

ART-C: A Neural Architecture for Self-Organization Under Constraints

Ji He*, Ah-Hwee Tan[†], and Chew-Lim Tan*

*School of Computing, National University of Singapore,
3 Science Drive 2, Singapore 117543

{heji,tancl}@comp.nus.edu.sg

[†]Kent Ridge Digital Labs,

21 Heng Mui Keng Terrace, Singapore 119613

ahhwee@krdl.org.sg

Abstract— This paper proposes a novel ART-based neural architecture known as ART-C (ART under Constraints) that performs online clustering of pattern sequences subject to the constraints on the recognition category representation. Experiments on two real-life data sets show that ART-C produces reasonably good clustering qualities, with the added advantage of high efficiency.

Keywords— Adaptive Resonance Theory, Constraint clustering, Machine learning.

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 size of its recognition categories with respect to the complexity of the input set. Although the self-adaptive capability is a key advantage of ART, in many real-life applications, such as personal online content management, or topic detecting and tracking, it is desirable to restrict the number of the generated clusters to a manageable scale. In principle, one could control the size of the category representation in ART by fine tuning a vigilance value. However, suggesting an appropriate vigilance for a given problem requires prior knowledge on the input data distribution.

In this paper, a novel neural architecture named ART-C (for “ART under Constraints”) is proposed to perform online clustering of input pattern sequences under the user’s constraints on the category representation, in terms of the number of output clusters. We introduce the architecture and the learning paradigm of ART-C, and evaluate its properties through controlled experiments on two real-life data sets.

The rest of this paper are organized as follows. Section II reviews the learning paradigm of ART network in necessary details. Section III proposes the architecture and the learning paradigm of the ART-C network. Section IV introduces our comparative experiments and reports the experimental results and our findings on two real-life data sets, namely the Iris data set and the Reuters-21578 corpus. The last section summarizes our conclusions.

II. ADAPTIVE RESONANCE THEORY (ART) NETWORKS

An ART network consists of three layers: the input layer (F_0), the comparison layer (F_1), and the recognition layer (F_2) (Figure 1). 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. The recognition layer F_2 stores the prototypes of input categories (clusters). Learning of the network modifies the weighted bottom-up (feed-forward) and top-down (feed-backward) connections between F_1 and the recognition layer F_2 . Interactions between F_1 and F_2 are controlled by the orienting subsystem using a vigilance threshold ρ . The learning paradigms of the fuzzy ART network and the ART-2 network are summarized in the following sub-sections.

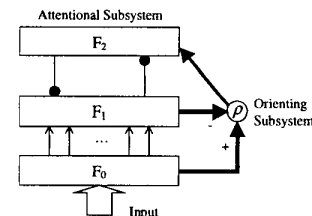


Fig. 1. The Adaptive Resonance Theory architecture

A. The Fuzzy ART Network

Parameters: The fuzzy ART dynamics are determined by the choice parameter $\alpha > 0$, the learning rate $\eta \in (0, 1]$, and the vigilance parameter $\rho \in [0, 1]$.

Input vectors: Fuzzy ART preprocesses the input vectors by either normalization or complement coding to avoid category proliferation. Complement coding preserves the input vector’s amplitude information and represents both the on-response and the off-response to the input vector. However, it doubles the number of network connections. Given an M -dimensional input \mathbf{a} , the complement coded F_1 input vector \mathbf{A} is a $2M$ -dimensional vector

$$\mathbf{A} = (\mathbf{a}, \mathbf{a}^c) \equiv (a_1, \dots, a_M, a_1^c, \dots, a_M^c) \quad (1)$$

where $a_i^c \equiv 1 - a_i$.

Network initialization: The recognition layer F_2 is initialized with the null set $\{\}$ (i.e. the initial F_2 layer does not contain any category). Some implementations (such as [2]) initialize F_2 with a so-called *uncommitted* node, which essentially equals to a null category set.

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

$$T_j = \frac{|\mathbf{A} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, \quad (2)$$

where the fuzzy AND operation \wedge is defined by

$$(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i) \quad (3)$$

and the norm $|\cdot|$ is defined by

$$|\mathbf{p}| \equiv \sum_i p_i \quad (4)$$

for vectors \mathbf{p} and \mathbf{q} .

