
Multi-view data ensemble clustering: a cluster-level perspective

Jiye Liang*

Department of Computer Science,
Taiyuan Normal University, China
and

Key Laboratory of Computational Intelligence and Chinese Information
Processing of Ministry of Education,
School of Computer and Information Technology,
Shanxi University, Taiyuan, China
Email: ljy@sxu.edu.cn

*Corresponding author

Qianyu Shi

Key Laboratory of Computational Intelligence
and Chinese Information,
Processing of Ministry of Education,
School of Computer and Information Technology,
Shanxi University,
Taiyuan, China
Email: 15735170907@139.com

Xingwang Zhao

Key Laboratory of Computational Intelligence and Chinese Information
Processing of Ministry of Education,
School of Computer and Information Technology,
Shanxi University,
Taiyuan, China
Email: zhaoxw84@163.com

Abstract: Ensemble clustering has recently emerged a powerful clustering analysis technology for multi-view data. From the existing work, these techniques held great promise, but most of them are inefficient for large data. Some researchers have also proposed efficient ensemble clustering algorithms, but these algorithms devote to data objects with the same feature spaces, which are not satisfied for multi-view data. To overcome these deficiencies, an efficient ensemble clustering algorithm for multi-view mixed data is developed from the cluster-level perspective. Firstly, a set of clustering solutions are produced with the K-prototypes clustering algorithm on each view multiple times, respectively. Then, a cluster-cluster similarity matrix is constructed by considering all the clustering solutions. Next, the METIS algorithm is conduct meta-clustering based on the similarity matrix. After that, the final clustering results are obtained by applying majority voting to assign the objects to their

corresponding clusters based on the meta-clustering. The corresponding time complexity of the proposed algorithm is analysed as well. Experimental results on several multi-view datasets demonstrated the superiority of our proposed algorithm.

Keywords: multi-view data; mixed data; k-prototypes clustering algorithm; ensemble clustering.

Reference to this paper should be made as follows: Liang, J., Shi, Q. and Zhao, X. (2018) 'Multi-view data ensemble clustering: a cluster-level perspective', *Int. J. Machine Intelligence and Sensory Signal Processing*, Vol. 2, No. 2, pp.97–120.

Biographical notes: Jiye Liang is a Professor with the Department of Computer Science at the Taiyuan Normal University. He is also a Director of the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education at the Shanxi University. He received his MSc and PhD degrees from the Xi'an Jiaotong University in 1990 and 2001, respectively. His current research interests include computational intelligence, granular computing, data mining and knowledge discovery. He has published more than 190 journal papers in his research fields.

Qianyu Shi is currently working toward the Master's degree in the School of Computer and Information Technology at the Shanxi University. Her research interests are in the areas of data mining and machine learning.

Xingwang Zhao is currently working toward the PhD degree in the School of Computer and Information Technology at the Shanxi University. His research interests are in the areas of data mining and machine learning.

1 Introduction

For a given dataset, clustering analysis is a process of automatically assigning similar objects to the same group based on the similarity measure between objects, so that objects of the same group are as similar as possible whereas objects belonging to different groups are as dissimilar as possible (Jain, 2010; Xu and Wunsch, 2005; Huang et al., 2012; Rana et al., 2013). Currently, clustering analysis techniques have been widely used in the field of bioinformatics, social networks, image processing, etc. (Han et al., 2011).

Multi-view data are instances that have multiple views, that is to say, objects are consistent, but there are certain differences between the feature spaces of the objects. For example, in order to evaluate economic growth of countries, there are multiple views including purchasing power, unemployment rate, gross domestic product (GDP), etc. As the second example, we can refer to a banking customer dataset, variables include a demographic view representing demographic information of customers, an account view showing the information about customer accounts, and a consumption view describing customer spending behaviours. Developing clustering algorithms for multi-view data is a very important task.

