Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.042222

ARTICLE

Automatic Aggregation Enhanced Affinity Propagation Clustering Based on Mutually Exclusive Exemplar Processing

Zhihong Ouyang*, Lei Xue, Feng Ding and Yongsheng Duan

Electronic Countermeasure Institute, National University of Defense Technology, Hefei, 230037, China

*Corresponding Author: Zhihong Ouyang. Email: ouyangzhihong@nudt.edu.cn

Received: 23 May 2023 Accepted: 24 August 2023 Published: 31 October 2023

ABSTRACT

Affinity propagation (AP) is a widely used exemplar-based clustering approach with superior efficiency and clustering quality. Nevertheless, a common issue with AP clustering is the presence of excessive exemplars, which limits its ability to perform effective aggregation. This research aims to enable AP to automatically aggregate to produce fewer and more compact clusters, without changing the similarity matrix or customizing preference parameters, as done in existing enhanced approaches. An automatic aggregation enhanced affinity propagation (AAEAP) clustering algorithm is proposed, which combines a dependable partitioning clustering approach with AP to achieve this purpose. The partitioning clustering approach generates an additional set of findings with an equivalent number of clusters whenever the clustering stabilizes and the exemplars emerge. Based on these findings, mutually exclusive exemplar detection was conducted on the current AP exemplars, and a pair of unsuitable exemplars for coexistence is recommended. The recommendation is then mapped as a novel constraint, designated mutual exclusion and aggregation. To address this limitation, a modified AP clustering model is derived and the clustering is restarted, which can result in exemplar number reduction, exemplar selection adjustment, and other data point redistribution. The clustering is ultimately completed and a smaller number of clusters are obtained by repeatedly performing automatic detection and clustering until no mutually exclusive exemplars are detected. Some standard classification data sets are adopted for experiments on AAEAP and other clustering algorithms for comparison, and many internal and external clustering evaluation indexes are used to measure the clustering performance. The findings demonstrate that the AAEAP clustering algorithm demonstrates a substantial automatic aggregation impact while maintaining good clustering quality.

KEYWORDS

Clustering; affinity propagation; automatic aggregation enhanced; mutually exclusive exemplars; constraint

1 Introduction

In the field of unsupervised learning, cluster analysis is a crucial technology. It aims at separating elements into distinct categories based on certain similarity assessment criteria and result evaluation indexes. Elements with high similarity are commonly classified into the same cluster, whereas elements in different clusters have lower similarity. With the fast development of information technology, clustering is required to address realistic challenges in numerous technical fields, including image segmentation, text mining, network analysis, target recognition, trajectory analysis, and gene analysis, which in turn enhance the continuous development of clustering approaches [1–4].

The existing clustering approaches can be generally divided into subsequent categories. The partitioning-based approach usually assumes that the data set must be classified into K clusters, and iteratively searches for the optimal centers and data partitioning through certain criteria. The benefits are comprehensible, but usually, there is a dearth of prior knowledge about the number of clusters. The representative approaches are K-means [5], K-medoids [6], and Fuzzy C-means (FCM) [7]. Hierarchical clustering approaches mainly combine data or split clusters according to certain criteria and ultimately represent the findings in a tree structure. Balanced iterative reducing and clustering using hierarchies [8] and clustering using representatives [9] are the representatives. Their most visible advantage is lucid logic, but the drawbacks are that the clustering process is irreversible and requires vast calculations. Density-based clustering approaches mainly consider the density of each data point within a certain range. Data points with superior density are commonly identified as centers, whereas those with very inferior density are considered outliers. These approaches can better adapt to clusters with disparate shapes and do not need to specify the number of clusters in advance. However, their clustering findings are sensitive to parameters, including distance range and density threshold. The representatives are density-based spatial clustering of applications with noise [10], ordering points to identify the clustering structure [11], and clustering by quick search and finding of density peaks (DP) [12]. The basic idea of grid-based clustering approaches is to divide the data space into a grid structure of numerous cells and achieve classification by processing cells [13]. The benefits are fast calculation and good data adaptability. The drawbacks are that the data structure is ignored and the clustering findings are also influenced by the cell shape and size. Statistical information grid [14] and clustering in quest [15] are well-known grid-based clustering approaches. In addition to the four types of clustering approaches, some approaches based on specific mathematical models have also been widely used, including Gaussian Mixture Model-based clustering [16], spectral clustering (SC) based on graph theory [17], and affinity propagation (AP) clustering based on message passing mechanism [18].

The AP clustering in which we are interested comes from the idea of belief propagation [19–21], and it is currently an extremely popular and potent exemplar-based clustering approach. To maximize total similarity, AP locates a set of exemplars and establishes corresponding relationships between exemplars and other data points. The message-passing mechanism is adopted to address the optimization challenge, which has been expressed in a more intuitive binary graphical model [22,23]. First, AP considers all data points as potential exemplars. The messages designated responsibility and availability are then transmitted back and forth between the data under the objective function and limitations. Finally, the exemplars are chosen and each data point finds its most suitable exemplar. AP offers several advantages, including simple initialization, the absence of a requirement to specify the cluster number, superior clustering quality, and high computational efficiency. AP has been widely employed in manifold aspects such as face recognition [24,25], document clustering [26,27], neural network classifier [28], image analysis [29,30], grid system data clustering [31,32], small cell networks working analysis [33], manufacturing process analysis [34], bearing fault diagnosis [35–37], K-Nearest Neighbor (KNN) positioning [38], psychological research [39], radio environment map analysis [40], building evaluation [41], building materials analysis [42], interference management [43], genome sequences analysis [44], map generalization [45], signal recognizing [46], vehicle counting [47], indoor positioning [48], android malware analysis [49], marine water quality monitoring [50], and groundwater management [51].

Classic AP also possesses some drawbacks including the sensitivity of preference selection. Generally, the median of all data similarity values is considered the preference value. Increasing this value leads to more exemplars, which represent more clusters. However, reducing the value leads to a decrease in the number of clusters. Furthermore, AP has superior performance on data sets with a regular distribution such as a spherical shape, but it is challenging to achieve good results in the tasks on data sets with a nonspherical distribution. Therefore, numerous studies have also concentrated on enhancing the AP clustering theory. The *K*-AP algorithm [52] has customizability in terms of cluster number. It introduces a constraint of *K* exemplars to make the clustering ultimately converge to *K* clusters. The rapid affinity propagation algorithm [53] enhances the clustering speed and quality, which separates the clustering process into coarsening phase, exemplar-clustering phase, and refining phase. The multi-exemplar affinity propagation algorithm [54] expands the single-exemplar model to a multi-exemplar one. It proposes the concept of super exemplar and addresses the multi-subclass challenge. The stability-based affinity propagation algorithm [55] concentrates on addressing the preference selection challenge. It offers a new clustering stability measure and automatically sets preference values, which can generate stable clustering results. The soft-constraint semi-supervised affinity propagation algorithm [56] adds supervision based on AP clustering and implements soft constraints, which can generate more accurate results. Another rapid affinity propagation algorithm [57] enhances the efficiency by compressing the similarity matrix. The adjustable preference affinity propagation algorithm [58] mainly concentrates on preference selection and parameter sensitivity of AP. The message-passing model is derived under additional preference-adjusting constraints, and it results in automatic preference adjustment and better clustering performance. Through density-adaptive preference estimation, an adaptive density distribution-inspired AP clustering algorithm [59] addresses the challenge of preference selection. Additionally, to address the nonspherical cluster problem, the algorithm uses a similarity measurement strategy based on the nearest neighbor search to describe the data set structure. The adaptive spectral affinity propagation algorithm [60] discusses why AP is unsuitable for nonspherical clusters and proposes a model selection procedure that can adaptively determine the number of clusters.

It is clear from the foregoing theoretical development of AP that the enhanced algorithms mainly concentrate on similarity matrix construction, preference selection, and application of nonspherical data clustering. However, there are relatively few studies on the aggregation ability of AP clustering. Aggregation is the important goal of clustering, and it is patently crucial. The enhancement of aggregation ability represents a reduction in cluster number, which is anticipated for numerous application scenarios. People prefer to obtain results such as sunrise and sunset rather than sunrise with fishing boats, sunrise with more clouds, sunset with faster waves, and peaceful sunset, as mentioned in the example of multi-subclass image clustering in [54]. Multiple clusters are undoubtedly significant since they offer more detailed classification and richer cluster information. We just want to study that reducing the cluster number is equally valuable as it can offer more general and comprehensive information.

However, numerous investigations have demonstrated that AP clustering cannot independently converge to a relatively small cluster size. As mentioned above, reducing preference values will lead to fewer clusters, although there is no explicit analytical correlation between the preference value and cluster number. By revising preferences and similarity matrices based on analyzing the data set structure or data point density, some enhanced algorithms can also reduce the number of clusters to some extent. However, we should recognize that the revision has certain subjectivity, as it indicates that we already hope some specific data points will become the final exemplars. Additionally, we cannot guarantee the accuracy of the structure analysis. The difficult tasks are how to set preference values for these exemplars and how to define the function to compute the similarity between exemplars and other data points, assuming that the analysis is accurate enough to exclude potential exemplars.

We concentrate on making the AP cluster show stronger aggregation while ensuring clustering quality. It can be imagined that the aggregation involves merging clusters, but classic AP clustering does not know which clusters can be merged; therefore, it needs a reliable information source to tell it. We anticipate that the information that can enhance aggregation is automatically generated and objective, without human involvement. An automatic aggregation enhanced affinity propagation (AAEAP) clustering algorithm based on mutually exclusive exemplar processing is proposed based on the foregoing considerations. The general idea of the AAEAP is as follows.

First, we select a dependable partitioning clustering approach such as FCM clustering, and allow it to generate M clusters when the AP clustering stabilizes and converges to M clusters. Fig. 1 shows various possible differences between AP and FCM clustering findings when the number of clusters is the same. In Fig. 1a, both AP and FCM generate two clusters, but the exemplars are disparate, leading to substantial differences in the classification of other data points. The exemplars of AP are just ordinary points in the two clusters of FCM. Fortunately, neither cluster of FCM contains both exemplars of AP. In Fig. 1b, the classification findings of AP and FCM are very similar and only a few data points demonstrate differences in selecting the cluster. These situations are understandable and tolerable. However, in Fig. 1c, there are substantial differences in the clustering findings between AP and FCM. Particularly, the blue cluster generated by FCM contains the blue and green exemplars of AP, and we consider it intolerable. Thus, these AP exemplars contained in the same FCM cluster are determined to be mutually exclusive. It is essential to modify the number of AP clusters, and these mutually exclusive exemplars should not exist as exemplars at the same time.

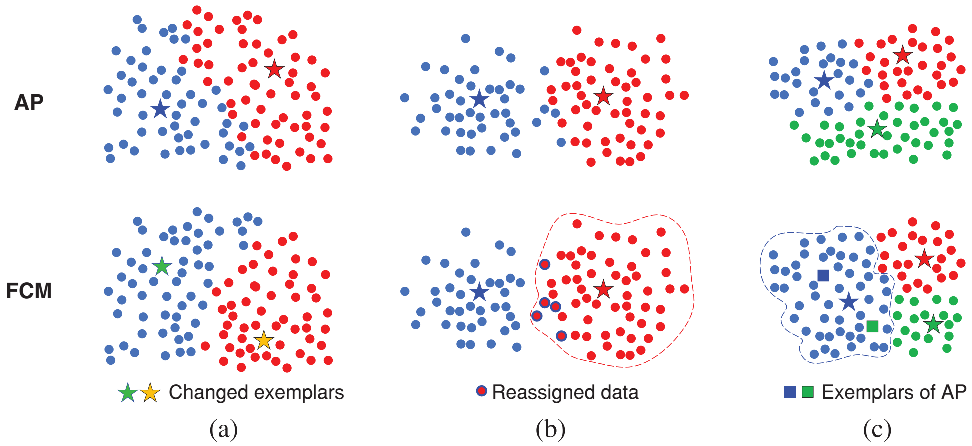

Figure 1: Possible changes in the findings of various clustering approaches. (a) Changes in exemplars. (b) Changes in data assignment. (c) Two exemplars classified in one cluster

By adding a novel constraint and altering the message iteration model, we addressed the foregoing challenge. Then, the adjusted clustering converges stably again based on the new model. Until there are no more conflicting situations, mutual exclusion exemplar detection and clustering of the entire clustering process must be repeated. Some standard classification data sets were adopted to examine and validate the proposed AAEAP algorithm, and six clustering assessment indexes were employed to compare the quality of the result between AAEAP and the other eight clustering algorithms.