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

$$T_J = \max\{T_j : \text{for all un-reset node } j \text{ in } F_2\}. \quad (5)$$

Resonance or reset: Resonance occurs if the *match* function M_J in the orienting subsystem meets the vigilance criteria:

$$M_J = \frac{|\mathbf{A} \wedge \mathbf{w}_J|}{|\mathbf{A}|} \geq \rho. \quad (6)$$

Learning then ensues, as defined below. If the vigilance constraint is violated, mismatch reset occurs. In a mismatch reset, the F_2 node J is excluded from the search process (as in Equation 5) for the duration of the input presentation and the search process is repeated until the network either finds an existing category whose prototype meets the vigilance criteria, or inserts the input prototype into F_2 as a new reference category. Insertion of a new category is normally done by creating an all-ones new node in F_2 as the winning node \mathbf{w}_J and temporarily set the learning rate to 1.0 in the following learning process.

Learning: Once the search ends, the attentional subsystem updates the weight vector \mathbf{w}_J according to the equation

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

B. The ART-2 Network

The ART-2 network generally follows the identical learning process as that of fuzzy ART. The major variances lie on the different set of *choice*, *match*, and *learning* functions for ART-2. ART-2 utilizes the cosine similarity during category search and resonance vigilance checking:

$$T_j = M_j = \frac{\mathbf{A} \cdot \mathbf{w}_j}{\|\mathbf{A}\| \|\mathbf{w}_j\|}, \quad (8)$$

where the L_2 -norm function $\|\cdot\|$ is defined by

$$\|\mathbf{x}\| = \sqrt{\sum_i x_i^2} \quad (9)$$

for vector \mathbf{x} . The learning function is given by

$$\mathbf{w}_J^{t+1} = (1 - \eta)\mathbf{w}_J^t + \eta\mathbf{A}. \quad (10)$$

Since ART-2 uses L_2 distance metrics in which manipulation is required, it is generally more computationally complex than fuzzy ART. However, it is normally not required to complement code the input vectors in the ART-2 architecture. In this sense, ART-2 is more memory compact than fuzzy ART.

C. Properties of ART

Among the properties of ART networks widely discussed in a large number of publications [3][4][5], highlighting the role of the orienting subsystem in the ART architecture helps to understand the rest of this paper. It is noted that the vigilance parameter ρ in the orienting subsystem governs the online generation of new neurons in the attentional subsystem, i.e. the number of created categories in F_2 layer. The higher the ρ value, the higher is the possibility for the vigilance criteria (Equation 6) to be violated, therefore the larger number of categories are generated and the more specific each category may be. In particular, $\rho = 1$ will yield one new category for every unique input. Hence it is applicable to control the granularity of the category representation by altering the vigilance value.

III. THE ART-C NETWORK

The ART-C (for "ART under Constraints") network proposed in this paper is an ART-based architecture capable of performing online clustering of arbitrary input sequences while keeping the size of the category field in the desired scale (Figure 2). Compared with the standard ART architecture, a constraining subsystem is added in the ART-C network, which interacts with the attentional subsystem and the orienting subsystem. During learning, the constraining subsystem adaptively estimates the distribution of the input data and self-adjusts the vigilance parameter for the orienting subsystem, which in turn governs the learning activities in the attentional subsystem.

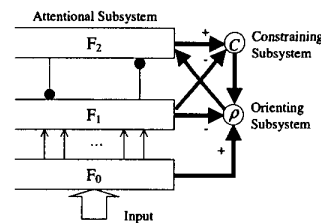


Fig. 2. The ART-C architecture

The ART module used in the ART-C architecture can be either ART-1, ART-2, or fuzzy ART. The fuzzy ART-C algorithm based on fuzzy ART is introduced in the following section.

A. The Fuzzy ART-C Network

Parameters: The fuzzy ART-C dynamics are determined by the choice parameter $\alpha > 0$, the learning rate $\eta \in$

(0, 1], and the constraint C on the number of recognition categories.

Network initialization: The architecture initializes the ART network with $\rho = 1.0$ (or a reasonable high value optionally given by the user).

