Adaptive Scaling of ClusterBoundaries for Large-scale Social Media Data Clustering

Lei Meng, Ah-Hwee Tan, Senior Member, IEEE, and Donald C. Wunsch II, Fellow, IEEE

Abstract—The large-scale and complex nature of social media data raises the need to scale clustering techniques to big data and make them capable of automatically identifying data clusters with few empirical settings. In this paper, we present our investigation and three algorithms based on the Fuzzy Adaptive Resonance Theory (Fuzzy ART) that have linear computational complexity, use a single parameter, i.e. the vigilance parameter to identify data clusters, and are robust to modest parameter settings. The contribution of this paper lies in two aspects. First, we theoretically demonstrate how complement coding, commonly known as a normalization method, changes the clustering mechanism of Fuzzy ART, and discover the vigilance region (VR) that essentially determines how a cluster in the Fuzzy ART system recognizes similar patterns in the feature space. The VR gives an intrinsic interpretation of the clustering mechanism and limitations of Fuzzy ART. Second, we introduce the idea of allowing different clusters in the Fuzzy ART system to have different vigilance levels in order to meet the diverse nature of the pattern distribution of social media data. To this end, we propose three vigilance adaptation methods, namely, the activation maximization rule (AMR), the confliction minimization rule (CMR), and the hybrid integration rule (HIR). With an initial vigilance value, the resulting clustering algorithms, namely, the AM-ART, CM-ART, and HI-ART, can automatically adapt the vigilance value of all clusters during the learning epochs in order to produce better cluster boundaries. Experiments on four social media data sets show that AM-ART, CM-ART, and HI-ART are more robust than Fuzzy ART to the initial vigilance value, and they usually achieve better or comparable performance and much faster speed than state-of-the-art clustering algorithms that also do not require a predefined number of clusters.

Index Terms—Clustering, Big social media data, Adaptive Resonance Theory, Vigilance region, Adaptive parameter tuning.

I. INTRODUCTION

The popularity of social websites has resulted in a dramatic increase in online multimedia documents, such as images, blogs, and tweets. Recently, clustering web multimedia data obtained from social websites has drawn much attention for social community discovery [12], [45], collective behavior analysis [13], and underlying topic discovery [3], [5]. However, in contrast to traditional image and text data sets, social media data sets are usually large-scale and may cover diverse content across different topics, making it difficult to manually evaluate the number of underlying topics and the pattern distributions of the data sets. These challenging issues raise the need for existing clustering approaches to be scalable to big data and capable of automatically identifying data clusters with few empirical settings, such as the number of clusters.

Unfortunately, most of the widely-used clustering approaches, such as K-means, spectral, probabilistic and matrix factorization approaches, require setting the number of clusters in a data set, which is still an open problem nowadays. To address this problem, numerous methods have been proposed, which can be categorized under the cluster tendency analysis approach [15], [16], [17] and the cluster validation approach [18], [19], [20], [21], [22], [23]. The first approach aims to identify the number of clusters in a data set by the neighboring relation of patterns, while the second one achieves this end by evaluating the quality of different cluster structures. However, both of these methods are typically slow and may not scale big social media data. As an alternative solution, clustering approaches are available that do not require a predefined number of clusters, including the hierarchical approach [26], [27], [28], the genetic approach [29], [30], the density-based approach [24], [31], affinity propagation (AP) [25], the approach by finding density peaks (Clusterdp) [49], and the adaptive resonance theory (ART) [9], [1], [2], [39], [40], [41]. However, the hierarchical and genetic approaches are similar to the cluster validation approach. The other algorithms typically have a quadratic time complexity of $O(n^2)$ and require one or more parameters to form clusters, which make their performance sensitive to the settings of these parameters.

In order to meet the requirements of clustering large-scale social media data, in this paper, we present our investigation and three algorithms based on the fuzzy adaptive resonance theory (Fuzzy ART) that have linear computational complexity, use a single parameter, i.e. the vigilance parameter to identify data clusters, and are robust to modest parameter settings. To achieve this goal, we first conduct a theoretical analysis in Fuzzy ART, in which we demonstrate that complement coding [2] significantly changes the clustering mechanism of Fuzzy ART and, with complement coding, the vigilance parameter of a cluster in Fuzzy ART forms a hyper-octagon region for the cluster, called a vigilance region (VR), in the high-dimensional feature space. The VR essentially determines how a cluster in the Fuzzy ART system recognizes similar patterns in the feature space and gives an intrinsic interpretation of the clustering mechanism and limitations of Fuzzy ART. Subsequently, with the deep understanding of the clustering mechanism of Fuzzy ART, we introduce the idea of allowing different clusters in the Fuzzy ART system to have different vigilance levels in order to meet the diverse nature of the pattern distribution of social media data. Three

Lei Meng and Ah-Hwee Tan are with the School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore, 639798. Email: [meng0027, asahtan]@ntu.edu.sg. Donald C. Wunsch II is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, Mo65409, USA. Email: dwunsch@mst.edu.
vigor adaptation rules, namely, the activation maximization rule (AMR), the confliction minimization rule (CMR), and the hybrid integration rule (HIR) are further proposed to make the vigilance values of all clusters in the Fuzzy ART system self-adaptable during the learning epochs. In this way, with an initial vigilance value, the clusters in the resulting algorithms, namely, the AM-ART, CM-ART, and HI-ART, are able to adaptively tune their boundaries to recognize similar patterns, and the performance of the proposed algorithms will be more robust than Fuzzy ART to the initial vigilance value.

AM-ART, CM-ART, and HI-ART were applied to four social media data sets, including a subset of the NUS-WIDE data set [7], a subset of the 20 Newsgroups data set [48], the Corel5k data set [50], and a subset of the BlogCatalog data set [51]. Their performance was studied in terms of their robustness to the initial vigilance parameter, the convergence speed, the noise immunity, the time-cost analysis, and the clustering performance comparison with Fuzzy ART [2], DBSCAN [31], Affinity Propagation [25], and Cluster_{dp} [49], in terms of purity [44], class entropy [46], and the Rand index [47]. Experimental results show that the proposed algorithms are more robust than Fuzzy ART to the vigilance parameter and usually perform better than the algorithms in comparison.

The remainder of this paper is organized as follows. Section II reviews previous studies on the automatic identification of the number of clusters in a data set. Section III summarizes the Fuzzy ART algorithm. Section IV presents our study on complement coding and the VR. Section V presents our proposed vigilance adaptation methods, including AMR, CMR, and HIR. The experiments are reported in Section VI. The last section concludes our work.