Ensemble clustering algorithms recently hold great promise for dealing with multi-view data due to their advantages of enhancing the clustering accuracy, robustness,

parallelism and so on (Fred and Jain, 2005; Lam-On and Boongoen, 2015; Ghosh and Acharya, 2011; Zhao et al., 2017). For a particular dataset, different clustering algorithms, or even the same algorithm with different parameters, may produce distinct solutions. It is impossible to select the most appropriate clustering algorithm with the correct parameter values for a given dataset without any further information in most cases. It is because the characteristic of the dataset is unknown but most of clustering algorithm focuses on their individual criteria about the underlying dataset. Ensemble clustering algorithms have recently emerged to overcome these limitations, and have been shown to be more effective and robust. The main objective of ensemble clustering is to combine multiple clusterings of a dataset into a unified and comprehensive final clustering result. This combined clustering result takes advantage of different criteria at the same time to achieve accuracy superior to that of any individual clustering. For multi-view data, ensemble clustering algorithms combine the clustering results from each view (Li et al., 2012; Bickel and Scheffere, 2004; Tzortzis and Likas, 2010; Cleuziou et al., 2009). For example, there are many departments to record information of students in the university. Students' affairs division records student achievements, but the centre of apartment holds student accommodations. If we want to cluster the students for analysis, all the data are not need to be send to a single site. Each department can perform clustering based on the local data views and send only the clustering results for combining. In this way, only a small amount of data are transferred and only clustering results of each view could be shared instead of original data. This provides benefits of saving bandwidth and preserving privacy at the same time. However, traditional ensemble clustering algorithms work on object level in the ensemble stage, which is faced with the problem of larger amount of computation, higher time complexity, especially when datasets have huge number of objects.

As a consequence, some researchers have also made some exploration with respect to this issue (Yu et al., 2012, 2014; Wu et al., 2012; Chung and Dai, 2013). For example, structure ensemble, which is firstly proposed by Yu et al. (2012, 2014), can solve the problem of cluster fusion in the ensemble learning well. Unlike traditional cluster ensemble approaches the main objective of which is to align individual labels obtained from different clustering solutions, the structure ensemble approach focuses on how to integrate multiple cluster structures extracted from different datasets into a unified structure. In the literature (Wu et al., 2012; Chung and Dai, 2013), a new concept called clustering fragment is incorporated for increased speed. A data fragment is any subset of the data that is not split by any of the clustering results. As the number of data fragments is usually far smaller than the number of data points, ensemble clustering based on data fragments is very likely to have a much lower time complexity than directly on data points without sacrificing effectiveness. The above mentioned algorithms are suitable for data clustering ensemble which are described by the same feature spaces. However, data objects from multiple views usually have different feature spaces. That is to say, there is diverse dimensionality in the multi-view data analysis. To tackle the above problems, it is particularly necessary to provide an efficient ensemble clustering algorithm for multi-view data.

In this paper, a novel ensemble clustering algorithm for multi-view data with mixed numerical and categorical attributes is developed. First, a set of clustering solutions are produced by executing the k-prototypes clustering algorithm (Huang, 1998) on each view multiple times, respectively. Then, considering all the clustering solutions, the algorithm

computes the relations between input clusters and displays this information in a similarity matrix (SM). Because our proposed ensemble clustering algorithm works on cluster level, it is very scalable and runs efficiently. Even datasets having large number of objects can be processed in practical run times. Next, the METIS algorithm (Karypis and Kumar, 1998) is used to conduct metaclustering based on the cluster-cluster SM. After that, each metacluster contains a set of clusters. And an object can occur multiple times in a metacluster. Finally, majority voting (Mimaroglu and Aksehirli, 2011) is performed on the metaclustering to produce a non-overlapping final clustering. As a result, each object is assigned to only one cluster where it exists most frequently. The effectiveness and efficiency of the proposed algorithm are empirically demonstrated over some multi-view datasets in terms of three benchmark evaluation measures. The experimental results show that the proposed algorithm achieves good performance in comparison with several traditional ensemble clustering algorithms for multi-view data.