Constraint checking: Presented with an input vector A in the F_1 layer, constraint checking occurs before the normal ART searching process for a winner category is carried out. A buffer of *match* scores M_j (see Equation 6) for every F_2 node j is calculated and held in F_1 . Constraint checking examines two threshold (binary) criteria on F_1 layer and F_2 layer correspondingly, namely the *match checking* function defined by:

$$Mc = \begin{cases} 1 & \max\{M_j : \text{for all } F_2 \text{ node } j\} < \rho \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

and the *category size checking* function defined by:

$$Sc = \begin{cases} 1 & c \geq C \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where c is the number of categories in F_2 . *Constraint reset* occurs when the following criteria is met:

$$Mc \cdot Sc > 0, \quad (13)$$

during which the network follows the process described as below to self-adjust the vigilance value and encode the input A . Otherwise, the network follows the standard fuzzy ART learning paradigm to encode the input A .

Constraint reset: A constraint reset signal is sent to both the orienting subsystem and the attentional subsystem¹, which consequently updates the vigilance value ρ and re-organizes the recognition layer F_2 towards the satisfaction of the constraints. The process is sequentially introduced as follows:

1. Insert input A into F_2 layer as a new reference prototype.
2. Calculate the pair-wise fuzzy *match* score $M_{i,j}$ of every F_2 node pairs (i, j) :

$$M_{i,j} = \frac{|\mathbf{w}_i \wedge \mathbf{w}_j|}{|\mathbf{w}_i|} \quad (14)$$

for all $i \neq j$. The search process locates a winner pair (I, J) according to:

$$M_{I,J} = \max\{M_{i,j} : \text{for all } i \text{ in } \{R\}\}, \quad (15)$$

where $\{R\}$ is the set of F_2 nodes indexed by the criteria:

$$\{R\} = \{F_2 \text{ node } i \text{ whose } \max\{M_{i,j} : \text{for all } j \neq i\} < \rho\}. \quad (16)$$

It is understandable that the toggling precondition in Equation 11 guarantees $\{R\} \neq \{\}$ in this stage. That is, the search process can always find the winner pair (I, J) .

3. Modify the vigilance value according to the match score between the winner pair:

$$\rho^{new} = M_{I,J}. \quad (17)$$

¹To simplify the illustration, the signal to the attentional subsystem is not reflected in Figure 2.

4. Update \mathbf{w}_J with \mathbf{w}_I , by utilizing the fuzzy ART learning function:

$$\mathbf{w}_J^{t+1} = (1 - \eta)\mathbf{w}_J^t + \eta(\mathbf{w}_I^t \wedge \mathbf{w}_J^t). \quad (18)$$

5. Delete node I from F_2 layer.

The constraint reset process practically causes two possible changes on the recognition layer, i.e. either the input is coded into the most similar category under a compromised vigilance criteria, or the input is directly inserted into the category field before two more similar categories (in terms of the match score) are merged. Both these two changes result in a more generalized category representation in F_2 . Moreover, the vigilance ρ is adaptively decreased, which leads the system to work on a coarser representation of the input sequence in order to satisfy the constraint on the number of output clusters.

Batch learning: Batch training of the network corresponds to a finite number of N inputs being repeatedly presented and encoded until the network converges. The network is said to converge only when both the following criteria are satisfied:

1. **Convergence of the ART network:** The number of mismatch resets r in a learning iteration reaches the convergence level $\varepsilon > 0$ such that $r/N \leq \varepsilon$.
2. **Satisfaction of the constraints:** The number of category prototypes c in F_2 satisfies $|c - C| \leq e$, where e is the *maximal error* in integer value for constraint satisfaction.

During online encoding of each input, the constraint reset mechanism decreases the system vigilance ρ only. This practically alters the system to work on a gradient fine-to-coarse representation of the input sequences. However, online decreasing of ρ increases the risk of dead unit generation, as some specialized F_2 categories may always be not selected under the relaxed vigilance criteria. Therefore, neuron pruning is normally required after each iteration of batch learning. In case that the actual number of F_2 categories c drops below the lower bound of constraint $(C - e)$ after neuron pruning, an additional signal is sent by the constraining subsystem so that the next iteration will utilize a slightly higher vigilance value:

$$\rho^{t+1} = \rho^t + \theta \quad (19)$$

where θ is a small positive value. This will cause a larger number of categories to be generated in the F_2 layer in the subsequent learning iteration.

