12-2015

Adaptive scaling of cluster boundaries for large-scale social media data clustering

Lei MENG

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

Donald C. WUNSCH

Citation

MENG, Lei; TAN, Ah-hwee; and WUNSCH, Donald C.. Adaptive scaling of cluster boundaries for large-scale social media data clustering. (2015). *IEEE Transactions on Neural Networks and Learning Systems*. 27, (12), 2656-2669.
Available at: https://ink.library.smu.edu.sg/sis_research/5235

Adaptive Scaling of Cluster Boundaries 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 the normalization method, changes the clustering mechanism of Fuzzy ART, and discovers 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 values 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 several state-of-the-art clustering algorithms that 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, the clustering of web multimedia data from social websites has drawn much attention for social community discovery [12], [45], collective behavior analysis [13], and underlying topic discovery [3], [5]. However, different from 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 algorithms 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 studying the neighborhood 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 (Cluster$_{dp}$) [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 requires one or more parameters in order 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 conducted a theoretical analysis in Fuzzy ART, in which we demonstrated 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.

vigilance 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 image data set [7], a subset of the 20 Newsgroups data set [48], the Corel5k data set [50], and 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]. The empirical results show that AM-ART, CM-ART and HI-ART are more robust than Fuzzy ART to the vigilance parameter and usually perform better than the algorithms in comparison.

The remainder of the 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 rules, 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 divided 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 intracluster 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, as 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 was provided in [30].

3) Density-based Clustering: 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$_{dp}$ [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$_{dp}$ 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 d-ifferent 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 the clustering scenario without any additional information remains a challenge.

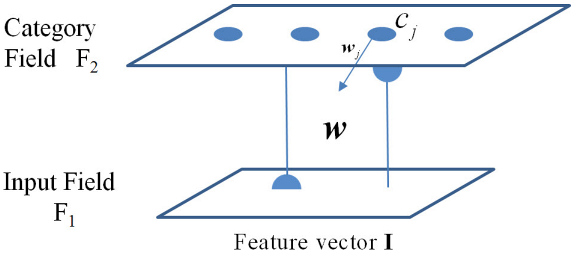


Fig. 1: Fuzzy ART architecture.

III. Fuzzy ART

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

Input vectors: Let $\mathbf{I} = \mathbf{x}$ denotes the input pattern in the input field F_1. With complement coding [2], $\mathbf{x}$ is further concatenated with its complement vector $\bar{\mathbf{x}}$ such that $\mathbf{I} = (\mathbf{x}, \bar{\mathbf{x}})$, where $\bar{\mathbf{x}} = \mathbf{1} - \mathbf{x}$.

Weight vectors: Let $\mathbf{w}_j$ denotes the weight vector associated with the jth 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 $\mathbf{I}$, Fuzzy ART calculates the choice function for all of the clusters in the category field F_2 and selects the most suitable cluster (winner) c_{j^*}, which has the largest value. The choice function for the jth cluster c_j is defined by

$$T_j = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, \tag{1}$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, and the norm $|.|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$.

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

$$M_{j^*} = \frac{|\mathbf{I} \wedge \mathbf{w}_{j^*}|}{|\mathbf{I}|}. \tag{2}$$

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 $\mathbf{w}_{j^*}$ will be updated through a learning function, defined by

$$\mathbf{w}_{j^*}^{(new)} = \beta(\mathbf{I} \wedge \mathbf{w}_{j^*}) + (1 - \beta)\mathbf{w}_{j^*}. \tag{3}$$

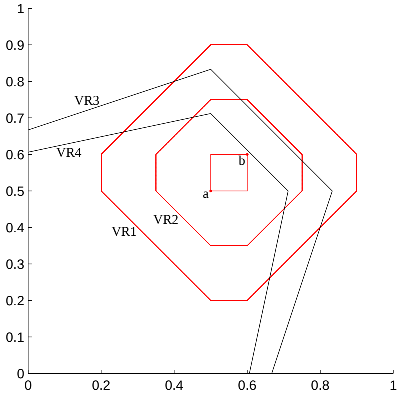


Fig. 2: An example of a cluster and its vigilance regions with or without complement coding in Fuzzy ART in 2D space.

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 $\mathbf{x}$ is represented as $\mathbf{I} = (\mathbf{x}, \bar{\mathbf{x}})$. Therefore, the weight vector $\mathbf{w}$ of a cluster also includes the feature part and the complement part. As demonstrated in [2], given a weight vector $\mathbf{w} = (\mathbf{a}, \bar{\mathbf{b}})$, the match area of the weight vector $\mathbf{w}$ corresponds to a rectangle in the 2D space, where $\mathbf{a}$ and $\mathbf{b}$ are the two diagonal corner points of the rectangle. As shown in Fig. 2, VR1 and VR2, are the VRs of the weight vector $\mathbf{w}$ generated under vigilance parameters $\rho = 0.75$ and 0.825 respectively. Without complement coding, weight vector $\mathbf{w}$ is the point $\mathbf{a}$, 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 $\mathbf{w}_j$ of cluster c_j is a subset of input pattern $\mathbf{I}$. We can prove that, incorporating complement coding in Fuzzy ART, the choice function considers the similarity between the input pattern and the weight hyper-rectangle of the selected cluster.