The remainder of this paper is organised as follows. In Section 2, some related works are discussed. Section 3 proposes an ensemble clustering algorithm for multi-view data. Experimental results are shown in Section 4. Finally, Section 5 provides conclusions.

2 Related work

2.1 *K*-prototypes clustering algorithm

In 1998, Huang (1998) proposed the *k*-prototypes algorithm, which is a simple integration of the *k*-means (Macqueen, 1967) and *k*-modes (Huang, 1998) algorithms. The *k*-prototypes algorithm is widely used because frequently encountered objects in real world database are mixed-type objects, and it is efficient in processing large datasets. In the *k*-prototypes algorithm, the dissimilarity measure takes into account both numerical attributes and categorical attributes. The dissimilarity measure on numerical attributes is defined by the squared Euclidean distance. For the categorical part, the computation of dissimilarity is performed by simple matching, which is the same as that of the *k*-modes algorithm.

Let x, y be two objects described by A attributes, $A = A^r \cup A^c$, A^r denotes numerical attributes, A^c denotes categorical attributes. Hence, x can be denoted as $(x^r, x^c)^T$ with $x^r = (x_1^r, x_2^r, \dots, x_{|A^r|}^r)$ and $x^c = (x_1^c, x_2^c, \dots, x_{|A^c|}^c)$. Similarly, y can be denoted as $(y^r, y^c)^T$ with $y^r = (y_1^r, y_2^r, \dots, y_{|A^r|}^r)$ and $y^c = (y_1^c, y_2^c, \dots, y_{|A^c|}^c)$. The dissimilarity of the numerical parts between two mixed-type objects x and y is calculated according to

$$D_{A^r}(x, y) = \sum_{q=1}^{|A^r|} (x_q^r - y_q^r)^2 \quad (1)$$

The dissimilarity of the categorical parts between two mixed-type objects x and y is calculated according to

$$D_{A^c}(x, y) = \sum_{q=1}^{|A^c|} \delta(x_q^c, y_q^c) \quad (2)$$

where

$$\delta(x_q^c, y_q^c) = \begin{cases} 1, & x_q^c \neq y_q^c \\ 0, & x_q^c = y_q^c \end{cases}$$

As a matter of fact, the dissimilarity used in the k-prototypes algorithm is calculated between an object and a prototype. Suppose that the clustering results of a given mixed dataset $D = \{x_1, x_2, \dots, x_n\}$ consisting of n objects, are $C^k = \{C_1, C_2, \dots, C_k\}$, whose cluster prototypes are $Z^k = \{z_1, z_2, \dots, z_k\}$, where k is the number of clusters. The dissimilarity between $x_i (1 \leq i \leq n)$ and the prototype $z_j (1 \leq j \leq k)$, can be measured by

$$D(x_i, z_j) = \frac{|A^r|}{|A|} \frac{D_{A^r}(x_i, z_j)}{\sum_{p=1}^k D_{A^r}(x_i, z_p)} + \frac{|A^e|}{|A|} \frac{D_{A^e}(x_i, z_j)}{\sum_{p=1}^k D_{A^e}(x_i, z_p)} \quad (3)$$

where $D_{A^r}(x_i, z_p)$ and $D_{A^e}(x_i, z_p)$ are calculated according to equations (1) and (2), respectively.

Based on the abovementioned dissimilarity (Liang et al., 2012b), K-prototypes clustering algorithm is described in Table 1.

