. | Set activations of all F2 units to zero.<br>Set activations of all F0 units to normalization input vector:$X = \Re x$,<br>*where* $\Re x = \frac{x}{\|x\|} = \frac{x}{\sqrt{\sum_{i=I}^{N} x_I^2}}$ |
| Step 4. | Send input signal from F0 to F1 layer. |
| Step 5. | For each F2 node that is not inhibited, calculate the bottom-up activation $T_i$<br>If $T_i \neq -1$,*then* $T_i = X \cdot w_i, i = 1,...,M$. |
| Step 6. | While reset is true, do Steps 7-8. |
| Step 7. | Find $I$ such that $T_I = max\{T_i | i = 1,...,M\}$ for all F2 nodes $i$. If $T_I$ = -1, then all nodes are inhibited (this pattern cannot be clustered). |
| Step 8. | Test for reset:<br>If $T_I < \rho$, then $T_I = -1$ (inhibit node I) and go to Step 6.<br>If $T_I \geq \rho$ , then proceed to Step 9. |
| Step 9. | Update the weights for node I:<br>$w_I^{new} = \Re(\eta X + (1-\eta) w_I^{old})$. |
| Step 10. | Test for stopping condition. |

## 11.4 Clustering Ensemble Model

### 11.4.1 Consensus Functions

Suppose we are given a set of $N$ data points $X = \{x_1, ...., x_N\}$ and a set of $H$ partitions $\Pi = \{\pi_1, \pi_2, ...., \pi_H\}$ of objects in X. Different partitions of X return a set of labels for each point $x_i, i = 1, ...., N$[29]

$$x_i \rightarrow \{\pi_1(x_i), \pi_2(x_i), ...., \pi_H(x_i)\} \tag{11.19}$$

where $H$ indicates different clusterings and $\pi_j(x_i)$ denotes a label assigned to $x_i$ by the $j$-th algorithm. A consensus function maps a set of partitions $\Pi = \{\pi_1, \pi_2, ...., \pi_H\}$ to a target partition $\lambda$. Generally, there are four types of consensus functions:

- *Co-association matrix*. The consensus function operates on the coassociation matrix. A voting-type algorithm could be applied to the coassociation matrix to obtain the final clustering.
- *Hypergraph approach*.The clusters in the ensemble partitions could be represented as hyperedges on a graph with $N$ vertices. Each hyperedge denotes a set of data points belonging to the same cluster. The problem of consensus clustering is then become to finding the minimum-cut of a hypergraph. Three hypergraph algorithms for ensemble clustering:CSPA, HGPA, and MCLA are presented in [25].
- *Mutual information algorithm*.The consensus function could be formulated as the mutual information between the empirical probability distribution of labels in the consensus partition and the labels in the ensemble.
- *Re − labeling approach*.All the partitions in the ensemble can be relabeled according to their best agreement with some chosen reference partition [29].

These existing consensus functions are complex and rely on uncertain statistical properties in finding consensus solutions. Neural network as an ensemble combiner is another method that motivates our study of ART ensemble aggregation. The next section introduces that model.

### 11.4.2 ART Ensemble Aggregation Model

Aggregation of ensemble of multiple clusterings can be viewed as a cluster-ing task itself. Fig. 2 shows an architecture diagram of ART ensemble model. In the first phase, three clustering components generate clustering result using ant colony algorithms with different moving speed such as constant, random, and randomly decreasing respectively. Each clustering in the combination is represented as a set of labels assigned by the clustering algorithm. The combined clustering is obtained as a result of ART clustering algorithm with validity index whose inputs are the cluster labels of the contributing clusterings.