II. RELATED WORK

A. Cluster Tendency Analysis
Cluster tendency analysis aims to identify the number of clusters in a data set before clustering. Most recent studies [15], [16], [17] have focused on investigating the dissimilarity matrix of patterns. Visual assessment of tendency (VAT) [15] reorders the dissimilarity matrix of patterns to form a reordered dissimilarity image (RDI), and the number of clusters is identified by counting the dark blocks along the diagonal pixels. Cluster count extraction (CCE) [16] and dark block extraction (DBE) [17] are further proposed to objectively identify the number of clusters instead of manual counting. CCE constructs a histogram using the off-diagonal pixel values of the filtered RDI obtained by VAT, and the number of clusters equals the number of spikes in the histogram. In contrast, DBE employs matrix transformation methods to project all of the pixel values of the obtained RDI to the main diagonal axis in order to obtain a projection signal, and the number of clusters equals the number of major peaks in the signal.

B. Clustering Validation
Cluster validation aims to find the best clustering by evaluating the quality of the different cluster structures generated. Considering the difference in the mechanism for cluster assignment, existing cluster validation indices can be categorized into those for hard clustering [52], where one pattern belongs to one cluster, and those for fuzzy clustering [18], where one pattern has the levels of fuzzy membership for all clusters. Regarding the hard clustering, existing methods typically follow three directions. First, validation indices, usually based on the intracluster compactness and the between-cluster separation, are employed to evaluate the quality of different clusterings generated by running a base algorithm with different numbers of clusters [19], [22], [53]. Second, the values of validation indices for different clusterings are plotted as a function of the number of clusters, and the best number of clusters located at the extreme or “elbow” values [20], [54], [55]. Third, multiple data sets are first produced by distorting the patterns in the given data set, using distortion tools like subsampling and adding random noise. Subsequently, a clustering algorithm is performed on each data set to identify the best number of clusters [23], [56], [57]. Existing methods for the fuzzy cluster validation [18], [21], [58] typically employ Fuzzy C-means as the base algorithm and evaluate the quality of the produced cluster assignments to identify the best number of clusters. A review of fuzzy cluster validation indices is available in [18].

C. Cluster Property Modeling

1) Hierarchical Clustering: Hierarchical clustering algorithms typically incorporate a cluster validity index to measure the cluster quality during each merging or splitting iteration. Li et al. [26] proposed an agglomerative fuzzy k-means algorithm that runs multiple times with a maximum number of clusters and a gradually increased penalty parameter. During these runs, the clusters that share centers are merged according to a validation method. Leung et al. [27] proposed a scale-based algorithm which considers a data set as an image, and each pattern is considered a light point on the image. The generation of a hierarchy is then simulated by blurring the image such that the light points gradually merge together. Several cluster validity indices, including lifetime, compactness, isolation and outlierness, are used to select the best cluster structure in the hierarchy. In [28], an agglomerative clustering algorithm was proposed based on an intrachuster dissimilarity measure, and a merge dissimilarity index (MDI) is presented to find the optimal number of clusters.

2) Genetic Clustering: The use of genetic algorithms [29], [30] to identify the best clustering typically depends on the evolution of cluster structures, during which the cluster quality is evaluated by certain cluster validity indices. In the symmetry-based genetic clustering algorithm (VGAPS-clustering) [29], each pattern in the population pool is a concatenation of the weight values of cluster centers, and different patterns may each have a different number of centers. After the maximum number of iterations, the pattern with the highest fitness is selected as the best cluster structure. A review of genetic clustering algorithms is provided in [30].

3) Density-based Clustering: The density-based clustering algorithms identify dense regions of patterns as clusters in the feature space. DBSCAN [31] forms the degree of density using two parameters, namely, the maximum distance for the search of neighbors and the minimum number of neighbors. Patterns
and their neighbors that satisfy the above requirements are deemed to be in the same cluster while others are considered noises. There are different variants of DBSCAN with different cluster density measures, such as GDBSCAN [32], OPTICS [33], DECODE [34], and KNNCLUST [35]. A review of density-based clustering algorithms can be found in [24].

4) **Affinity Propagation:** Affinity propagation (AP) [25] is an exemplar-based clustering algorithm that identifies a set of representative patterns as “exemplars” to the other patterns in the same cluster. Exemplars are identified by recursively updating two messages of patterns, namely, the “availability” indicating the qualification of a pattern to be an exemplar, and the “responsibility” indicating the suitability of a pattern to be a member of the exemplars’ clusters. Two algorithms [36], [37] have been proposed to improve the efficiency of affinity propagation. Fast sparse affinity propagation (FSAP) [36] generates a sparse graph using the K-nearest neighbor method, rather than the original similarity matrix, in order to reduce the computational cost. In [37], the computational cost is saved by pruning the edges that can be directly calculated after the convergence of affinity propagation.

5) **Clustering by Finding Density Peaks:** Cluster \(d_{dp} [49]\) identifies clusters of patterns by measuring the distances between patterns in order to find the density peaks. With a predefined value of search radius, the density peaks are evaluated by following two criteria: 1) Density peaks should have more neighbors than those of their neighbors; and 2) all density peaks should be far away from each other. Cluster \(d_{dp}\) plots a decision graph for the user to identify the density peaks as cluster centers, and the remaining patterns are assigned to the nearest cluster centers.

6) **Adaptive Resonance Theory:** Adaptive resonance theory (ART) [9] is a learning theory that simulates how a human brain captures, recognizes and memorizes information about objects and events. It has resulted in the development of a series of unsupervised learning models, such as ART 1 [1], ART 2 [39], ART 2-A [40], ART 3 [41], and Fuzzy ART [2], as well as supervised learning models, such as ARTMAP [6] and Fuzzy ARTMAP [42].

Fuzzy ART, as the base model studied in this paper, incrementally processes input patterns one at a time by performing real-time searching and matching of existing clusters (memory prototypes) in the category space. The vigilance parameter is used to restrain the minimum degree of similarity for patterns in the same cluster. When none of existing clusters is deemed similar to the input pattern, a new cluster is generated to encode this novel pattern. Fuzzy ART has been used in different variants to resolve many image and text mining problems, such as web document management [3], tag-based web image organization [5], image-text association [4], and heterogeneous data co-clustering [14].

Existing studies on the adaptation of or doing without the vigilance parameter in ART-based algorithms do not allow different clusters to have different vigilance levels and typically require additional information, such as the number of clusters [10], [11] and class labels [6], [43]. Therefore, adapting the vigilance parameter in ART under clustering scenario without any additional information remains a challenge.