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

Proof 1:

$$\begin{aligned} T_j &= \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} \\ &= \frac{|\mathbf{x} \wedge \mathbf{a}| + |\bar{\mathbf{x}} \wedge \bar{\mathbf{b}}|}{|\mathbf{w}_j|} \\ &= \frac{|\mathbf{x} \wedge \mathbf{a}| + |\overline{\mathbf{x} \vee \mathbf{b}}|}{|\mathbf{a} + \bar{\mathbf{b}}|} \\ &= \frac{|\mathbf{a}|}{|\mathbf{a} + \bar{\mathbf{b}}|} \cdot \frac{|\mathbf{x} \wedge \mathbf{a}|}{\alpha + |\mathbf{a}|} + \frac{|\bar{\mathbf{b}}|}{|\mathbf{a} + \bar{\mathbf{b}}|} \cdot \frac{|\overline{\mathbf{x} \vee \mathbf{b}}|}{\alpha + |\bar{\mathbf{b}}|}. \end{aligned} \tag{4}$$

As shown in (4), the choice function evaluates both the degree to which $\mathbf{a}$ is a subset of $\mathbf{x}$ and to which $\mathbf{x}$ is a subset of $\mathbf{b}$. The final choice value is obtained by their weighted summation, which is normalized by their respective norms. Therefore, given $\mathbf{x} = (x_1, ..., x_m)$, $\mathbf{a} = (a_1, ..., a_m)$, and $\mathbf{b} = (b_1, ..., b_m)$, choice function T_j achieves its maximum for c_j when, for $\forall i \in [1, m]$, $a_i \leq x_i \leq b_i$. 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 $\mathbf{I}$ is a subset of weight vector $\mathbf{w}_{j^*}$ of cluster c_{j^*}. In template matching, input pattern $\mathbf{I}$ is considered to be similar to the winner cluster c_{j^*} if

$$M_{j^*} = \frac{|\mathbf{I} \wedge \mathbf{w}_{j^*}|}{|\mathbf{I}|} \geq \rho. \tag{5}$$

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 $\mathbf{w}_{j^*} = (w_1, ..., w_m)$ of cluster c_{j^*}, the vigilance parameter ρ, and an arbitrary input pattern $\mathbf{I} = (x_1, ..., x_m)$ in the Fuzzy ART system. If Fuzzy ART does not employ complement coding, (5) is equivalent to

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

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

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

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 VR of a cluster and subsequently use them to interpret the clustering process of Fuzzy ART.

Property 2: Given the weight vector $\mathbf{w}_j = (w_1, ..., w_m)$ of cluster c_j in the Fuzzy ART system with complement coding, the VR of c_j consists of $3^{\frac{m}{2}} - 1$ hyper-planes.

Proof 2: Similar to (4), given $\mathbf{w}_j = (a_1, ..., a_{\frac{m}{2}}, \bar{b}_1, ..., \bar{b}_{\frac{m}{2}})$, and $\mathbf{I} = (x_1, ..., x_{\frac{m}{2}}, \bar{x}_1, ..., \bar{x}_{\frac{m}{2}})$, (5) can be expressed as

$$\Sigma_{i=1}^{\frac{m}{2}} \min(x_i, a_i) + \Sigma_{i=1}^{\frac{m}{2}} \overline{\max(x_i, b_i)} \geq \frac{m\rho}{2}. \tag{8}$$

In view that the m dimensional vector $\mathbf{w}_j$ is a hyper-rectangle in the $\frac{m}{2}$ dimensional space, and for $\forall i \in [1, \frac{m}{2}]$, $x_i \in [0, a_i) \cup [a_i, b_i) \cup [b_i, 1]$. Therefore, the feature space is divided into $3^{\frac{m}{2}}$ subsections. Considering that (8) is an identical equation in the weight hyper-rectangle, the number of hyper-planes for constructing the VR is $3^{\frac{m}{2}} - 1$.

Property 3: Patterns falling into the weight hyper-rectangle have the same value of match function (2).

Proof 3: Given a cluster c_j and its weight vector $\mathbf{w}_j = (a_1, ..., a_{\frac{m}{2}}, \bar{b}_1, ..., \bar{b}_{\frac{m}{2}})$ and $\mathbf{I} = (x_1, ..., x_{\frac{m}{2}}, \bar{x}_1, ..., \bar{x}_{\frac{m}{2}})$ falling into the weight hyper-rectangle, we have for $\forall i \in [1, \frac{m}{2}]$, $a_i \leq x_i \leq b_i$. In this case, according to (8), the value of the match function depends only on weight vector $\mathbf{w}_j$ such that $|\mathbf{I} \wedge \mathbf{w}_{j^*}| = \mathbf{w}_{j^*}$. Therefore, all of the patterns in the weight hyper-rectangle have the same match value.

The situation may also be interpreted as all of those patterns having the same ℓ_1 distance to $\mathbf{a}$ and $\mathbf{b}$, as

$$\begin{aligned} |\mathbf{x} - \mathbf{a}| + |\mathbf{x} - \mathbf{b}| &= \sum_i (x_i - a_i) + \sum_i (b_i - x_i) \\ &= \sum_i (b_i - a_i). \end{aligned} \tag{9}$$

Property 4: Patterns falling into the weight hyper-rectangle of the winner do not result in the expansion of the weight hyper-rectangle during the learning step (3).

Proof 4: In *Property* 2, if $\mathbf{I}$ falls into the weight hyper-rectangle of cluster c_j, we have $|\mathbf{I} \wedge \mathbf{w}_{j^*}| = \mathbf{w}_{j^*}$. In this case, (3) is equivalent to

$$\mathbf{w}_{j^*}^{(new)} = \beta \mathbf{w}_{j^*} + (1 - \beta)\mathbf{w}_{j^*} = \mathbf{w}_{j^*}. \tag{10}$$

Therefore, weight vector $\mathbf{w}_{j^*}$ undergoes no change after encoding input pattern $\mathbf{I}$.

Property 5: The weight hyper-rectangle of a cluster reflects the cluster size, which is controlled by the learning rate β.