B. Computational Complexity of The ART-C Network

The computational complexity of standard ART has been widely discussed in the literature. We hereby discuss the additional time and memory cost brought by the constraining subsystem.

During constraint checking, a maximum of C match scores are calculated. Pairwise matching of F_2 nodes requires at most $C \cdot (C + 1)$ times of match function calculation. When C is large, this operation could be time consuming. However, since encoding of each input changes at most two prototypes of the F_2 nodes, by using a buffering mechanism that stores the pair-wise match scores into a C

by C matrix, the calculations for each pairwise matching phase can be reduced to a maximum of $4C$ times. Therefore, for online encoding of each input pattern, the computational complexity of the constraining subsystem can be estimated as: $O(C)$ in times of match score calculation and $O(C^2)$ in terms of memory.

C. Properties of ART-C

This section describes the main properties of the ART-C network.

1. **Online satisfaction of constraints:** Given a finite number of input presentations, the system is capable of estimating the distribution of the input in an online manner and suggesting an appropriate ρ value depending on the desired granularity of the category representation. Estimation of ρ value does not require prior knowledge of the problem. This empowers the system with the capability of handling incrementally incoming pattern stream.

2. **Stability of learning:** Once a ρ value is determined, further learning process of the network follows the standard ART learning paradigm. Therefore the ART-C learning paradigm is stable.

3. **Order of presentation and over-fitting:** Learning of ART-C is affected by the order of input presentations and may over-fit on noises and outliers. These deficiencies are inherited from the traditional ART networks.

IV. EXPERIMENTS

The goal of our experiments on the ART-C network is twofold. First, we demonstrate ART-C's capability of satisfying user's constraints on real-life data sets. Second, based on a set of widely used evaluation measures, we compare the performance of ART-C with the state-of-the-art clustering methods in the literature.

A. Iris Data Set

The Iris data set [6] is popularly used in mathematical analysis, clustering, and classification research. Suggested by [7], we used two features (i.e. *petal length* and *petal width*) of each data point for the ease of demonstration (Figure 3).

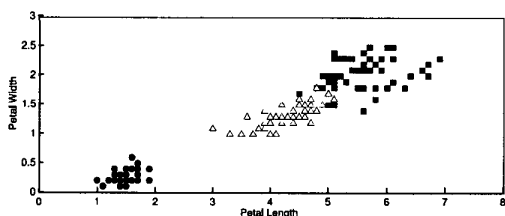


Fig. 3. The Iris data set using two features, i.e. *petal length* and *petal width*. Data points from different categories are identified with different shapes.

Our experiment paradigm on the Iris data set is described as follows. Given an arbitrary input sequence and a constraint C on the number of clusters, we recorded the number of clusters (c_1) and the vigilance value of fuzzy ART-C (ρ) after it converged. The vigilance value ρ and the same input sequence were subsequently used to train

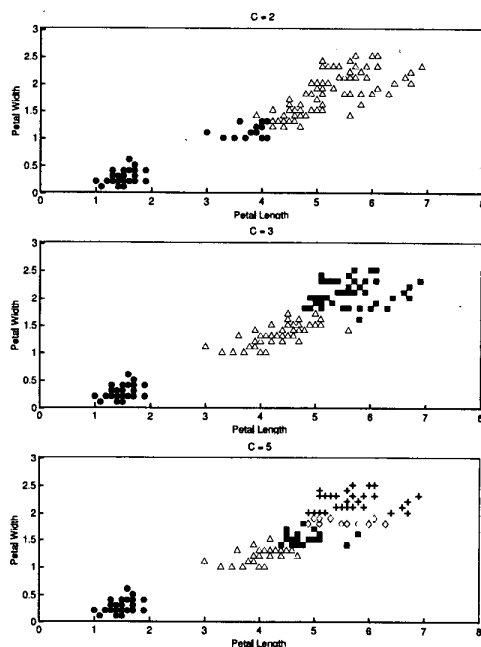


Fig. 4. The outputs of fuzzy ART-C on the Iris data set when we set $C = 2, 3,$ and 5 respectively. The network converged at $\rho = 0.5958, 0.7187,$ and 0.7542 correspondingly.