The key contributions of the proposed work are as follows:

- We propose a method that can improve the aggregation ability of AP clustering. The core is to employ a partitioning clustering algorithm to detect mutually exclusive exemplars and then reliably guide AP to combine clusters.
- The detection information output by the partitioning clustering approach is mapped as a mutual exclusion and aggregation clustering constraint, and the new message iteration model is derived in detail.
- The overall aggregation improved clustering process is automated and does not need manual intervention, nor does it involve potential exemplar selection and preference revision, which enhances the algorithm's applicability.

The remainder of this study is organized as follows. Section 2 provides a brief review of AP. Section 3 introduces the proposed AAEAP algorithm. Section 4 presents the experimental findings on standard classification data sets. Section 5 concludes the study work.

2 Affinity Propagation

AP clustering is an exemplar-based clustering algorithm that simultaneously considers all data points as potential exemplars and exchanges messages between them until a high-quality exemplar set and corresponding cluster emerge. AP was originally derived as an instance of the max-product algorithm in a loopy factor graph [19]. A simplified max-sum (*log*-domain max-product) message update form was obtained by reducing the *n*-ary messages to binary messages [18,22,23], making the message iteration process of the AP clustering clearer and easier to expand.

Given a data set $X = \{x_1, x_2, \cdots, x_N\}$, a set of exemplars and their clusters are generated through AP clustering. The results are expressed using a binary matrix $C = \left\{c_{ij}\right\}_{N\times N}\, i,j \in \{1, \cdots, N\}$, where c_{ij} represents a binary variable, $c_{ij} = 1$, if data point i selects data point j as an exemplar, otherwise $c_{ij} =$ 0. If data point i represents an exemplar, then $c_{ii} = 1$, otherwise $c_{ii} = 0$.

The complete process of AP clustering is as follows: First, a data similarity measurement function is defined to compute the similarity $s(i,j)$ between data points x_i and x_j, which can be a negative Euclidean distance $s(i,j) = -\left\|x_i - x_j\right\|^2$ or user-defined, forming a similarity matrix $S = \{s(i,j)\}_{N\times N}\, i,j \in \{1, \cdots, N\}$. The diagonal elements of the similarity matrix S represent preference parameters, denoted as $p_k = s(k,k), k \in \{1, \cdots, N\}$. Based on the S function, S_{ij} is defined to denote the similarity between data points and their exemplars.

$$S_{ij}\left(c_{ij}\right) = \begin{cases} s(i,j) & if c_{ij} = 1 \\ 0 & otherwise \end{cases} \tag{1}$$

Meanwhile, we provide two basic constraints, I and E. We term I as the 1-of-N constraint, which indicates that each data point can only be assigned to one exemplar. I can be naturally defined as follows:

$$I_i(c_{i1}, \cdots, c_{iN}) = \begin{cases} -\infty & if \sum_{j=1}^{N} c_{ij} \neq 1 \\ 0 & otherwise \end{cases} \tag{2}$$

E represents the exemplar consistency constraint, which indicates that once a data point is selected as an exemplar by another point, it must select itself as an exemplar. E is defined as

$$E_j\left(c_{1j},\cdots,c_{Nj}\right)=\begin{cases}-\infty & if \quad c_{jj}=0 \quad and \quad \exists i\neq j \quad s.t.c_{ij}=1\\ 0 & otherwise\end{cases} \tag{3}$$

Based on Eqs. (1)–(3), the max-sum objective function is expressed as

$$F_{ms}=\sum_{i=1}^{N}\sum_{j=1}^{N}S_{ij}\left(c_{ij}\right)+\sum_{i=1}^{N}I_i\left(c_{i1},\cdots,c_{iN}\right)+\sum_{j=1}^{N}E_j\left(c_{1j},\cdots,c_{Nj}\right) \tag{4}$$

As previously mentioned, the max-sum is the representation of the *log*-domain max-product, and the 1-N constraint and the exemplar consistency constraint in the max-product model are changed as

$$I_i\left(c_{i1},\cdots,c_{iN}\right)=\begin{cases}0 & if \sum_{j=1}^{N}c_{ij}\neq 1\\ 1 & otherwise\end{cases} \tag{5}$$

$$E_j\left(c_{1j},\cdots,c_{Nj}\right)=\begin{cases}0 & if \ c_{jj}=0 \quad and \quad \exists i\neq j \quad s.t.c_{ij}=1\\ 1 & otherwise\end{cases} \tag{6}$$

The max-product objective function is expressed as

$$F_{mp}=\prod_{i=1}^{N}\prod_{j=1}^{N}e^{S_{ij}\left(c_{ij}\right)}\cdot\prod_{i=1}^{N}I_i\left(c_{i1},\cdots,c_{iN}\right)\cdot\prod_{j=1}^{N}E_j\left(c_{1j},\cdots,c_{Nj}\right) \tag{7}$$

The goal of AP clustering is to maximize F by finding a set of exemplars and corresponding data partitions. The process of finding high-quality exemplars is accomplished through the recursive transmission of messages. Fig. 2 shows the transmission mechanisms for the two types of messages, $r(i,j)$ and $a(i,j)$. $r(i,j)$ is referred to as responsibility, which is a message sent by data point i to candidate exemplar j, reflecting the accumulated evidence for how well-suited data point j is to serve as the exemplar for i. $a(i,j)$ is referred to as availability, which represents a message sent by candidate exemplar j to data point i, reflecting the accumulated evidence for how suitable it would be for data point i to select j as its exemplar. In other words, $r(i,j)$ shows how strongly a data point favors one candidate exemplar over other candidates, and $a(i,j)$ shows to what degree each candidate exemplar is available as a cluster center for one data point.

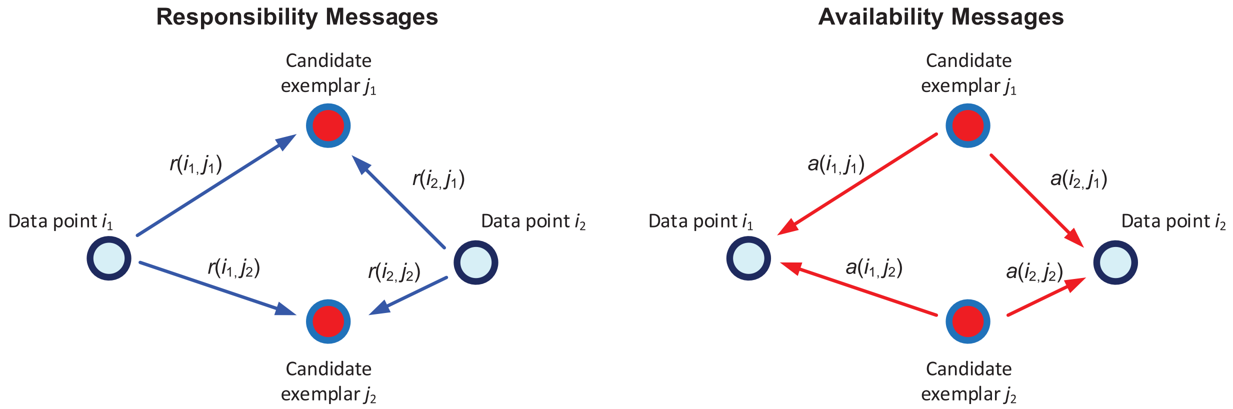

Figure 2: Sending responsibility and availability messages

The messages are initialized as 0 and updated as follows.

$$\begin{aligned} &\forall i,j' r\left(i,j\right) = s\left(i,j\right) - \max_{j' \neq j}\left[s\left(i,j'\right) + a\left(i,j'\right)\right] \\ &\forall i,j' a\left(i,j\right) = \begin{cases} \sum_{i' \neq i} \max\left[0, r\left(i',j\right)\right], & i = j \\ \min\left[0, r\left(j,j\right) + \sum_{i' \neq i,j} \max\left[0, r\left(i',j\right)\right]\right], & i \neq j \end{cases} \end{aligned} \tag{8}$$

It is necessary to increase the damping factor λ, with a value range of [0, 1], to prevent oscillation during the message update process. By integrating the message r_t and a_t of the t iteration with the message $r_{t'}, a_{t'}$ based on r_t and a_t, the message r_{t+1} and a_{t+1} of the $t + 1$ iteration is obtained.

$$\begin{aligned} &r_{t+1}\left(i,j\right) = \left(1 - \lambda\right) \cdot r_{t'}\left(i,j\right) + \lambda \cdot r_t\left(i,j\right) \\ &a_{t+1}\left(i,j\right) = \left(1 - \lambda\right) \cdot a_{t'}\left(i,j\right) + \lambda \cdot a_t\left(i,j\right) \end{aligned} \tag{9}$$

Responsibility and availability update until convergence. AP ultimately outputs an assignment vector $\mathrm{c} = [c_1, \cdots, c_N]$ and $c_i = \mathrm{argmax}_j\left[r\left(i,j\right) + a\left(i,j\right)\right]$.

3 Automatic Aggregation Enhanced Affinity Propagation

We propose an AAEAP clustering algorithm to overcome the difficulty of AP convergence to a small exemplar size. First, the entire framework of the algorithm is introduced. Then, the basic model is offered, including constraints, objective function, and factor graph. The message iteration is finally derived.

3.1 Overall Framework

The distinctive feature of AAEAP is that it uses a dependable partitioning clustering approach to detect whether there is a mutual exclusion situation in the exemplars produced by AP clustering. If there is, by adding a new constraint, the two mutually exclusive exemplars will not become exemplars and the number of clusters decreases. This will lead to the merging of clusters and present an aggregative state. The algorithm converges to a smaller exemplar size when there are no longer mutually exclusive exemplars. The basic framework of AAEAP is as follows.

The input of the algorithm is an unclassified data set $X = \{x_1, \cdots, x_N\}$, and each data x_i is a one-dimensional sequence having d features. Based on the negative Euclidean distance given by the classic AP, the similarity between two sequences is calculated, and ultimately the similarity matrix S is obtained. The computation approach of the similarity matrix S can be enhanced based on [53,59,60] to make AP more adaptable for data sets with nonspherical structures, although this is not the concentration of this research. The required parameters for AP clustering are initialized, including the maximum iteration number NI_{max}, stable convergence number NI_{cvg}, damping factor λ, and similarity matrix S. It is also required to clarify constraints I and E, which determine the way messages are iterated.

Then, the algorithm enters the message iteration. The algorithm will complete the first convergence based on constraints I and E, and generate a set of XEs containing M exemplars due to the null initialization of the mutually exclusive exemplars MEs. A dependable partitioning clustering approach including K-means, K-medoids, or FCM is used to detect mutually exclusive exemplars in XEs. Particularly, by employing the partitioning clustering approach to generate M clusters simultaneously, each cluster is checked to determine whether there are two or more AP exemplars. If so, the included exemplars are considered mutually exclusive. There may be situations where numerous pairs of mutually exclusive exemplars are detected. Only one pair is randomly selected for each iteration to simplify the computation. The foregoing process of partitioning clustering and detecting is processed through the function *IdentifyExclusion*, which assigns a pair of mutually exclusive exemplars to *MEs*.