Proof 5: Given input pattern $\mathbf{I} = (\mathbf{x}, \bar{\mathbf{x}})$, winner c_{j^*} and its corresponding weight vector $\mathbf{w}_{j^*} = (\mathbf{a}, \bar{\mathbf{b}})$. If $\mathbf{I}$ is categorized into c_{j^*}, $\mathbf{w}_{j^*}$ is updated according to (3) such that

$$\begin{aligned} \mathbf{w}_{j^*}^{(new)} &= (\mathbf{a}^{(new)}, \bar{\mathbf{b}}^{(new)}) = \beta(\mathbf{I} \wedge \mathbf{w}_{j^*}) + (1 - \beta)\mathbf{w}_{j^*} \\ &= \beta(\mathbf{x} \wedge \mathbf{a}, \overline{\mathbf{x} \vee \mathbf{b}}) + (1 - \beta)(\mathbf{a}, \bar{\mathbf{b}}) \\ &= (\beta(\mathbf{x} \wedge \mathbf{a}) + (1 - \beta)\mathbf{a}, \beta(\overline{\mathbf{x} \vee \mathbf{b}}) + (1 - \beta)\bar{\mathbf{b}})). \end{aligned} \tag{11}$$

We observe that the update of weight vector $\mathbf{w}_{j^*}$ is essentially the movement of $\mathbf{a}$ and $\mathbf{b}$ towards the input pattern $\mathbf{I}$. Specifically, $\mathbf{a}$ moves towards $\mathbf{I}$ in the dimensions $\{i | x_i < a_i\}$, while $\mathbf{b}$ moves towards $\mathbf{I}$ 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.

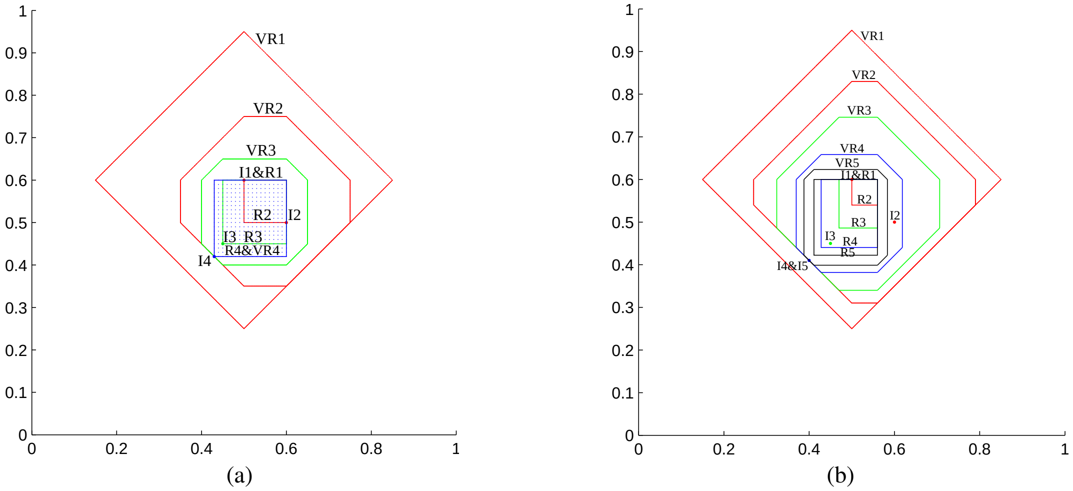
Proof 6: As demonstrated in *Property* 2, a VR in the $\frac{m}{2}$ dimensional space is constructed by $3^{\frac{m}{2}} - 1$ functions, each of which is calculated using (8). Because the learning function (3) suppresses the values of features, we have $a_i^{(new)} \leq a_i$ and $b_i^{(new)} \geq b_i$ for $\forall i \in [1, \frac{m}{2}]$. 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 in Fig. 3. Fig. 3a depicts the evolution of a cluster in Fuzzy ART under learning parameter $\beta = 1$. When the cluster has only one pattern I1, the weight rectangle R1 is situated exactly at point I1. In this case, the corresponding VR1 is a square diamond centered by I1. After encoding I2, R2 becomes a rectangle, and the corresponding VR2 becomes an octagon, which satisfies *Property* 2. During the presentation of I3 and I4, the weight rectangle expands to cover all of the patterns, which satisfies *Property* 5. It is notable that I4 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 I4 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 $\mathbf{a}$ may gradually move to the origin, which causes category proliferation. With complement coding, the VR of



(a)

(b)

Fig. 3: A 2D example on the evolution of a cluster in Fuzzy ART under different learning parameter values, (a) $\beta = 1$ and (b) $\beta = 0.6$. (a) Sequential presentation of I1(0.5,0.6), I2(0.6,0.5), I3(0.45,0.45) and I4(0.43,0.42). R1-R4 indicate the expansion of the cluster's weight rectangle, and VR1-VR4 indicate the corresponding VRs. (b) Sequential presentation of I1(0.5,0.6), I2(0.6,0.5), I3(0.45,0.45), I4(0.4,0.41), and I5(0.4,0.41).

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 improving its limitations. 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 VR covers all of the 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 ρ, 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 ρ, which both determines the VRs of clusters for accepting patterns and limits the size of all of the 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 ART to scale the size of all clusters. Based on the above consideration, in our preliminary study [38], we proposed two heuristic methods, the activation maximization rule (AMR) and the confliction minimization rule (CMR), which 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 VR and further propose a hybrid method to integrate 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 ρ_{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 σ, 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 σ 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 miscategorized 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 to prevent the generation of small clusters and the over-generalization of cluster weights by evening out the sizes of two very close clusters.