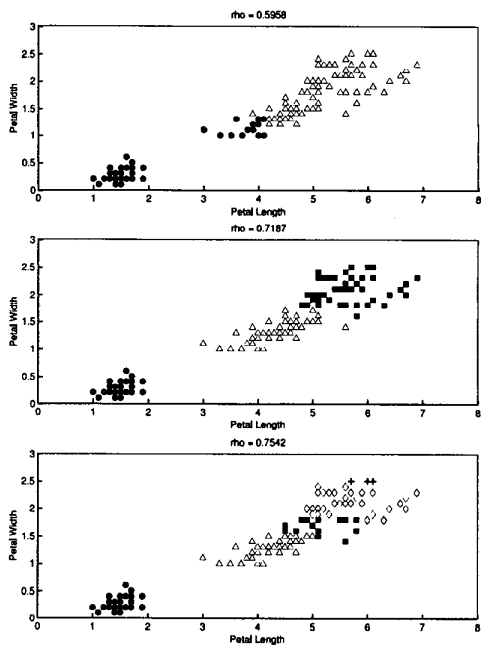


Fig. 5. The outputs of fuzzy ART on the Iris data set when we set $\rho = 0.5958, 0.7187,$ and 0.7542 respectively. The network outputted 2, 3, and 5 clusters correspondingly.

fuzzy ART, which generated c_2 number of clusters after convergence. By comparing c_1 with C , we examined the capability of ART-C in estimating the distribution of the input and satisfying the user's constraints. While by com-

paring the clusters generated by fuzzy ART-C with those generated by fuzzy ART, we empirically examined the quality of ART-C's output, and validated the theoretical soundness of this new architecture.

Figure 4 depicts the outputs of fuzzy ART-C with $C = 2, 3,$ and 5 respectively. As a comparison, figure 5 illustrates the outputs of fuzzy ART using the corresponding vigilance values suggested by fuzzy ART-C in the prior experiments. In all of our experiments, fuzzy ART-C effectively adjusted its vigilance value ρ to produce the required number of clusters. Working under $C = 2$ and 3 , fuzzy ART-C generated the identical outputs with those of fuzzy ART using the same vigilance values. With $C = 5$, the outputs of the two networks were reasonably comparable. The results demonstrated the capability of ART-C in the effective estimation of the input distribution and satisfaction of user's constraints on the category representation, without losing the clustering quality of ART.

B. Reuters-21578 Corpus

Our experiments on the Reuters-21578 corpus compared the performance of ART-C with those of Self-Organizing Map (SOM) [8][9] and k-means [10], both have been extensively studied in the literature. The major differences among the trio can be summarized as below: SOM performs incremental and soft learning with relatively low learning rate that requires a large number of learning iterations; k-means is an optimization-based off-line learning method; while ART-C follows an incremental *winner-take-all* (WTA) learning strategy which is capable of both fast and slow learning.

B.1 Data Preparation

The training and testing sets for the top 10 categories from the Reuters-21578 corpus were used in our experiments. We adopted the bag-of-words representation of document features. *CHI* (χ) statistics [11] was employed as the ranking metric for feature selection. Based on a bag of 335 top-ranking keyword features, the content of each document was represented as an in-document term frequency (TF) vector, which was then processed using an inverse-document frequency (IDF) based weighting method [12] and subsequently normalized. Null vectors (i.e. vectors with all attributes valued 0) were removed from the data set.

B.2 Evaluation Measures

Based on a study of the various clustering validity methods [7][13], we adopted two quality evaluation measures in our experiments, namely: *cluster scattering* and *cluster separation*. The definitions of these measures are introduced as follows.

