GoT: a Growing Tree Model for Clustering Ensemble

Feijiang Li, Yuhua Qian, Jieting Wang
Institute of Big Data Science and Industry, Shanxi University
feijiangli@email.sxu.edu.cn, yuhuaqian@126.com, jietingwang@email.sxu.edu.cn

Abstract

The clustering ensemble technique that integrates multiple clustering results can improve the accuracy and robustness of the final clustering. In many clustering ensemble algorithms, the co-association matrix (CA matrix), which reflects the frequency of any two samples being partitioned into the same cluster, plays an important role. However, generally, the CA matrix is highly sparse with low value density, which may limit the performance of an algorithm based on it. To handle these issues, in this paper, we propose a growing tree model (GoT). In this model, the CA matrix is firstly refined by the shortest path technique so that its sparsity will be mitigated. Then, a set of representative prototype examples is discovered. Finally, to handle the low value density of the CA matrix, the prototypes gradually connect to their neighborhood, which likes a set of trees growing up. The rationality of the discovered prototype examples is illustrated by theoretical analysis and experimental analysis. The working mechanism of the GoT is visually shown on synthetic data sets. Experimental analyses on eight UCI data sets and eight image data sets show that the GoT outperforms nine representative clustering ensemble algorithms.

Introduction

Data clustering is an interesting unsupervised technique in machine learning (Vega-Pons and Ruiz-Shulcloper 2011; Zhou 2019), which aims to partition a data set into homogeneous groups or clusters based on the similarity between samples. The clustering ensemble technique has drawn much attentions due to its ability of improving clustering effectiveness and robustness. The clustering ensemble technique discovers the group structure of data through combining multiple diverse clustering results without involving the original data set (Strehl and Ghosh 2003). Due to the flexible process, clustering ensemble technique has been widely applied in many challenging tasks, such as high dimensional data clustering (Li et al. 2018), large-scale data clustering (Yu et al. 2019), temporal data clustering (Yang and Chen 2010), image segmentation (Zhang et al. 2008), etc.

Given a set of clustering results, the frequency that two samples appear in the same cluster is used to measure the relation between two samples. All the pairwise frequencies form the co-association matrix (also known as CA matrix). For a set of clustering results $\Pi = \{\pi_1, \pi_2, \ldots, \pi_l\}$ on a data set with n samples $U = \{x_1, x_2, \ldots, x_n\}$, the CA matrix is

$$CA = \{a_{ij}\}_{n\times n}, \tag{1}$$

where

$$a_{ij} = \frac{1}{l}\sum_{b=1}^{l} \mathbb{I}\left(c^b(x_i), c^b(x_j)\right), \tag{2}$$

and

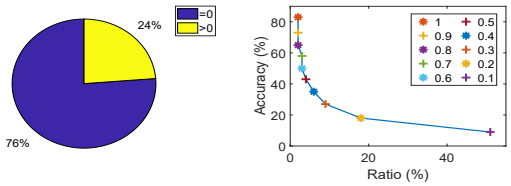
$$\mathbb{I}\left(c^b(x_i), c^b(x_j)\right) = \begin{cases} 1, c^b(x_i) = c^b(x_j) \\ 0, c^b(x_i) \neq c^b(x_j). \end{cases}$$

To obtain a consensus clustering π_*, the CA matrix is utilized by a large number of clustering ensemble techniques. However, some limitations of the CA matrix may affect the performance of clustering ensemble methods based on it. The limitations mainly come from two aspects:

- High Sparsity. The CA matrix is highly sparse, which makes most of the samples' relations can not be reflected.
- Low Value Density. A higher co-association value offers more reliable information, while most of the elements of the CA matrix are small values.

To show the above limitations, we utilize the example in (Huang, Lai, and Wang 2016). They generate 10 base clustering results on the MNIST handwritten digits data set and construct the CA matrix. Firstly, we show the fraction of the zero value in the CA matrix in Figure 1 (a). From Figure 1 (a), it is easy to see that this CA matrix contains a large number of zero values, which indicates a highly sparse matrix. Figure 1 (b) shows the low value density limitation by showing the relations between the ratio and accuracy of co-association values. The accuracy of a co-association value is the proportion of the links with the co-association value that makes a correct decision. As shown by Figure 1 (b), a higher co-association value obtains a higher accuracy value, while a lower co-association value obtains a higher ratio, which means that the CA matrix contains many unreliable information.

The sparsity of the CA matrix means that a number of sample pairs have no relation. As a result, the clusters may



(a) High Sparsity (b) Low Value Density

Figure 1: The distribution of the co-association values of the CA matrix for the MNIST dataset

be broken up, and then the prototype of each cluster is hard to be discovered. The low value density of the CA matrix means that it can not offer sufficient reliable global structure information. Then directly discovering the group structure inherent in the CA matrix may limit the ensemble performance.

In this paper, we focus on tackling the limitations of the CA matrix. To handle the sparsity of the CA matrix, we draw into the shortest path technique to refine the CA matrix. To handle the low value density limitation of the CA, we propose a growing tree model for clustering ensemble.