AAEAP:

```
Input the unclassified data set $X = \{x_1, \cdots, x_N\}$;
Calculate similarity matrix $S = \{s(i,j)\}$;
Assign the initial values of $NI_{\max}, NI_{cvg}, \lambda, S$ and set the $I, E$ constraints;
Initialize mutual exclusion exemplars $MEs = \phi$;
while $iter < NI_{\max}$ do
  if $MEs = \phi$
    Update $R = \{r(i,j)\}$ and $A = \{a(i,j)\}$ under $I, E$ until convergence;
  else
    Define the mutual exclusion and aggregation constraint $D$;
    Update $R$ and $A$ under $I, E, D$ until convergence;
  Generate $M$ exemplars $XEs = \{idx_1, \cdots, idx_M\}$;
  Identify mutual exclusion exemplars $MEs = IdentifyExclusion(X, XEs)$;
  if $MEs = \phi$
    break;
end while
Output the assignment vector $\mathrm{c} = [c_1, \cdots, c_N]$;
Evaluate the clustering results;
```

We must define a new constraint D based on *MEs*, which we call the mutual exclusion and aggregation constraint because of the existence of mutually exclusive exemplars. Its primary capability is to prevent mutually exclusive exemplars from becoming exemplars simultaneously again and reduce the overall number of exemplars by 1. A detailed description will be offered in the next section. The algorithm begins the second message iteration under the constraints of I, E, and D. The expected aggregation impact will occur when stable convergence is achieved. Afterward, the previous steps will be repeated, including detecting the existence of mutually exclusive exemplars, updating constraint conditions D, and restarting message iteration. The algorithm ends until there are no mutually exclusive exemplars left. The output of the AAEAP is the assignment vector $\mathrm{c} = [c_1, \cdots, c_N]$, which is the same as the classic AP, and $c_i = \mathrm{argmax}_j [r(i,j) + a(i,j)]$. Finally, the clustering results should be examined to confirm the effectiveness of the algorithm.

3.2 Basic Model

Fig. 3 shows the factor graph of the AAEAP algorithm. The similarity function S and the three constraints I, E, and D together influence the variable nodes. The similarity function S only influences each variable node in the graph separately, constraint I influences the rows in the graph, constraint E influences the columns, and constraint D affects the diagonal.

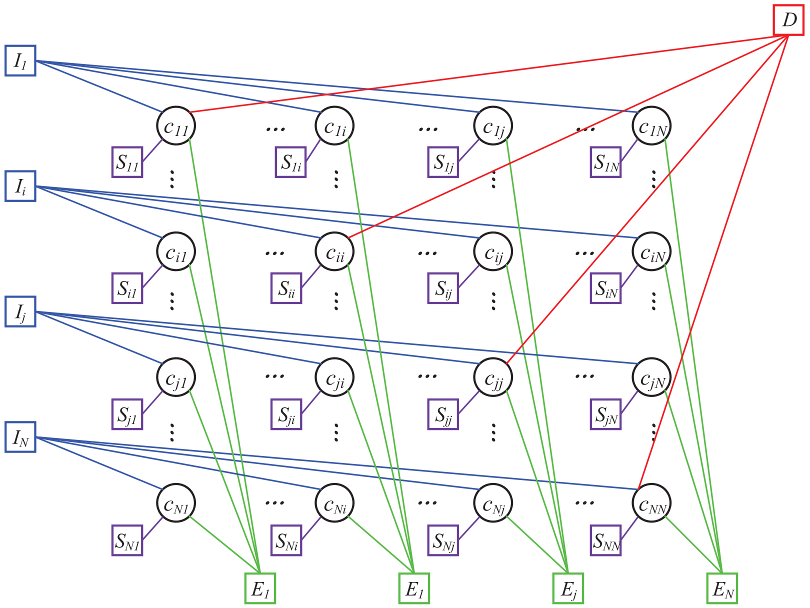

Figure 3: Factor graph of the AAEAP

We present the model of the AAEAP algorithm in the max-product form. First, the constraints of AAEAP are given in detail. The 1-N constraint I and the exemplar consistency constraint E are the same as those of the classic AP. They are defined as

$$I_i\left(c_{i1},\cdots,c_{iN}\right)=\begin{cases}0 & if\ \sum_{j=1}^{N}c_{ij}\neq 1\\ 1 & otherwise\end{cases}$$

$$E_j\left(c_{1j},\cdots,c_{Nj}\right)=\begin{cases}0 & if\quad c_{jj}=0\quad and\quad \exists i\neq\quad js.t.c_{ij}=1\\ 1 & otherwise\end{cases} \tag{10}$$

The following concentrates on describing the newly added mutual exclusion and aggregation constraint D. Constraint D has two functions, one of which is to prevent mutually exclusive exemplars from becoming exemplars the next time, that is, they cannot coexist. The next clustering exemplar set XEs will either only have the exemplar p, only exemplar q, or neither exemplars p nor q, assuming that p and q are mutually exclusive exemplars detected and recommended by partitioning clustering. The second function is to cause clustering to aggregation. Exemplars p and q serve as the core and representative of their clusters, reflecting the basic characteristics of the clusters. Crucially, it shows that there are substantial differences in the clusters generated around p and q. Therefore, mutual exclusion can be seen as a problem to the AP clustering findings, showing that some data points of the clusters represented by p and q can be combined, while the remaining data points may select other exemplars, leading to a decrease in 1 in the exemplar number.

According to the above considerations, constraint D is defined as

$$D\left(c_{11},\cdots,c_{NN}\right)=\begin{cases}0 & if\quad \sum_{i=1}^{N}c_{ii}\neq M-1\quad or\quad c_{pp}\cdot c_{qq}=1\quad p,q\in\{1,\cdots,N\}\\ 1 & otherwise\end{cases} \tag{11}$$

M denotes the exemplar number generated by the previous iteration, p and q are two exemplars that satisfy $c_{pp} = 1$ and $c_{qq} = 1$ in the above equation. However, c_{pp} and c_{qq} cannot both be 1, and the exemplar number for the next iteration is constrained to M-1 when they are identified as mutually exclusive exemplars. Constraint D is dynamically changing, as there are three variables p, q, and M, which need to be defined before each iteration to determine the current messaging model.

Then, the max-product objective function defined according to constraints I, E, and D is

$$F_{mp} = \prod_{i=1}^{N}\prod_{j=1}^{N} e^{S_{ij}(c_{ij})} \cdot \prod_{i=1}^{N} I_i\left(c_{i1}, \cdots, c_{iN}\right) \cdot \prod_{j=1}^{N} E_j\left(c_{1j}, \cdots, c_{Nj}\right) \cdot D\left(c_{11}, \cdots, c_{NN}\right) \tag{12}$$

3.3 Message Iteration

Fig. 4 illustrates the message iteration of AAEAP. As shown in Fig. 4a, there are eight types of messages related to diagonal nodes. As illustrated in Fig. 4b, there are six types of messages related to other nodes.

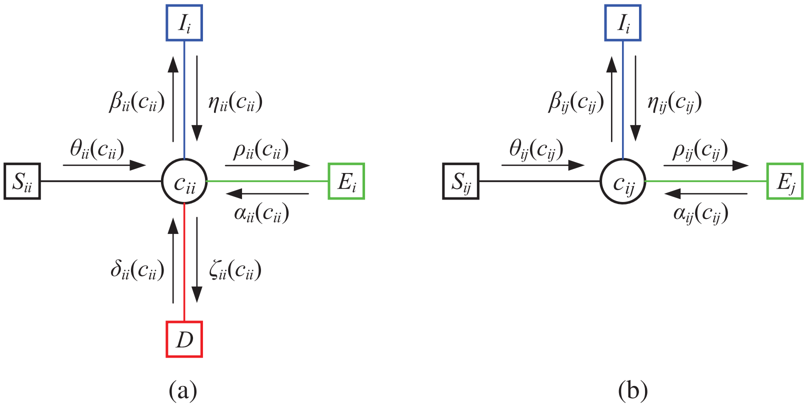

Figure 4: Messages of the AAEAP. (a) Messages associated with c_{ii}. (b) Messages associated with c_{ij}

Based on the max-product algorithm in the factor graph described in [23,52], the message representation from variable node x_i to function node f_m is

$$\mu_{x_i \to fm}\left(x_i\right) = \prod\nolimits_{fn \in Ne\left(x_i\right) \backslash fm} \mu_{fn \to x_i}\left(x_i\right) \tag{13}$$

where $Ne\left(x_i\right) \backslash f_m$ represents the set of functions related to the variable x_i excluding the function f_m, which can be considered as the neighborhood of f_m.

The message from function node f_m to variable node x_i is expressed as

$$\mu_{fm \to x_i}\left(x_i\right) = \max\left[f_m\left(X_{Ne(fm)}\right) \cdot \prod_{x_j \in Ne(fm) \backslash x_i} \mu_{x_j \to fm}\left(x_j\right)\right] \tag{14}$$

where $Ne\left(f_m\right)$ represents the set of all variable nodes related to the f_m function, while $Ne\left(f_m\right) \backslash x_i$ does not include variable x_i.

Based on Eq. (13), the messages sent by the variable nodes to the constraint functions in Fig. 4 are

$$\begin{cases} \beta_{ii}\left(c_{ii}\right) = \mu_{c_{ii}\to I_i}\left(c_{ii}\right) = \delta_{ii}\left(c_{ii}\right)\cdot\theta_{ii}\left(c_{ii}\right)\cdot\alpha_{ii}\left(c_{ii}\right) \\ \beta_{ij}\left(c_{ij}\right) = \mu_{c_{ij}\to I_i}\left(c_{ij}\right) = \theta_{ij}\left(c_{ij}\right)\cdot\alpha_{ij}\left(c_{ij}\right) \\ \rho_{ii}\left(c_{ii}\right) = \mu_{c_{ii}\to E_i}\left(c_{ii}\right) = \delta_{ii}\left(c_{ii}\right)\cdot\theta_{ii}\left(c_{ii}\right)\cdot\eta_{ii}\left(c_{ii}\right) \\ \rho_{ij}\left(c_{ij}\right) = \mu_{c_{ij}\to E_j}\left(c_{ij}\right) = \theta_{ij}\left(c_{ij}\right)\cdot\eta_{ij}\left(c_{ij}\right) \\ \zeta_{ii}\left(c_{ii}\right) = \mu_{c_{ii}\to D}\left(c_{ii}\right) = \theta_{ii}\left(c_{ii}\right)\cdot\alpha_{ii}\left(c_{ii}\right)\cdot\eta_{ii}\left(c_{ii}\right) \end{cases} \tag{15}$$

Based on Eq. (14), the messages θ sent by the similarity function S to the variable nodes are

$$\begin{cases} \theta_{ii}\left(c_{ii}\right) = \mu_{S_{ii}\to c_{ii}}\left(c_{ii}\right) = S_{ii}\left(c_{ii}\right) \\ \theta_{ij}\left(c_{ij}\right) = \mu_{S_{ij}\to c_{ij}}\left(c_{ij}\right) = S_{ij}\left(c_{ij}\right) \end{cases} \tag{16}$$

