5-2004

Modified ART 2A growing network capable of generating a fixed number of nodes

Ji HE

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Chew-Lim TAN

Citation

HE, Ji; TAN, Ah-hwee; and TAN, Chew-Lim. Modified ART 2A growing network capable of generating a fixed number of nodes. (2004). *IEEE Transactions on Neural Networks*. 15, (3), 728-737.
Available at: https://ink.library.smu.edu.sg/sis_research/5238

Modified ART 2A Growing Network Capable of Generating a Fixed Number of Nodes

Ji He, *Member, IEEE*, Ah-Hwee Tan, and Chew-Lim Tan, *Senior Member, IEEE*

***Abstract*—This paper introduces the Adaptive Resonance Theory under Constraint (ART-C 2A) learning paradigm based on ART 2A, which is capable of generating a user-defined number of recognition nodes through online estimation of an appropriate vigilance threshold. Empirical experiments compare the cluster validity and the learning efficiency of ART-C 2A with those of ART 2A, as well as three closely related clustering methods, namely online K-Means, batch K-Means, and SOM, in a quantitative manner. Besides retaining the online cluster creation capability of ART 2A, ART-C 2A gives the alternative clustering solution, which allows a direct control on the number of output clusters generated by the self-organizing process.**

***Index Terms*—Adaptive Resonance Theory (ART), clustering, constraint learning, neural networks.**

I. Introduction

Adaptive Resonance Theory (ART) [1] is a family of neural networks that develop stable recognition categories (clusters) by self-organization in response to arbitrary sequences of input patterns. Through dynamic creation of recognition categories for encoding distinct input samples, an ART module is capable of self-adjusting the scale of its recognition field, in terms of the number of committed nodes, with respect to the complexity of the problem domain. Its *fast commitment* mechanism and capability of learning at moderate speed guarantees a high efficiency. However, given a data set, the scale of ART recognition field (i.e., the number of output clusters) depends on a global threshold parameter called *vigilance*. While in principle, one could control ART's recognition representation by fine tuning the vigilance parameter, in practice, suggesting an appropriate vigilance value requires prior knowledge on the scale and the distribution of the problem data set, which is unlikely to be available.

This paper proposes a novel ART learning paradigm named ART-C (Adaptive Resonance Theory under Constraint). Specifically this paper introduces ART-C 2A based on ART 2A [2], which contains several improvements over its predecessor previously introduced by He *et al.* [3]. Our aim is to combine the neuron initialization and the online clustering capabilities of ART 2A with the predictability in allowing a direct control on the number of the output clusters. This capability is achieved by a *constraint reset* mechanism in ART-C 2A that adaptively adjusts the global vigilance threshold of the system and reorganizes the category representation in response to an intuitive constraint. Given a specific data set, the constraint reset mechanism does not affect the network's learning when the proper vigilance is accurately estimated. Hence ART-C 2A's output is practically comparable to that of ART 2A using a preestimated vigilance value which intends to output the same number of clusters over the data set.

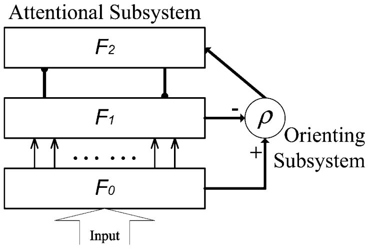

Fig. 1. The ART architecture.

The rest of the paper is organized as follows. Section II provides a brief review of the ART 2A network and analyzes its key learning characteristics which motivate our work. Section III introduces and analyzes the ART-C 2A learning paradigm. Section IV compares ART-C 2A with related work. Section V reports our benchmark on the performance of ART-C 2A by comparing it with that of ART 2A, as well as three closely related clustering methods, namely online K-Means, batch K-Means, and SOM, and extends our discussions on the experimental results. Section VI summarizes our concluding remarks.

II. Analysis of the ART 2A Network

There exist a large variety of ART networks in the literature. Our review focuses on the ART 2A network [2] which is closely related to our work. An ART 2A network mainly follows the conventional ART architecture [1], which consists of three layers depicted in Fig. 1: the input layer (F_0), the comparison layer (F_1), and the recognition layer (F_2). The input layer F_0 receives and stores the input patterns. Neurons in the input layer F_0 and comparison layer F_1 are one-to-one connected with hard-coded links, which corresponds to a normalization preprocessing to prevent category proliferation. The comparison layer F_1 stores the short-term memory for the current input pattern while the recognition layer F_2 stores the prototypes of recognition categories (clusters) as the long-term memory. In Carpenter *et al.* original prototype [2], the F_2 layer initially contains a number of so-called *uncommitted* nodes, which one by one will conditionally get *committed* upon input presentation. This however may give a wrong impression that ART uses "*a*

Manuscript received December 2, 2002; revised July 10, 2003 and October 30, 2003.

J. He and C.-L. Tan are with the School of Computing, National University of Singapore, Singapore 117543 (e-mail: mail@jihe.net; tancl@comp.nus.edu.sg).

A.-H. Tan is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: sashtna@ntu.edu.sg).

Digital Object Identifier 10.1109/TNN.2004.826220

fixed number of output nodes which limit the number of clusters that can be produced" [4]. As an alternative interpretation, a number of subsequent studies (such as [5], [6]) refer the F_2 layer initially as a null set (i.e., contains no node) which dynamically grows by creating new recognition categories (committed nodes) using distinct inputs. We follow the later interpretation in the rest of this paper, as it highlights ART's capability of expanding the scale of its recognition field indefinitely.

The ART network follows a winner-take-all competitive learning process. Learning of the conventional ART network involves the modification of the weighted bottom-up (feed-forward) and top-down (feed-backward) connections between F_1 and F_2. The interactions between F_1 and F_2 are controlled by the orienting subsystem using a vigilance threshold ρ. Such a learning process is simplified in ART 2A by using a feed-forward only connection between F_1 and F_2, and a symmetric dot product as the similarity measure in the category *choice* function and *match* function. The learning process of ART 2A network is summarized.

Parameters

The ART 2A dynamics are determined by the vigilance parameter $\rho \in [0, 1]$ and the learning rate $\eta \in [0, 1]$.

Network initialization

The recognition layer F_2 is initialized with the null set $\vee$ (i.e., contains no category).