In the proposed growing tree model, a set of prototype samples are firstly discovered based on the refined CA matrix, which are treated as trees' roots. Then the trees' roots grow up through connecting to the samples in the data set. Different from the traditional connecting process that connects each sample to its nearest prototype, we gradually connect the samples around the prototypes and expand the prototypes set. Firstly, the tree roots connect to the samples which have strong relation with them. Then, the assigned samples form new leaf nodes and connect to the samples which have strong relation with them. This process is repeated until all samples in the data set have been connected to the trees. In this process, high co-association values play important role.

The main contributions of this paper are summarized as follows:

- The shortest path technique is introduced to mitigate the sparsity of the CA matrix.
- A prototype example discovering method is developed and its rationality is illustrated by theoretical analysis and experimental analysis.
- A growing trees algorithm is proposed to handle the low value density limitation of the CA matrix. Combining prototype example discovering method and growing trees algorithms, a growing tree model (GoT) is developed to integrate multiple clustering results.
- The working mechanism and effectiveness of the growing tree model are illustrated by experimental analyses.

The remained of this paper is organized as follows. We firstly introduce the related work. Then the approach of handling the sparsity of CA matrix and discovering prototype examples are described. Following that, the growing tree model is described. Finally, the experimental results are reported.

Related Work

In the CA matrix-based clustering ensemble algorithms, the CA matrix is generally treated as two types of expressions: pairwise similarity matrix and graph matrix.

If the CA matrix is treated as a pairwise similarity matrix, many clustering algorithms based on the pairwise similarity matrix can be utilized to generate a consensus clustering result. The generally used clustering algorithm is the hierarchical type clustering. Fred and Jain (Fred and Jain 2005) first introduced the definition of the CA matrix and proposed an evidence accumulation clustering ensemble method, which extracted the consensus clustering by hierarchical clustering. Following that, Huang *et al.* (Huang, Lai, and Wang 2015) proposed a weighted evidence accumulation clustering. Zheng *et al.* (Zheng, Li, and Ding 2014) proposed a framework for hierarchical clustering ensemble. The main technique used in the clustering ensemble method based on the sample's stability that proposed in (Li et al. 2019) is also a hierarchical clustering.

If the CA matrix is treated as a graph matrix, the clustering ensemble problem can be handled by solving a graph partition problem. The generally used graph partition methods include METIS graph partition package (Karypis and Kumar 1998), normalized cut algorithm (N-cut) (Shi and Malik 2000), and T-cut algorithm (Li, Wu, and Chang 2012). Strehl and Ghosh (Strehl and Ghosh 2003) proposed three graph-based clustering ensemble methods, which are Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper-Graph Partitioning Algorithm (HGPA), and Meta-CLustering Algorithm (MCLA). The graph partition methods used by CSPA and HGPA are METIS and HMETIS, respectively. Yu *et al.* (Yu et al. 2015) utilized affinity propagation algorithm with different distance measures and attributes to generate base clustering results, and used the N-cut algorithm to generate a consensus clustering on the CA matrix. In (Huang, Lai, and Wang 2015), Huang *et al.* proposed two novel consensus functions, in which the graph partition with multi-granularity link analysis (GP-MGLA) function utilized the T-cut algorithm to generate the clustering result. Yu et al. (Yu, Wong, and Wang 2007) applied the consensus clustering method based on N-cut graph partition to discover cancer from microarray data. Tao *et al.* (Tao et al. 2019) learnt low-rank representation (LRR) for the CA matrix and developed a robust spectral ensemble clustering method.

Although a number of CA matrix-based clustering ensemble approaches have been proposed, there is still room for improving the clustering ensemble performance, which is arisen by handling the two limitations of the CA matrix.

Discovering Prototypes Based on the Refined Pairwise Relation Matrix

One major limitation of the sparsity of CA matrix is that the clusters may be broken up and thus the prototype for each cluster is hard to be discovered. Prototypes are helpful for discovering the underlying structure of a data set. In order to discover a set of reasonable prototype samples from a set of base clustering results, in this section, we first refine the CA

matrix to mitigate its sparsity and then calculate the tendency that a sample to be a prototype sample. In addition, we verify the rationality of the discovered prototype samples by experimental analysis and theoretical analysis.

Refining Pairwise Relations

We introduce the concept of shortest path distance to refine the CA matrix. The CA matrix can be treated as a graph, in which the nodes are the data samples and the edge weights are the co-association values. In an undirected graph, the shortest path distance is a path from one source node to the destination node with the minimum sum of weight.

To utilize the shortest path distance, the weight should be transformed into a distance. To obtain a distance d_{ij} between x_i and x_j, the transformation for co-association value a_{ij} can be performed by $d_{ij} = 1 - a_{ij}$. Then, a distance matrix (DM) is obtained as $DM = \{d_{ij}\}_{n\times n}$.

By applying any shortest path algorithm on the DM, a refined distance matrix will be obtained, which is simply noted as $RM = \{p_{ij}\}_{n\times n}$. In the following realization, we simply apply the Dijkstra's algorithm (Dijkstra 1959).