The messages η sent to the variable nodes by the I constraint function are

$$\begin{cases} \eta_{ii}\left(c_{ii}\right) = \mu_{I_i\to c_{ii}}\left(c_{ii}\right) = \max\left[I_i\left(c_{i1},\cdots,c_{ii},\cdots,c_{iN}\right)\cdot\prod_{k\neq i}\beta_{ik}\left(c_{ik}\right)\right] \\ \eta_{ij}\left(c_{ij}\right) = \mu_{I_i\to c_{ij}}\left(c_{ij}\right) = \max\left[I_i\left(c_{i1},\cdots,c_{ij},\cdots,c_{iN}\right)\cdot\prod_{k\neq j}\beta_{ik}\left(c_{ik}\right)\right] \end{cases} \tag{17}$$

The messages α sent by the E constraint function to the variable node are

$$\begin{cases} \alpha_{ii}\left(c_{ii}\right) = \mu_{E_i\to c_{ii}}\left(c_{ii}\right) = \max\left[E_i\left(c_{1i},\cdots,c_{ii},\cdots,c_{Ni}\right)\cdot\prod_{k\neq i}\rho_{ki}\left(c_{ki}\right)\right] \\ \alpha_{ij}\left(c_{ij}\right) = \mu_{E_j\to c_{ij}}\left(c_{ij}\right) = \max\left[E_j\left(c_{1j},\cdots,c_{ij},\cdots,c_{Nj}\right)\cdot\prod_{k\neq i}\rho_{kj}\left(c_{kj}\right)\right] \end{cases} \tag{18}$$

Additionally, the δ messages sent by the D constraint function to the variable node are expressed as

$$\delta_{ii}\left(c_{ii}\right) = \mu_{D\to c_{ii}}\left(c_{ii}\right) = \max\left[D\left(c_{11},\cdots,c_{ii},\cdots,c_{NN}\right)\cdot\prod_{k\neq i}\zeta_{kk}\left(c_{kk}\right)\right] \tag{19}$$

The above messages are binary messages that can be normalized by a scalar ratio [23,52]: $\beta_{ij}\left(1\right) = \beta_{ij}$ and $\beta_{ij}\left(0\right) = 1$. Then

$$\begin{cases} \eta_{ii} = \dfrac{\eta_{ii}\left(1\right)}{\eta_{ii}\left(0\right)} = \dfrac{\prod_{k\neq i}\beta_{ik}\left(0\right)}{\max\limits_{j\neq i}\left[\beta_{ij}\left(1\right)\cdot\prod_{k\neq i,j}\beta_{ik}\left(0\right)\right]} = \dfrac{\beta_{ij}\left(0\right)\cdot\prod_{k\neq i,j}\beta_{ik}\left(0\right)}{\max\limits_{j\neq i}\left[\beta_{ij}\left(1\right)\cdot\prod_{k\neq i,j}\beta_{ik}\left(0\right)\right]} = \dfrac{1}{\max\limits_{j\neq i}\beta_{ij}} \\ \eta_{ij} = \dfrac{\eta_{ij}\left(1\right)}{\eta_{ij}\left(0\right)} = \dfrac{\prod_{k\neq j}\beta_{ik}\left(0\right)}{\max\limits_{l\neq j}\left[\beta_{il}\left(1\right)\cdot\prod_{k\neq j,l}\beta_{ik}\left(0\right)\right]} = \dfrac{1}{\max\limits_{l\neq j}\beta_{il}} = \dfrac{1}{\max\left\{\beta_{ii},\max\limits_{l\neq i,j}\beta_{il}\right\}} \end{cases} \tag{20}$$

Similarly, $\rho_{ij}\left(0\right) = 1\rho_{ij}\left(1\right) = \rho_{ij}$ and $\rho_{ij}\left(0\right) = 1$, then

$$\begin{cases} \alpha_{ii} = \dfrac{\alpha_{ii}\left(1\right)}{\alpha_{ii}\left(0\right)} = \dfrac{\max\left[\prod_{k\neq i}\rho_{ki}\left(c_{ki}\right)\right]}{\prod_{k\neq i}\rho_{ki}\left(0\right)} = \prod_{k\neq i}\max\left\{1,\rho_{ki}\right\} \\ \alpha_{ij} = \dfrac{\alpha_{ij}\left(1\right)}{\alpha_{ij}\left(0\right)} = \dfrac{\rho_{jj}\left(1\right)\cdot\max\left[\prod_{k\neq i,j}\rho_{kj}\left(c_{kj}\right)\right]}{\max\left\{\rho_{jj}\left(0\right)\cdot\prod_{k\neq i,j}\rho_{kj}\left(0\right),\rho_{jj}\left(1\right)\cdot\max\left[\prod_{k\neq i,j}\rho_{kj}\left(c_{kj}\right)\right]\right\}} \\ \quad = \min\left\{1,\rho_{jj}\cdot\prod_{k\neq i,j}\max\left\{1,\rho_{kj}\right\}\right\} \end{cases} \tag{21}$$

Deriving δ_{ii} messages is relatively complex, although $\zeta_{ij}\left(1\right) = \zeta_{ij}$ and $\zeta_{ij}\left(0\right) = 1$, it is crucial to consider both the sub-constraints of mutual exclusion and the reduction of the exemplar number. Suppose p and q are two mutually exclusive exemplars, without loss of generality, if $i = p$ and the exemplar number is reduced to M-1, then

$$\begin{cases} \delta_{ii}(1) = \zeta_{qq}(0) \cdot \max\left[\prod_{j\in J_1} \zeta_{jj}(1) \cdot \prod_{k\in K_1} \zeta_{kk}(0)\right] \\ \delta_{ii}(0) = \max\left\{\zeta_{qq}(1) \cdot \max\left[\prod_{j\in J_1} \zeta_{jj}(1) \cdot \prod_{k\in K_1} \zeta_{kk}(0)\right], \zeta_{qq}(0) \cdot \max\left[\prod_{j\in J_2} \zeta_{jj}(1) \cdot \prod_{k\in K_2} \zeta_{kk}(0)\right]\right\} \end{cases} \tag{22}$$

Define $R = \{1, \cdots, p-1, p+1, \cdots, q-1, q+1, \cdots, N\}$. In Eq. (21), $J_1 \subset R$ has M-2 elements, $K_1 \subset R$ has N-M elements, and they satisfy$J_1 \cap K_1 = \varnothing, J_1 \cup K_1 = R$, while $J_2 \subset R$ has M-1 elements, $K_2 \subset R$ has N-M-1 elements, and they satisfy $J_2 \cap K_2 = \varnothing, J_2 \cup K_2 = R$. $\max\left[\prod_{j\in J_1} \zeta_{jj}(1) \cdot \prod_{k\in K_1} \zeta_{kk}(0)\right]$ is the selection of M-2 nodes with a value of 1 and the remaining N-M nodes with a value of 0 from N-2 variable nodes excluding c_{pp} and c_{qq} in the factor graph. This selection scheme can maximize the continuous product of $\zeta_{jj}(1)$ messages and $\zeta_{kk}(0)$ messages. Furthermore, it can be obtained that

$$\begin{aligned} \delta_{ii} = \frac{\delta_{ii}(1)}{\delta_{ii}(0)} &= \min\left\{\frac{1}{\zeta_{qq}}, \frac{\max\left[\prod_{j\in J_1} \zeta_{jj}(1) \cdot \prod_{k\in K_1} \zeta_{kk}(0)\right]}{\max\left[\prod_{j\in J_2} \zeta_{jj}(1) \cdot \prod_{k\in K_2} \zeta_{kk}(0)\right]}\right\} \\ &= \min\left\{\frac{1}{\zeta_{qq}}, \frac{\max\left[\prod_{j\in J_1} \zeta_{jj}(1) \cdot \prod_{k\in K_1} \zeta_{kk}(0)\right] \Big/ \prod_{r\in R} \zeta_{rr}(0)}{\max\left[\prod_{j\in J_2} \zeta_{jj}(1) \cdot \prod_{k\in K_2} \zeta_{kk}(0)\right] \Big/ \prod_{r\in R} \zeta_{rr}(0)}\right\} \\ &= \min\left\{\frac{1}{\zeta_{qq}}, \frac{\prod_{r=1}^{M-2} \phi_r}{\prod_{r=1}^{M-1} \phi_r}\right\} = \min\left\{\frac{1}{\zeta_{qq}}, \frac{1}{\phi_{M-1}}\right\} \end{aligned} \tag{23}$$

where Φ represents the ζ message set of $\zeta_{jj}, j \in R$ messages arranged in descending order, Φ_1 represents the maximum value in Φ, and $\Phi_{M\text{-}1}$ is the M-1 th largest value in Φ.

And if $i \notin \{p, q\}$, then

$$\begin{cases} \delta_{ii}(1) = \max\left[\prod_{j\in J_3} \zeta_{jj}(1) \cdot \prod_{k\in K_3} \zeta_{kk}(0)\right] \\ \delta_{ii}(0) = \max\left[\prod_{j\in J_4} \zeta_{jj}(1) \cdot \prod_{k\in K_4} \zeta_{kk}(0)\right] \end{cases} \tag{24}$$

Define $R' = \{1, \cdots, i-1, i+1, \cdots, N\}$. In Eq. (23), $J_3 \subset R', \{p, q\} \not\subset J_3$ has M-2 elements, $K_3 \subset R'$ has N-M elements and they satisfy $J_3 \cap K_3 = \varnothing, J_3 \cup K_3 = R'$. Meanwhile, $J_4 \subset R', \{p, q\} \not\subset J_4$ has M-1 elements, $K_4 \subset R'$ has N-M-1 elements and they satisfy $J_4 \cap K_4 = \varnothing, J_4 \cup K_4 = R'$. So

$$\delta_{ii} = \frac{\delta_{ii}(1)}{\delta_{ii}(0)} = \frac{\max\left[\prod_{j\in J_3}\zeta_{jj}(1)\cdot\prod_{k\in K_3}\zeta_{kk}(0)\right]\Big/\prod_{r\in R'}\zeta_{rr}(0)}{\max\left[\prod_{j\in J_4}\zeta_{jj}(1)\cdot\prod_{k\in K_4}\zeta_{kk}(0)\right]\Big/\prod_{r\in R'}\zeta_{rr}(0)} = \frac{\prod_{r=1}^{M-2}\gamma_r}{\prod_{r=1}^{M-1}\gamma_r} = \frac{1}{\gamma_{M-1}} \tag{25}$$

where Γ denotes the set of ζ messages arranged in descending order of $\zeta_{jj}, j \in R'$ messages. Since p and q are two mutually exclusive exemplars and the min $\left\{\zeta_{pp}, \zeta_{qq}\right\}$ element must be eliminated from Γ, $\Gamma_{M\text{-}1}$ is the M-1 th largest value in Γ.