III. **Fuzzy ART**

The architecture of Fuzzy ART (Fig. 1) consists of the input field \(F_1\) for receiving input patterns and the category field \(F_2\) for clusters. The generic network dynamics of Fuzzy ART are described as follows.

**Input vectors:** Let \(I = x\) denotes the input pattern in the input field \(F_1\). Note that for \(x = (x_1, ..., x_m), x_i \in [0, 1] (i = 1, ..., m)\). With complement coding [2], \(x\) is further concatenated with its complement vector \(\bar{x}\) such that \(I = (x, \bar{x})\), where \(\bar{x} = 1 - x\).

**Weight vectors:** Let \(w_j\) denotes the weight vector associated with the \(j\)th cluster \(c_j (j = 1, ..., J)\) in the category field \(F_2\).

**Parameters:** The Fuzzy ART dynamics are determined by choice parameter \(\alpha > 0\), learning parameter \(\beta \in [0, 1]\), and vigilance parameter \(\rho \in [0, 1]\).

The clustering process of Fuzzy ART has three key steps:

1) **Category choice:** For each input pattern \(I\), Fuzzy ART calculates a choice value according to the choice function for each of the clusters in the category field \(F_2\) and identifies the cluster having the largest value, i.e. the winner \(c_j^*\). The choice function for the \(j\)th cluster \(c_j\) is defined by

\[
T_j = \frac{|I \land w_j|}{\alpha + |w_j|},
\]

where the fuzzy AND operation \(\land\) is defined by \((p \land q)_i \equiv \min(p_i, q_i)\), and the norm \(|.|\) is defined by \(|p| \equiv \sum p_i\).

2) **Template matching:** The similarity between the input pattern \(I\) and the winner \(c_j^*\) is evaluated using a match function \(M_{j^*}\), which is defined by

\[
M_{j^*} = \frac{|I \land w_{j^*}|}{|I|}.
\]

If the winner satisfies the vigilance criteria such that \(M_{j^*} \geq \rho\), a resonance will occur, which leads to the learning step. Otherwise, a new winner will be selected among the rest of the clusters in the category field. If no winner satisfies the vigilance criteria, a new cluster will be generated to encode the input pattern.

3) **Prototype learning:** If \(c_j^*\) satisfies the vigilance criteria, its corresponding weight vector \(w_{j^*}\) will be updated through a learning function, defined by

\[
w_{j^*}^{(new)} = \beta(I \land w_{j^*}) + (1 - \beta)w_{j^*}.
\]
IV. COMPLEMENT CODING AND VIGILANCE REGION IN FUZZY ART

The vigilance region (VR) of a cluster in Fuzzy ART, which is calculated based on the vigilance criteria, is geometrically defined by a region associated to the cluster in the feature space. It geometrically interprets the vigilance criteria of Fuzzy ART where the input patterns falling into VRs are considered to be similar to the corresponding clusters.

The shapes and functional behaviors of a VR depend on the use of complement coding. With complement coding, an input pattern $x$ is represented as $I = (x, \bar{x})$. Therefore, the weight vector $w$ of a cluster also includes the feature part and the complement part. As demonstrated in [2], given a weight vector $w = (a, b)$, the match area of the weight vector $w$ corresponds to a rectangle in the 2D space, where $a$ and $b$ are the two diagonal corner points of the rectangle. As shown in Fig. 2, VR1 and VR2, are the VRs of the weight vector $w$ generated under vigilance parameters $\rho = 0.75$ and $0.825$ respectively. Without complement coding, weight vector $w$ is the point $a$, and VR3 and VR4 are the VRs generated under the same vigilance parameters. Therefore, with complement coding, the VR is a hyper-octagon centered by the weight hyper-rectangle, which shrinks as the cluster size expands; otherwise, without complement coding, the VR is an irregular hyper-polygon with axes.

A. Complement Coding in Fuzzy ART

Complement coding [2] is employed in Fuzzy ART as a normalization method for the input patterns, which prevents cases in which the values of the weight vector of a cluster decrease to such a low level that the cluster is no longer representative for its category, and a set of new clusters must be generated to encode input patterns for this category; this is known as the problem of category proliferation. However, complement coding significantly changes the clustering mechanism of Fuzzy ART.

1) Effect of Complement Coding on Category Choice:

Choice function (1) evaluates the degree to which the weight vector $w_j$ of cluster $c_j$ is a subset of the input pattern $I$. We can prove that, by incorporating complement coding in Fuzzy ART, the choice function geometrically evaluates the similarity between the input pattern and the weight hyper-rectangle of the selected cluster in the feature space.

Property 1: Given the input pattern $I = (x, \bar{x})$, weight vector $w_j = (a, b)$ of cluster $c_j$, and $\alpha \approx 0$, choice function $T_j$ evaluates the similarity between the original input pattern $x$ and the weight hyper-rectangle of cluster $c_j$.

Proof 1:

$$T_j = \frac{|I \cap w_j|}{\alpha + |w_j|} = \frac{|x \cap a + \bar{x} \cap b|}{|w_j|} = \frac{|x \cap a| + |\bar{x} \cap b|}{|a + b|} = \frac{|a|}{|a + b|} \frac{|x \cap a|}{\alpha + |a|} + \frac{|b|}{|a + b|} \frac{\bar{x} \cap b}{\alpha + |b|}. \tag{4}$$

As shown in (4), the choice function evaluates both the degree to which $a$ is a subset of $x$ and to which $x$ is a subset of $b$. The final choice value is obtained by their weighted summation, which is normalized by their respective norms. Therefore, given $x = (x_1, ..., x_m)$, $a = (a_1, ..., a_m)$, and $b = (b_1, ..., b_m)$, choice function $T_j$ achieves its maximum for $c_j$ when, for all $i \in [1, m]$, $a_i \leq x_i \leq b_i$. For example, in Fig. 2, the choice function for this cluster achieves its maximum when the input pattern falls into the weight rectangle.

Therefore, with complement coding, the choice function evaluates the similarity of the input pattern to the weight hyper-rectangle of the selected cluster $c_j$.

2) Effect of Complement Coding on Template Matching:

Match function (2) evaluates the degree to which the input pattern $I$ is a subset of weight vector $w_{j*}$ of cluster $c_{j*}$. In template matching, the input pattern $I$ is considered to be similar to the winner cluster $c_{j*}$ if

$$M_{j*} = \frac{|I \cap w_{j*}|}{|I|} \geq \rho. \tag{5}$$

The VR, therefore, is identified to show the extent to which an input pattern can be categorized into a specific cluster. Given the weight vector $w_{j*} = (w_1, ..., w_m)$ of cluster $c_{j*}$, the vigilance parameter $\rho$, and an arbitrary input pattern $I = (x_1, ..., x_m)$ in the Fuzzy ART system, if Fuzzy ART does not employ complement coding, (5) is equivalent to

$$\sum_{i=1}^{m} \min(x_i, w_i) - \rho \sum_{i=1}^{m} x_i \geq 0. \tag{6}$$

As shown in Fig. 2, when $m = 2$, (6) is an irregular polygon constructed by three functions and the axes.