For matrix $DM = \{d_{ij}\}_{n\times n}$ and its refined distance matrix $RM = \{p_{ij}\}_{n\times n}$, it is true that $p_{ij} \leq d_{ij}$. Then, the sparsity of the CA matrix will be mitigated by transforming to the RM. Thus, the RM may be beneficial for discovering a set of prototype samples.

Discovering Prototype Examples

To discover representative prototype examples, we following the idea in (Rodriguez and Laio 2014) that the tendency of a sample to be a prototype sample contains two factors, which are local density and representative capacity. In what follows, we introduce a method to measure the tendency of a sample to be a prototype sample based on a set of clustering results.

Based on a set of base clustering results, the density of sample x_i is measured by:

$$\rho_i = \frac{1}{l}\sum_{b=1}^{l}\frac{1}{n}|c^b(x_i)|, \tag{3}$$

where $|c^b(x_i)|$ is the number of samples in the cluster that contains x_i in the bth clustering result.

The representative capacity of x_i is:

$$\delta_i = \min_{j:\rho_j>\rho_i}\{p_{ij}\}. \tag{4}$$

For the sample which has the maximal local density, its representative capacity is the maximal distance value in the RM, i.e. $\delta = \max\{p_{ij}\}$.

The tendency of x_i to be a prototype sample is defined as:

$$r_i = \rho_i\delta_i. \tag{5}$$

For a density set with many equal values, the above approach will generate many equal representative values in a local area. Using these representative values may not identify prototype examples for all local areas. To handle this problem, if $\rho_j = \rho_i$, we set

$$\rho_j = \rho_j - \epsilon, \tag{6}$$

where ϵ is a small value.

One can discover k prototype examples $Z = \{z_1, z_2, \ldots, z_k\}$ by selecting the samples with top k tendency values. The algorithm of discovering prototype examples is shown as Algorithm 1.

Algorithm 1 Discovering prototype examples

INPUT: $\Pi = \{\pi_1, \pi_2, \ldots, \pi_l\}$, k.
OUTPUT: $Z = \{z_1, z_2, \ldots, z_k\}$.
Process:

```
Constructing the co-association matrix CA based on Formula 1.
Utilizing a shortest path technique to generate the refined distance matrix RM = {p_ij}_{n×n}.
Calculating local density of each sample by Formula 3.
for i = 1 to n do
    Calculating representative value r_i of sample x_i by Formula 5.
end for
for i = 1 to n do
    Updating local density by Formula 6.
end for
Selecting the samples with top k maximal representative values to form the prototype examples set Z = {z_1, z_2, ..., z_k}.
```

Analysis About Discovering Prototype

To show the superiority of discovered prototype examples based on the RM, we first conduct a simple experiment to visually compare the prototype examples that are discovered based on the DM and RM. Then, we theoretically analyse the discovered prototype examples on the assumption of two clusters with uniform distribution.

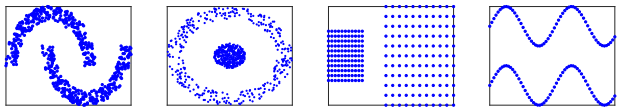
Based on the DM, δ is calculated as:

$$\delta_i^{DM} = \min_{j:\rho_j>\rho_i}\{d_{ij}\}. \tag{7}$$

Here, we denote the δ based on RM as δ^{RM}. Embedding δ^{DM} or δ^{RM} into the Algorithm 1, a set of prototype examples can be discovered.

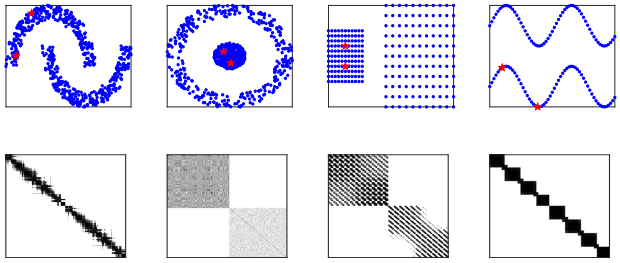
To visually compare the prototype examples discovered based on the DM and RM, we employ four synthetic data sets, the distributions of which are shown by Figure 2. For each data set, we generate 50 base clustering results based on multiple k-means algorithm with random initial centers. In each base clustering result, the number of clusters is set as $\min\{\sqrt{n}, 50\}$, where n is the number of samples in the data sets.

Figure 3 and Figure 4 show a concrete result base on the same clustering results set. In Figure 3 and Figure 4, the red stars represent the discovered prototype examples. From Figure 3 and Figure 4, it can be seen that the DMs contain more than 2 black blocks on the four synthetic data sets, while the RMs contain clear 2 black blocks. As the DM shows, the clusters are broken up. The corresponding RM shows that the broken clusters are connected. Then, reasonable prototype examples are discovered based on the RM.