Input normalization

Given the nonzero input vector $\mathbf{A}^0$ presented to F_0, the links between F_0 and F_1 form a built-in Euclidean normalization according to

$$\mathbf{A} = \Re\mathbf{A}^0 \tag{1}$$

where the Euclidean normalization function $\Re$ is given by

$$\Re\mathbf{x} \equiv \frac{\mathbf{x}}{||\mathbf{x}||} = \frac{\mathbf{x}}{\sqrt{\sum_i x_i^2}}. \tag{2}$$

Category choice

Given an F_1 input vector $\mathbf{A}$, for each F_2 node j, the *choice* function T_j is defined by

$$T_j = \mathbf{A} \cdot \mathbf{w}_j \tag{3}$$

where $\mathbf{w}_j$ is the weight vector of node j. The system is said to make a choice when at most one F_2 node can become active. The choice is indexed at J where

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \tag{4}$$

Resonance or reset

Mismatch reset happens when the network fails to locate a winner category (when the first input is presented), or when the choice score T_J does not reach the vigilance value

$$T_J < \rho \tag{5}$$

during which a new category K is created by copying $\mathbf{A}$ as its weight vector

$$\mathbf{w}_K = \mathbf{A}. \tag{6}$$

Otherwise the network is said to reach *resonance*, during which learning ensues, as defined.

Learning

Once the search ends and a resonance is achieved, the attentional subsystem updates the weight vector $\mathbf{w}_J$ according to

$$\mathbf{w}_J^{t+1} = \Re(\eta\mathbf{A} + (1 - \eta)\mathbf{w}_J^t). \tag{7}$$

The minor differences between the above and the original ART 2A learning process proposed in [2] deserve some explanations herein. In Carpenter *et al.* version, a *match* function, which uses the dot product as well, is used in the resonance checking step. We consolidate the *choice* function and *match* function into one, as they are essentially equivalent to each other. In [2], a small threshold θ is used to cut off the attribute value of the input vector such that if $a_i < \theta$, a_i is reset to be 0. Similar cut-off is applied on the weight vector as well during learning. This is claimed to "*distinguish features that are irrelevant in given categories*" [2]. As such, once an attribute value of the weight vector drops below the threshold, the value will remain zero in the further learning. However, suggesting an appropriate "irrelevance" threshold value requires prior knowledge and may be quite subjective. In addition, most clustering systems assume there is an effective feature selection preprocessing and all features presented to the system are equally important. Therefore we follow a common practice to disable this threshold for simplicity of analysis. Carpenter *et al.* also use a small constant α such that the so-called *uncommitted* nodes are enforced to have a nominal, minor "similarity" of $\alpha \sum_i a_i$ with the input pattern. It follows that in some simulations, even when $\rho = 0$, some uncommitted nodes may be activated and the system may generate a few categories [2]. Readers should note this is practically equivalent to the result produced by the paradigm we summarized above, using a very small ρ value.

The network's learning in (7) contains a convex combination of the input and the existing category. The Euclidean normalization $\Re$ in (1) and (7) limits the learning on a unit hypersphere to avoid category proliferation. Under this condition, the choice function in (3) is equivalent to the cosine similarity between the input and the recognition category. That is

$$T_j = \mathbf{A} \cdot \mathbf{w}_j = ||\mathbf{A}||||\mathbf{w}_j|| \cos\alpha = \cos\alpha \quad \text{given } ||\mathbf{A}|| = ||\mathbf{w}_j|| = 1 \tag{8}$$

where α is the angle between vector $\mathbf{A}$ and $\mathbf{w}_j$. The network's mismatch reset mechanism brings interesting characteristics to ART 2A's learning paradigm. The vigilance threshold ρ guards whether an input will be incorporated into its most similar recognition category, or will be used to generate a new category. Specifically, this threshold forms a circular decision boundary with a radius of $\sqrt{2(1-\rho)}$ around the weight vector of each category.[1] When the Euclidean distance between two nearest categories is less than $2\sqrt{2(1-\rho)}$ (i.e., there is an overlap between the two corresponding circular regions), the partitioning boundary between these two categories is given by their perpendicular bisector on the hypersphere (Fig. 2).

[1] Given $\mathbf{A} \cdot \mathbf{B} = \rho$ and $||\mathbf{A}|| = ||\mathbf{B}|| = 1$, $||\mathbf{A} - \mathbf{B}|| = \sqrt{||\mathbf{A}||^2 + ||\mathbf{B}||^2 - 2\mathbf{A} \cdot \mathbf{B}} = \sqrt{2(1-\rho)}$.

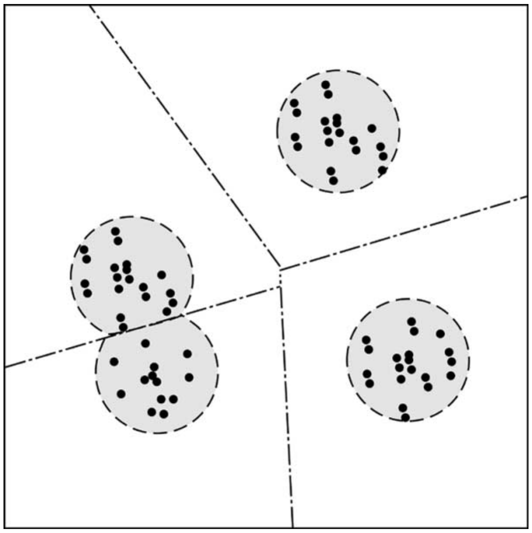

Fig. 2. The decision boundaries of the ART 2A network (dashed lines), the committed region (gray), and the uncommitted region (white) being viewed on the unit hypersphere. Cross markers $(+)$ identify the weights of the recognition categories. Vector quantization is done by relaxing the decision boundaries with $\rho = 0$ (dot-dashed lines) after the network convergence on the input sequence.

Mismatch reset of the network occurs if an input pattern falls outside of the *committed region*, i.e., the combination of circular regions around all categories (gray areas in Fig. 2). Learning of such an input is done by creating a new category with the input, which turns a new circular region around it into a committed sub-region. This fast commitment paradigm guarantees stable encoding of new distinct inputs. We would highlight that this characteristic particularly reflects the plasticity of the network.

Network resonance only happens if the input pattern falls into the committed region. Learning activity of the network in the committed region, is closely related to the naive competitive learning paradigm. This close relationship makes ART 2A, like other clustering algorithms in the same category, capable of serving as a *vector quantizer*. The interesting point is that, the resonance check of the network provides an upper bound for the variance of the newly learnt cluster prototype $\mathbf{w}^{(t+1)}$ from $\mathbf{w}^{(t)}$, specifically $\|\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\| \leq \sqrt{2(1-\rho)}$. As such, the network is capable of learning at either slow or intermediate speed without causing cluster oscillation. Combination of this fast learning capability with the fast commitment mechanism ensures ART 2A's capability of achieving stable encoding of input sequences with very few learning iterations in practice.