In contrast, if Fuzzy ART employs complement coding, the number of dimensions of the feature space will be $\frac{m}{2}$. Therefore, (5) can be expressed as

$$\sum_{i=1}^{m} \min(x_i, w_i) \geq \frac{m \rho}{2}. \tag{7}$$

When $m = 4$, as shown in Fig. 2, the VR of $c_{j*}$ becomes a regular polygon, namely, an octagon.
B. Vigilance Region in Fuzzy ART

1) Properties of Weight Hyper-rectangle and Vigilance Region: In Section IV-A, we prove that, with complement coding, the VR of a cluster becomes a hyper-octagon centered by the weight vector of the cluster, namely, the weight hyper-rectangle. In this section, we analyze the properties of the weight hyper-rectangle and the VR of a cluster, which are further used to interpret the clustering process of Fuzzy ART.

Property 2: Given the weight vector \( w_j = (w_{1j},...,w_{mj}) \) of cluster \( c_j \) in the Fuzzy ART system with complement coding, the VR of \( c_j \) consists of \( 3^m - 1 \) hyper-planes.

**Proof 2:** Similar to (4), given \( w_j = (a_1,\ldots,a_m,\bar{b}_1,\ldots,\bar{b}_m) \), and \( I = (x_1,\ldots,x_m,\bar{x}_1,\ldots,\bar{x}_m) \), (5) can be expressed as

\[
\Sigma_{i=1}^m \min(x_i,a_i) + \Sigma_{i=1}^m \max(x_i,b_i) \geq \frac{m\rho}{2}.
\]

We observe that the update of weight vector \( w_j \) is essentially the movement of \( a \) and \( b \) towards the input pattern \( I \). Specifically, \( a \) moves towards \( I \) in the dimensions \( \{i|x_i < a_i\} \), while \( b \) moves towards \( I \) in the dimensions \( \{i|x_i > b_i\} \). Therefore, when \( \beta = 1 \), the weight hyper-rectangle of \( c_j \) covers all of the patterns in \( c_j \), which indicates the boundaries of \( c_j \). When \( \beta < 1 \), the weight hyper-rectangle expands towards the new patterns to some extent, making it unable to cover all the patterns. However, the weight hyper-rectangle may reflect the cluster size on a smaller scale.

Property 6: The VR shrinks as the weight hyper-rectangle expands to control the minimum intracluster similarity.

**Proof 6:** As demonstrated in Property 2, a VR in the \( \frac{m}{2} \) dimensional space can be constructed by \( 3^m - 1 \) functions, each of which is calculated using (8). Because the learning function (3) suppresses the values of features, we have \( a_i^{\text{new}} \leq a_i \) and \( b_i^{\text{new}} \geq b_i \) for all \( i \). Therefore, after the weight hyper-rectangle expands, the constant in the left part of (8) decreases so that functions in the subsections either remain the same or move towards the weight hyper-rectangle.

The shrinking of the VR can also be understood from another perspective. As the VR indicates, the boundaries that the weight hyper-rectangle expands towards go in all directions. When the weight hyper-rectangle expands in one direction, its distance from the VR is determined by the function in that direction, and the functions in other directions should shrink to meet this distance so that the updated VR remains a regular hyper-octagon centered by the weight hyper-rectangle.

2) Interpretation of Fuzzy ART using Vigilance Region: Given the properties of the weight hyper-rectangle and the VR, we interpret the clustering process of Fuzzy ART by a 2D example. Fig. 3a depicts the evolution of a cluster in Fuzzy ART under learning parameter \( \beta = 1 \). When the cluster has only one pattern \( I_1 \), the weight rectangle \( R_1 \) is situated exactly at point \( I_1 \). In this case, the corresponding VR1 is a square diamond centered by \( I_1 \). After encoding \( I_2 \), \( R_2 \) becomes a rectangle, and the corresponding VR2 becomes an octagon, which satisfies Property 2. During the presentation of \( I_3 \) and \( I_4 \), the weight rectangle expands to cover all of the patterns, which satisfies Property 5. It is notable that \( I_4 \) lies directly on the edge of VR3 so that VR4 overlaps with R4. Based on Property 3 and Property 4, patterns falling into R4 have the same match function value, and this cluster will no longer expand. Also, the bottom-left edge of VR2-VR4, where \( I_4 \) lies, never shrinks, which can be interpreted by Property 6.

Fig. 3b illustrates that, with \( \beta = 0.6 < 1 \), R1 expands towards I2 but cannot cover both I1 and I2. It is notable that a repeated presentation I5 of I4 still causes the cluster to learn. Therefore, when \( \beta < 1 \), the continuous presentation of the same pattern to a cluster results in the gradual expansion of the weight rectangle of the cluster towards the input pattern; however, the rectangle cannot cover that pattern.

3) Discussion: The VR provides a geometric understanding of how Fuzzy ART works. As shown in Fig. 2, without complement coding, the VR of Fuzzy ART in a 2D space is an open region, so the weight vector of the cluster denoted by point \( a \) may gradually move to the origin, which causes the problem of category proliferation. With complement coding,
the VR of a cluster in Fuzzy ART is a regular polygon, which shrinks as the cluster size expands. Therefore, Fuzzy ART with complement coding tends to partition the high-dimensional feature space into regions of hyper-rectangles.

The geometric interpretation of Fuzzy ART is also helpful for deducing and further improving the limitations of Fuzzy ART. First, given that the VR of a new cluster is usually much larger than the weight rectangle of the cluster and shrinks quickly after the encoding of the subsequent patterns, it may be difficult to cover a group of patterns using a single cluster, even if the first VR covers all patterns. Second, a small VR may result in the generation of multiple clusters to cover a group of patterns. Third, a large VR may incur an incorrect categorization of patterns, since the sequence of input patterns is unknown. Therefore, the performance of Fuzzy ART depends greatly on the value of vigilance parameter $\rho$, and with different sequences of input patterns, the clustering results may differ.