(a) Rings (b) Target (c) Squares (d) Lines

Figure 2: The four synthetic data sets



Rings Target Squares Lines

Figure 3: The experimental results based on the DM matrix

In what follows, we analyse the prototypes that are discovered based on the DM and RM with three assumptions, which are:

1. Clusters c_1 and c_2 are two clusters with uniform distribution. c_1 is denser than c_2.
2. For samples x_i and x_j, the co-association value a_{ij} is the fraction of their common $\sqrt{n}$ nearest neighbor ($\sqrt{n}NN$).
3. The samples in different clusters have no common $\sqrt{n}NN$ neighbor.

We analyse the discovering prototypes on three data points x_i, x_j and x_k, in which x_i and x_j come from cluster c_1 and have no common $\sqrt{n}$ nearest neighbors, x_k comes from c_2. With the above assumptions, the following results will be obtained.

Discovery 1. *Based on the DM, the two discovered prototype examples come from the same cluster.*

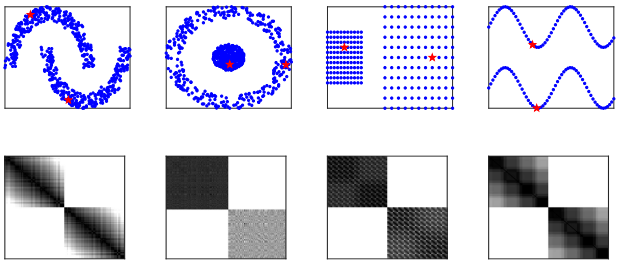
Proof. Due to $x_i, x_j \in c_1$ and $x_k \in c_2$, we have $\rho_i = \rho_j > \rho_k$.

Without loss of generality, we assume x_i is selected as the sample with the maximum density. Based on Formula 6, it holds that $\rho_i > \rho_j > \rho_k$.

With the conditions, it is easy to obtain $a_{ik} = 0$, $a_{jk} = 0$, and $a_{ij} = 0$. Then, with $d = 1 - a$, we have $d_{ij} = 1$, $d_{ik} = 1$, $d_{jk} = 1$.

The δ^{DM} for samples x_i, x_j, x_k are calculated as:

$$\begin{aligned}\delta_i^{DM} &= \max\{d_{ij}, d_{ik}, d_{jk}\} = 1;\\ \delta_j^{DM} &= d_{ij} = 1;\\ \delta_k^{DM} &= \min\{d_{ik}, d_{jk}\} = 1.\end{aligned}$$



Rings Target Squares Lines

Figure 4: The experimental results based on the RM matrix

It is true that $\delta_i^{DM} = \delta_j^{DM} = \delta_k^{DM}$. Then, based on formula $r^{DM} = \rho\delta^{DM}$, we have $r_i^{DM} > r_j^{DM} > r_k^{DM}$.

The two selected prototype examples are x_i and x_j, which come from the same cluster. □

Based on the RM, the discovered prototype examples can represent the clusters very well, which is shown by the following property.

Property 1. *Based on the RM, the two discovered prototype examples come from different clusters.*

Proof. The same as the proof of Discover 1, we can obtain $\rho_i > \rho_j > \rho_k$.

Due to $x_i, x_j \in c_1$, then in the DM, there exists a path $p(i, j)$ that the distance d between any pair of adjacent nodes in $p(i, j)$ is 0. Therefore, $p_{ij} = 0$.

Combining $\{x_i, x_j\} \in c_1$, $x_k \in c_2$, and $\forall x_e \in c_1, \forall x_f \in c_2$, $d_{ef} = 1$, we have $p_{ik} = 1$, $p_{jk} = 1$.

The δ^{RM} of samples x_i, x_j, x_k are:

$$\begin{aligned}\delta_i^{RM} &= \max\{p_{ij}, p_{ik}, p_{jk}\} = 1;\\ \delta_j^{RM} &= p_{ij} = 0;\\ \delta_k^{RM} &= \min\{p_{ik}, p_{jk}\} = 1.\end{aligned}$$

It is true that $\delta_i^{DM} = \delta_k^{DM} > \delta_j^{DM}$. Based on formula $r^{RM} = \rho\delta^{RM}$, we have $r_i^{DM} > r_k^{DM} > r_j^{DM}$.

Then, x_i and x_k are selected as the prototype examples, which come from different clusters. □

The above experimental analysis and theoretical analysis have shown the advantage of the prototype examples discovered based on the RM. With the discovered prototype examples, we then propose a growing tree method to assign the samples in the data set.

The Proposed Growing Tree Model

After discovering a set of prototype samples, most prototype-based algorithm assigns other samples to its nearest prototype based on a distance or similarity measure. The low value density of the CA matrix may affect the effectiveness of the assignment process. To handle this challenge, we propose a growing tree model, which takes the advantages of the reliable information.