With the analysis above, it is understandable that the vigilance threshold ρ affects the number of ART 2A recognition categories generated on a specific input sequence in a major way. Specifically, $\rho = 1$ causes each unique input to be encoded as one separate category, whereas $\rho = 0$ causes all inputs to be encoded into the same category.

This characteristics of ART 2A motivates our study of the ART-C 2A learning paradigm, which dynamically adjusts the vigilance parameter during its learning in respect to a user-defined constraint on the category representation, in terms of the number of recognition categories. In the Section III, we give the details of the ART-C 2A paradigm.

III. The ART-C 2A Learning Paradigm

Unlike a conventional ART 2A network that mainly controls its learning activity with a vigilance threshold ρ, ART-C 2A's learning is mainly guided by an intuitive constraint C on the maximal number of recognition categories in the F_2 layer. The solution introduces an extra *constraint reset* mechanism to the ART 2A network, which self-adjusts the vigilance threshold of the network through an adaptive estimation of the input distribution in response to the constraint C. The dynamically adjusted vigilance threshold in turn drives the learning activities to satisfy the user-defined constraint. The ART-C 2A learning paradigm is introduced below.

Parameters

The ART-C 2A dynamics are determined by the constraint C on the number of recognition categories and the learning rate $\eta \in [0, 1]$.

Network initialization

The recognition layer F_2 is initialized with the null set $\vee$. The vigilance ρ for the orienting subsystem is initialized with 1.0.

Learning of each input representation

Learning of each input presentation follows the ART 2A learning paradigm. This includes the same built-in input normalization, category choice, resonance check, and learning stages.

Constraint checking

Constraint checking is performed after the learning of each input representation by comparing the number of existing recognition categories N with the predefined constraint C

$$\hbar = \begin{cases} 1, & \text{if } N > C \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

With $\hbar = 0$, the constraint is said to be satisfied, upon which the network carries on to learn the next input representation. Otherwise, *constraint reset* occurs.

Constraint reset

Constraint reset reorganizes the recognition categories in the F_2 layer toward the satisfaction of the constraint and adjusts the ρ value based on the current category distribution. The process is introduced as follows.

1) *Search of nearest category pair:* For each category pair (i, j) in the F_2 layer, their similarity is defined by the dot product of their corresponding weights $\mathbf{w}_i$ and $\mathbf{w}_j$ such that

$$T_{(i,j)} \equiv \mathbf{w}_i \cdot \mathbf{w}_j = \cos\alpha \tag{10}$$

where α is the angle between $\mathbf{w}_i$ and $\mathbf{w}_j$, given $\|\mathbf{w}_i\| = \|\mathbf{w}_j\| = 1$. The *nearest neighbor* of each category i, indexed as $J(i)$, is the category that has the maximal similarity with i

$$T_{(i,J(i))} = \max\{T_{(i,j)} : j = 1, \ldots, N,\ j \neq i\}. \tag{11}$$

The *nearest neighbor similarity* of category i, marked as $\tau(i)$ then refers to the similarity between category i and its nearest neighbor $J(i)$

$$\tau(i) \equiv T_{(i,J(i))}. \tag{12}$$

The *nearest category pair*, indexed as (I, J), is identified by the category I that has the maximal nearest neighbor similarity to its nearest neighbor J

$$\tau(I) = T_{(I,J(I))} = \max\{T_{(i,J(i))} : i = 1, \ldots, N\}. \qquad (13)$$

2) *Adjustment of the vigilance:* The vigilance value $\rho^{(new)}$ for subsequent learning is decreased according to

$$\rho^{(new)} = \max\{\tau(i) : \text{all } i \text{ whose } \tau(i) < \rho^{(old)}\} \qquad (14)$$

thus, $\rho^{(new)} < \rho^{(old)}$.

3) *Merging of the nearest category pair:* Merging of the nearest category pair (I, J) is done by inserting a new category L with the weight vector as the mean of these two categories

$$\mathbf{w}_L = \Re(0.5\mathbf{w}_I + 0.5\mathbf{w}_J) \qquad (15)$$

where $\Re$ is the Euclidean normalization as given by (2). In addition, the categories I and J are deleted from the recognition layer F_2 after the creation of the new category.

Each constraint reset cycle decreases the number of recognition categories in F_2 by one. Theoretically the constraint checking and constraint reset processes should be repeated till the network satisfies the constraint (i.e., $\hbar = 0$). However, considering the nature of the ART 2A learning is to create at most one new recognition category for encoding of each input, constraint reset practically occurs only when $N = C + 1$. Therefore constraint reset happens at most once on each input representation, after which the number of recognition categories in the F_2 layer is decreased to C. It is also understandable that constraint reset can only happen right after a mismatch reset, which is the direct cause that increases the number of recognition categories from C to $C + 1$.

Computational complexity of the network

The computational complexity of the conventional ART 2A network has been widely discussed in the literature. We hereby discuss the additional computational cost introduced by the ART-C 2A's constraint reset process. Apparently the cost of this process is dominated by the calculation of the pairwise similarities among the existing $C + 1$ recognition categories, as given by (10). The computation cost can be estimated as $O(C^2)$, in terms of the number of dot-product calculations. Compared with the category choice process, which is estimated as $O(C)$, constraint reset is computationally intensive. When C is large, this operation could be time consuming. However, one should note that constraint reset happens only conditionally, depending on the distributions of the input data and the recognition categories, as well as the vigilance ρ. The decreasing vigilance and category redistribution reduce the possibility of constraint reset in subsequent learning. When the input data are reasonably densely distributed and the number of input data M satisfies $M \gg C^2$, the additional cost of the constraint reset process can be ignored.

IV. Related Work

The idea of controlling the category representation of an ART network using varying vigilance values has been investigated in the literature. The varying vigilance plays an essential role in the supervised ARTMAP networks [7]. Most closely related to the ART-C 2A learning paradigm may be the HART modular designs (HART-J and HART-S) [5]. HART generates hierarchical representation of the input sequence. Learning activities in various layers of the hierarchy are guarded with different ρ values and produce category representation of the same input sequence with varying details, either from fine to coarse representation (HART-J) or from coarse to fine representation (HART-S). HART however lacks the capability of producing a predefined number of categories in any layer, as the modular design presets a vigilance value for each layer and limits the learning activities in each layer strictly like a conventional ART.