Cluster Scattering: Halkidi et. al. proposed a set of validity indices based on the variance of a data set [7]. The variance of a data set $\{X\}$ is defined by

$$\sigma(X) = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})^2 \quad (20)$$

where n is the number of members in $\{X\}$ and $\bar{\mathbf{x}} = \frac{1}{n} \sum_i \mathbf{x}_i$ is the mean of $\{X\}$. The average scattering for the clusters generated by a system is defined as

$$Sct = \frac{1}{C} \sum_i \frac{\|\sigma(c_i)\|}{\|\sigma(X)\|}, \quad (21)$$

where C is the number of clusters generated on the data set $\{X\}$, $\sigma(c_i)$ is the variance of the cluster c_i , and $\sigma(X)$ is the variance of the data set $\{X\}$.

Cluster Separation: We borrowed the idea of the metric *total separation between clusters* [7] and redefined the measure *cluster separation* for our experiments, based on the clustering evaluation function introduced by [13]. The cluster separation is defined by

$$Sep = \frac{1}{C(C-1)} \sum_{i=1}^C \sum_{j=1, j \neq i}^C \exp\left(-\frac{\|\mathbf{x}_{c_i} - \mathbf{x}_{c_j}\|^2}{2\sigma^2}\right), \quad (22)$$

where C is the number of clusters and \mathbf{x}_{c_i} is the prototype of the center of cluster c_i .

The cluster scattering index evaluates the similarity of data within clusters, while the cluster separation score reflects the distance among clusters. For both measures, a smaller value indicates a better performance. Although it is possible to combine the two indices into one for evaluation convenience, our experiments examine the two measures separately for a better understanding of the properties of the three methods.

In addition, the time complexity of each system, in terms of the number of learning iterations and the CPU time used on each experiment, were reported and compared. Based on these, we empirically examined the learning efficiency of each method.

B.3 Evaluation Paradigm

Our experiments compared the performances of ART2-C (based on ART-2), SOM, and k-means respectively. All trio used the cosine similarity measure and utilized a typical set of parameters.

We conducted two batches of experiments, with a constraint of 25 and 81 clusters respectively. Each batch of experiments contained ten runs. Every run randomly shuffled the presenting sequence of input set and trained the three systems to convergence. $2\sigma^2 = 1.0$ (as in Equation 22) was used for computational simplicity in evaluating cluster separation. t-statistics was employed to validate the significance of our comparative observations across the ten runs.

B.4 Results and Discussions

Table I reports the experimental results on ART2-C, SOM, and k-means respectively. I and T stand for the number of learning iterations and the cost of training time (in seconds) respectively. Value pairs before and after the "±" signs denote the means and the standard deviations over ten observations respectively.

Working on either 25 or 81 clusters, the cluster scattering indices (Sct) of ART2-C outputs were generally higher

TABLE I

EXPERIMENTAL RESULTS FOR ART2-C, SOM, AND K-MEANS ON THE REUTERS-21578 CORPUS, WHEN THE NUMBER OF CLUSTERS WERE SET TO 25 AND 81 RESPECTIVELY.

	cluster no = 25		
	ART2-C	SOM	k-means
<i>I</i>	2.2±0.4	10.9±2.0	11.2±1.8
<i>T</i> (s)	42.724±15.537	103.356±36.543	90.018±13.738
<i>Sct</i>	0.5187±0.0044	0.4681±0.0086	0.4560±0.0106
<i>Sep</i>	0.2063±0.0143	0.2312±0.0286	0.2248±0.0617
	cluster no = 81		
	ART2-C	SOM	k-means
<i>I</i>	2.8±1.0	11.8±1.6	12.3±1.3
<i>T</i> (s)	88.057±45.738	323.850±74.285	310.836±33.656
<i>Sct</i>	0.4594±0.0037	0.3957±0.0116	0.4126±0.0041
<i>Sep</i>	0.1874±0.0243	0.1968±0.0187	0.2135±0.0217

TABLE II

STATISTICAL SIGNIFICANCE OF OUR CROSS-METHOD COMPARISONS BETWEEN ART2-C, SOM, AND K-MEANS ON THE REUTERS-21578 CORPUS. ">>" AND ">" (OR "<<" AND "<") DENOTE THE LEFT-SIDE VALUE IS GREATER (OR SMALLER) THAN THE RIGHT-SIDE VALUE AT SIGNIFICANCE LEVEL 0.01 AND 0.05 RESPECTIVELY.