Algorithm 2 Growing Trees

INPUT: $CA = \{a_{ij}\}_{n\times n}$, $Z = \{z_1, z_2, \ldots, z_k\}$.
OUTPUT: $F = \{t_1, t_2, \ldots, t_k\}$.
Process:

Initial $F = \{t_1, t_2, \ldots, t_k\}$, in which $t_i = \{z_i\}, i = 1, 2, \ldots, k$.
while $\sum_{i=2}^{k} |t_i| < n$ **do**
  Calculating the margin of each sample by Formula 8.
  Selecting the samples to be assigned by Formula 9.
  Assigning the selected samples by Formula 10.
  Updating the tree graph with Formula 11.
end while

In the growing tree model, each prototype sample is treated as a root. Then, the root gradually grows to trunks and leaves, which are the data samples. Suppose the obtained prototype examples set is $Z = \{z_1, z_2, \ldots, z_k\}$. We first built a forest with k trees $F = \{t_1, t_2, \ldots, t_k\}$, in which each tree is rooted by a prototype sample. In the beginning, each tree only has one node, which is the prototype sample $t_i = \{z_i\}$. Then, a tree will grow by reaching its near samples gradually.

Have obtained a set of trees $F = \{t_1, t_2, \ldots, t_k\}$, the samples that F reaches to should have a high confidence level to be correctly assigned. To quantify this confidence level, the margin of sample x_i is introduced, which is the difference between its most intimate tree and second intimate tree:

$$m(x_i) = ot(x_i, t_{p^*}) - \max_{p'\neq p^*, t_{p'}\in F} ot(x_i, t_{p'}), \qquad (8)$$

where $p^* = \arg_p \max_{t_p\in F} ot(x_i, t_p)$, and $ot(x_i, t_j) = \max_{x_b\in t_j}\{a(x_i, x_b)\}$.

With the margins set, we can select the samples to be assigned into the tree structure F with a threshold th:

$$\{x_i | m(x_i) > th\}. \qquad (9)$$

The threshold th can be learned by Otsu's algorithm (Otsu 1975).

The assignment of sample x_i is handled by assigning it to its nearest tree and linking it to its nearest node:

$$v(x_i) = t_j \qquad (10)$$

where$j = \arg_j \max_{t_j\in F} ot(x_i, t_j)$.

If sample x_i is assigned to tree t_j, t_j will be expanded by:

$$t_j = t_j \cup \{x_i\} \qquad (11)$$

After an assignment step, the size of the trees in the forest will increase, and the number of the samples that are not assigned is reduced. To generate the final clustering ensemble result, we can iteratively execute the assignment step until all samples are assigned to the forest. The iterative assignment forms the growing tree algorithm, which is shown as Algorithm 2.

Combining Algorithm 1 and Algorithm 2, we develop the growing tree model for clustering ensemble (using GoT for short). In the forest $F = \{t_1, t_2, \ldots, t_k\}$ that is obtained by Algorithm 2, each tree t is corresponding to a cluster in the ensemble result $\pi = \{c_1, c_2, \ldots, c_k\}$.

Algorithm 3 The Growing Tree Model (GoT)

INPUT: $\Pi = \{\pi_1, \pi_2, \ldots, \pi_l\}$, k.
OUTPUT: π^*.
Process:

$(Z, CA) \leftarrow Algorithm\,1\,(\Pi, k)$.
$(F) \leftarrow Algorithm\,2\,(Z, CA)$.
Generating clustering result π^* based on F.

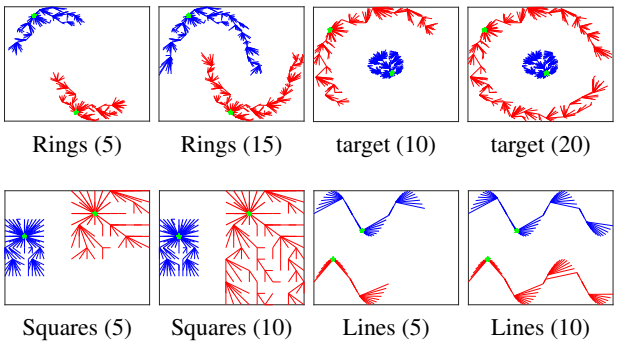


Figure 5: The processes of GoT on the synthetic data

Experimental Analysis

We execute two types of experiments to evaluate the performance of GoT. Firstly, we show the working mechanism of GoT on two-dimensional synthetic data sets. Then, we compare GoT with other state-of-art clustering ensemble algorithms on UCI benchmark data sets and image data sets. In the experiment, the base clustering results are generated by k-means algorithm with random initial centers. The clusters number of each result is set as $\min\{\sqrt{n}, 50\}$. The ensemble size is set as $l = 50$.

The Working Mechanism of GoT

We utilize the above four two-dimensional synthetic data sets to visually show the working mechanism of GoT.

The processes and ensemble results of the GoT on the four synthetic data sets are shown by Figure 5. For each data set, we show three results of special iteration. In each sub-fig, the discovered prototypes are represented by green stars, the handled samples are connected with their representative prototype. In Figure 5, the line width is related to the times of iteration that the corresponding sample is assigned. The earlier assigned samples are connected by a wider line. From Figure 5, it can be seen that a reasonable prototype examples set can be discovered for each synthetic data set and the clusters are expanded from the discovered prototypes to border prototypes, which likes the growth of trees.