One key idea employed in ART-C 2A is the redistribution of the representation categories during constraint reset through merging of categories. A number of hierarchical agglomerative clustering algorithms, such as UPGMA [8] and neighbor-joining [9], apply a similar paradigm. Hierarchical agglomerative clustering algorithms typically represent M input samples as M reference clusters. Each clustering cycle identifies the most similar pair of clusters and merges them. The process may repeat until there is only one cluster left. While they are able to generate a predefined number of C clusters over M input samples, they are notably computational intensive in maintaining the pair-wise similarity matrix as typically $M \gg C$ [10]–[12]. The advantage of ART-C 2A over this class of algorithms lies in its combination of competitive learning of individual inputs with the calculation of pair-wise category similarities. While online learning of individual inputs maintains a high efficiency, merging of the nearest category pair in ART-C 2A enables a quick redistribution of recognition categories, with a notably lower computational cost.

V. Experiments

Our experiments study the characteristics of ART-C 2A by comparing it with the conventional ART 2A learning, as well as three alternative clustering algorithms, namely online K-Means [13], batch K-Means [13] and SOM [14]. All these methods have been extensively studied and widely applied in the literature. Among them, ART-C 2A, ART 2A, online K-Means (or naive competitive learning in some literatures), and SOM share the common competitive learning principle. On various real-life data sets, our experiments evaluate the cluster validity and learning efficiency of these five algorithms, using various quantitative evaluation measures.

A. Cluster Validity Measures

Our experiments adopted two sets of cluster validity measures summarized below. More discussions on these measures can be found in [15].

1) Cluster Validity Measures Based on Cluster Distribution: Since the nature of clustering is to reorganize the input samples such that data points in the same cluster are more similar[2] to each other than to points in a different cluster, it is a natural way to evaluate the intracluster homogeneity and

[2] The similarity measure is chosen subjectively based on the system's ability to create "interesting" clusters.

the intercluster separation of the clustering output in a global fashion.

Cluster compactness

The cluster compactness measure is based on the generalized definition of the *deviation* of a data set given by

$$dev(\mathbf{X}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d^2(\mathbf{x}_i, \overline{\mathbf{x}})} \tag{16}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is a distance metric between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ that reflects their dissimilarity, N is the number of members in $\mathbf{X}$, and $\overline{\mathbf{x}} = (1/N)\sum_i \mathbf{x}_i$ is the mean of $\mathbf{X}$. A smaller deviation indicates a higher homogeneity of the vectors in the data set, in terms of the distance measure $d()$. In particular, when $\mathbf{X}$ is one-dimensional (1-D) and $d()$ is the Euclidean distance, $dev(\mathbf{X})$ becomes the standard deviation of the data set $\sigma(\mathbf{X})$. The cluster compactness for the output clusters $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_C$ generated by a system is then defined as

$$\mathrm{Cmp} = \frac{1}{C}\sum_i \frac{\mathrm{dev}(\mathbf{c}_i)}{\mathrm{dev}(\mathbf{X})} \tag{17}$$

where C is the number of clusters generated on the data set $\mathbf{X}$, $dev(\mathbf{c}_i)$ is the deviation of the cluster $\mathbf{c}_i$, and $dev(\mathbf{X})$ is the deviation of the data set $\mathbf{X}$.

Cluster separation

The cluster separation measure used here borrows the idea in [16] and the clustering evaluation function introduced by [17]. The cluster separation of a clustering system's output is defined by

$$\mathrm{Sep} = \frac{1}{C(C-1)}\sum_{i=1}^{C}\sum_{j=1, j\neq i}^{C} \exp\left(-\frac{d^2(\mathbf{x}_{c_i}, \mathbf{x}_{c_j})}{2\sigma^2}\right) \tag{18}$$

where σ is a Gaussian constant, C is the number of clusters, $\mathbf{x}_{c_i}$ is the centroid of the cluster $\mathbf{c}_i$, and $d(\mathbf{x}_{c_i}, \mathbf{x}_{c_j})$ is the distance between the centroid of $\mathbf{c}_i$ and the centroid of $\mathbf{c}_j$.

Following a similar practice in [16], we combine the cluster compactness and cluster separation measures into one for the ease of evaluating the overall performance of a clustering system. The combination, named overall cluster quality, is defined as

$$\mathrm{Ocq}(\beta) = \beta \cdot \mathrm{Cmp} + (1-\beta) \cdot \mathrm{Sep} \tag{19}$$

where $\beta \in [0, 1]$ is the weight that balances cluster compactness and cluster separation. For example, $\mathrm{Ocq}(0.5)$ gives equal weights to the two measures.

2) Cluster Validity Measures Based on Class Conformity: This category of validity measures assumes that there is a desirable distribution of the data set with which it is possible to perform a direct comparison of the clustering output. Following the data distribution, one can assign a class label to each data point. The target of the clustering system can then be correspondingly interpreted as to replicate the underlying class structure through unsupervised learning. In an optimal clustering output, data points with the same class labels are clustered into the same cluster and data points with different class labels appear in different clusters. Two validity measures based on class conformity are summarized later.

Cluster entropy

Boley [18] introduced an information entropy approach to evaluate the quality of a set of clusters according to the original class labels of the data points. For each cluster $\mathbf{c}_i$, a cluster entropy Enc_i is computed by

$$\mathrm{Enc}i = -\sum_j \frac{n(l_j, c_i)}{n(c_i)} \log \frac{n(l_j, c_i)}{n(c_i)} \tag{20}$$

where $n(l_j, c_i)$ is the number of the samples in cluster $\mathbf{c}_i$ with a predefined label l_j and $n(c_i) = \sum_j n(l_j, c_i)$ is the number of samples in cluster $\mathbf{c}_i$. The overall cluster entropyEnc is then given by a weighted sum of the individual cluster entropies by

$$\mathrm{Enc} = \frac{1}{\sum_i n(c_i)} \sum_i n(c_i)\mathrm{Enc}_i. \tag{21}$$

The cluster entropy reflects the quality of individual clusters in terms of the homogeneity of the data points in a cluster. A smaller value indicates a higher homogeneity. It, however, does not measure the compactness of a clustering solution in terms of the number of clusters generated. A clustering system that generates many clusters would tend to have very low cluster entropies but is not necessarily desirable. To counter this deficiency, we use another entropy measure below to measure how data points of the same class are represented by the various clusters created.