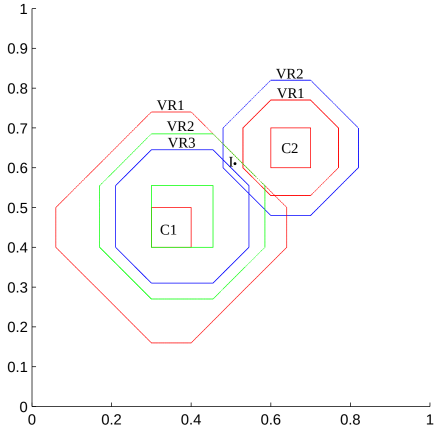


Fig. 4: A 2D example on how AMR adapts vigilance values of two clusters in Fuzzy ART with complement coding.

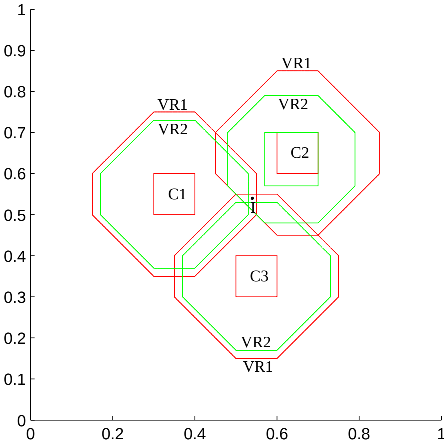


Fig. 5: A 2D example on how CMR adapts vigilance values of clusters to reduce overlap between their VRs.

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 $\mathcal{W}inc = \{c_j | M_j \geq \rho\}$ 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 $\{c_j | c_j \in \mathcal{W}inc \wedge j \neq j^*\}$ 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 time 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 $\mathcal{W}inc = \{c_j | M_j \geq \rho\}$ 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 $\rho_{j^*}^{(new)} = (1+\sigma)\rho_{j^*}$;

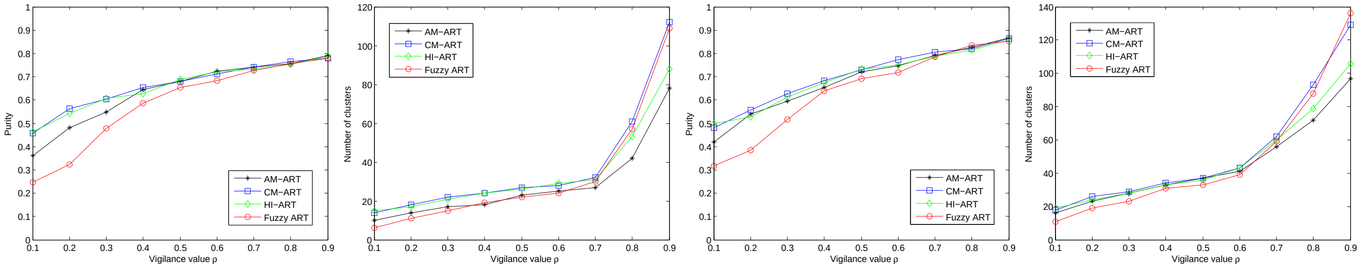
3) **Confliction Minimization**: Update the vigilance parameters of all winner candidates except for the winner $\{c_j | c_j \in \mathcal{W}inc \wedge j \neq j^*\}$ using $\rho_j^{(new)} = M_j + \Delta$;

4) **Activation Maximization**: Search in the remaining clusters to identify the set of clusters $\mathcal{R}c = \{c_j | c_j \in F_2 \wedge c_j \notin \mathcal{W}inc \wedge T_j \geq T_{j^*}\}$, and for $\forall c_j \in \mathcal{R}c$, set $\rho_j^{(new)} = (1-\sigma)\rho_j$.

D. Computational Complexity Analysis

As presented in Section III, given an input pattern, 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 $O(n_f)$, $O(n_c n_f)$, and $O(n_f)$, where n_c denotes the number of clusters and n_f denotes the number of features. Therefore, given n_i input patterns, the overall time complexity of ART is $O(n_i n_c n_f)$.

AMR requires ART to adapt the vigilance parameter values of the selected winners, of which the time complexity is $O(n_c)$. Therefore, the time complexity of AM-ART is $O(n_i n_c n_f)$. CM-ART reverses procedures of category choice and template matching and adapts the vigilance parameter values of all winner candidates, whose time complexity is $O(n_c)$. Therefore, the time complexity of CM-ART is also $O(n_i n_c n_f)$. Similarly, HI-ART makes a integration of procedures of AM-ART and CM-ART. Therefore, its time complexity is $O(n_i n_c n_f)$.



(a) (b) (c) (d)

Fig. 6: Sensitivity of AM-ART, CM-ART, HI-ART, and Fuzzy ART to the vigilance parameter ρ measured by Purity and number of generated clusters on the (a & b) NUS-WIDE and (c & d) 20 Newsgroups data sets.

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 netnews newsgroups, each of which contains nearly 1,000 documents. We collected 9,357 messages 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[1]. Regarding the feature extraction, we filtered any words that occurred less than 30 times, and each message 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 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 α, β, and ρ. Besides, AM-ART has the restraint parameter σ, CM-ART has the parameter Δ, 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 ρ 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 ρ typically results in the generation of a few small clusters, typically 10% of the total number of the generated clusters. Besides, a small cluster typically contains several or tens of patterns. We followed this method to select the initial value of ρ.

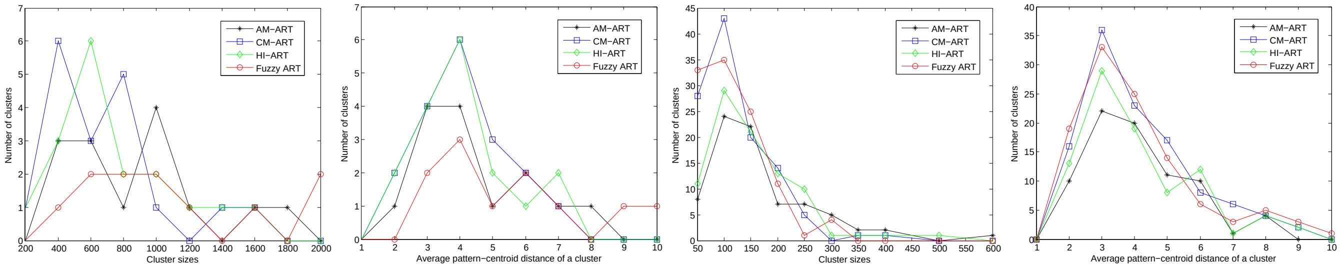
C. Robustness to Vigilance Parameter

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 ρ. 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 can take 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.