V. RULES FOR ADAPTING VIGILANCE PARAMETER IN FUZZY ART

As discussed in IV-B3, the performance of Fuzzy ART depends greatly on the value of vigilance parameter $\rho$, which both determines the VRs of clusters for recognizing similar patterns and limits the size of all clusters. However, because large-scale web multimedia data usually contain a large number of groups of patterns in arbitrary shapes in the feature space, it is not advisable to use a single value of the vigilance parameter in Fuzzy ART to scale the size of all clusters. Based on the above consideration, in our preliminary study [38], we proposed two heuristic methods, i.e., the activation maximization rule (AMR) and the confliction minimization rule (CMR). They allow different clusters in ART to have individual vigilance parameters, which are self-adapted during the clustering process. In the following sections, we offer a geometric interpretation of AMR and CMR using the VR and further propose a hybrid integration rule (HIR), which integrates both AMR and CMR.

A. Activation Maximization Rule

The activation maximization rule (AMR) comes from the observation that input patterns are likely to incur resonances for the same cluster with a small vigilance value, while a large vigilance value may lead to the reset of input patterns for all clusters in the category field, requiring the creation of a new cluster. Therefore, AMR is proposed to restrain the continuous activation of the same cluster and promote the activation of clusters that usually incur resets.

Fuzzy ART with AMR (AM-ART) adapts the vigilance parameter $\rho_{j^*}$ of the winner $c_j^*$ when

1) Resonance occurs: $\rho_{j^*}^{(new)} = (1 + \sigma)\rho_{j^*}$;

2) Reset occurs: $\rho_{j^*}^{(new)} = (1 - \sigma)\rho_{j^*}$.

The restraint parameter $\sigma \in [0, 1]$ controls the degree to which the vigilance parameter increases or decreases. With a small $\sigma$, AMR incurs small changes in the vigilance values of clusters, so the performance of ART still highly depends on the initial value of the vigilance parameter. In contrast, a large $\sigma$ may help to make AM-ART to be robust to the initial vigilance value but may result in unstable vigilance values of clusters.

Fig. 4 illustrates how AMR works. C1 and C2 are two clusters with different values of vigilance parameter. When the input pattern I is presented, the first winner, C2, incurs a reset due to its small VR, and the next winner, C1, encodes I. Without AMR, the VR of C1 shrinks from VR1 to VR2, and that of C2 remains the same. In this situation, a new input pattern located close to I will be misclassified to C1 again. In contrast, with AMR, the VR of C1 shrinks to VR3 while that of C2 expands to VR2. Therefore, C2 is likely to encode the following new input pattern located close to I than C1.

Based on the above discussion, AMR may help to improve the clustering performance of Fuzzy ART when the initial vigilance value is not suitable. Besides, AMR may also help...
B. Confliction Minimization Rule

The confliction minimization rule (CMR) minimizes the overlap between VRs of close clusters to produce better cluster boundaries. CMR is based on the idea that, in Fuzzy ART, the incorrect recognition of patterns usually is caused by a small vigilance value, so the VR of a cluster may cover patterns from other classes. Therefore, well-partitioned boundaries between clusters can minimize the risk of miscategorization.

Fuzzy ART with CMR (CM-ART) has three key steps:

1) Candidate Selection: Select all winner candidates \( Winc = \{ c_j | M_j \geq \rho \} \) in the category field \( F_2 \) through the match function (2). If no candidates are selected, CMR stops;

2) Winner Identification: Identify the winner \( c_j^{*} \) from all candidates using the choice function (1) such that \( j^{*} = \arg \max_j T_j \);

3) Confliction Minimization: Update the vigilance parameters of all winner candidates except for the winner \( \{ c_j | c_j \in Winc \land j \neq j^{*} \} \) using \( \rho_{j}^{(new)} = M_j + \Delta \) (\( \Delta \approx 0 \) is a very small positive value).

Fig. 5 illustrates how CMR reduces the overlap between the VRs of clusters. C1-C3 are three clusters and I is an input pattern falling at the overlap between VRs of all clusters. C2 encodes the input pattern I and its VR shrinks to VR2. Without CMR, the VRs of C1 and C3 remains the same, and the overlap between all three clusters has no decrease. While with CMR, the VRs of C1 and C3 shrinks from VR1 to VR2, and the overlap undergoes significant reduction.

C. Hybrid Integration of AMR and CMR

AMR and CMR are inspired by different considerations for ART and have different mechanisms when embedding in ART, so we cannot simply combine them into a single framework. However, we may simultaneously integrate the objectives of AMR and CMR. Specifically, AMR essentially rewards the clusters that have larger choice values than the winner but incur resets due to large vigilance values, while penalizing the clusters that incur resonances to avoid a potentially low vigilance value. In contrast, CMR minimizes the overlap between the VRs of clusters.

The implementation of the hybrid method, called the hybrid integration rule (HIR), may follow the procedures of either AMR or CMR. For the purpose of computational efficiency, Fuzzy ART with HIR (HI-ART) is implemented according to the procedures of CM-ART, as listed below:

1) Candidate Selection: Select all winner candidates \( Winc = \{ c_j | M_j \geq \rho \} \) in category field \( F_2 \) through match function (2). If no candidates are selected, for \( \forall c_j \in F_2 \), set \( \rho_{j}^{(new)} = (1 - \sigma)\rho_{j} \), and HIR stops;

2) Winner Identification: Identify the winner \( c_j^{*} \) from all candidates through choice function (1) such that \( j^{*} = \arg \max_j T_j \). Set \( \rho_{j}^{(new)} = (1 + \sigma)\rho_{j}^{*} \);

3) Confliction Minimization: Update the vigilance parameters of all winner candidates except for the winner \( \{ c_j | c_j \in Winc \land j \neq j^{*} \} \) using \( \rho_{j}^{(new)} = M_j + \Delta \);

4) Activation Maximization: Search in the remaining clusters to identify the set of clusters \( Rc = \{ c_j | c_j \in F_2 \land c_j \notin Winc \land T_j \geq T_j^{*} \} \), and for \( \forall c_j \in Rc \), set \( \rho_{j}^{(new)} = (1 - \sigma)\rho_{j} \).

D. Computational Complexity Analysis

As presented in Section III, given an input pattern, Fuzzy ART undergoes procedures including complement coding, finding matching cluster by category choice and template matching, and prototype learning or new cluster creation. Their corresponding time complexities are \( O(n_{c}) \), \( O(n_{c}n_{f}) \), and \( O(n_{f}) \), where \( n_{c} \) denotes the number of clusters and \( n_{f} \) denotes the number of features. Therefore, given \( n_{c} \) input patterns, the overall time complexity of Fuzzy ART is \( O(n_{c}n_{c}n_{f}) \).