Class entropy

For each class $\mathbf{l}_j$, a class entropy Enl_j is computed by

$$\mathrm{Enl}_j = -\sum_i \frac{n(l_j, c_i)}{n(l_j)} log \frac{n(l_j, c_i)}{n(l_j)} \tag{22}$$

where $n(l_j, c_i)$ is the number of samples in cluster $\mathbf{c}_i$ with a predefined label l_j and $n(l_j) = \sum_i n(l_j, c_i)$ is the number of the samples with class label l_j. The overall class entropyEnl is then given by a weighted sum of individual class entropies by

$$\mathrm{Enl} = \frac{1}{\sum_j n(l_j)} \sum_j n(l_j)\mathrm{Enl}_j. \tag{23}$$

Similar to the combination paradigm above, we define a combined overall entropy measure to facilitate our comparison

$$\mathrm{Ens}(\beta) = \beta \cdot \mathrm{Enc} + (1-\beta) \cdot \mathrm{Enl} \tag{24}$$

where $\beta \in [0, 1]$ is the weight that balances the two measures.

It is understandable that for all six quality measures above, a smaller score indicates a better performance.

B. Evaluation Session

All five clustering algorithms, namely ART-C 2A, ART 2A, online K-Means, batch K-Means, and SOM, are implemented in-house with C++ and share a common set of functions for vector manipulation. K-Means (both online and batch versions) and SOM utilized Euclidean distances. Their reference clusters were initialized with random vectors that slightly perturbed from the mean vector of the input set.

SOM used a 2-D square map with corresponding square topological neighborhood (resonance domain). Its neighborhood size was initialized with half the total number of nodes. Gaussian neighborhood function was used. Various neighborhood shrinking strategies were tried prehand and the Gaussian shrinking function, which produced slightly better overall performance than others, was used. In addition, in each set of experiments dealing with different data set and different output map size, the Gaussian constants for these functions were fine tuned in order to obtain an locally optimal output.[3]

The learning rates of ART-C 2A, ART 2A, online K-Means and SOM were initialized with 0.05. We applied a simple linear function for the learning rate fading such that the learning rate $\eta^{(t+1)} = 0.9\eta^{(t)}$ if the network's recognition accuracy reached a threshold of 0.8. All five algorithms were said to reach convergence if the cluster assignment for the input samples did not show a relative change of 0.5%.

We utilized the Euclidean distance for the evaluation of cluster compactness (Cmp) and cluster separation (Sep). This makes the cluster compactness measure (Cmp) equivalent to the *average cluster scattering* index used in Halkidi *et al.* study [16]. $2\sigma^2 = 1.0$ as in (18) was used to simplify our evaluation. On each data set, we evaluated the five algorithms on a varying number of output clusters. To obtain a statistically valid comparison, a batch of 10 experiments using the same parameters for each algorithm were conducted. Each experiment randomly reshuffled the sequence of the input and trained the system to converge. The mean and the standard deviation of each evaluation measure over the 10 experiments are reported. t-test was used to evaluate the statistical significance of our comparison observation when appropriate.

We note that the number of output clusters affects the score of all the evaluation measures used in our experiments. Strictly, two systems are not comparable if they work on different number of output clusters. The difficulty in our experiments is to suggest an appropriate ρ value for ART 2A in order to obtain a fixed C number of output clusters over a specific input sequence. To simplify our experiments, we manually tried various ρ values on one random input sequence, then used the ρ value which produced C output clusters on this input sequence in all 10 experiments. While on different input sequences the actual number of ART 2A output clusters may slightly vary from C, we found the variance was within an acceptable level that does not affect the validity of our comparison.

C. Gene Expression Data Sets

Our first batch of the experiments compared the performance of these algorithms on two gene expression data sets, namely the yeast cell cycle data set (YEAST)[4] and the human hematopoietic differentiation data set with features under mixed conditions (HL60_U937_NB4_Jurkat).[5]

Following a common preprocessing procedure [19], a variance filter was used to eliminate the relatively constant gene expressions. 1109 and 1423 gene expressions from the two data sets, respectively, passed the filter. They were normalized using the standard normal distribution with a mean of 0 and a standard variance of 1 within each observation panel [19]. The intermediate ninth condition of the YEAST data set was excluded from our experiments for the ease of normalization. This preprocessing, which produces the same input data sets for all five clustering methods in our experiments, is done before the built-in normalization of ART 2A and ART-C 2A, given in (1).

Both the two gene expression data sets are small scale, have a small number of features, and are densely distributed. Prior studies are capable of identifying a few number of expression patterns on these data sets only. Therefore on each data set, we set the target number of the output clusters to be relatively small. Table I reports the five algorithms' cluster validity measures based on cluster distribution, together with the number of iterations to reach convergence and the CPU time costs, when $C = 9$ and $C = 25$, which correspond to a 3×3 and a 5×5 map in SOM, respectively.

In all four batches of experiments, the cluster validity measures produced by ART-C 2A, in terms of both cluster compactness and cluster separation, were very similar to those of ART 2A. Specifically, t-test did not suggest any significant difference between our observations on each evaluation measure.

In terms of cluster compactness, the validity measures of these five algorithms did not show significant differences on the two data sets with $C = 9$. However, with $C = 25$, both online K-Means and batch K-Means produced significantly lower scores than the rest trio. In general, across these four batches of observations, online K-Means and batch K-Means slightly outperform ART-C 2A and ART 2A in terms of cluster compactness. SOM did not seem to produce outstanding performance compared to the other four algorithms. In terms of cluster separation, the validity measures of both ART-C 2A and ART 2A were significantly lower than those of the rest trio. The difference among the latter three algorithms were not significant in our experiments.

In terms of efficiency, ART-C 2A incurred slightly more computational cost than ART 2A. With $C = 9$, the numbers of iterations used by ART-C 2A and ART 2A were relatively close to those of SOM and Online K-Means, which in turn were significantly fewer than that of batch K-Means. With $C = 25$, both ART-C 2A and ART 2A showed a significantly higher efficiency than online K-Means, batch K-Means and SOM, in the number of iterations as well as the CPU time cost.

D. Reuters-21 578 Text Document Collection

The Reuters-21 578 (REUTERS) text document collection[6] was originally released for evaluation of text categorization methods. The class labels available on each document enable keyword feature selection and quality evaluation using class conformity based measures. The training and testing documents from the top 10 categories of the corpus were used in our experiments. For the ease of our evaluation, documents with multiple class labels were duplicated so that each copy was associated

[3] Detailed parameter settings are not reported due to page constraint.