Through normalization, we have obtained $\eta_{ii}, \eta_{ij}, \alpha_{ii}, \alpha_{ij}, \delta_{ii}$. Because $\theta_{ij} = e^{s(i,j)}$ it is easy to obtain $\beta_{ii} = \delta_{ii}\cdot\theta_{ii}\cdot\alpha_{ii}, \beta_{ij} = \theta_{ij}\cdot\alpha_{ij}$, the expressions for other messages are

$$\zeta_{ii} = \theta_{ii}\cdot\alpha_{ii}\cdot\eta_{ii} = e^{s(i,i)}\cdot\alpha_{ii}\cdot\frac{1}{\max\limits_{j\neq i}\left[e^{s(i,j)}\cdot\alpha_{ij}\right]} \tag{26}$$

$$\begin{cases} \rho_{ii} = \delta_{ii}\cdot\theta_{ii}\cdot\eta_{ii} = \delta_{ii}\cdot e^{s(i,i)}\cdot\dfrac{1}{\max\limits_{j\neq i}\beta_{ij}} = \delta_{ii}\cdot e^{s(i,i)}\cdot\dfrac{1}{\max\limits_{j\neq i}\left[e^{s(i,j)}\cdot\alpha_{ij}\right]} \\ \rho_{ij} = \theta_{ij}\cdot\eta_{ij} = e^{s(i,j)}\cdot\dfrac{1}{\max\limits_{l\neq j}\beta_{il}} = e^{s(i,j)}\cdot\dfrac{1}{\max\left\{\delta_{ii}\cdot e^{s(i,i)}\cdot\alpha_{ii}, \max\limits_{l\neq i,j}\left[e^{s(i,l)}\cdot\alpha_{il}\right]\right\}} \end{cases} \tag{27}$$

Responsibility messages are expressed as $r(i,j) = \log\rho_{ij}$ and availability messages are expressed as $a(i,j) = \log\alpha_{ij}$ [19–23,52]. As a reference, mutual exclusion and aggregation messages are expressed as $u(i) = \log\delta_{ii}, v(i) = \log\zeta_{ii}$. They are initialized as 0 and updated, respectively, as follows.

$$\begin{cases} r(i,i) = u(i) + s(i,i) - \max\limits_{j\neq i}\left[s(i,j) + \alpha(i,j)\right] \\ r(i,j) = s(i,j) - \max\left\{u(i) + s(i,i) + \alpha(i,i), \max\limits_{l\neq i,j}\left[s(i,l) + \alpha(i,l)\right]\right\} \\ a(i,i) = \sum_{k\neq i}\max\{0, r(k,i)\} \\ a(i,j) = \min\left\{0, r(j,j) + \sum_{k\neq i,j}\max\{0, r(k,j)\}\right\} \\ v(i) = s(i,i) + a(i,i) - \max\limits_{j\neq i}\left[s(i,j) + a(i,j)\right] \\ u(i) = \begin{cases} -\max\left\{v(q), V'(M-1)\right\}, i \in \{p,q\}, i = p \\ -V''(M-1), i \notin \{p,q\} \end{cases} \end{cases} \tag{28}$$

where V' represents the set of $v(j), j \in R$ arranged in descending order, and $V'(M-1)$ is the M-1 th largest value in V'. Similarly, V'' represents the set of $v(j), j \in R'$ arranged in descending order, and $V''(M-1)$ represents the M-1 th largest value in V''.

4 Results and Discussion

The experiments were conducted on some standard classification data sets, and the clustering findings were evaluated using internal and external clustering efficiency evaluation indexes to verify the aggregation and accuracy of AAEAP. The clustering algorithms for comparison include classic AP, five enhanced AP algorithms, SC, and DP.

4.1 Experimental Setting

The data sets for our experiments are from the UC Irvine Machine Learning Repository [61], Knowledge Extraction based on Evolutionary Learning (KEEL) [62], and S-sets [63]. Table 1 presents the brief information on these data sets.

Table 1: Characteristics of the data sets

Data set	Instances	Attributes	Classes	Subject area	Sources
Iris	150	4	3	Life	UCI
Wine	178	13	3	Physical	UCI
Segment	210	19	7	Other	UCI
Glass	214	9	6	Physical	UCI
Yeast	1484	8	2	Life	UCI
Banana	5300	2	2	Life	KEEL
Phoneme	5404	5	2	Life	KEEL
S1	5000	2	15	Synthetic	S-sets

We conduct min-max normalization on each column to ensure that the impact of each attribute on the clustering process and results are balanced before clustering.

$$x'_{ij} = \frac{x_{ij} - \min\limits_{k} x_{kj}}{\max\limits_{k} x_{kj} - \min\limits_{k} x_{kj}} \tag{29}$$

Besides classic AP, SC, and DP, there are also five enhanced AP algorithms for comparison, including K-AP, AP clustering based on cosine similarity (CSAP) [48], adjusted preference AP clustering based on twice the median (TMPAP) [46], adjusted preference AP clustering based on quantile (QPAP) [50], and adaptive density distribution inspired affinity propagation clustering (ADDAP) [59]. K-AP concentrates on customizing the number of clusters, CSAP concentrates on modifying the similarity matrix S, TMPAP and QPAP concentrate on modifying the preferences, while ADDAP measures the similarities based on nearest neighbor searching and modifies the preferences based on density.

Both the clustering algorithms for comparison and the proposed AAEAP algorithm have been edited and implemented in Matlab R2016b. The pertinent settings are as follows: the similarity between data points is measured by Euclidean distance except for CSAP and ADDAP, the preferences of AAEAP, AP, and CSAP are established as the median of the total inter-point similarities, the damping factors of AAEAP, AP, and five enhanced AP algorithms are 0.9, the M parameter of QPAP is considered the value corresponding to the 5th quantile to reduce the number of exemplars, and the partitioning clustering approach necessary for AAEAP to detect mutually exclusive exemplars is FCM clustering. All experiments were conducted on Windows 7, with Intel® Core™ i7-9700, memory size 16 GB.

4.2 Clustering Evaluations

An integral component of the clustering process is the validation of the clustering findings, and numerous indexes have been proposed to quantitatively evaluate the performance of clustering algorithms. Effectiveness evaluation indexes can be widely classified into two categories: internal and external evaluation indexes. The internal evaluation indexes primarily examine and assess the clustering results from aspects, including compactness, separation, and overlap, based on the structural information of the data set. The external evaluation indexes are mainly based on available prior information from the data set, including the cluster labels of all data points. The performance is evaluated by comparing the degree of correspondence between clustering results and external information.

We use six extensive evaluation indexes, including Silhouette Coefficient (Sil) [64], In-Group Proportion (IGP) [65], Rand Index (RI) [66], Adjusted Rand Index (ARI) [66], F-measure (FM) [67], and Normalized Mutual Information (NMI) [54] to analyze the clustering results of the proposed AAEAP algorithm. Internal evaluation indexes are Sil and IGP, whereas external evaluation indexes are RI, ARI, FM, and NMI.

Sil is an evaluation index based on compactness within a cluster and separation between clusters. For the data point i, compute the average distance from it to other data points within the cluster, denoted as $a(i)$. Compute the average distance from it to each other cluster, and use the minimum value denoted as $b(i)$. Then, its silhouette coefficient can be expressed as

$$s(i) = (b(i) - a(i)) / \max\{a(i), b(i)\} \tag{30}$$

The value range of $s(i)$ is $[-1, 1]$. When $s(i)$ approaches 1, it shows that the data point has a high degree of correspondence with the assigned cluster and is distant from other clusters. Furthermore, when $s(i)$ approaches -1, it indicates that the data point is assigned to the wrong cluster. Traditionally, the Sil index of the overall clustering results is the average of $s(i)$ for all data points.

IGP is defined as the proportion of each data point and its nearest neighboring point belonging to the same cluster.

$$IGP\_M = \sum_{u=1}^{nc} IGP(u, X) / nc, IGP(u, X) = \frac{\#\{j \,|\, Class_X(j) = Class_X(j_{1NN}) = u\}}{\#\{j \,|\, Class_X(j) = u\}} \tag{31}$$

where X is the data set, u represents one cluster, j represents a data point in u, j_{1NN} is the nearest neighbor point from j, $Class_X(j) = Class_X(j_{1NN}) = u$ denotes j and j_{1NN} belongs to the same cluster, and # denotes the number of data points that meet the above conditions. Compute $IGP(u, X)$ for all nc clusters, with a larger mean IGP_M showing superior clustering quality.

RI is an external evaluation index that necessitates real classification information C. Assuming K is the clustering results, a denotes the number of data pairs in the same cluster in both C and K, and b denotes the number of data pairs that are not in the same cluster whether in C or K, then RI is expressed as

$$RI = (a + b) / C_n^2 \tag{32}$$

where n represents the number of data points, C_n^2 denotes the number of data pairs that can be generated in the data set. The range of RI values is [0,1]. A larger value indicates that the clustering results are more consistent with the real classification.

ARI is an enhancement of RI, which examines clustering by computing the number of data pairs assigned to the same or different clusters in real labels and clustering findings. Compared to RI, ARI has higher discrimination. ARI is expressed as

$$ARI = 2(ad - bc) / [(a + b)(d + b) + (a + c)(d + c)] \tag{33}$$

where a denotes the number of data pairs that belong to the same cluster in both real labels and clustering findings, b denotes the number of data pairs that belong to the same cluster in real labels but do not in clustering findings, c denotes the number of data pairs that do not belong to the same cluster in real labels but belong to the same cluster in clustering findings, and d denotes the number of data pairs that are not in the same cluster, whether in real labels or clustering results. The range of ARI values is [-1, 1], the larger the value, the better the clustering impact, and ARI equals 1, which signifies that clustering findings are completely consistent with real labels.

FM integrates *precision* and *recalls* to examine the clustering impact, which is expressed as

$$F = \sum_{k=1}^{nc} \frac{n_k}{N} \max_{m} \left(F\left(k, m\right)\right), F\left(k, m\right) = \frac{2 \times precision \times recall}{precision + recall} \tag{34}$$

where $precision = n_k^m / n_k, recall = n_k^m / n_m$. n_k denotes the number of data points in the k cluster of clustering findings, and n_m denotes the number of data points in the m cluster of real classification and n_k^m denotes the number of data points shared by the k and m clusters. The larger the FM index value, the better the clustering impact.

NMI assesses the similarity between the clustering results and the real labels from the perspective of information theory. Assuming U denotes the clustering results containing k clusters, V denotes the real labels containing m clusters, and $MI(U, V)$ is the mutual information between the clustering results and the real labels, then NMI is expressed as

$$NMI = \frac{MI\left(U, V\right)}{F\left(H\left(U\right), H\left(V\right)\right)} = \sum_{c=1}^{k} \sum_{p=1}^{m} n_c^p \log\left(\frac{n \cdot n_c^p}{n_c \cdot n_p}\right) \Big/ \sqrt{\left(\sum_{c=1}^{k} n_c \log\left(\frac{n_c}{n}\right)\right)\left(\sum_{p=1}^{m} n_p \log\left(\frac{n_p}{n}\right)\right)} \tag{35}$$

where F is the geometric mean, n is the number of data points, n_c is the data point number of the c cluster in clustering results, n_p represents the data point number of the p cluster in real labels, and n_c^p represents the number of the intersection of the c and p clusters.

4.3 Experimental Results

To demonstrate the automatic aggregation process of the AAEAP algorithm, we selected Iris and Wine data sets. Since the data dimensions of two data sets are greater than 3, we used the classic t-distributed stochastic neighbor embedding (t-SNE) approach [68] to achieve dimensionality reduction and then visualize the data to show the clustering process and aggregation impact.

Fig. 5 demonstrates the findings of the Iris data set. AAEAP converges based on the classic AP when the cluster number *NClass* is 11. Then, AAEAP iterates eight times to automatically detect mutually exclusive exemplars and aggregates, stably converging to three clusters, consistent with the real category number of the Iris data set.