Table 1 K-prototypes clustering algorithm

Input: a mixed dataset D , the number of clusters k in the final clustering.	
Output: final clustering.	
1	Select k initial prototypes from the mixed dataset D randomly, one for each cluster.
2	Allocate an object to the cluster whose prototype is the nearest to it according to equation (3).
3	Repeat
4	Update the prototype of the cluster after each allocation.
5	Retest the dissimilarity of objects against the current prototypes according to equation (3). If an object is found such that its nearest prototype belongs to another cluster rather than its current one, reallocate the object to that cluster.
6	Until no object has changed clusters after a full cycle test of the whole dataset or a given stopping criterion is fulfilled.
7	Return the final clustering.

2.2 Ensemble clustering

Ensemble clustering combines different clustering results of a dataset to obtain a single consolidated clustering, and it has exhibited great potential in improving the robustness as well as the quality of clustering results. Therefore, ensemble clustering has become one of the important research directions in the ensemble learning branch in recent years (Zhang and Suganthan, 2015; Pan et al., 2015). In general, the whole process of ensemble clustering involves two critical steps: generation of base clusterings, and aggregation of clustering solutions (Zhao et al., 2017).

The objective of the first step is to generate a set of different clustering results which are as diverse as possible. The existing approaches that are applied to generate multiple clusterings on a dataset can be categorised into five types:

- applying different clustering algorithms as the basic clustering algorithms to obtain different solutions (Strehl and Ghosh, 2002)
- applying the same clustering algorithm but changing the input parameter values, e.g., different the number of clusters k , to generate different clustering solutions (Iam-on and Boongoen, 2010)
- running a random initialisation clustering algorithm multiple times, e.g., adopting different initial centres for k -prototypes (Fred and Jain, 2005)
- using selected subsets of features, i.e., projection (Topchy et al., 2003)
- sub-sampling, re-sampling or adding noise to the original data (Valentini, 2007).

The objective of the second step is to obtain a unified and comprehensive final clustering result which is as accurate as possible. The examples of existing well-known ensemble methods are:

- The feature-based method that simply transforms the problem of cluster ensembles to categorical data clustering, i.e., cluster label of each clustering solution is treated as a new feature describing each object, the quality of final clustering result depends on the clustering algorithms for categorical data (Nguyen and Caruana, 2007; Topchy et al., 2005).
- The voting-based method that learns from classifier ensemble (Terrades et al., 2009), but there is no supervised information in ensemble clustering which is different from classifier ensemble. The final partition is found through relabeling the base clustering results (Boulis and Ostendorf, 2004).
- The co-association-based method that makes use of co-occurrences in the same clusters to create a co-association matrix, in which entries are viewed as pairwise similarity among objects, so any similarity-based clustering algorithm, e.g., hierarchical clustering, can be applied to this matrix to find consensus clustering (Fred and Jain, 2005).
- The graph-based algorithms that employ a graph partitioning methodology, Strehl and Ghosh (2002) present the three graph-based ensemble clustering algorithms: the cluster-based similarity partitioning algorithm (CSPA), the hyper-graph partitioning algorithm (HGPA), and the metaclustering algorithm (MCLA), which will be introduced in the Section 4.3. Each algorithm constructs a graph based on a pairwise matrix from the initial set of base clusterings, and then applies graph-clustering techniques to get the final clustering result. Fern and Brodley (2003, 2004) use the binary cluster-association matrix to generate a bipartite graph which models both objects and clusters as vertices, and considers both the similarities between objects and the similarities between clusters, then this graph is partitioned to achieve the final clustering result, using either METIS or spectral graph partitioning (SPEC) (Ng et al., 2001).

From the above, the time complexity of classical ensemble clustering algorithms is highly sensitive to the data size. When the number of objects is large, these algorithms will be unsuitable for practical application.