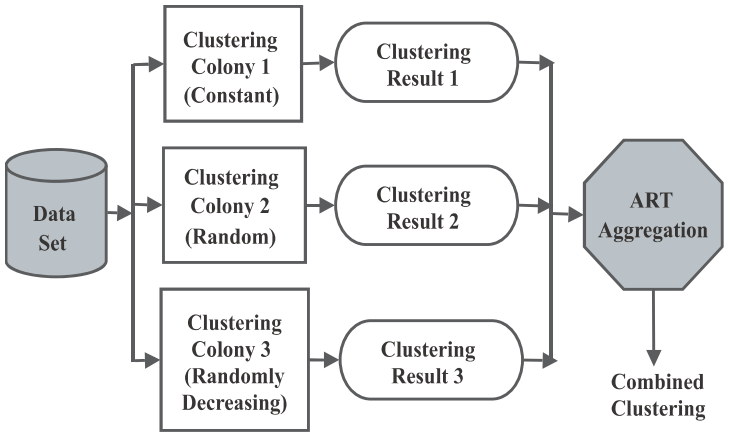


**Fig. 11.2.** System architecture of ART ensemble model

Let $X = \{x_1, ...., x_N\}$ denote a set of objects, and a set of 3 partitions $\Pi = \{\pi_1, \pi_2, \pi_3\}$ of objects in X is obtained by ant colony algorithm with different settings. For each label vector $\pi_i \in N^n$ with $c^{(i)}$ clusters, the binary membership matrix $A^{(i)} \in N^{n \times c^{(i)}}$is constructed, in which each cluster is represented as a row. All entries of a column in the binary membership matrix $A^{(i)}$ are 1, if the column corresponds to an object with known label. Columns for objects with unknown label are all zero. For example in Table 3 [25], there are 8 objects $x_i (i = 1, 2, ...., 8)$ corresponding to 3 label vectors of clusterings. The first and second clusterings are logically identical. The third one involves a dispute about objects 3 and 5. These clusterings are represented as the binary membership matrixes A shown in Table 4, where $c^{(1,2,3)} = 3$.

The binary membership matrix A is as input of ART neural network. After ART clustering, final target clustering $\lambda$ can be obtained. The clustering validity index is also used to find the best number of clusters and reduce outliers. For dispute points such as objects 3 and 5, the combined clustering result may match most cases in clustering, i.e. object 3 belongs to cluster 1 and object 5 belongs to cluster 2 like in clusterings 1 and 2.

More precisely, we use $\|x\|$ instead of 1 in matrix A that aims to enhance the accuracy of clustering ensemble. The idea is based on, combining several clustering results and nature attributes of data set. $\|x\|$ is defined by

$$\|x\| = \sqrt{\sum_{j=l}^{q} x_j^2} \tag{11.20}$$

where $q$ is the number of attributes.

The algorithm for clustering ensemble using ART is summarized in Table 5.

**Table 11.3.** Label vectors

| | $\pi_1$ | $\pi_2$ | $\pi_3$ |
|---|---|---|---|
| $x_1$ | 1 | 2 | 1 |
| $x_2$ | 1 | 2 | 1 |
| $x_3$ | 1 | 2 | 2 |
| $x_4$ | 2 | 3 | 2 |
| $x_5$ | 2 | 3 | 3 |
| $x_6$ | 3 | 1 | 3 |
| $x_7$ | 3 | 1 | 3 |
| $x_8$ | 3 | 1 | 3 |

**Table 11.4.** 3 binary membership matrixes A

| | $\pi_1$ | $\pi_2$ | $\pi_3$ | $\pi_4$ | $\pi_5$ | $\pi_6$ | $\pi_7$ | $\pi_8$ |
|---|---|---|---|---|---|---|---|---|
| $A^{(1)}$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| $A^{(2)}$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| $A^{(3)}$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

**Table 11.5.** Algorithm 3

| Clustering ensemble algorithm | |
|---|---|
| Step 0. | Apply ant colony algorithm with different settings to generate diversity clusterings: $\Pi = \{\pi_1, \pi_2, ...., \pi_H\}$ . |
| Step 1. | Compute the binary membership matrix A by label vectors $\pi_i$, $i = 1, 2, ...., H$, and use $\|\|x\|\|$ instead of 1 as input of ART network. |
| Step 2. | Use ART-2A model to ensemble clustering. |
| Step 3. | Calculate the validity index *CDbw* by formula (15) so as to find the optimal number of clusters and reduce outliers. |

## 11.5 Experimental Analysis

We have designed experiments to study the performance of the clustering ensemble model by comparing it with the ACC-VI algorithm, the ART-2A algorithm and the LF algorithm on various artificial and real data sets. We evaluated the clustering performance using cluster validity index *CDbw*.