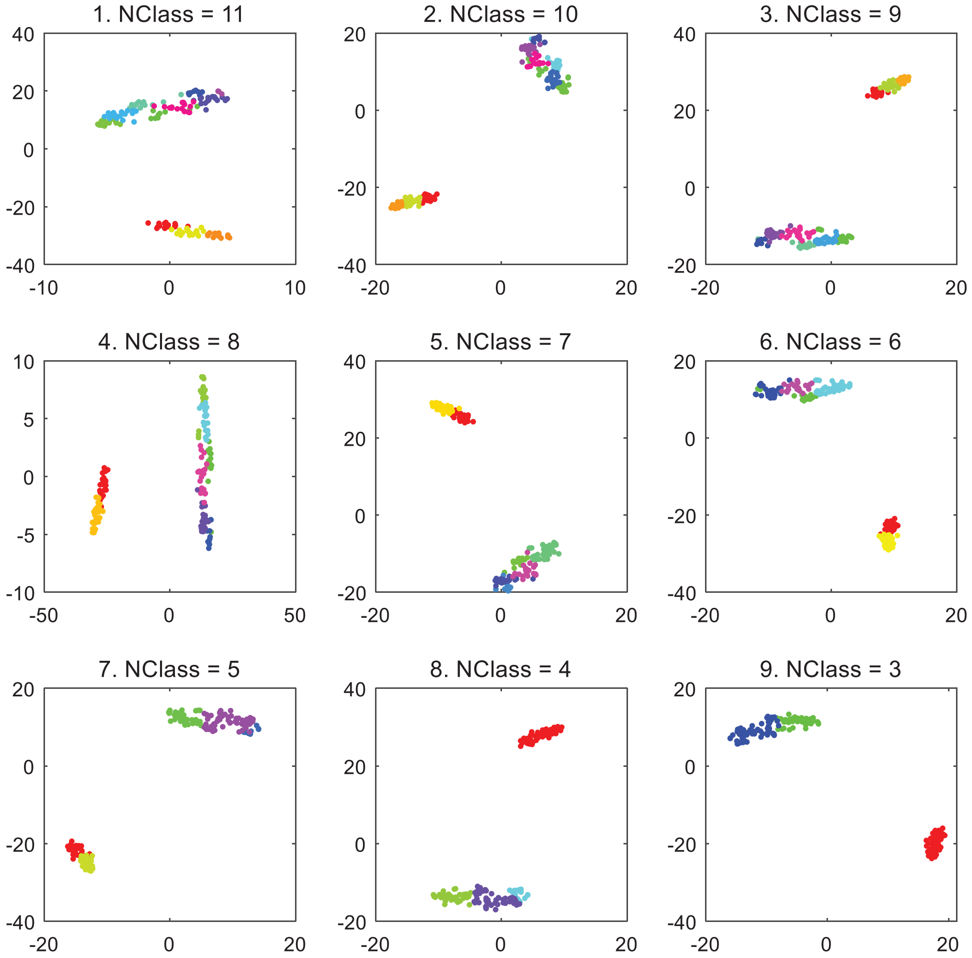

Figure 5: Aggregation effect on Iris data set

Fig. 6 demonstrates the clustering and aggregation impact on the Wine data set. Based on classic AP, AAEAP converges to 21 clusters and then achieves automatic aggregation through mutually exclusive exemplar detection. Finally, with an equivalent number of real categories, AAEAP stably converges into three clusters. However, from the experiments, we discover that AAEAP cannot converge to three clusters every time, and there are also cases of six clusters. Similarly, there are cases of aggregation into three or four clusters for the Yeast data set. The main reason we examined is that AAEAP uses FCM partitioning clustering to detect mutually exclusive exemplars in the experiments, and the random initialization of FCM clustering causes instability in its classification, directly influencing the detection results of mutually exclusive exemplars; therefore, causing changes in AAEAP clustering results. A simple solution is to combine numerous FCM partitioning clustering results to weaken the randomness effect of FCM, and then conduct mutual exclusion detection and ensure the stability and comprehensiveness of the detection. Additionally, it shows the reliability of mutual exclusion detection, which we emphasized earlier. It is easy to imagine that dependable detection will bring accurate aggregation and better clustering quality.

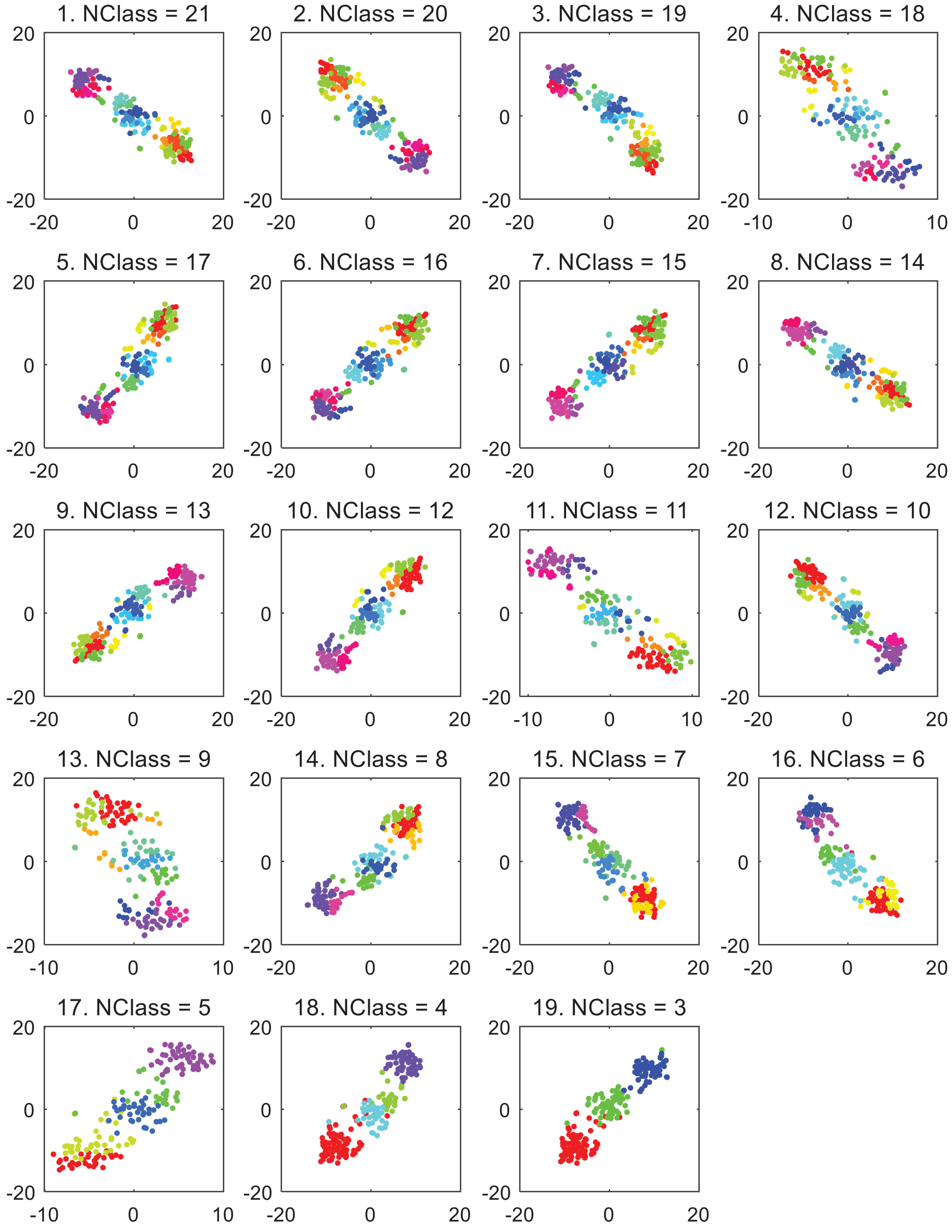

Figure 6: Aggregation effect on Wine data set

Then, to measure the AAEAP clustering quality, we use clustering effectiveness evaluation indexes. Table 2 demonstrates the experimental findings on the Iris, Wine, Yeast, and S1 data sets. AAEAP can converge to the true cluster number, while AP, CSAP, QPAP, and TMPAP cannot. Compared to them, AAEAP has substantial benefits in terms of aggregation performance. ADDAP aims at obtaining the maximum Sil index value, but it cannot obtain the true numbers of clusters for these four data sets. The evaluation indexes of ADDAP are typically superior to those of AP, CSAP, QPAP, and TMPAP. However, AAEAP has better clustering quality than ADDAP, particularly external evaluation indexes, demonstrating that the AAEAP clustering results are closer to the real categories. The evaluation indexes of *K*-AP, SC, and DP are mostly inferior to those of AAEAP for the Iris, Wine, and Yeast data sets, while their indexes for the S1 data set are perfect and slightly superior to those of AAEAP.

Table 2: Clustering results on Iris, Wine, Yeast, and S1 data sets

Data set	Results	AAEAP	AP	CSAP	QPAP	TMPAP	ADDAP	*K*-AP	SC	DP
Iris	NClass	3	11	3	11	8	2	3	3	3
	Sil	0.491	0.325	0.320	0.332	0.330	**0.625**	0.505	0.487	0.450
	IGP	0.953	0.838	0.898	0.843	0.899	0.980	0.954	0.993	**1.000**
	RI	**0.895**	0.754	0.785	0.757	0.790	0.772	0.880	0.809	0.820
	ARI	**0.762**	0.321	0.515	0.331	0.440	0.558	0.730	0.606	0.610
	FM	**0.909**	0.445	0.734	0.454	0.560	0.780	0.895	0.820	0.827
	NMI	**0.770**	0.606	0.593	0.608	0.674	0.725	0.758	0.716	0.706
Wine	NClass	3	22	14	20	8	2	3	3	3
	Sil	**0.289**	0.128	0.104	0.137	0.111	0.276	0.260	0.043	0.224
	IGP	0.928	0.807	0.790	0.809	0.779	0.932	**0.950**	0.909	0.664
	RI	**0.873**	0.708	0.725	0.718	0.766	0.677	0.826	0.541	0.713
	ARI	**0.717**	0.175	0.246	0.213	0.382	0.353	0.616	0.136	0.450
	FM	**0.903**	0.275	0.379	0.320	0.533	0.742	0.860	0.604	0.742
	NMI	**0.731**	0.535	0.569	0.551	0.625	0.453	0.697	0.270	0.599
Yeast	NClass	2	116	65	77	59	3	2	2	2
	Sil	0.235	0.130	0.105	0.130	0.122	0.204	0.248	**0.497**	−0.101
	IGP	0.937	0.663	0.746	0.731	0.729	0.914	0.933	**0.969**	0.500
	RI	0.516	**0.776**	0.774	0.776	0.775	0.611	0.488	0.307	0.224
	ARI	0.119	0.020	0.036	0.031	0.038	**0.129**	0.091	0.013	0.000
	FM	**0.510**	0.088	0.134	0.117	0.140	0.440	0.456	0.466	0.476
	NMI	0.154	**0.289**	0.272	0.285	0.282	0.168	0.131	0.064	0.012
S1	NClass	15	97	16	92	65	13	15	15	15
	Sil	0.711	0.371	0.077	0.369	0.351	0.640	**0.712**	0.711	0.652
	IGP	0.998	0.941	0.998	0.941	0.950	0.999	0.998	0.998	**1.000**
	RI	0.998	0.947	0.932	0.947	0.952	0.980	0.998	**0.999**	0.990
	ARI	0.985	0.315	0.470	0.326	0.416	0.860	0.986	**0.988**	0.920
	FM	0.993	0.322	0.629	0.333	0.420	0.908	0.993	**0.994**	0.946
	NMI	0.986	0.774	0.675	0.778	0.806	0.955	0.986	**0.988**	0.968

Note: For Iris, Wine, Yeast, and S1 data sets, the real number of categories are 3, 3, 2, and 15, respectively.

Table 3 shows the findings on Segment and Glass data sets, where AAEAP does not converge to the real category numbers and exhibits excessive aggregation. Therefore, we also delineate the clustering evaluation findings when *K*-AP, SC, and DP converge to the real category numbers.