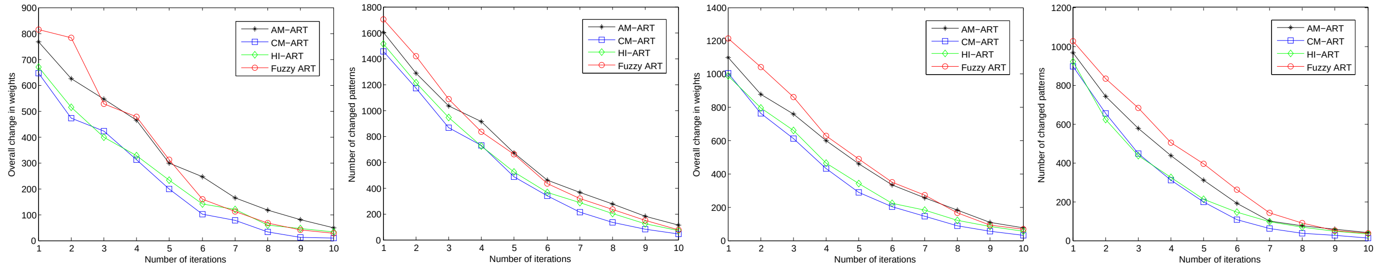
A case study was further conducted by analyzing the size and coherence of the clusters generated by each algorithm to provide a deeper understanding of how the proposed algo-

[1] http://qwone.com/∼jason/20Newsgroups/



(a) (b) (c) (d)

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 sizes and average pattern-centroid distance under (a & b) $\rho = 0.2$ and (c & d) $\rho = 0.9$.



(a) (b) (c) (d)

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

rithms work. As shown in 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 explained 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 explained why they can generate fewer clusters than the other algorithms and demonstrated the effectiveness of AMR in simplifying the network with a high vigilance value.

D. Convergence Analysis

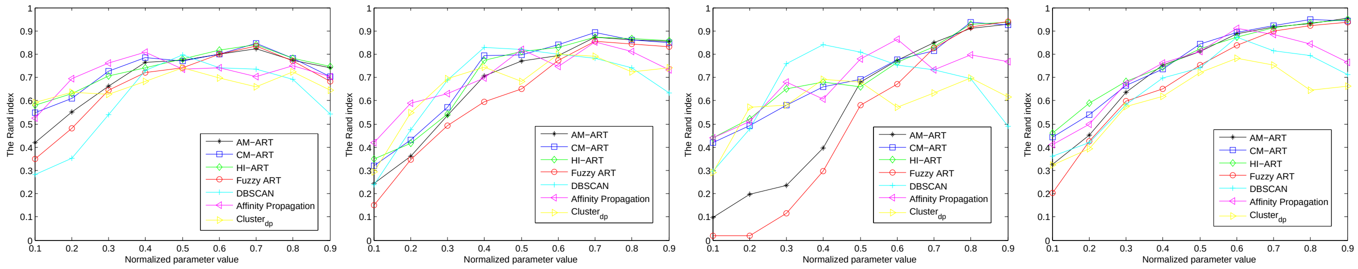
This section presents a comparative study on the convergence property of the proposed algorithms and Fuzzy ART, which is measured by the overall change in weights and the number of patterns moving across clusters with respect to the repeat presentation of patterns.

We reported the results on the NUS-WIDE and 20 Newsgroups data sets and similar findings were observed on the other data sets. In the experiments, we set $\rho = 0.7$ and 0.6 for the NUS-WIDE and 20 Newsgroups data sets, respectively. 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 converges faster than Fuzzy ART during the first rounds of iterations due to CMR but achieves 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 on the cluster assignments of patterns.

E. Clustering Performance Comparison

In this section, we presented an evaluation of the clustering performance of AM-ART, CM-ART and HI-ART, and compared them 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 were 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 ART-based algorithms require the input values to be in the range of [0, 1]. Experimental results indicated that the normalization of data has an unobvious effect on the performance of other algorithms. To ensure a fair comparison, we utilized several 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



(a) (b) (c) (d)

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.

select the search radius ε, 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 p using the MATLAB function "preferenceRange.m"[2]. The values of $dampfact$, $convits$ 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. Cluster$_{dp}$ requires a search radius d_c and a cluster center selection procedure by the user. We set d_c to the value of ε used in DBSCAN as both have the same meaning and selected the cluster centers from the decision graph that producing 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 reported the performance of each algorithm under different parameter settings in terms of the Rand index on all data sets, which provided 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_c. 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, which were consistent with our findings in Section VI-C. It is notable that Fuzzy ART achieved very low performance when $\rho < 0.3$, 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

[2] http://genes.toronto.edu/index.php?q=affinity%20propagation

TABLE I: 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.