[4] The YEAST data set is available via http://genomics.stanford.edu

[5] The HL60_U937_NB4_Jurkat data set is available via http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi

[6] The REUTERS corpus is available via http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

TABLE I
Experimental Results for ART-C 2A, ART 2A, SOM, Online K-Means, and Batch K-Means on the YEAST and the HL60_U937_NB4_Jurkat Data Sets, When the Number of Clusters C Were Set to 9 and 25. I, T, Cmp, Sep, and Ocq Indicate the Number of Learning Iterations, the Cost of Training Time (in ms), *Cluster Compactness, Cluster Separation,* and *Overall Cluster Quality*, Respectively. All Values Are Shown With the Mean and the Standard Deviation Over Ten Runs

	YEAST, $C = 9$				
Method	I	T (ms)	Cmp	Sep	Ocq(0.5)
ART-C 2A	7.5 ± 2.3	134 ± 43	**0.7562 ± 0.0326**	**0.1416 ± 0.0063**	**0.4489 ± 0.0183**
ART 2A	6.8 ± 3.1	112 ± 52	0.7595 ± 0.0206	0.1422 ± 0.0071	0.4509 ± 0.0118
SOM	8.9 ± 3.0	169 ± 44	0.7749 ± 0.0324	0.1619 ± 0.0103	0.4684 ± 0.0210
Online K-Means	6.5 ± 1.3	95 ± 18	0.7780 ± 0.0035	0.1512 ± 0.0031	0.4646 ± 0.0006
Batch K-Means	12.3 ± 3.1	144 ± 36	0.7665 ± 0.0073	0.1639 ± 0.0063	0.4652 ± 0.0026
	YEAST, $C = 25$				
Method	I	T (ms)	Cmp	Sep	Ocq(0.5)
ART-C 2A	7.3 ± 3.5	315 ± 176	0.7059 ± 0.0207	0.1658 ± 0.0045	0.4359 ± 0.0102
ART 2A	6.2 ± 3.1	245 ± 117	0.7240 ± 0.0183	**0.1655 ± 0.0050**	0.4448 ± 0.0087
SOM	11.1 ± 3.2	597 ± 173	0.6745 ± 0.0218	0.1983 ± 0.0138	0.4364 ± 0.0174
Online K-Means	14.0 ± 1.9	535 ± 74	**0.5592 ± 0.0228**	0.1834 ± 0.0047	**0.3713 ± 0.0122**
Batch K-Means	12.8 ± 1.8	387 ± 52	0.6496 ± 0.0198	0.1850 ± 0.0046	0.4173 ± 0.0113
	HL60_U937_NB4_Jurkat, $C = 9$				
Method	I	T (ms)	Cmp	Sep	Ocq(0.5)
ART-C 2A	8.4 ± 6.3	178 ± 135	**0.7062 ± 0.0311**	0.1627 ± 0.0147	0.4345 ± 0.0153
ART 2A	6.8 ± 3.0	132 ± 57	0.7090 ± 0.0314	**0.1551 ± 0.0133**	**0.4321 ± 0.0163**
SOM	7.0 ± 2.4	164 ± 37	0.7327 ± 0.0407	0.1917 ± 0.0233	0.4622 ± 0.0320
Online K-Means	7.3 ± 1.9	129 ± 34	0.7188 ± 0.0064	0.1903 ± 0.0133	0.4546 ± 0.0077
Batch K-Means	14.6 ± 5.1	204 ± 72	0.7168 ± 0.0083	0.1897 ± 0.0120	0.4532 ± 0.0079
	HL60_U937_NB4_Jurkat, $C = 25$				
Method	I	T (ms)	Cmp	Sep	Ocq(0.5)
ART-C 2A	8.3 ± 4.9	430 ± 258	0.6267 ± 0.0189	0.1742 ± 0.0071	0.4004 ± 0.0079
ART 2A	6.6 ± 4.4	302 ± 216	0.6573 ± 0.0303	**0.1736 ± 0.0061**	0.4154 ± 0.0140
SOM	10.0 ± 4.2	641 ± 213	0.6541 ± 0.0473	0.2299 ± 0.0307	0.4420 ± 0.0384
Online K-Means	14.0 ± 2.9	635 ± 130	**0.5370 ± 0.0314**	0.2141 ± 0.0075	**0.3755 ± 0.0144**
Batch K-Means	13.1 ± 2.6	482 ± 93	0.6069 ± 0.0121	0.2238 ± 0.0087	0.4154 ± 0.0083

with one class label accordingly. A bag-of-words representation of document features was adopted in our experiments. The $CHI(\chi)$ statistics [20] was employed as the ranking metric for feature selection. 365 keywords that passed the preset threshold $\chi \geq 15$ were selected as the features. During document feature extraction, the content of each document was first represented as an in-document term frequency (TF) vector, then processed using an inverse document frequency (IDF) based term weighting method and subsequently Euclidean normalized [20]. After removing null (i.e., all-zero) vectors, we obtained a set of 9530 document vectors for our benchmark.

To obtain a better understanding of each algorithm's learning efficiency, we tested them on 10 subsets of the REUTERS data set constructed as follows. Documents from each class were evenly split into 10 folds. The ith subset used in our experiments contained document folds $1, \ldots, i$ from each category. In this way, all 10 subsets used in our experiments had nearly identical document class distribution, while the number of data samples in each subset varied from 957 to 9530.

In contrast to the two gene expression data sets, the REUTERS data set is relatively high-dimensional, large-scale, noisy, and sparsely distributed. Therefore, we did not expect a cluster algorithm to replicate the exact 10 clusters corresponding to the labeled classes in our experiments. Instead, we tested the algorithms on each subset with $C = 25$, $C = 49$, and $C = 81$, corresponding to a 5×5, 7×7, and 9×9 map in SOM, respectively. The comparative experiments with different C values showed very similar results. Experimental results with $C = 49$ are reported in Fig. 3. Cluster validity was measured using both cluster distribution and class conformity.