AM-ART requires to adapt the vigilance values of the selected winners, incurring a time complexity of \( O(n_{c}) \); CM-ART reverses procedures of category choice and template matching and adapts the vigilance parameter values of all winner candidates, of which the time complexity is \( O(n_{c}) \); HI-ART makes an integration of procedures of AM-ART and CM-ART. Therefore, our proposed algorithms have the same time complexity to Fuzzy ART, i.e. \( O(n_{c}n_{c}n_{f}) \).
VI. EXPERIMENTS

A. DATA SETS

We conducted our experiments on four data sets:

1) **NUS-WIDE** data set [7] consists of 269,648 Flickr images with their raw surrounding text and ground-truth labels from 81 concepts. We collected 10,800 images in total from nine classes, including dog, bear, cat, bird, flower, lake, sky, sunset, and wedding, each of which contains 1,200 images. Each image was represented as a 426-D vector by concatenating three types of visual features, including Grid Color Moment (225 features), Edge Direction Histogram (73 features) and Wavelet Texture (128 features).

2) **20 Newsgroups** data set [48] consists of approximately 20,000 messages from 20 different newsgroups, each of which contains nearly 1,000 documents. We collected 9,357 documents from 10 classes, including alt.atheism, comp.graphics, comp.windows.x, rec.sport.baseball, rec.sport.hockey, sci.med, sci.space, misc.forsale, talk.politics.guns, and talk.politics.misc, from the processed MATLAB version of the 20news-bydate data set. Regarding the feature extraction, we filtered any words that occurred less than 30 times, and each document was represented by a bag-of-words vector of 6,823 features, weighted by the term frequency-inverse document frequency (tf-idf) algorithm.

3) **Corel5K** data set [59] consists of 4999 images from 50 equal-sized classes. We utilized the whole data set for experiments, and extracted the 426 visual features for image representation, as used for the NUS-WIDE data set.

4) **BlogCatalog** data set [60] consists of the friendship network and the raw blog data (blog content, category, and tags) of 88,784 social network users. We used a polished version of the data set as processed in [45]. Specifically, we collected the blog content of 10,000 users from 10 equal-sized classes, including travel, music, writing, sports, shopping, computers, finance, film, fashion, and books. By filtering the infrequent words, each user is represented by a 5685-D vector, of which the features are weighted by the tf-idf algorithm.

B. PARAMETER SELECTION

Similar to Fuzzy ART, the proposed AM-ART, CM-ART, and HI-ART have parameters $\alpha$, $\beta$, and $\rho$. Besides, AM-ART has the restraint parameter $\sigma$, CM-ART has the parameter $\Delta$, and HI-ART has both. In the experiments, we consistently used $\alpha = 0.01$, $\beta = 0.6$, $\sigma = 0.1$, and $\Delta = 0.01$, as the proposed algorithms and Fuzzy ART has general robustness to the parameters with the above values [8], [38], [45].

The vigilance parameter $\rho$ is essentially a ratio value that controls the minimum intracluster similarity of patterns, so its value is data-dependent. However, an empirical method [45] has shown that a suitable value of $\rho$ typically results in the generation of a few small clusters with tens of patterns, typically 10% of the total number of the generated clusters. We followed this method to select the initial value of $\rho$.

C. ROBUSTNESS TO VIGILANCE PARAMETER

In this section, we evaluated the performance of AM-ART, CM-ART, and HI-ART on improving the robustness of Fuzzy ART to the vigilance parameter $\rho$. The performance was measured in terms of purity [44] and the number of clusters generated. Here, purity measures how well the algorithm recognizes the data objects of the same class, and the number of clusters measures how well the algorithm partitions the data set with the lowest network complexity. We reported the performance on the NUS-WIDE and 20 Newsgroups data sets in Fig. 6, and similar observations were found in the experiments on the Corel5K and BlogCatalog data sets.

As shown in Fig. 6a, when $\rho < 0.4$, AM-ART, CM-ART, and HI-ART achieved much better performance in purity and identified more clusters than Fuzzy ART. When $\rho > 0.7$, all algorithms achieved comparable performance; however, as shown in Fig. 6b, the higher purity was achieved by increasing network complexity. Meanwhile, we observed that AM-ART and HI-ART generated significantly fewer clusters than Fuzzy ART and CM-ART. These findings indicated that AMR, CMR, and HIR enabled the proposed algorithms to be more robust than Fuzzy ART to the vigilance parameter, especially when the initial vigilance value is low. AMR can effectively simplify the generated cluster network when the initial vigilance value is high. More importantly, HI-ART has advantage of both AM-ART and CM-ART, which demonstrates the viability of developing hybrid methods for vigilance adaptation. Similar findings can be observed in Fig. 6c and Fig. 6d.

A case study was further conducted to provide a deeper understanding of how the proposed algorithms work by analyzing the clusters generated by each algorithm. As shown in
Fig. 7: Distributions of clusters generated by AM-ART, CM-ART, HI-ART, and Fuzzy ART on the NUS-WIDE data set in terms of cluster size and average pattern-centroid distance under $\rho = 0.2$ (a & b) and $\rho = 0.9$ (c & d).

Fig. 8: Convergence analysis of AM-ART, CM-ART, HI-ART, and Fuzzy ART measured by the change in weights and the number of patterns moving across clusters in each iteration on the NUS-WIDE (a & b) and 20 Newsgroups (c & d) data sets.

Fig. 7a and 7b, under $\rho = 0.2$, AM-ART, CM-ART and HI-ART identified relatively more smaller clusters having better coherence than Fuzzy ART. These facts explain the lower performance of Fuzzy ART. In contrast, as illustrated in Fig. 7c and Fig. 7d, when $\rho = 0.9$, all algorithm generated clusters of similar quality while HI-ART and AM-ART generated much fewer small clusters than CM-ART and Fuzzy ART. This explains why they can generate fewer clusters than the other algorithms and demonstrates the effectiveness of AMR in simplifying the network with a high vigilance value.