### 11.5.1 Artificial Data Set (2D3C)

We artificially generated the data set (2D3C), containing three 2D-Gaussian distributed clusters of different sizes (50,100,75), different densities (variance) and shapes (one with elliptical Gaussian distributions in elongation level and rotated orientation) shown in Fig. 3.

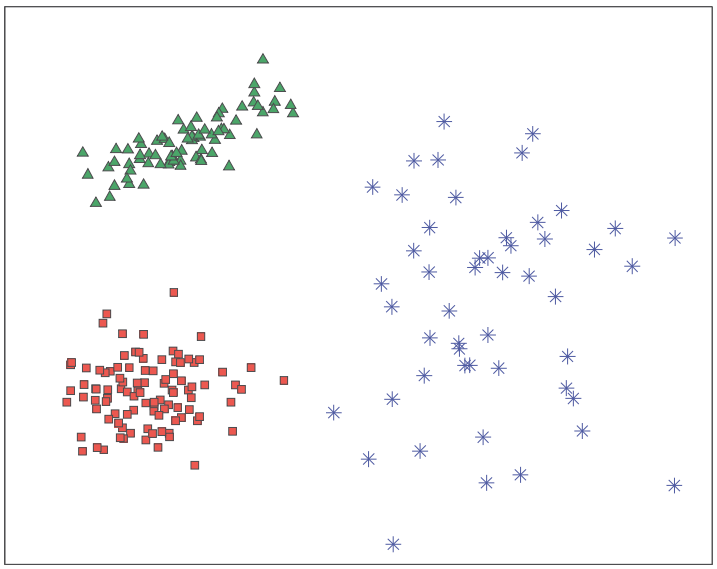

**Fig. 11.3.** Artificial data set (2D3C)

Table 6 presents the *CDbw* values on artificial data set (2D3C) for the proposed ensemble algorithm, the ACC-VI algorithm, the ART-2A algorithm, and the LF algorithm, respectively. It is noted that *CDbw* takes its maximum value 10.42 for the partitioning of three classes defined by the ACC-VI algorithm, 16.76 for the partitioning of three classes defined by the proposed ensemble algorithm, and 15.53 for the partitioning of three classes defined by the ART-2A algorithm, respectively. While the clustering results of the LF Algorithm into 5 clusters is presented by highlight 12.95 in the fourth column. It is obvious that 3 is considered as the correct number of clusters. This is also the number of actual clusters of (2D3C). The biggest value 16.76 on *CDbw* shows that the ensemble clustering algorithm is optimal.

### 11.5.2 Real Data Set (Iris)

The real data set used is the Iris data set, which has been widely used in pattern classification, downloaded from the UCI machine learning repository [23]. The data

**Table 11.6.** Optimal number of clusters found by *CDbw* for different clustering algorithm

| No clusters | ACC-VI | Ensemble | ART-2A | LF Algorithm |
|---|---|---|---|---|
| 2 | 1.26 | 5.52 | 8.22 | 1.10 |
| **3** | **10.42** | **16.76** | **15.53** | 9.02 |
| 4 | 4.17 | 12.50 | 11.98 | 8.70 |
| 5 | 9.85 | 13.03 | 12.78 | **12.95** |

set contains 3 classes of 50 instances each in a 4 dimensional space, where each class refers to a type of iris plant. One class is linearly separable from the other 2, the latter are not linearly separable from each other.