It is interesting that all five algorithms produced rather consistent cluster validity scores in response to the varying number of input samples. This is probably due to the similar data distribution in each subset used in the experiments. In terms of

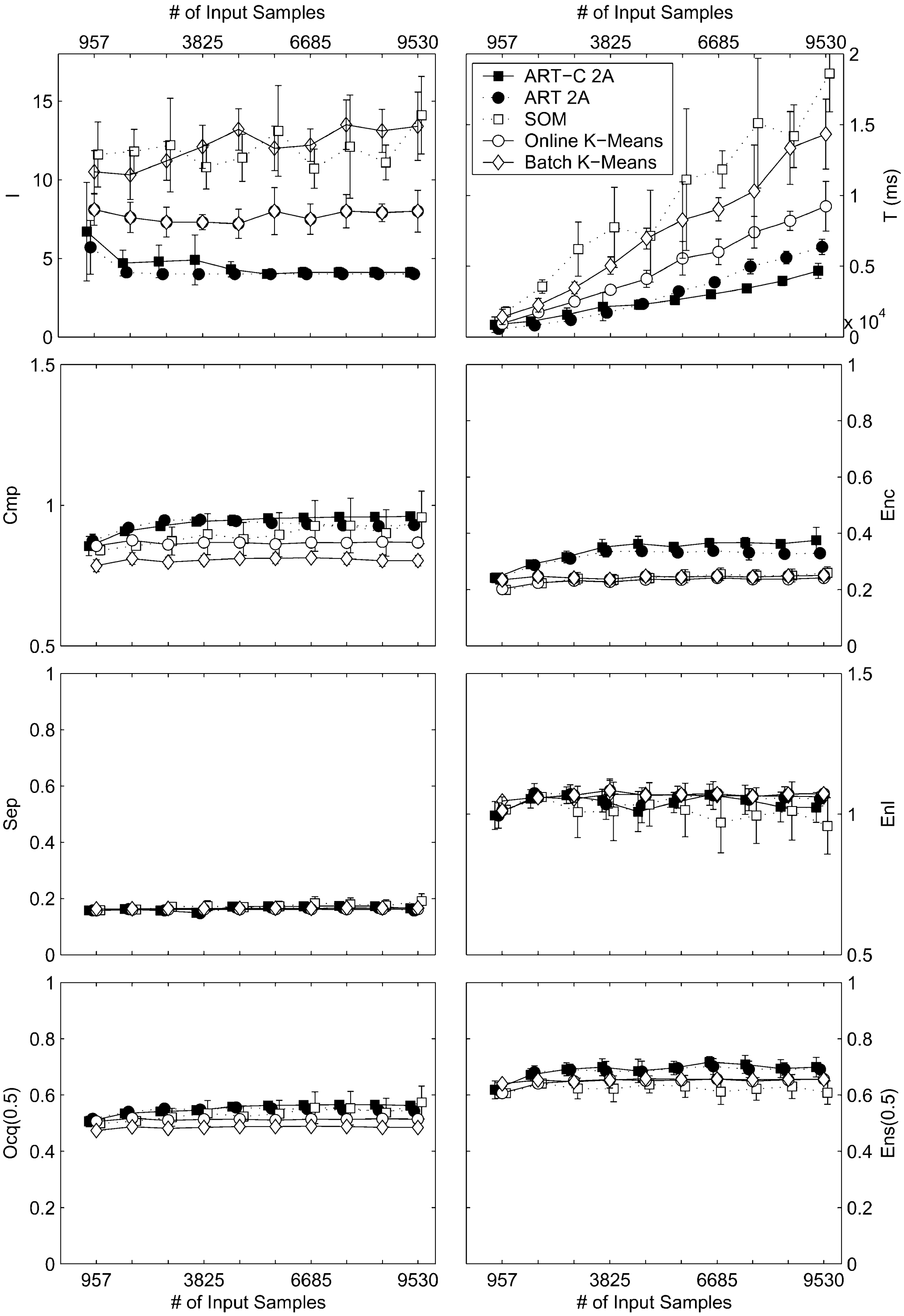

Fig. 3. Experimental results for ART-C 2A, ART 2A, SOM, Online K-Means, and batch K-Means on the reuters-21 578 data set with 49 clusters. I and T indicate the number of learning iterations and the cost of training time (ms), respectively. Sep, Cmp, and Ocq indicate *cluster separation*, *cluster compactness*, and *overall cluster quality*, respectively. Enc, Enl, and Ens indicate *cluster entropy*, *class entropy*, and *overall entropy*, respectively. All values are shown with the mean and the standard deviation over 10 runs.

cluster compactness (Cmp), batch K-Means produced significantly better scores than online K-Means and SOM in all the experiments, while the latter two in turn performed slightly better than ART-C 2A and ART 2A in most experiments. In terms of cluster separation (Sep), all five algorithms performed quite similarly in all experiments.

Using the set of validity measures based on class conformity, ART-C 2A and ART 2A produced significantly higher cluster entropy (Enc) scores than those of online K-Means, batch K-Means, and SOM, while the performance of the latter three methods were quite close. As for class entropies (Enl), all algorithms produced similar scores. Interestingly, these observations generally harmonize with the comparable results using the cluster distribution based measures.

In terms of efficiency, both ART-C 2A and ART 2A showed a significantly higher efficiency than online K-Means by about one time, which in turn was significantly faster than batch K-Means and SOM in all experiments. This is reflected by both the number of iterations and the CPU time cost.

E. Discussions

Our benchmark on the cluster validity of the five clustering algorithms led to mixed results. Generally speaking, the performance of ART-C 2A is quite comparable to that of ART 2A. Compared with online K-Means and batch K-Means, on the gene expression data sets, ART-C 2A and ART 2A output with comparable intra-cluster compactness and better inter-cluster separation. While on the REUTERS data set, ART-C 2A and ART 2A output with worse intracluster compactness and comparable intercluster separation, both reflected by the cluster distribution based measures and class conformity based measures.

As another large family of selforganizing neural networks, SOM did not show notably outstanding performance over others, even with the optimized parameter settings in our controlled experiments. However, readers shall note the application domain of SOM is mainly on topology preserving mapping and visualization, rather than clustering.

Additionally, we must point out that the observations above reflect the nature of clustering. As a matter of fact, the ill-posted clustering problem "precludes an absolute judgement as to the relative efficacy of all clustering techniques" [21].

We are particularly interested in the relatively high efficiency of ART-C 2A and ART 2A reflected in our controlled experiments. This may be due to their capabilities of dynamically initializing the reference clusters using distinct input samples through the network's mismatch reset cycle. Mismatch reset ensures stable encoding of new samples through one scan. The constraint reset process in ART-C 2A also serves to move cluster centroids quickly from a high density area to a low density area, with minor impact to the learning history. These factors guarantee the high efficiency of ART-C 2A, which is comparable to that of ART 2A. However, when working with too few number of output clusters, which corresponds to a very low vigilance threshold, such an advantage is not notable in our experiments, as both ART-C 2A and ART 2A work rather like the competitive learning in this scenario.