Table 3: The clustering results on segment and glass data sets

Data set	Method	NClass	Sil	IGP	RI	ARI	FM	NMI
Segment	AAEAP	6	0.290	0.937	0.870	0.512	**0.685**	0.640
	AP	24	0.349	0.804	0.892	0.394	0.488	0.678
	CSAP	16	**0.351**	0.983	**0.904**	0.521	0.616	0.681
	QPAP	21	0.327	0.880	0.891	0.394	0.495	0.659
	TMPAP	15	0.293	0.902	0.900	0.486	0.589	**0.685**
	ADDAP	4	0.341	0.970	0.797	0.421	0.660	0.648
	K-AP	6	0.290	0.941	0.869	0.508	0.678	0.638
		7	0.309	0.927	0.863	0.472	0.643	0.621
	SC	6	0.009	0.942	0.401	0.012	0.303	0.120
		7	0.118	0.943	0.496	0.064	0.363	0.279
	DP	6	0.281	0.994	0.842	0.461	0.678	0.642
		7	0.297	**0.995**	0.877	**0.525**	0.673	0.672
Glass	AAEAP	5	0.361	0.892	0.669	0.175	0.507	0.327
	AP	26	0.291	0.606	**0.752**	0.158	0.328	**0.426**
	CSAP	16	**0.372**	0.706	0.709	0.182	0.407	0.411
	QPAP	16	0.247	0.775	0.748	0.183	0.388	0.401
	TMPAP	14	0.235	0.742	0.745	0.194	0.417	0.399
	ADDAP	4	0.333	0.921	0.642	0.125	0.459	0.240
	K-AP	5	0.355	0.892	0.661	0.159	0.498	0.300
		6	0.348	0.889	0.663	0.151	0.462	0.304
	SC	5	0.352	**0.979**	0.536	0.144	0.540	0.325
		6	0.322	0.967	0.545	0.130	0.509	0.312
	DP	5	0.138	0.791	0.703	**0.219**	0.564	0.284
		6	0.127	0.795	0.534	0.163	**0.569**	0.286

Note: For Segment and Glass data sets, the real number of categories are 7 and 6.

Although the real category number of the Segment data set is 7, Table 3 shows that when *K*-AP converges into 6 categories, the evaluation indexes are significantly superior to those of 7 categories because of fewer incorrect element classifications, and the entire clustering performance of AAEAP is better than *K*-AP. However, some evaluation indexes of AP, CSAP, QPAP, and TMPAP are superior to those of AAEAP, but their aggregation performances are poor. The excessive aggregation of ADDAP is more serious, while the evaluation index values are generally good, particularly the internal indexes. The evaluation indexes are mostly inferior to those with seven categories when SC and DP converge into six categories. The performance of AAEAP is superior to that of SC and proximate to DP. The real number of categories is six for the Glass data set, and there is also a situation where the evaluation indexes when *K*-AP converges into five categories are typically superior to the evaluation indexes with the real category number. Generally, the performance of AAEAP is better than that of *K*-AP. ADDAP obtains fewer clusters, while its evaluation indexes are often inferior to those of AAEAP. Although AP, CSAP, QPAP, and TMPAP cannot converge into the real number of categories, their evaluation indexes are not bad. The evaluation indexes of SC and DP have their respective benefits, and the index values of AAEAP are often between the two. Fig. 7 demonstrates the entire performance of AAEAP, with no weaknesses in its evaluation indexes. They are substantially superior to the means of indexes obtained by other algorithms and close to the maximum values.

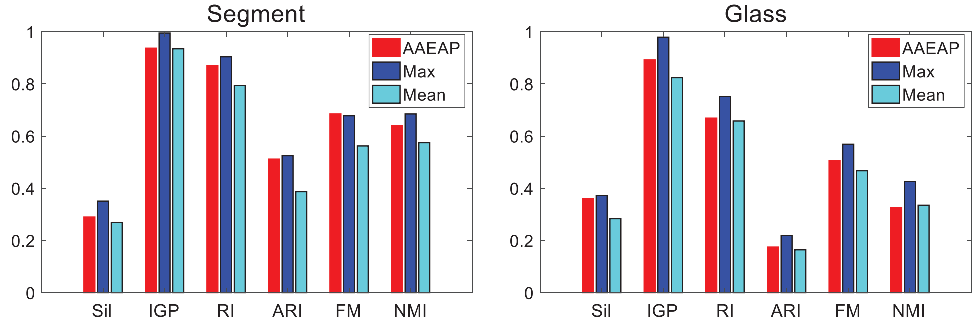

Figure 7: Overall performances of AAEAP on segment and glass data sets

Table 4 shows the findings on Banana and Phoneme data sets, where AAEAP does not converge to the real category numbers and exhibits a lack of aggregation. Therefore, we also delineate the clustering evaluation findings when *K*-AP, SC, and DP converge to the real category numbers.

Table 4: The clustering results on Banana and Phoneme data sets

Data set	Method	NClass	Sil	IGP	RI	ARI	FM	NMI
Banana	AAEAP	5	0.341	0.988	0.530	0.066	0.389	0.078
	AP	176	0.336	0.905	0.499	0.008	0.022	0.253
	CSAP	23	−0.077	0.997	0.502	0.012	0.128	0.054
	QPAP	77	0.130	0.731	**0.776**	0.031	0.117	0.285
	TMPAP	111	0.323	0.925	0.501	0.013	0.033	0.252
	ADDAP	6	0.240	0.989	0.674	**0.352**	0.521	**0.628**
	K-AP	2	**0.391**	0.992	0.515	0.029	0.586	0.025
		5	0.353	0.987	0.526	0.058	0.388	0.388
	SC	2	**0.391**	0.996	0.515	0.030	0.586	0.021
		5	0.365	0.989	0.505	0.017	0.341	0.024
	DP	2	0.082	**1.000**	0.509	0.011	**0.666**	0.010
		5	−0.006	0.995	0.500	0.003	0.435	0.017
Phoneme	AAEAP	6	0.316	0.975	0.487	**0.078**	0.375	0.151
	AP	265	0.249	0.870	0.417	0.003	0.015	0.164
	CSAP	88	0.156	0.904	0.423	0.009	0.047	0.135
	QPAP	264	0.249	0.868	0.417	0.003	0.015	0.165
	TMPAP	145	0.242	0.888	0.419	0.005	0.026	0.148
	ADDAP	4	**0.331**	0.983	0.503	0.075	0.497	0.164
	K-AP	2	0.197	0.984	0.500	−0.052	0.616	0.097
		6	0.307	0.974	0.483	0.070	0.352	0.144
	SC	2	0.248	0.997	0.536	0.059	0.627	**0.204**

(Continued)

Table 4 (continued)

Data set	Method	NClass	Sil	IGP	RI	ARI	FM	NMI
		6	0.278	0.967	0.484	0.057	0.434	0.163
	DP	2	0.074	**1.000**	0.581	0.006	**0.807**	0.001
		6	−0.238	0.990	**0.585**	0.061	0.757	0.041

Note: For Banana and Phoneme data sets, the real number of categories is 2.

The deviation between the numbers of clusters obtained by AP, CSAP, QPAP, TMPAP, and the real category numbers is significant, and the problem of insufficient aggregation of these four algorithms is obvious. AAEAP converges to the numbers of clusters different from the real numbers since FCM does not offer more mutually exclusive exemplars. However, AAEAP still shows strong automatic aggregation ability even for large data sets compared with AP, CSAP, QPAP, and TMPAP. The performance of ADDAP is superior to that of most algorithms since the density of data is considered and the optimal classification is obtained through iteration. The density distribution is distinctly nonuniform for Banana and Phoneme data sets, enabling ADDAP to offer full play to the superiority. The index values of *K*-AP, SC, and DP are mostly inferior to those of AAEAP, regardless of whether they obtain the real category numbers or the same number of clusters as AAEAP. Fig. 8 demonstrates the overall performance of AAEAP. The evaluation indexes of AAEAP are superior to the means of indexes obtained by other algorithms and close to the maximum values.

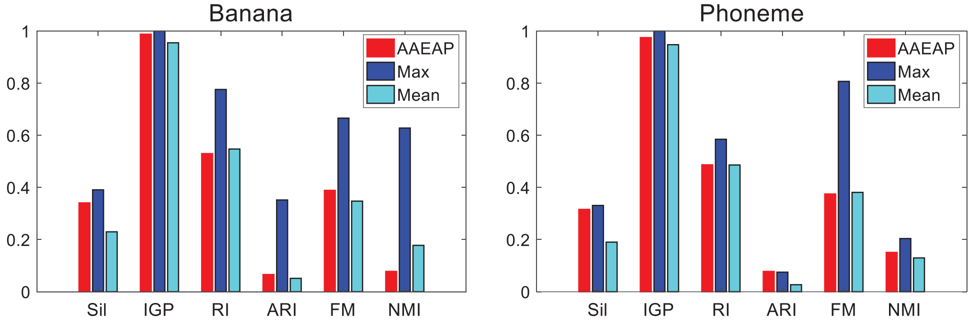

Figure 8: Overall performances of AAEAP on Banana and Phoneme data sets

5 Conclusions

This study proposes an AAEAP clustering algorithm according to mutually exclusive exemplar processing. Its main objective is to enable AP clustering to automatically aggregate, and the information that enhances aggregation comes from real-time partitioning clustering results, rather than prior knowledge or human intervention. This is also the distinction between the AAEAP algorithm and semi-supervised AP clustering algorithms. Potential mutually exclusive exemplar pairs are identified by cross-checking the partitioning and AP clustering findings. Based on them, the current clusters are disassembled and clustering is restarted based on the mutual exclusion and aggregation constraint, achieving aggregation incrementally. From the experimental findings, it can be discerned that the automatic aggregation impact of AAEAP is substantial, and the entire clustering evaluation index values are superior. However, we also discover that the quality of clustering results is related to whether partitioning clustering can offer stable and reliable mutual exclusion detection information. The more accurate the information, the better the AAEAP clustering impact. Future studies will concentrate on enhancing AAEAP to make it more adaptable to cluster on nonspherical data sets.

Acknowledgement: The authors wish to acknowledge the contribution of the Beijing Institute of Tracking and Telemetry Technology, China. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers.

Funding Statement: This research was supported by Research Team Development Funds of L. Xue and Z. H. Ouyang, Electronic Countermeasure Institute, National University of Defense Technology.

Author Contributions: Study conception and design: Z. H. Ouyang, L. Xue; data collection: Y. S. Duan; analysis and interpretation of results: Z. H. Ouyang, F. Ding; draft manuscript preparation: Z. H. Ouyang, Y. S. Duan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data will be made available on request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Xu and D. C. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[2] Z. Dafir, Y. Lamari and S. C. Slaoui, "A survey on parallel clustering algorithms for big data," *Artifcial Intelligence Review*, vol. 54, no. 4, pp. 2411–2443, 2021.

[3] K. Golalipour, E. Akbari, S. S. Hamidi, M. Lee and R. Enayatifar, "From clustering to clustering ensemble selection: A review," *Engineering Applications of Artificial Intelligence*, vol. 104, pp. 104388, 2021.

[4] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka *et al.,* "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Engineering Applications of Artificial Intelligence*, vol. 110, pp. 104743, 2022.

[5] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[6] H. S. Park and C. H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[7] J. C. Bezdek, "Modified objective function algorithms," in *Pattern Recognition with Fuzzy Objective Function Algorithms*. Boston, MA, USA: Springer, pp. 155–201, 1981.

[8] T. Zhang, R. Ramakrishnan and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Record*, vol. 25, no. 2, pp. 103–114, 1996.

[9] S. Guha, R. Rastogi and K. Shim, "CURE: An efficient clustering algorithm for large databases," *ACM SIGMOD Record*, vol. 27, no. 2, pp. 73–84, 1998. https://doi.org/10.1145/276305.276312

[10] M. Ester, H. P. Kriegel, J. Sander and X. W. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland, OR, USA, pp. 226–231, 1996.

[11] M. Ankerst, M. M. Breunig, H. P. Kriegel and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. of the 1999 ACM SIGMOD Int. Conf. on Management of Data*, Philadelphia, PA, USA, pp. 49–60, 1999.