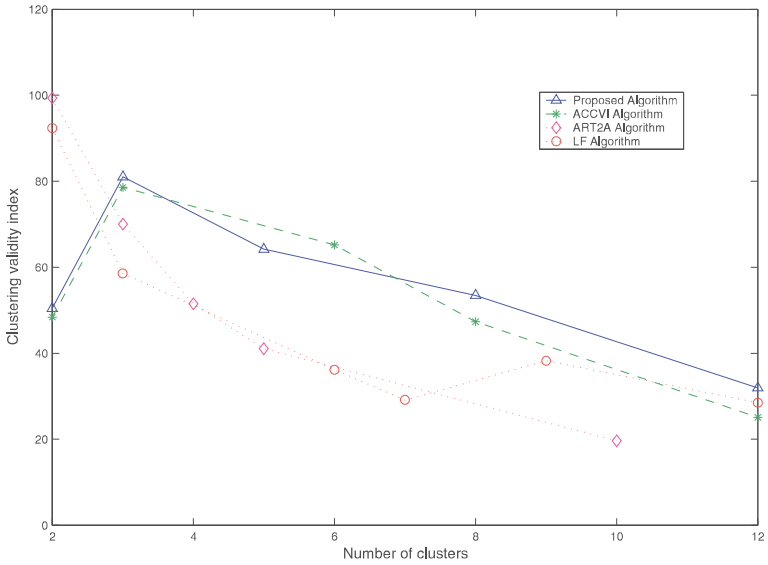


**Fig. 11.4.** *CDbw* as a function of number of clusters on Iris data set for the proposed algorithm, the ACC-VI algorithm, the ART-2A algorithm, and the LF algorithm, respectively

Fig. 11.4, *CDbw* indicates that the Iris data are divided into three clusters by the proposed ensemble algorithm and the ACC-VI algorithm. It is more consistent with the inherent three clusters of data, compared to two clusters by the ART-2A algorithm and the LF algorithm. The ensemble model is a little better than the ACC-VI algorithm with *CDbw* at its peak.

### 11.5.3 Reuter-21578 Document Collection

The Reuters-21578 document collection is a standard text-clustering corpus composed of 21578 news articles in 1987 [36]. We sampled 5 different document collections each of size 1000 that have only TOPICS labels. Each document is processed by removing a set of common words using a "stop-word" list, and the suffixes are removed using a Porter stemmer. Then the document is represented as a vector space model using TF-IDF-weighting [35].

Fig. 5-9 illustrates *CDbw* as a function of the number of clusters for the samples using the proposed algorithm, the ACC-VI Algorithms, the ART-2A Algorithm, and the LF Algorithm, respectively. The maximum value of *CDbw* indicates the optimal number of clusters for each algorithm. Table 7 summarized the highest *CDbw* in Fig. 5-9, where the highlighted results presented the optimal number of clusters. For example, the best number of clusters equals to 12 for the proposed ensemble algorithm and the ACC-VI algorithm, 21 for the ART-2A algorithm, and 8 for the LF algorithm, respectively. At the same time,*CDbw* can also be considered to evaluate the performance of different algorithms. From the results shown in Table 4, we can see that the proposed algorithm has produced the maximum *CDbw* value compared to the 3 other algorithms. Note that not all algorithms produced results for all the number of clusters considered.

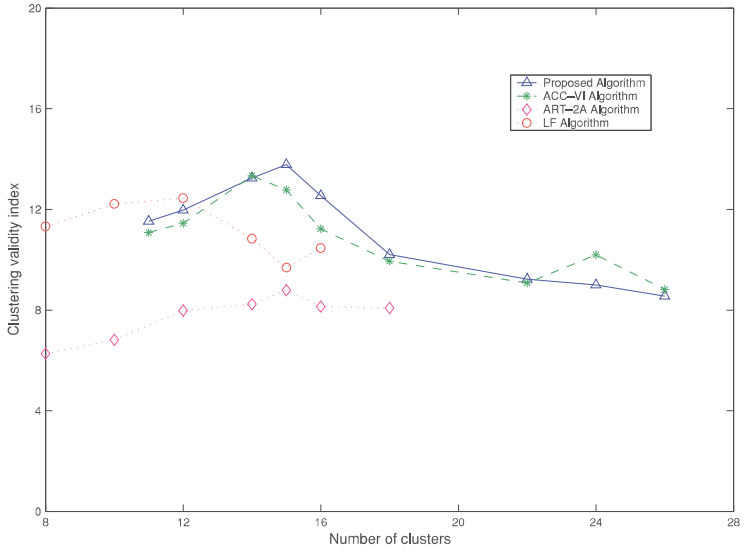


**Fig. 11.5.** *CDbw* as a function of number of clusters on the first sample collection of 1000 documents each for the proposed algorithm, the ACC-VI algorithm, the ART-2A algorithm, and the LF algorithm, respectively