Despite the advantage above, readers shall note that both ART-C 2A and ART 2A have a built-in Euclidean normalization on the input and the category representation in order to avoid category proliferation. As such, the input vector length information is ignored by the networks. This limits the application of ART-C 2A and ART 2A to problems where the input vector length information is not of critical importance.

VI. Conclusion

As our concluding remarks, the ART-C 2A learning paradigm retains the efficient cluster creation capability of ART 2A, and allows a user to directly control the number of the output clusters by imposing a constraint on ART 2A's category learning. The constraint reset mechanism of ART-C 2A adaptively adjusts the network's vigilance threshold which guides the network's learning and redistributes the recognition categories to satisfy the constraint. As such, unlike a conventional ART 2A module which requires prior knowledge in estimating an appropriate *vigilance* parameter, the knowledge in estimating an optimal *number of clusters* over the data set is required by an ART-C 2A module. We consider this is a good alternative to the conventional ART 2A module and is of great value for various real-life applications. While this paper focuses on ART-C 2A, which is based on ART 2A, the same idea may be applied to other ART modules.

Acknowledgment

We would like to thank our colleague, H. Pan, for the extensive discussions on the cluster validity measures based on class conformity. In addition, we are very grateful to the anonymous referees and the Associate Editor for their valuable comments on our work.

References

[1] G. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vis. Graph. Image Processing*, vol. 34, pp. 54–115, 1987.

[2] G. Carpenter, S. Grossberg, and D. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks*, vol. 4, pp. 493–504, 1991.

[3] J. He, A. Tan, and C. Tan, "ART-C: A neural architecture for self-organization under constraints," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2002, pp. 2550–2555.

[4] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Comput. Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[5] G. Bartfai and R. White, "Incremental learning and optimazation of hierarchical clusterings with ART-based modular networks," in *Innovations in ART Neural Networks*, L. Jain, B. Lazzerini, and U. Halici, Eds. New York: Physica-Verlag, 2000, pp. 87–132.

[6] A. Tan, "Adaptive resonance associative map," *Neural Netw.*, vol. 8, no. 3, pp. 437–446, 1995.

[7] G. Carpenter, S. Grossberg, and J. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by self-organizing neural network," *Neural Networks*, vol. 4, pp. 565–588, 1991.

[8] C. D. Michener and R. R. Sokal, "A quantitative approach to a problem in classification," *Evolution*, vol. 11, pp. 130–162, 1957.

[9] N. Saitou and M. Nei, "The neighbor-joining method: A new method for reconstructing phylogenetic trees," *Molec. Biol. Evolut.*, vol. 4, pp. 406–425, 1987.

[10] G. Patane and M. Russo, "Fully automatic clustering system," *IEEE Trans. Neural Networks*, vol. 13, pp. 1285–1298, 2002.

[11] B. Fritzke. (1997) Some Competitive Learning Methods. [Online]. Available: http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/

[12] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. Neural Networks*, vol. 21, pp. 450–465, 1999.

[13] J. MacQueen, "Some methods for classification and analysis of multivariante observations," in *Proc. 5th Berkely Symp. Mathematics and Probability*, Berkeley, CA, 1967, pp. 281–297.

[14] T. Kohonen, *Self-Organization and Associative Memory*, 2 ed, T. Huang and M. Schroeder, Eds. Berlin, Germany: Springer-Verlag, 1997.

[15] J. He, A. Tan, C. Tan, and S. Sung, "On quantitative evaluation of clustering systems," in *Information Retrieval and Clustering*, W. Wu, H. Xiong, and S. Shekhar, Eds. Boston, MA: Kluwer, 2003.

[16] M. Halkidi, M. Vazirgiannis, and I. Batistakis, "Quality scheme assessment in the clustering process," in *Proc. 4th Eur. Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2000, pp. 265–276.

[17] E. Gokcay and J. Principe, “A new clustering evaluation function using Renyi’s information potential,” in *Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, 2000.

[18] D. Boley, “Principal direction divisive partitioning,” *Data Mining and Knowledge Discov.*, vol. 2, no. 4, pp. 325–344, 1998.

[19] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub, “Interpreting patterns of gene expression with self-organizing maps: Methods and application to homatopoietc differentiation,” *Proc. Nat. Acad. Sci.*, vol. 96, pp. 2907–2912, 1999.

[20] Y. Yang and X. Liu, “A re-examination of text categorization methods,” in *Proc. 22nd Ann. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 1999, pp. 42–49.

[21] A. Baraldi and E. Alpaydm, “Constructive feedforward ART clustering networks—Part I,” *IEEE Trans. Neural Networks*, vol. 13, pp. 645–661, 2002.

Ji He (M’01) received the B.S. degree in electronic engineering, in 1997 and the M.S. degree in information management, in 2000, both from Shanghai Jiaotong University, China.

He is a Research Engineer with the Institute of Engineering Science, Singapore. He is currently working toward the Ph.D. degree at the Department of Computer Science, School of Computing, National University of Singapore and the Institute for Infocomm Research. His research interests include text mining, knowledge discovery, machine learning, and neural networks.

Ah-Hwee Tan received the B.S. degree (first class honors) and the M.S. degree in computer and information science from the National University of Singapore, in 1989 and 1991, respectively, and the received the Ph.D. degree in cognitive and neural systems from Boston University, Boston, MA, in 1994.

He is an Associate Professor with the School of Computer Engineering, Nanyang Technological University. He was a Research Manager and Senior Member of Research Staff with the Kent Ridge Digital Labs, Laboratories for Information Technology, and Institute for Infocomm Research, where he led research and development projects in knowledge discovery, document analysis, and information mining.

Dr. Tan is an editorial board member of *Applied Intelligence* and a member of the Singapore Computer Society, ACM, and ACMSIGKDD.

Chew-Lim Tan (SM’85) received the B.Sc. (Hons.) degree in physics from the University of Singapore in 1971, the M.Sc. degree in radiation studies from the University of Surrey, U.K., in 1973, and the Ph.D. degree in computer science from the University of Virginia, in 1986.

He is an Associate Professor with the Department of Computer Science, School of Computing, National University of Singapore. His research interests include document image and text processing, neural networks, genetic programming, and expert systems.

Dr. Tan is an Associate Editor of *Pattern Recognition*. He has served on the program committees for ICPR 2002, GREC 2001, GREC 2003, WDA 2001, WDA 2003, DIAR 2003, DIAL 2004, and ICDAR 2005.