The Effectiveness of GoT

To illustrate the effectiveness of GoT, we compare its performance with the other eight representative clustering ensemble methods, which are PTGP (Huang, Lai, and Wang 2016), PTA (Huang, Lai, and Wang 2016), EAC (Fred and Jain 2005), WTQ (Iam-On et al. 2011), HGPA (Strehl and Ghosh

Number	UCI data sets	N	D	K	Image data sets	N	K	Source
1	Iris	150	4	3	MerchData	75	5	Matlab
2	Wine recognition data	178	13	3	jp-office-04	161	8	UIUC Ponce Research group
3	Glass Identification Database	214	9	6	17flowers	1360	17	Oxford Visual geometry Group
4	Ecoli	336	7	8	SceneClass13	3859	13	(Lazebnik, Schmid, and Ponce 2006)
5	LIBRAS Movement Database	360	91	15	Sort_1000pics	1000	10	(Ciocca et al. 2014)
6	Cardiotocography Data Set	2126	40	10	coil-20-proc	1440	20	Columbia University Image Library
7	Image Segmentation data	2310	19	7	Caltech 101	9146	101	(Li et al. 2013)
8	Statlog Landsat Satellite Data Set	6435	36	6	MNIST-test	10000	10	(Lecun et al. 1998)

Table 1: Description of the UCI data sets and image data sets

Data	PTGP	PTA	EAC	WTQ	HGPA	CSPA	USENC	DREC	GoT
1	0.683±0.021	0.328±0.039	0.665±0.042	0.594±0.018	0.853±0.000	0.845±0.000	0.701±0.003	0.861±0.000	**0.871±0.000**
2	0.729±0.011	0.503±0.017	0.361±0.054	0.754±0.022	0.814±0.000	0.771±0.000	0.807±0.000	0.784±0.002	**0.821±0.002**
3	0.208±0.001	0.205±0.001	0.261±0.000	0.230±0.001	0.189±0.000	0.181±0.000	0.204±0.000	0.252±0.001	**0.272±0.000**
4	0.370±0.008	0.412±0.004	0.540±0.008	0.421±0.017	0.332±0.000	0.306±0.000	0.385±0.002	0.375±0.003	**0.558±0.006**
5	0.283±0.001	0.298±0.000	0.332±0.001	0.286±0.001	0.305±0.000	0.281±0.000	0.322±0.000	0.312±0.000	**0.342±0.000**
6	0.916±0.007	0.994±0.001	**1.000±0.000**	0.579±0.017	0.606±0.000	0.543±0.000	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
7	0.461±0.007	0.492±0.002	0.212±0.000	0.457±0.002	0.485±0.004	0.492±0.001	0.511±0.000	0.484±0.001	**0.516±0.000**
8	0.528±0.007	**0.565±0.003**	0.002±0.000	0.480±0.002	0.193±0.002	0.450±0.001	0.551±0.001	0.522±0.003	0.556±0.000

Table 2: The index ARI from ten clustering ensemble methods for the eight UCI data sets

Data	PTGP	PTA	EAC	WTQ	HGPA	CSPA	USENC	DREC	GoT
1	0.751±0.008	0.494±0.029	0.756±0.009	0.693±0.007	0.801±0.000	0.802±0.000	0.775±0.001	0.850±0.000	**0.852±0.000**
2	0.728±0.007	0.589±0.010	0.470±0.059	0.769±0.008	0.782±0.000	0.763±0.000	0.783±0.000	0.776±0.001	**0.805±0.002**
3	0.369±0.001	0.372±0.001	0.395±0.000	0.395±0.001	0.336±0.001	0.332±0.000	0.384±0.001	0.417±0.002	**0.436±0.001**
4	0.550±0.002	0.596±0.001	0.626±0.001	0.584±0.002	0.544±0.000	0.516±0.000	0.575±0.001	0.564±0.001	**0.655±0.001**
5	0.568±0.000	0.584±0.000	0.618±0.000	0.575±0.001	0.576±0.000	0.554±0.000	0.605±0.000	0.600±0.000	**0.620±0.000**
6	0.946±0.003	0.998±0.000	**1.000±0.000**	0.797±0.004	0.823±0.000	0.769±0.000	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
7	0.628±0.002	0.639±0.000	0.582±0.000	0.604±0.001	0.602±0.002	0.620±0.000	**0.645±0.000**	0.611±0.001	0.634±0.000
8	0.604±0.002	0.620±0.001	0.028±0.000	0.556±0.001	0.268±0.002	0.539±0.000	0.612±0.000	0.601±0.001	**0.624±0.000**

Table 3: The index NMI from ten clustering ensemble methods for the eight UCI data sets

2003), CSPA (Strehl and Ghosh 2003), USENC (Huang et al. 2019) and DREC (Zhou, Zheng, and Pan 2019).

To relieve the influence of different base clustering results sets on the evaluation of the performance of a clustering ensemble method, for each data set, we generate 50 ensemble sets and report the average performance of each compared method. To evaluate the ensemble result, we employ the Adjusted Rand Index (ARI) (Hubert and Arabie 1985) and Normalized Mutual Information (NMI) (Strehl and Ghosh 2003).

Eight real data sets from UCI and eight benchmark image data sets are used in this comparison experiment. The detailed information about the eight UCI data sets and the eight image data sets are shown in Table 1.