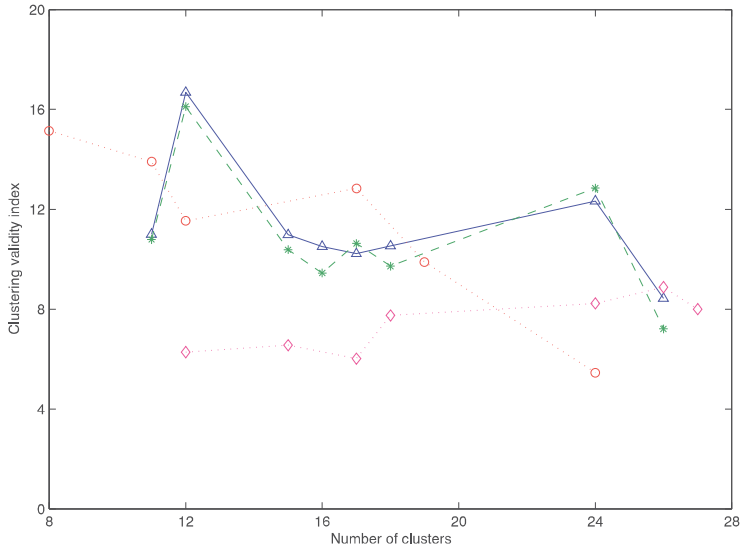


**Fig. 11.6.** *CDbw* as a function of number of clusters on the second sample collection of 1000 documents each for the proposed algorithm, the ACC-VI algorithm, the ART-2A algorithm, and the LF algorithm, respectively

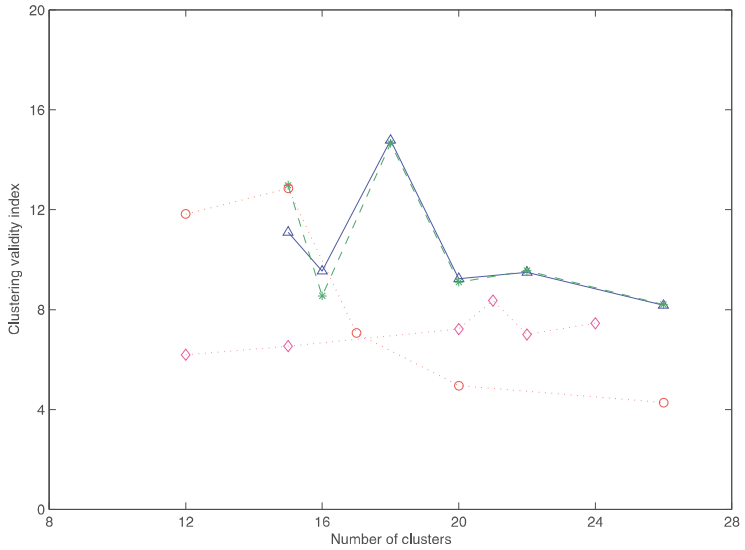


**Fig. 11.7.** *CDbw* as a function of number of clusters on the third sample collection of 1000 documents each for the proposed algorithm, the ACC-VI algorithm, the ART-2A algorithm, and the LF algorithm, respectively