D. Convergence Analysis

This section presents a study on the convergence property of the proposed algorithms and Fuzzy ART. We reported their performance on the NUS-WIDE and the 20 Newsgroups data sets under $\rho = 0.7$ and 0.6, respectively, and similar findings were observed on the other data sets. As shown in Fig. 8, all algorithms experienced large changes during the first six rounds. This circumstance likely is due to the generation of new clusters. CM-ART and HI-ART usually obtain comparable convergence speeds, which are faster than AM-ART and Fuzzy ART. This is because CMR promotes the shrinking of the VRs of neighboring clusters by reducing their overlap, resulting in the fast stabilization of cluster assignments. AM-ART usually converges slower than Fuzzy ART, because AMR increases the vigilance value of competitive winner candidates and decreases that of the winner so that patterns may jump across those winner candidates when they are presented multiple times. HI-ART converged faster than Fuzzy ART during the first rounds of iterations due to CMR but achieved a convergence speed similar to that of Fuzzy ART after the network becomes stable due to AMR. Interestingly, in contrast to its performance on the NUS-WIDE data set, AM-ART converged faster than Fuzzy ART on the 20 Newsgroups data set. This may be due to the larger dispersion of patterns in the feature space, which caused the increase in the size of the VRs to have less of an effect.

E. Clustering Performance Comparison

In this section, we compared the clustering performance of AM-ART, CM-ART, and HI-ART to existing clustering approaches that also automatically identify the number of clusters in data, including DBSCAN, affinity propagation, Cluster$_{dp}$, and fuzzy ART. All algorithms were implemented in MATLAB. Hierarchical and genetic clustering approaches are not considered here because they require heavy computation and are not scalable to large-scale data sets.

We applied min-max normalization to the data sets because the ART-based algorithms required the input values to be in the range of [0, 1]. Experimental results indicated that the normalization of data had an unobvious effect on the performance of other algorithms. To ensure a fair comparison, we utilized practical parameter tuning strategies for the algorithms. For DBSCAN, we determined the minimum cluster size $minPts$ by evaluating the sizes of small clusters generated by Fuzzy ART under high vigilance values $\rho \in [0.7, 0.9]$. Subsequently, we followed the method suggested in [31] to select the search radius $\varepsilon$, namely, plotting the $k$-distance graph ($k$ is the value of $minPts$) and choosing the “bend” value. For Affinity propagation, we selected the preference value $\rho$ using the MATLAB function “preferenceRange.m”\(^2\). The values of

\(^2\)http://genes.toronto.edu/index.php?q=affinity%20propagation
**Fig. 9:** Clustering performance comparison of the proposed algorithms and four baseline algorithms under different parameter settings measured by the Rand index on the (a) NUS-WIDE, (b) 20 Newsgroups, (c) Corel5K, and (d) BlogCatalog data sets.

*dampfact, convits* and *maxits* were first set to the suggested values by the authors and then changed with respect to the preference value *p* to ensure convergence. For Cluster*dp*, we set the search radius *d* to the value of ε used in DBSCAN as both have the same meaning and selected the cluster centers from the decision graph that produced the best performance.

We used three external clustering performance measures, including purity [44], class entropy [46] and the Rand index [47]. Purity evaluates the precision aspect, i.e., how well an algorithm recognizes patterns belonging to the same class, and a higher value indicates better performance. Class entropy evaluates the recall aspect, i.e., how well an algorithm partitions the data set with the minimum number of clusters, and a lower value indicates better performance. The Rand index considers both aspects. Internal performance measures, such as sum of squared error (SSE), were not used because they make assumptions based on cluster shapes, so they are not suitable to evaluate the performance of DBSCAN and Cluster*dp*.

We first report the performance of each algorithm under different parameter settings in terms of the Rand index on all data sets, which provides an overall picture of the performance of each algorithm. Specifically, for the ART-based algorithms, we plotted the curve of performance as a function of the vigilance parameter *ρ*; for DBSCAN, we plotted the curve as a function of the minimum cluster size *minPts*; for Affinity Propagation, we plotted the curve as a function of the preference value *p*; for Cluster*dp*, we plotted the curve as a function of the search radius *d*.. The other parameters of each algorithm were fixed or tuned as aforementioned so that the best performance was achieved under each condition of the functions. Additionally, for DBSCAN and the ART-based algorithms whose results may be affected by the input data sequence, the performance was the mean value obtained by repeating the experiments ten times with different sequences of patterns, while that of Affinity Propagation and Cluster*dp* is obtained on a single run. The results were shown in Fig. 9. To facilitate the comparison, the x-axis values of each algorithm were normalized to be in the range of [0, 1]. We observed that the performance of the ART-based algorithms typically increased with respect to the increase in the vigilance value *ρ*, which indicated that better performance can be achieved by setting higher intracluster similarity threshold to some extent. However, a too high vigilance value would result in a deteriorated performance by the high network complexity, as shown in Fig. 9a and 9b.

Besides, HI-ART and CM-ART usually outperform AM-ART and Fuzzy ART when the vigilance value is low, consistent with our findings in Section VI-C. It was notable that Fuzzy ART achieved very low performance when *ρ* < 0.3, which was caused by the fact that all patterns in the Corel5K data set were clustered into a single cluster. In contrast, AM-ART achieved an improved performance on this case while CM-ART and HI-ART had a big improvement over Fuzzy ART. Compared with the ART-based algorithms, DBSCAN could achieve a stable performance when the values of *minPts* were near to the best setting. However, we observed that the best parameter value varied with different data sets and the best performance of DBSCAN was typically lower than these achieved by the ART-based algorithms. Affinity Propagation could perform comparably to the ART-based algorithms and achieved a more stable performance under different parameter settings, especially in Fig. 9a and 9d. However, the performance of Affinity Propagation could fluctuate a lot as shown in Fig. 9b and 9c, making it difficult to manually select the best parameter settings. Cluster*dp* typically performed the worst among all algorithms. Although a fairly stable performance was achieved in Fig. 9a, its best performance is almost 10% lower than those achieved by other algorithms. This should be caused by the noisy features of patterns so that the neighboring relation between patterns belonging to the same cluster may not be well-reflected by the calculated distance. Besides, Cluster*dp* suffered from the problem of selecting qualified cluster centers from the decision graph on all data sets. In our experiments, almost all patterns were in a mass while few of them satisfied the requirements of having many neighbors and long distance to other more qualified cluster centers.