Following the above experimental settings, the two indices scores from the ten clustering ensemble algorithms for the eight UCI data sets are obtained. Table 2 shows the average ARI scores and Table 3 shows the average NMI scores. In Table 2 and Table 3, the maximum score for each data is underlined in bold type. As Table 2 and Table 3 shown, for the eight UCI data sets, the GoT is marked on seven data sets both from the view of ARI and NMI.

To extract vectorized features from each image in the eight data sets, we utilize two techniques: the pre-trained VGG-16 convolutional neural network (Simonyan and Zisserman 2014) and T-SNE (Maaten and Hinton 2008) dimension reduction method. The feature extraction process is outlined in Figure 6. Because the image size of MNIST is much less than 224×224, for this data set, we only utilize T-SNE to obtain 50-dimensional features. Specially, the cluster

Data	PTGP	PTA	EAC	WTQ	HGPA	CSPA	USENC	DREC	GoT
1	0.734±0.028	0.659±0.029	0.817±0.006	0.682±0.019	0.888±0.000	0.889±0.001	0.859±0.004	0.875±0.000	**0.895±0.000**
2	0.969±0.005	0.859±0.011	**1.000±0.000**	0.608±0.020	0.797±0.002	0.722±0.001	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
3	0.373±0.001	0.412±0.001	0.182±0.002	0.351±0.001	0.417±0.000	0.387±0.000	0.401±0.000	0.387±0.000	**0.419±0.001**
4	0.523±0.003	0.571±0.000	0.431±0.001	0.499±0.004	0.481±0.002	0.550±0.000	0.571±0.000	0.553±0.001	**0.599±0.001**
5	0.919±0.002	0.855±0.003	0.894±0.002	0.666±0.025	0.933±0.000	0.932±0.000	0.934±0.000	0.935±0.000	**0.944±0.000**
6	0.891±0.001	0.907±0.000	0.878±0.000	0.442±0.010	0.921±0.001	0.918±0.001	0.912±0.000	0.901±0.001	**0.934±0.000**
7	0.275±0.000	0.283±0.000	0.329±0.000	0.282±0.000	0.163±0.001	0.096±0.000	0.320±0.000	0.302±0.000	**0.398±0.001**
8	0.742±0.002	0.738±0.002	**0.786±0.000**	0.545±0.008	0.144±0.001	0.120±0.002	0.768±0.000	0.755±0.000	0.779±0.002

Table 4: The index ARI from ten clustering ensemble methods for the eight image data sets

Data	PTGP	PTA	EAC	WTQ	HGPA	CSPA	USENC	DREC	GoT
1	0.828±0.008	0.785±0.009	0.868±0.001	0.803±0.005	0.888±0.000	0.897±0.000	0.888±0.001	0.899±0.000	**0.906±0.000**
2	0.989±0.001	0.945±0.002	1.000±0.000	0.845±0.004	0.897±0.000	0.838±0.000	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
3	0.563±0.000	0.587±0.000	0.547±0.000	0.544±0.000	0.577±0.000	0.545±0.000	0.577±0.000	0.578±0.000	**0.589±0.000**
4	0.700±0.000	0.720±0.000	0.697±0.000	0.691±0.001	0.648±0.001	0.686±0.000	0.721±0.000	0.696±0.000	**0.730±0.000**
5	0.939±0.000	0.916±0.000	0.934±0.000	0.848±0.004	0.938±0.000	0.938±0.000	0.940±0.000	0.940±0.000	**0.949±0.000**
6	0.961±0.000	0.964±0.000	0.966±0.000	0.844±0.001	0.964±0.000	0.963±0.000	0.973±0.000	0.972±0.000	**0.982±0.000**
7	0.719±0.000	0.724±0.000	0.736±0.000	0.713±0.000	0.669±0.000	0.646±0.000	0.735±0.000	0.722±0.000	**0.744±0.000**
8	0.846±0.000	0.830±0.000	0.845±0.000	0.752±0.001	0.511±0.001	0.480±0.003	0.841±0.000	**0.850±0.000**	**0.850±0.000**

Table 5: The index NMI from ten clustering ensemble methods for the eight image data sets

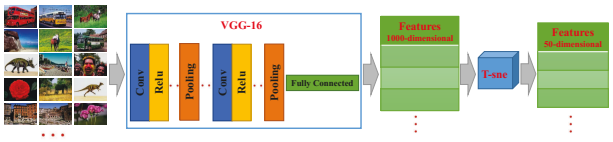


Figure 6: The process of feature extraction for image data

number of base clustering result for the Caltech 101 data is set as 200.

The experimental results of the ARI score and NMI score for the image data sets are shown in Table 4 and Table 5, respectively. It is obvious that the GoT obtains much more times of the top score than the other algorithms, which illustrates the effectiveness of the Got model.