[12] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[13] S. H. Yue, M. M. Wei, J. S. Wang and H. X. Wang, "A general grid-clustering approach," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1372–1384, 2008.

[14] W. Wang, J. Yang and R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proc. of the 23rd Int. Conf. on Very Large Data Bases*, Athens, Greece, pp. 186–195, 1997.

[15] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, "Automatic subspace clustering of high dimensional data," *Data Mining and Knowledge Discovery*, vol. 11, no. 1, pp. 5–33, 2005.

[16] P. D. McNicholas and T. B. Murphy, "Model-based clustering of microarray expression data via latent Gaussian mixture models," *Bioinformatics*, vol. 26, no. 21, pp. 2705–2712, 2010.

[17] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[18] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[19] F. R. Kschischang, B. J. Frey and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[20] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 736–744, 2001.

[21] C. Yanover, T. Meltzer and Y. Weiss, "Linear programming relaxations and belief propagation—An empirical study," *Journal of Machine Learning Research*, vol. 7, pp. 1887–1907, 2006.

[22] I. E. Givoni and B. J. Frey, "A binary variable model for affinity propagation," *Neural Computation*, vol. 21, no. 6, pp. 1589–1600, 2009.

[23] D. Dueck, "Affinity propagation: Clustering data by passing messages," Ph.D. dissertation, University of Toronto, Canada, 2009.

[24] C. H. Du, J. Yang, Q. Wu and T. H. Zhang, "Face recognition using message passing based clustering method," *Journal of Visual Communication and Image Representation*, vol. 20, no. 8, pp. 608–613, 2009.

[25] I. Dagher, S. Mikhael and O. Al-Khalil, "Gabor face clustering using affinity propagation and structural similarity index," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4719–4727, 2021.

[26] Y. He, Q. Chen, X. Wang, R. Xu, X. Bai *et al.,* "An adaptive affinity propagation document clustering," in *Proc. of the 7th Int. Conf. on Informatics and Systems*, Cairo, Egypt, pp. 1–7, 2010.

[27] R. C. Guan, X. H. Shi, M. Marchese, C. Yang and Y. C. Liang, "Text clustering with seeds affinity propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 4, pp. 627–637, 2011.

[28] Z. Q. Zhao, J. Gao, H. Glotin and X. Wu, "A matrix modular neural network based on task decomposition with subspace division by adaptive affinity propagation clustering," *Applied Mathematical Modelling*, vol. 34, no. 12, pp. 3884–3895, 2010.

[29] J. Zhang, X. Tuo, Z. Yuan, W. Liao and H. Chen, "Analysis of fMRI data using an integrated principal component analysis and supervised affinity propagation clustering approach," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 11, pp. 3184–3196, 2011.

[30] H. M. Ge, L. G. Wang, H. Z. Pan, Y. X. Zhu, X. Y. Zhao *et al.,* "Affinity propagation based on structural similarity index and local outlier factor for hyperspectral image clustering," *Remote Sensing*, vol. 14, no. 5, pp. 1995, 2022.

[31] X. L. Zhang, C. Furtlehner, C. Germain and M. Sebag, "Data stream clustering with affinity propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1644–1656, 2014.

[32] Q. H. Gao, Y. W. Wang, X. Z. Cheng, J. G. Yu, X. Chen *et al.,* "Identification of vulnerable lines in smart grid systems based on affinity propagation clustering," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5163–5171, 2019.

[33] J. Zhang, X. Tuo, Z. Yuan, W. Liao and H. Chen, "A practical semi-dynamic clustering scheme using affinity propagation in cooperative picocells," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4372–4377, 2015.

[34] L. Wang, Y. J. Zhang and S. S. Zhong, "Typical process discovery based on affinity propagation," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 10, no. 1, pp. 1–13, 2016.

[35] Z. X. Wei, Y. X. Wang, S. L. He and J. D. Bao, "A novel intelligent method for bearing fault diagnosis based on affinity propagation clustering and adaptive feature selection," *Knowledge-Based Systems*, vol. 116, no. C, pp. 1–12, 2017.

[36] Q. Hu, Q. Zhang, X. S. Si, G. X. Sun and A. S. Qin, "Intelligent fault diagnosis approach based on composite multi-scale dimensionless indicators and affinity propagation clustering," *IEEE Sensors Journal*, vol. 20, no. 19, pp. 11439–11453, 2020.

[37] M. Li, Y. X. Wang, Z. G. Chen and J. Zhao, "Intelligent fault diagnosis for rotating machinery based on potential energy feature and adaptive transfer affinity propagation clustering," *Measurement Science and Technology*, vol. 32, no. 9, pp. 94012, 2021.

[38] J. X. Bi, Y. J. Wang, X. Li, H. X. Qi, H. J. Cao *et al.,* "An adaptive weighted KNN positioning method based on omnidirectional fingerprint database and twice affinity propagation clustering," *Sensors*, vol. 18, no. 8, pp. 2502, 2018.

[39] M. J. Brusco, D. Steinley, J. Stevens and J. D. Cradit, "Affinity propagation: An exemplar-based tool for clustering in psychological research," *British Journal of Mathematical and Statistical Psychology*, vol. 72, no. 1, pp. 155–182, 2019.

[40] H. Y. Xia, S. Zha, J. J. Huang and J. B. Liu, "Radio environment map construction by adaptive ordinary Kriging algorithm based on affinity propagation clustering," *International Journal of Distributed Sensor Networks*, vol. 16, no. 5, pp. 1–10, 2020.

[41] Y. M. Han, C. Y. Fan, Z. Q. Geng, B. Ma, D. Cong *et al.,* "Energy efficient building envelope using novel RBF neural network integrated affinity propagation," *Energy*, vol. 209, pp. 118414, 2020.

[42] S. Sajid, L. Chouinard and N. Carino, "Condition assessment of concrete plates using impulse-response test with affinity propagation and homoscedasticity," *Mechanical Systems and Signal Processing*, vol. 178, pp. 109289, 2022.

[43] L. C. Wang, Y. S. Chao, S. H. Cheng and Z. Han, "An integrated affinity propagation and machine learning approach for interference management in drone base stations," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 83–94, 2020.

[44] A. Busch, T. H. Bachmann, M. Y. Abdel-Glil, A. Hackbart, H. Hotzel *et al.,* "Using affinity propagation clustering for identifying bacterial clades and subclades with whole-genome sequences of Francisella tularensis," *PLoS Neglected Tropical Diseases*, vol. 14, no. 9, pp. e0008018, 2020.

[45] X. F. Yan, H. Chen, H. R. Huang, Q. Liu and M. Yang, "Building typification in map generalization using affinity propagation clustering," *ISPRS International Journal of Geo-Information*, vol. 10, no. 11, pp. 732, 2021.

[46] S. H. Xu, Z. J. Qin, W. T. Zhang, X. M. Xiong, H. Li *et al.,* "A novel method of recognizing disturbance events in Φ-OTDR based on affinity propagation clustering and perturbation signal selection," *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13272–13282, 2021.

[47] P. M. Harikrishnan, A. Thomas, V. P. Gopi, P. Palanisamy and K. A. Wahid, "Inception single shot multi-box detector with affinity propagation clustering and their application in multi-class vehicle counting," *Applied Intelligence*, vol. 51, no. 7, pp. 4714–4729, 2021.

[48] A. Alitaleshi, H. Jazayeriy and J. Kazemitabar, "Affinity propagation clustering-aided two-label hierarchical extreme learning machine for Wi-Fi fingerprinting-based indoor positioning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 6, pp. 3303–3317, 2022.

[49] M. Katebi, A. RezaKhani, S. Joudaki and M. E. Shiri, "RAPSAMS: Robust affinity propagation clustering on static android malware stream," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 15e6980, 2022.

[50] X. Fang, C. S. Luo, D. R. Zhang, H. F. Zhang, J. Qian *et al.,* "Pre-selection of monitoring stations for marine water quality using affinity propagation: A case study of Xincun Lagoon, Hainan, China," *Journal of Environmental Management*, vol. 325, pp. 116666, 2023.

[51] X. Guo, Z. X. Yang, C. Li, H. X. Xiong and C. M. Ma, "Combining the classic vulnerability index and affinity propagation clustering algorithm to assess the intrinsic aquifer vulnerability of coastal aquifers on an integrated scale," *Environmental Research*, vol. 217, pp. 114877, 2023.

[52] X. L. Zhang, W. Wang, K. Nørvåg and M. Sebag, “K-AP: Generating specified K clusters by efficient affinity propagation,” in *Proc. of 2010 IEEE Int. Conf. on Data Mining*, Sydney, NSW, Australia, pp. 1187–1192, 2010.

[53] F. H. Shang, L. C. Jiao, J. R. Shi, F. Wang and M. G. Gong, “Fast affinity propagation clustering: A multilevel approach,” *Pattern Recognition*, vol. 45, no. 1, pp. 474–486, 2012.

[54] C. D. Wang, J. H. Lai, C. Y. Suen and J. Y. Zhu, “Multi-exemplar affinity propagation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2223–2237, 2013.

[55] D. W. Chen, J. Q. Sheng, J. J. Chen and C. D. Wang, “Stability-based preference selection in affinity propagation,” *Neural Computing and Applications*, vol. 25, no. 7, pp. 1809–1822, 2014.

[56] N. M. Arzeno and H. Vikalo, “Semi-supervised affinity propagation with soft instance-level constraints,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 5, pp. 1041–1052, 2015.

[57] L. L. Sun, C. H. Guo, C. R. Liu and H. Xiong, “Fast affinity propagation clustering based on incomplete similarity matrix,” *Knowledge and Information Systems*, vol. 51, no. 3, pp. 941–964, 2017.

[58] P. Li, H. F. Ji, B. L. Wang, Z. Y. Huang and H. Q. Li, “Adjustable preference affinity propagation clustering,” *Pattern Recognition Letters*, vol. 85, pp. 72–78, 2017.

[59] Z. Y. Fan, J. Jiang, S. Q. Weng, Z. H. He and Z. W. Liu, “Adaptive density distribution inspired affinity propagation clustering,” *Neural Computing and Applications*, vol. 31, no. 1, pp. 435–445, 2019.

[60] L. Tang, L. L. Sun, C. H. Guo and Z. Zhang, “Adaptive spectral affinity propagation clustering,” *Journal of Systems Engineering and Electronics*, vol. 33, no. 3, pp. 647–664, 2022.

[61] UCI Datasets, 2007. [Online]. Available: https://archive-beta.ics.uci.edu/datasets

[62] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García *et al.*, “KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework,” *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2–3, pp. 255–287, 2011.

[63] P. Fränti and O. Virmajoki, “Iterative shrinking method for clustering problems,” *Pattern Recognition*, vol. 39, no. 5, pp. 761–775, 2006.

[64] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, no. 1, pp. 53–65, 1987.

[65] A. V. Kapp and R. Tibshirani, “Are clusters found in one dataset present in another dataset?” *Biostatistics*, vol. 8, no. 1, pp. 9–31, 2007.

[66] A. D’Ambrosio, S. Amodio, C. Iorio, G. Pandolfo and R. Siciliano, “Adjusted concordance index: An extensionl of the adjusted rand index to fuzzy partitions,” *Journal of Classification*, vol. 38, no. 1, pp. 112–128, 2021.

[67] L. M. Wang, Q. Ji and X. M. Han, “Adaptive semi-supervised affinity propagation clustering algorithm based on structural similarity,” *Tehnicki Vjesnik*, vol. 23, no. 2, pp. 425–435, 2016.

[68] L. J. P. van der Maaten and G. E. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.