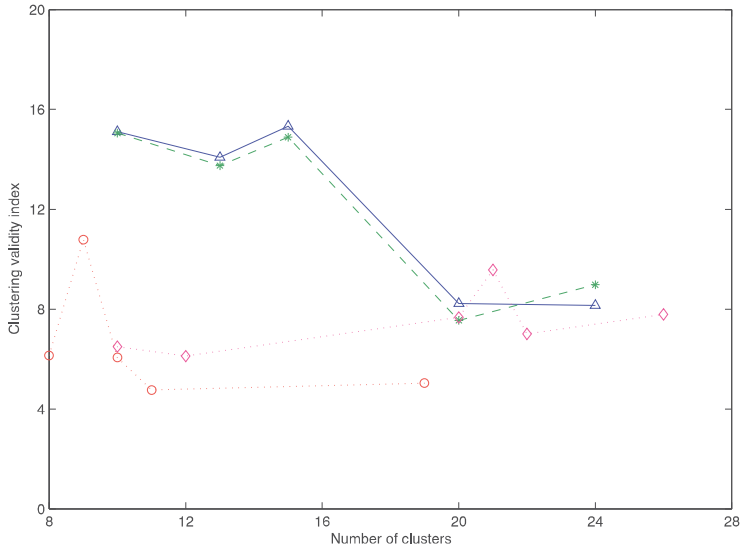


**Fig. 11.8.** *CDbw* as a function of number of clusters on the fourth sample collection of 1000 documents each for the proposed algorithm, the ACC-VI algorithm, the ART-2A algorithm, and the LF algorithm, respectively

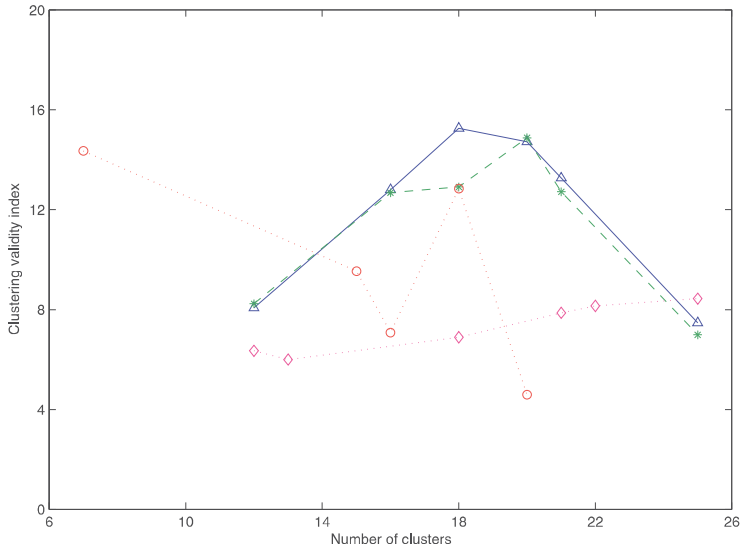


**Fig. 11.9.** *CDbw* as a function of number of clusters on the fifth sample collection of 1000 documents each for the proposed algorithm, the ACC-VI algorithm, the ART-2A algorithm, and the LF algorithm, respectively

**Table 11.7.** Optimal number of clusters found by *CDbw* on 5 sample collection of 1000 documents each for different clustering algorithm

| No clusters | ACC-VI | Ensemble | ART-2A | LF Algorithm |
|---|---|---|---|---|
| 7 | | | | 14.35 |
| 8 | | | | **15.15** |
| 9 | | | | 10.79 |
| 10 | 15.05 | | | |
| 12 | **16.11** | **16.68** | | 12.45 |
| 14 | 13.34 | | | |
| 15 | | 15.33 | 8.79 | 12.85 |
| 15 | | 13.78 | | |
| 18 | | 15.26 | | |
| 18 | 14.66 | 14.78 | | |
| 20 | 14.88 | | | |
| 21 | | | 8.36 | |
| 21 | | | **9.57** | |
| 25 | | | 8.44 | |
| 26 | | | 8.88 | |