		DBSCAN	Affinity Propagation	Cluster$_{dp}$	Fuzzy ART	AM-ART	CM-ART	HI-ART
NUS-WIDE	Purity	0.6598 ± 0.015	0.6827	0.6193	0.7264 ± 0.026	0.7313 ± 0.023	**0.7436 ± 0.023**	0.7348 ± 0.025
	Class entropy	0.7188 ± 0.011	**0.7063**	0.7497	0.7287 ± 0.024	0.7148 ± 0.022	0.7266 ± 0.026	0.7159 ± 0.021
	Rand index	0.7970 ± 0.012	0.8084	0.7408	0.8305 ± 0.027	0.8244 ± 0.019	**0.8461 ± 0.026**	0.8419 ± 0.023
20 Newsgroups	Purity	0.7084 ± 0.017	0.7225	0.6518	0.7165 ± 0.027	0.7476 ± 0.022	**0.7735 ± 0.019**	0.7491 ± 0.024
	Class entropy	**0.5604 ± 0.016**	0.5779	0.5978	0.5679 ± 0.027	0.5873 ± 0.021	0.6081 ± 0.024	0.5936 ± 0.026
	Rand index	0.8303 ± 0.013	0.8522	0.7907	0.8527 ± 0.021	0.8745 ± 0.023	**0.8918 ± 0.018**	0.8794 ± 0.024
Corel5K	Purity	0.6792 ± 0.012	0.6926	0.5708	**0.7983 ± 0.026**	0.7627 ± 0.018	0.7863 ± 0.022	0.7715 ± 0.020
	Class entropy	0.4940 ± 0.009	0.5358	0.5639	0.5216 ± 0.016	**0.4758 ± 0.021**	0.5391 ± 0.017	0.5034 ± 0.015
	Rand index	0.8408 ± 0.018	0.8639	0.6977	**0.9391 ± 0.024**	0.9284 ± 0.014	0.9380 ± 0.019	0.9369 ± 0.021
BlogCatalog	Purity	0.7762 ± 0.017	0.8023	0.7129	0.8431 ± 0.017	**0.8635 ± 0.013**	0.8599 ± 0.023	0.8492 ± 0.016
	Class entropy	0.5121 ± 0.019	**0.4889**	0.5307	0.5321 ± 0.022	0.5003 ± 0.021	0.5218 ± 0.019	0.4963 ± 0.017
	Rand index	0.8720 ± 0.016	0.9120	0.7836	0.9361 ± 0.014	**0.9561 ± 0.018**	0.9492 ± 0.018	0.9484 ± 0.015

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 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 general more clusters to produce clusters with higher quality. This is due to the fact that the distributions of patterns belonging to the same class are not regular because of the noisy patterns in the data sets. 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 compared existing algorithms in terms of purity and the Rand index, and also perform reasonably well in terms of class entropy.

F. Case Study on Noise Immunity

The noisy and diverse nature of the 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 $y = awgn(x, snr)$ 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 snr is the signal-to-noise ratio. We empirically set $snr = 20$ 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.

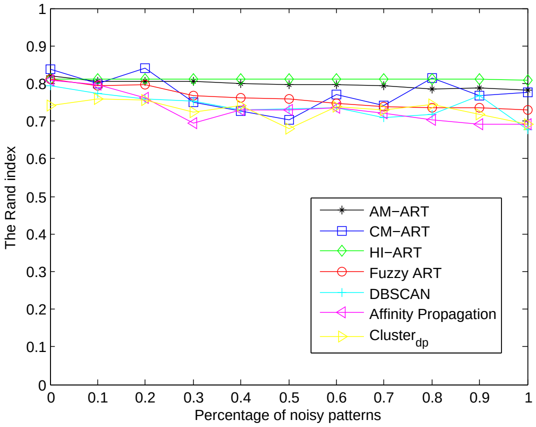


Fig. 10: Performance of AM-ART, CM-ART, HI-ART, and the algorithms in comparison on noisy data generated from the NUS-WIDE data set.

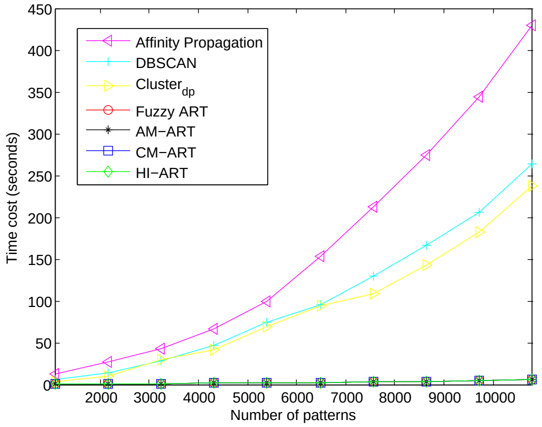


Fig. 11: Time cost of AM-ART, CM-ART, HI-ART, and the algorithms in comparison on the NUS-WIDE data set.

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. Specifically, to ensure an unbiased evaluation, we divided the 10,800 patterns in the data set into 10 subsets, each of which contained 1,080 patterns of equally sized subsets from the nine classes, and tested the time cost of each algorithm by incrementally adding a subset at each time. To ensure a fair comparison, we followed the parameter settings of each algorithm used in the previous section but tuned them slightly to force them to generate the same number of clusters. All algorithms were run on a 3.40GHz Intel(R) Core(TM) i7-4770 CPU with 16GB RAM. Fig. 11 illustrates that, compared with Affinity Propagation, DBSCAN, and Cluster$_{dp}$, the time cost of the four ART-based algorithms increased slightly as the number of input patterns increased. It is notable that AM-ART, CM-ART, HI-ART, and Fuzzy ART were able to cluster 10,800 patterns in 6 seconds. This demonstrated the scalability of the ART-based algorithms for big data sets. Moreover, the largest difference in their time cost was only less than 0.2 seconds, which demonstrated that the incorporation of AMR, CMR and HIR into Fuzzy ART incurs little computation.