We further conducted a case study on the best clustering results of all algorithms achieved in Fig. 9 by comparing their performances in terms of purity, class entropy, and the Rand index. For DBSCAN and the ART-based algorithms, the means and standard derivations obtained from ten runs were reported and their differences were further measured by t-test. As shown in Table I, the proposed CM-ART and AM-ART typically obtained the best performance across all data sets in terms of purity and the Rand index, which was usually significantly better than that achieved by DBSCAN, Affinity Propagation, and Cluster*dp* at the significant level *p* = 0.001. Fuzzy ART usually obtains comparative performance to the proposed algorithms and obtained the best performance on
TABLE I: The best clustering performance of DBSCAN, Affinity Propagation, Cluster$_{dp}$, Fuzzy ART, AM-ART, CM-ART, and HI-ART on the four data sets in terms of purity, class entropy, and the Rand index.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>DBSCAN</th>
<th>Affinity Propagation</th>
<th>Cluster$_{dp}$</th>
<th>Fuzzy ART</th>
<th>AM-ART</th>
<th>CM-ART</th>
<th>HI-ART</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUS-WIDE</td>
<td>0.7307 ± 0.023</td>
<td>0.6926 ± 0.027</td>
<td>0.7265 ± 0.026</td>
<td>0.7143 ± 0.023</td>
<td>0.7266 ± 0.026</td>
<td>0.7348 ± 0.025</td>
<td>0.7109 ± 0.021</td>
</tr>
<tr>
<td>20 Newsgroups</td>
<td>0.6903 ± 0.033</td>
<td>0.8028 ± 0.021</td>
<td>0.8252 ± 0.023</td>
<td>0.8745 ± 0.023</td>
<td>0.8918 ± 0.023</td>
<td>0.7663 ± 0.022</td>
<td>0.7915 ± 0.020</td>
</tr>
<tr>
<td>Corel5K</td>
<td>0.7408 ± 0.013</td>
<td>0.8084 ± 0.028</td>
<td>0.8352 ± 0.022</td>
<td>0.9007 ± 0.023</td>
<td>0.7983 ± 0.024</td>
<td>0.7627 ± 0.018</td>
<td>0.5391 ± 0.017</td>
</tr>
<tr>
<td>BlogCatalog</td>
<td>0.3021 ± 0.010</td>
<td>0.3809 ± 0.019</td>
<td>0.3800 ± 0.020</td>
<td>0.5321 ± 0.022</td>
<td>0.5003 ± 0.024</td>
<td>0.5218 ± 0.019</td>
<td>0.4963 ± 0.017</td>
</tr>
</tbody>
</table>

the Corel5K data set in terms of purity and the Rand index. However, it did not perform significantly differently than CM-ART at the significant level $p = 0.1$. We also observed that Affinity Propagation and DBSCAN usually obtain the best performance of class entropy, which indicated that the ART-based algorithms may have to generate more clusters to guarantee the higher quality of clusters. This may be due to the irregular distributions of patterns in the feature space resulted by the noisy patterns. Besides, the proposed AM-ART, CM-ART, and HI-ART usually achieve the performance not significantly different to the best performance in terms of class entropy at the significant levels $p = 0.05$. The above findings revealed that the proposed algorithms usually perform better than or comparable to the existing algorithms in terms of purity and the Rand index, and also perform reasonably well in terms of the class entropy.

F. Case Study on Noise Immunity

The noisy and diverse nature of social media data raises a challenge for the robustness of clustering algorithms to noise. Here, noise is not only defined by the noisy patterns that are isolated from clusters of the same class, but also defined by the noisy features that result in the noisy or ill-represented patterns. In this section, we reported the performance of our proposed algorithms and the baselines on noisy data.

To quantitatively evaluate the effectiveness of the proposed algorithms on noisy data, we followed a widely used method to add noise to different proportions of the original data to produce noisy data sets in different noisy levels. Specifically, we used the Matlab function $y = a u g n (x, s n r)$ to add additive white Gaussian noise to our collected data from the NUS-WIDE data set, where $x$ and $y$ are the original and the generated noisy data patterns respectively, and $s n r$ is the signal-to-noise ratio. We empirically set $s n r = 20$ to ensure that the generated noisy patterns will generally blur but not break the original distribution of patterns to certain extent. For each class of data patterns, we randomly selected the same number of patterns to add noise. In total, we generated ten noisy data sets with different proportions of noisy patterns. The average performance of all algorithms obtained from ten runs on the original and ten noisy data sets were reported in Fig. 10. Regarding the ART-based algorithms, we observed that Fuzzy ART has a relatively stable decrease in performance when applied to noisier data sets while AM-ART, CM-ART, and HI-ART behave differently. AM-ART and HI-ART shows a much better robustness than Fuzzy ART, especially HI-ART whose performance is almost not affected by the noisy patterns; while the performance of CM-ART has a fluctuation on different noisy data sets. With an investigation of the generated cluster structures, we found that the performance of Fuzzy ART decreased mainly because of the increase in the generated clusters while the clusters generated by Fuzzy ART still had a high quality in terms of precision; AM-ART and HI-ART alleviate this case by generating high-quality but much fewer clusters than Fuzzy ART; the performance of CM-ART was affected by the case when the noisy patterns were selected as cluster centers. This produced much more complex cluster boundaries and resulted in the over-generation of clusters. However, by incorporating both AMR and CMR, HI-ART can largely alleviate this problem. In comparison, the performance of DBSCAN, Affinity Propagation, and Cluster$_{dp}$ also decreased and fluctuated along with the increase in the percentage of noisy data. It demonstrated the robustness of the proposed AM-ART and HI-ART to noise.

G. Time Cost Analysis

This section presents an evaluation on the time cost of the proposed algorithms and the baselines on the NUS-WIDE data set with respect to the increase in the number of input patterns.

Fig. 10: Performance of AM-ART, CM-ART, HI-ART, and the algorithms in comparison on noisy data generated from the NUS-WIDE data set.
We demonstrated in experiments that by incorporating AMR, CMR and HIR into Fuzzy ART, the resulting AM-ART, CM-ART and HI-ART usually perform better than or comparable to state-of-the-art clustering algorithms that require no pre-defined number of clusters. More importantly, we found AMR, CMR and HIR could greatly improve the performance of Fuzzy ART when the vigilance value was low, especially CMR; AMR could significantly reduce the number of small clusters when the vigilance value was large; HIR could greatly improve the robustness of Fuzzy ART to noisy data. These enable AM-ART, CM-ART and HI-ART to be more robust to the initial vigilance value than Fuzzy ART.

Several avenues exist for further study. First, with the geometrical interpretation of the VR and Fuzzy ART, the clustering mechanism could be further improved by, for example, improving the shape of the VR to produce better cluster boundaries. Second, existing vigilance adaptation rules still require an initial vigilance value. Therefore, developing algorithms for choosing a suitable initial vigilance value is an important future work. Third, HI-ART has shown comparable performance and better noise immunity than AM-ART and CM-ART, which demonstrates the viability of integrating multiple ideas for tuning the vigilance parameter. Therefore, exploring new rules to develop hybrid methods for Fuzzy ART is a promising way to make contributions towards parameter-free clustering algorithms.

ACKNOWLEDGMENT

This research is supported in part by the National Research Foundation, Prime Minister’s Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office.

REFERENCES