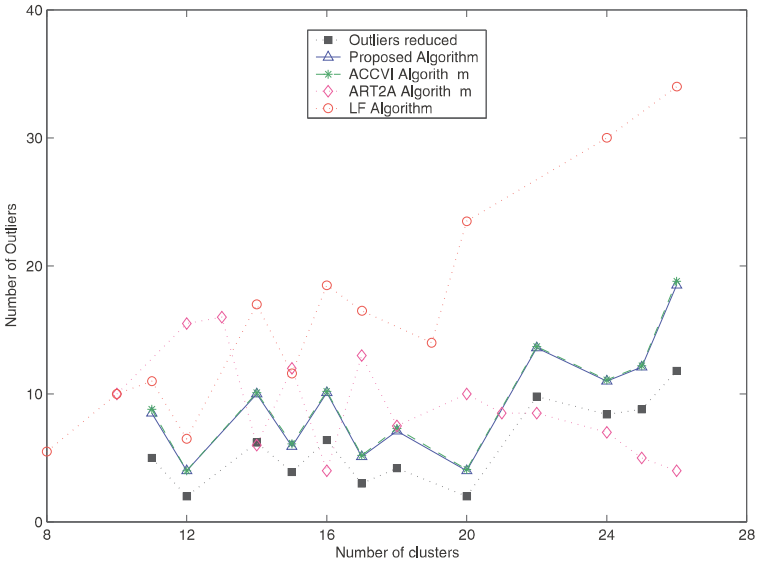


**Fig. 11.10.** The average number of outliers on 5 document collection of 1000 documents each

Fig. 10 gives the average number of outliers on the same data sets. It is noted that the proposed algorithm has lower outliers after using the outlier reduction strategy.

## 11.6 Conclusions

In this chapter we proposed a clustering ensemble model using ant colony algorithm with validity index and ART network. This model uses the parallel and independent ant colonies combined by ART network as well as clustering validity index to improve the performance of the clustering. As shown by the results of the experiment, the proposed ensemble model improved the quality of the clustering.

**Acknowledgements**

This work was partially funded by the Key Basic Application Founding of Sichuan Province (04JY029-001-4) and the Science Development Founding of Southwest Jiaotong University (2004A15).

## References

1. Ayad H, Kamel M (2003) Finding natural clusters using multi-clusterer com-biner based on shared nearest neighbors. In: Multiple Classifier Systems: Fourth International Workshop, MCS 2003, UK, Proceedings, pp166-175
2. Ayad H, Basir O, Kamel M (2004) A probabilistic model using information theoretic measures for cluster ensembles. In: Multiple Classifier Systems: Fifth International Workshop, MCS 2004, Cagliari, Italy, Proceedings, pp144-153
3. Bartfai G (1996) An ART-based Modular Architecture for Learning Hierar-chical Clusterings. J Neurocomputing, 13:31-45
4. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm Intelligence - From Natural to Artificial System. Oxford University Press, New York
5. Carpenter G A, Grossberg S (1987a) A massively parallel architecture for a self-organizing neural pattern recognition machine. J Computer Vision, Graphics, and Image Processing, 37:54-115
6. Carpenter G A, Grossberg S (1987b) ART 2: Self-organization of stable cate-gory recognition codes for analog input patterns. J Applied Optics, 26(23):4919-4930
7. Carpenter G A, Grossberg S, Rosen D B (1991a) ART2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition. J Neu-ral Networks, 4:493-504
8. Carpenter G A, Grossberg S, Rosen D B (1991b) Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance sys-tem. J Neural Networks, 4:759-771
9. Deneubourg J ., Goss S, Franks N, Sendova-Franks A, Detrain C, Chretien L (1991) The Dynamics of Collective Sorting: Robot-like Ant and Ant-like Ro-bot. In: Meyer J A, Wilson S W (eds) Proc. First Conference on Simulation of Adaptive Behavior: From Animals to Animats. Cambridge, MA: MIT Press, pp356-365