VII. Conclusion

In this paper, we investigated making the vigilance parameter in Fuzzy ART self-adaptable in order to allow Fuzzy ART to consistently produce high-quality clusters under modest parameter settings for large-scale social media data sets. The contributions of this study are two-fold: First, we theoretically demonstrated the effect of complement coding on the Fuzzy ART clustering mechanism and offered a geometric interpretation of the cluster mechanism of Fuzzy ART using the identified vigilance region (VR). Second, we introduced 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, and proposed three adaptation rules, namely, the activation maximization rule (AMR), the confliction minimization rule (CMR), and the hybrid integration rule (HIR), for the vigilance parameter so that the clusters in the Fuzzy ART system would have individual vigilance levels and be able to adaptively tune their VR boundaries during the clustering process.

We demonstrated in experiments that by incorporating AMR, CMR and HIR into Fuzzy ART, the resulting AM-ART, CM-ART and HI-ART usually performed better than or comparably 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 is 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, algorithms for choosing a suitable initial vigilance value is required. 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 also a promising way to make contribution towards parameter-free clustering algorithms.

Acknowledgment

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

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. 37, no. 1, pp. 54-115, 1987.

[2] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System," *Neural Networks*, vol. 4, no. 6, pp. 759-771, 1991.

[3] A.-H. Tan, H.-L. Ong, H. Pan, J. Ng, and Q Li, "Towards Personalized Web Intelligence," *Knowledge and Information Systems*, vol. 6, no. 5, pp. 595-616, 2004.

[4] T. Jiang and A.-H. Tan, "Learning Image-Text Associations," *IEEE Transctions on Knowledge and Data Engineering*, vol. 21, no. 2, pp. 161-177, 2009.

[5] L. Meng and A.-H. Tan, "Semi-Supervised Hierarchical Clustering For Personalized Web Image Organization," *International Joint Conference on Neural Networks*, pp. 251-258, 2012.

[6] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "ARTMAP: Supervised Real-Time Learning And Classification Of Nonstationary Data By A Self-Organizing Neural Network," *Neural Networks*, vol. 4, no. 5, pp. 565-588, 1991.

[7] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A Real-World Web Image Database from National University of Singapore," *International Conference on Image and Video Retrieval*, 2009.

[8] A.-H. Tan, "Adaptive Resonance Associative Map," *Neural Networks*, vol. 8, no. 3, pp. 437-446, 1995.

[9] S. Grossberg, "How Does a Brain Build a Cognitive Code?" *Psychological Review*, vol. 87, no. 1, pp. 1-51, 1980.

[10] J. He, A.-H. Tan, and C.-L. Tan, "ART-C: A Neural Architecture for Self-Organization under Constraints," *International Joint Conference on Neural Networks*, pp. 2550-2555, 2002.

[11] J. He, A.-H. Tan, and C.-L. Tan, "Modified ART 2A Growing Network Capable of Generating a Fixed Number of Nodes," *IEEE Transactions on Neural Networks*, vol. 15, no. 3, pp. 728-737, 2004.

[12] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community Detection in Social Media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515-554, 2012.

[13] L. Tang and H. Liu, "Scalable Learning of Collective Behavior Based on Sparse Social Dimensions," *ACM International Conference on Information and Knowledge Management*, pp. 1107-1116, 2009.

[14] L. Meng, A.-H. Tan, and D. Xu, "Semi-Supervised Heterogeneous Fusion for Multimedia Data Co-clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2293-2306, 2014.

[15] J. C. Bezdek and R. Hathaway, "VAT: A Tool for Visual Assessment of (Cluster) Tendency," *International Joint Conference on Neural Networks*, pp. 2225-2230, 2002.

[16] I. Sledge, J. Huband, and J. C. Bezdek, "(Automatic) Cluster Count Extraction from Unlabeled Datasets," *International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 3-13, 2008.

[17] L. Wang, C. Leckie, K. Ramamohanarao, and J. Bezdek, "Automatically Determining the Number of Clusters in Unlabeled Data Sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 3, pp. 335-350, 2012.

[18] W. Wang and Y. Zhang, "On Fuzzy Cluster Validity Indices," *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2095-2117, 2007.

[19] J. Liang, X. Zhao, D. Li, F. Cao, and C. Dang, "Determining the Number of Clusters Using Information Entropy for Mixed Data," *Pattern Recognition*, vol. 45, no. 6, pp. 2251-2265, 2012.

[20] C. A. Sugar and G. M. James, "Finding the Number of Clusters in A Data Set: An Information Theoretic Approach," *Journal of the American Statistical Association*, vol. 98, no. 463, pp. 750-763, 2003.

[21] H. Sun, S. Wang, and Q. Jiang, "FCM-Based Model Selection Algorithms for Determining the Number of Clusters," *Pattern Recognition*, vol. 37, no. 10, pp. 2027-2037, 2004.

[22] R. Kothari and D. Pitts, "On Finding the Number of Clusters," *Pattern Recognition Letters*, vol. 20, no. 4, pp. 405-416, 1999.

[23] J.-S. Lee and S. Olafsson, "A Meta-Learning Approach for Determining the Number of Clusters with Consideration of Nearest Neighbors," *Information Sciences*, vol. 232, pp. 208-224, 2013.

[24] H.-P. Kriegel, P. Kroger, J. Sander, and A. Zimek, "Density-Based Clustering," *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231-240, 2011.

[25] B. J. Frey and D. Dueck, "Clustering by Passing Messages Between Data Points," *Science*, vol. 315, pp. 972-976, 2007.

[26] M. J. Li, M. K. Ng, Y. Cheung, and Z. X. Huang, "Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1519-1534, 2008.

[27] Y. Leung, J. S. Zhang, and Z. B. Xu, "Clustering by Scale-Space Filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1394-1410, 2000.

[28] H. Yan, K. K. Chen, L. Liu, and J. Bae, "Determining the Best K for Clustering Transactional Datasets: A Coverage Density-Based Approach," *Data & Knowledge Engineering*, vol. 68, no. 1, pp. 28-48, 2009.