Conclusion

Many clustering ensemble algorithms are based on the CA matrix. However, the CA matrix is highly sparse and low value density, which may affect the performance of these algorithms. In this paper, we have introduced the shortest path technique to mitigate the sparsity of the co-association matrix. In addition, we have proposed a growing tree model to integrate multiple clustering results. We have theoretically and experimentally illustrated the rationality of prototype examples. The working mechanism of the growing tree model has been visually shown by eight two-dimensional data sets. The experimental results have shown that the proposed model is more effective in integrating multiple clustering results than the other nine representative clustering algorithms on eight UCI data sets and eight image data sets.

Acknowledgments

This work was supported by National Key R&D Program of China (No. 2018YFB1004300), National Natural Science Foundation of China (Nos. 61672332, 61872226, 61802238, 61976129), Program for the San Jin Young Scholars of Shanxi, Natural Science Foundation of Shanxi Province (Grant No. 201701D121052).

References

Ciocca, G.; Cusano, C.; Santini, S.; and Schettini, R. 2014. On the use of supervised features for unsupervised image categorization: an evaluation. *Computer Vision and Image Understanding* 122: 155–171.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1(1): 269–271.

Fred, A. L.; and Jain, A. K. 2005. Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence* 27(6): 835–850.

Huang, D.; Lai, J.-H.; and Wang, C.-D. 2015. Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis. *Neurocomputing* 170: 240–250.

Huang, D.; Lai, J.-H.; and Wang, C.-D. 2016. Robust ensemble clustering using probability trajectories. *IEEE transactions on knowledge and data engineering* 28(5): 1312–1326.

Huang, D.; Wang, C.; Wu, J.; Lai, J.; and Kwoh, C. K. 2019. Ultra-Scalable Spectral Clustering and Ensemble Clustering. *IEEE Transactions on Knowledge and Data Engineering* 1–1.

Hubert, L.; and Arabie, P. 1985. Comparing partitions. *Journal of classification* 2(1): 193–218.

Iam-On, N.; Boongoen, T.; Garrett, S.; and Price, C. 2011. A link-based approach to the cluster ensemble problem. *IEEE transactions on pattern analysis and machine intelligence* 33(12): 2396–2409.

Karypis, G.; and Kumar, V. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing* 20(1): 359–392.

Lazebnik, S.; Schmid, C.; and Ponce, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 2169–2178. IEEE.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE* 86: 2278 – 2324. doi: 10.1109/5.726791.

Li, F.; Qian, Y.; Wang, J.; Dang, C.; and Jing, L. 2019. Clustering ensemble based on sample's stability. *Artificial Intelligence* 273: 37–55.

Li, F.; Qian, Y.; Wang, J.; Dang, C.; and Liu, B. 2018. Clusters Quality Evaluation and Selective Clustering Ensemble. *ACM Transactions on Knowledge Discovery From Data* 12(5): 60.

Li, F. F.; Fergus, R.; Perona, P.; and Zekrifa, D. M. S. 2013. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. volume 106, 59–70.

Li, Z.; Wu, X.-M.; and Chang, S.-F. 2012. Segmentation using superpixels: A bipartite graph partitioning approach. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, 789–796. IEEE.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

Otsu, N. 1975. A threshold selection method from gray-level histograms. *Automatica* 11(285-296): 23–27.

Rodriguez, A.; and Laio, A. 2014. Clustering by fast search and find of density peaks. *Science* 344(6191): 1492–1496.

Shi, J.; and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8): 888–905.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

Strehl, A. L.; and Ghosh, J. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3(3): 583–617.

Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2019. Robust spectral ensemble clustering via rank minimization. *ACM Transactions on Knowledge Discovery from Data* 13(1): 4.

Vega-Pons, S.; and Ruiz-Shulcloper, J. 2011. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence* 25(03): 337–372.

Yang, Y.; and Chen, K. 2010. Temporal data clustering via weighted clustering ensemble with different representations. *IEEE transactions on knowledge and data engineering* 23(2): 307–320.

Yu, H.; Chen, Y.; Lingras, P.; and Wang, G. 2019. A three-way cluster ensemble approach for large-scale data. *International Journal of Approximate Reasoning* 115: 32 – 49.

Yu, Z.; Li, L.; Liu, J.; Zhang, J.; and Han, G. 2015. Adaptive noise immune cluster ensemble using affinity propagation. *IEEE Transactions on Knowledge and Data Engineering* 27(12): 3176–3189.

Yu, Z.; Wong, H.-S.; and Wang, H. 2007. Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics* 23(21): 2888–2896.

Zhang, X.; Jiao, L.; Liu, F.; Bo, L.; and Gong, M. 2008. Spectral clustering ensemble applied to SAR image segmentation. *IEEE Transactions on Geoscience and Remote Sensing* 46(7): 2126–2136.

Zheng, L.; Li, T.; and Ding, C. H. Q. 2014. A Framework for Hierarchical Ensemble Clustering. *ACM Transactions on Knowledge Discovery From Data* 9(2): 9.

Zhou, J.; Zheng, H.; and Pan, L. 2019. Ensemble clustering based on dense representation. *Neurocomputing* 357(SEP.10): 66–76.

Zhou, Z. 2019. Abductive learning: towards bridging machine learning and logical reasoning. *Science in China Series F: Information Sciences* 62(7): 1–3.