	cluster no = 25		cluster no = 81	
	ART2-C vs. SOM	ART2-C vs. k-means	ART2-C vs. SOM	ART2-C vs. k-means
<i>T</i>	<	<<	<<	<<
<i>Sct</i>	>>	>>	>>	>>
<i>Sep</i>	<<	<	<	<<

(which indicated worse performance) than those of SOM and k-means. Our explanations are as follows: The learning paradigms of SOM and k-means minimize the mean square error of the data points within the individual clusters. Therefore both SOM and k-means are more capable of generating compact clusters. In terms of cluster separation (*Sep*), the outputs of ART2-C were better than those of SOM and k-means. It may be that, whereas SOM and k-means modify existing cluster prototypes to encode new samples, ART adaptively inserts recognition categories to encode new input samples that are significantly distinct from existing prototypes. This unique neuron initialization mechanism appeared to be effective in representing diverse data patterns in the input set. Compared with SOM and k-means, the various clusters generated by ART2-C in our experiments were more significantly dissimilar to each other. t-statistical validations (Table II) suggested very high significance for our observations. Noting the difficulty in combining the two performance measures into one, the performance of ART2-C is roughly comparable to those of SOM and k-means.

Besides the clustering quality of the trio, we are particularly interested in the learning efficiency of each method. Table I and II showed that ART2-C was significantly faster than SOM and k-means in our controlled experiments, in terms of the average training time in each experiment. This demonstrated the promising learning efficiency of ART-C network architecture and suggested its strength in handling massive real-life data on the fly.

The training time of ART2-C however varied signifi-

cantly in some experiments as reflected in Table I by the relatively large standard deviations of the number of learning iterations and training time. The observation showed that ART-C was sensitive to the sequence of the input order. This is one of the structural drawbacks of ART-C. SOM on the other hand also seemed to show the similar deficiency in our experiments.

V. CONCLUSIONS

We have proposed a novel Adaptive Resonance Theory-based neural architecture known as ART-C (ART under Constraints) for online clustering of arbitrary input sequences under constraints. Our studies and comparative experiments on the Iris data set and the Reuters-21578 corpus suggest that:

- ART-C is capable of effectively satisfying user defined constraints on the category representation, in terms of the number of output clusters.
- ART-C has retained the key properties of ART. They include:
 1. online learning efficiency,
 2. reasonably good clustering quality, and
 3. sensitivity to the order of input representations and possible over-fitting on noises and outliers.

REFERENCES

- [1] G.A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine." *Computer Vision, Graphics, and Image processing*, vol. 34, pp. 54-115, 1987.
- [2] A.-H. Tan, "Adaptive Resonance Associative Map," *Neural Networks*, vol. 8, no. 3, pp. 437-446, 1995.
- [3] G. Bartfai and R. White, "Incremental learning and optimization of hierarchical clusterings with ART-based modular networks," in *Innovations in ART Neural Networks*, L.C. Jain, B. Lazzarini, and U. Halici, Eds., pp. 87-132. Physica-Verlag, 2000.
- [4] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, pp. 786-801, 1999.
- [5] M. Georgiopoulos, I. Dagher, G. Heilman, and G. Bebis, "Properties of learning of a fuzzy ART variant," *Neural Networks*, vol. 12, no. 6, pp. 837-850, 1999.
- [6] C.L. Blake and C.J. Merz, "UCI repository of machine learning databases," 1998.
- [7] M. Halkidi, M. Vazirgiannis, and I. Batistakis, "Quality scheme assessment in the clustering process," in *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2000.
- [8] T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, Berlin, second edition, 1997.
- [9] A. Flexer, "On the use of self-organizing maps for clustering and visualization," in *Principles of Data Mining and Knowledge Discovery*, 1999, pp. 80-88.
- [10] J.T. Tou and R.C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, Massachusetts, 1974.
- [11] Y. Yang and J.P. Pedersen, "A comparative study on feature selection in text categorization," in *the Fourteenth International Conference on Machine Learning (ICML'97)*, 1997, pp. 412-420.
- [12] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999, pp. 42-49.
- [13] E. Gokcay and J.C. Principe, "A new clustering evaluation function using Renyi's information potential," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2000.