[29] S. Bandyopadhyay and S. Saha, "A Point Symmetry-Based Clustering Technique for Automatic Evolution of Clusters," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1441-1457, 2008.

[30] S. Bandyopadhyay, "Genetic Algorithms for Clustering and Fuzzy Clustering," *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 6, pp. 524-531, 2011.

[31] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 226-231, 1996.

[32] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169-194, 1998.

[33] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering Points to Identify the Clustering Structure," *ACM SIGMOD International Conference on Management of Data*, pp. 49-60, 1999.

[34] T. Pei, A. Jasra, D. J. Hand, A.-X. Zhu, and C. Zhou, "DECODE: A New Method for Discovering Clusters of Different Densities In Spatial Data," *Data Mining and Knowledge Discovery*, vol. 18, no. 3, pp. 337-369, 2009.

[35] T. N. Tran, R. Wehrens, and L. M. C. Buydens, "KNN-Kernel Density-Based Clustering for High-Dimensional Multivariate Data," *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 513-525, 2006.

[36] Y. Jia, J. Wang, C. Zhang, and X.-S. Hua, "Finding Image Exemplars Using Fast Sparse Affinity Propagation," *ACM International Conference on Multimedia*, pp. 639-642, 2008.

[37] Y. Fujiwara, G. Irie, and T. Kitahara, "Fast Algorithm for Affinity Propagation," *International Joint Conference on Artificial Intelligence*, pp. 2238-2243, 2011.

[38] L. Meng, A.-H. Tan, and D. C. Wunsch II, "Vigilance Adaptation in Adaptive Resonance Theory," *International Joint Conference on Neural Networks*, 2013.

[39] G. A. Carpenter and S. Grossberg, "ART 2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns," *Applied Optics*, vol. 26, no. 23, pp. 4919-4930, 1987.

[40] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "ART 2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition," *Neural Networks*, vol. 4, pp. 493-504, 1987.

[41] G. A. Carpenter and S. Grossberg, "ART 3: Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures," *Neural Networks*, vol. 3, no.2, pp. 129-152, 1990.

[42] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A Neural Network Architecture For Incremental Supervised Learning Of Analog Multidimensional Maps," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698-713, 1992.

[43] D. G. Amorim, M. F. Delgado, and S. B. Ameneiro, "Polytope ARTMAP: Pattern Classification Without Vigilance Based on General Geometry Categories," *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1306-1325, 2007.

[44] Y. Zhao and G. Karypis, "Criterion Functions for Document Clustering: Experiments and Analysis," *Technical Report, Department of Computer Science, University of Minnesota*, 2001.

[45] L. Meng and A.-H. Tan, "Community Discovery in Social Networks via Heterogeneous Link Association and Fusion," *SIAM International Conference on Data Mining*, pp.803-811, 2014.

[46] J. He, A.-H. Tan, C.-L. Tan, and S.-Y. Sung, "On Quantitative Evaluation of Clustering Systems," *Clustering and Information Retrieval, Kluwer Academic Publishers*, pp.105-133, 2003.

[47] R. Xu and D. C. Wunsch II, "BARTMAP: A Viable Structure for Biclustering," *Neural Networks*, vol. 24, no. 7, pp. 709-716, 2011.

[48] K. Lang, "Newsweeder: Learning to Filter Netnews," *International Conference on Machine Learning*, pp. 331-339, 2005.

[49] A. Rodriguez and A. Laio, "Clustering by Fast Search and Find of Density Peaks," *Science*, vol. 344, no. 6191, pp. 1492-1496, 2014.

[50] P. Duygulu, K. Barnard, J. F. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," *European Conference on Computer Vision*, pp. 97-112, 2002.

[51] X. Wang, L. Tang, H. Gao and H. Liu, "Discovering Overlapping Groups in Social Media," *IEEE International Conference on Data Mining*, pp. 569-578, 2010.

[52] B. Mirkin, "Choosing the Number of Clusters," *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 252-260, 2011.

[53] S. Saha, and S. Bandyopadhyaya, "A Symmetry Based Multiobjective Clustering Technique for Automatic Evolution of Clusters," *Pattern Recognition*, vol. 43, no. 3, pp. 738-751, 2010.

[54] M. M.-T. Chiang, and B. Mirkin, "Intelligent Choice of the Number of Clusters in K-Means Clustering: An Experimental Study with Different Cluster Spreads," *Journal of Classification*, vol. 27, no. 1, pp. 3-40, 2010.

[55] M. Yan, and K. Ye, "Determining the Number of Clusters Using the Weighted Gap Statistic," *Biometrics*, vol. 63, no. 4, pp. 1031-1037, 2007.

[56] G. B. Mufti, P. Bertrand, and L. E. Moubarki, "Determining the number of groups from measures of cluster stability," *International Symposium on Applied Stochastic Models and Data Analysis*, pp. 404-412, 2005.

[57] G. J. McLachlan, and N. Khan, "On a Resampling Approach for Tests on the Number of Clusters with Mixture Model-based Clustering of Tissue Samples," *Journal of Multivariate Analysis*, vol. 90, no. 1, pp. 90-105, 2004.

[58] I. Ozkan, and I. B. Turksen, "MiniMax ε-stable Cluster Validity Index for Type-2 Fuzziness," *Information Sciences*, vol. 184, no. 1, pp. 64-74, 2012.

[59] P. Duygulu, K. Barnard, J. F. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," *European Conference on Computer Vision*, pp. 97-112, 2002.

[60] X. Wang, L. Tang, H. Gao, and H. Liu, "Discovering Overlapping Groups in Social Media," *IEEE International Conference on Data Mining*, pp. 569-578, 2010.