10. Dudoit S, Fridlyand J (2003) Bagging to improve the accuracy of a clustering procedure. J Bioinformatics, 19(9):1090-1099
11. Fred A L N (2002) Finding Consistent Clusters in Data Partitions. In: Roli F, Kittler J (Eds) Proc. 3rd Int. Workshop on Multiple Classifier Systems, LNCS 2364, pp309-318
12. Grossberg S (1976) Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors II: Feedback, expectation,olfaction,and illusions.J Biological Cybernetics,23:121-134 187-202
13. Halkidi M, Vazirgiannis M (2002) Clustering validity assessment using multi representatives. In: Proc. of SETN Conference
14. Halkidi M, Vazirgiannis M, Batistakis Y (2000) Quality scheme assessment in the clustering process. In: Proc. 4th Eur. Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD), pp165-276
15. Handl J, Meyer B (2002) Improved ant-based clustering and sorting in a document retrieval interface. In: Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature. LNCS2439, Berlin, Germany: Springer-Verlag, pp913-923
16. He J, Tan A, Tan C (2004) Modified ART 2A Growing Network Capable of Generating a Fixed Number of Nodes. J IEEE Trans. on Neural Networks, 15(3):728-737
17. He Y, Hui S C, Fong A C M (2002) Mining a web citation database for docu-ment clustering. J Applied Artificial Intelligence, 16:283-302
18. Hussin M F, Kamel M (2003) Document clustering using hierarchical SOMART neural network. In: Proc of the Int'l Joint Conf on Neural Network, Portland, Oregon, USA, pp2238-2241
19. Jiang Y, Zhou Z (2004) SOM Ensemble-Based Image Segmentation. J Neural Processing Letters, 20:171-178
20. Kanade P M, Hall L O (2003) Fuzzy Ants as a Clustering Concept. In: Proc. of the 22nd Int. Conf. of the North American Fuzzy Information Processing Society, pp227-232
21. Lumer E, Faieta B (1994) Diversity and Adaptation in Populations of Clustering Ants. In: Proc. Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3.Cambridge, MA: MIT Press,pp499-508
22. Monmarch¨| N, Slimane M, Venturini G (1999) Antclass: Discovery of Clus-ters in Numeric Data by a Hybridization of an Ant Colony with the Kmeans Algorithm. Technical Report 213, Laboratoire d'Informatique, E3i, Univer-sity of Tours
23. Murpy P M, Aha D W (1994) UCI repository of machine learning databases. Irvine, CA: University of California. [Online] Available: http://www.ics.uci.edu/mlearn/MLRepository.html
24. Ramos V, Merelo J J (2002) Self-organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning. In: Alba E, Herrera F, Merelo J J (eds) AEB?2002 - 1st Spanish Conference on Evolutionary and Bio-Inspired Algorithms, Centro Univ. de M¨|rida, M¨|rida, Spain, pp284-293
25. Strehl A, Ghosh J (2002) Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J Machine Learning Research, 3:583-617
26. Theodoridis S, Koutroubas K (1999) Pattern Recognition. Academic Press
27. Tomida S, Hanai T, Honda H, Kobayashi T (2002) Analysis of expression profile using fuzzy adaptive resonance theory. J Bioinformatics, 18(8):1073-1083
28. Topchy A, Jain A K, Punch W (2003) Combining Multiple Weak Clusterings. In: Proc. IEEE Intl. Conf. on Data Mining, Melbourne, FL, pp331-338

29. Topchy A, Jain A K, Punch W (2004) A Mixture Model of Clustering En-sembles. In: Proc. SIAM Intl. Conf. on Data Mining, pp379-390
30. Wu B, Zheng Y, Liu S, Shi Z (2002) CSIM: a Document Clustering Algo-rithm Based on Swarm Intelligence. In: IEEE World Congress on Computational Intelligence, pp477-482
31. Wu S, Chow T (2003) Self-organizing-map based clustering using a local clustering validity index. J Neural Processing Letters, 17:253-271
32. Yang S, Browne A, Picton P D (2002) Multistage Neural Network Ensembles. In: Roli F, Kittler J (Eds) Proc. 3rd Int. Workshop on Multiple Classifier Systems, LNCS 2364, pp91-97
33. Yang Y, Kamel M (2003) Clustering ensemble using swarm intelligence. In: IEEE Swarm Intelligence Symposium, Indianapolis, USA, pp65-71
34. Yang Y, Kamel M (2005) A Model of Document Clustering using Ant Col-ony Algorithm and Validity Index. In: Int. Joint Conf. on Neural Network (IJCNN'05), Montreal, Canada, pp1732-1737
35. Yang Y, Kamel M, Jin F (2005) Topic discovery from document using ant-based clustering combination. In: Web Technologies Research and Develop-ment - APWeb 2005, 7th Asia-Pacific Web Conference, Shanghai, China, LNCS3399, UK, Springer, pp100-108
36. [Online] Available: http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

# Index