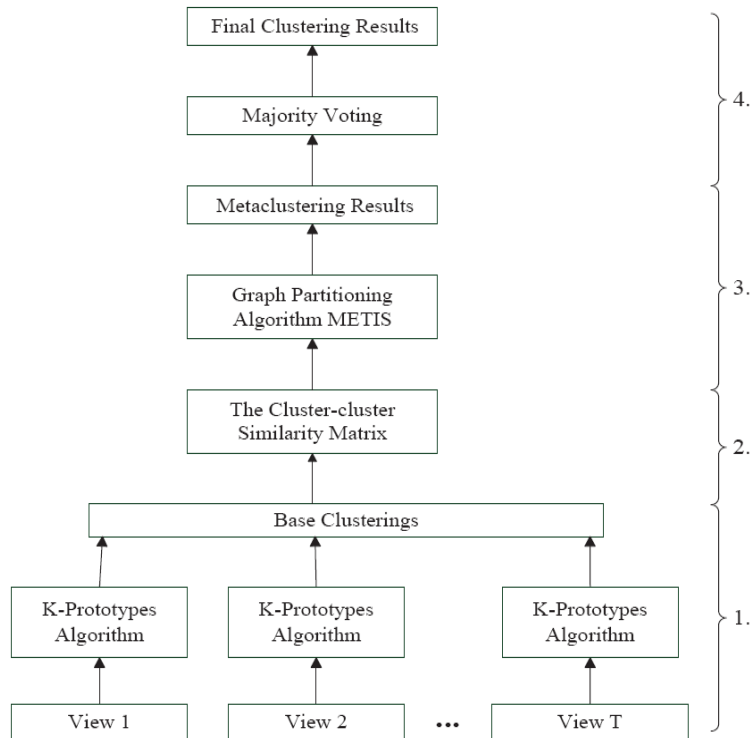
3 An ensemble clustering algorithm for multi-view data

Ensemble clustering algorithm for multi-view data includes four major steps of:

- 1 generating base clusterings: respectively executing the k-prototypes clustering algorithm on each view multiple times to form a set of base clusterings (II)
- 2 obtaining ensemble relationship: combining all base clusterings to generate a cluster-cluster SM
- 3 using the METIS algorithm to conduct metaclustering based on the SM
- 4 applying majority voting to assign the objects to their corresponding clusters based on the metaclustering, step 3 and step 4 are collectively known as determining final clustering result.

The framework of the proposed algorithm is illustrated in Figure 1.

Figure 1 The framework of the proposed algorithm



3.1 Generating base clusterings

Consider the dataset D consisting of n objects coming from T views. The multi-view data $D = (D^1, D^2, \dots, D^T)$ can be represented as $D^1 = (d_1^1, d_2^1, \dots, d_n^1), \dots, D^T = (d_1^T, d_2^T, \dots, d_n^T)$ where $D^t(1 \leq t \leq T)$ denotes the set of objects from the t^{th} view, and d_i^t is the i^{th} object in D^t . Let the cluster number of all base clusterings be k . Respectively execute k -prototypes clustering algorithm M_t times on each view D^t based on random initial cluster centres to get different base clusterings $\pi_j^t = \{C_{j1}^t, C_{j2}^t, \dots, C_{jk}^t\}$ with $1 \leq t \leq T$ and $1 \leq j \leq M_t$, $C_{jq}^t(1 \leq q \leq k)$ is the q^{th} cluster in the j^{th} base clustering π_j^t of the t^{th} view. Therefore, the set of base clusterings can be denoted as $\Pi(D) = \{\pi_1^1, \pi_2^1, \dots, \pi_{M_1}^1, \dots, \pi_1^T, \dots, \pi_{M_T}^T\}$. The goal of multi-view clustering is to find a new and better clustering $\pi^*(D) = \{C_1^*, C_2^*, \dots, C_k^*\}$ by using the information provided in $\Pi(D)$.

3.2 Obtaining ensemble relationship

In this step, the similarities of different clusters need to be computed. In the literature, many similarity or dissimilarity measure between clusters have been proposed. For example, in the agglomerative clustering algorithms, the difference between clusters is measured by using the distances between the closest or farthest data objects of two clusters, i.e., single link and complete link. They are sensitive to noise due to the dependence on a few objects. The distance between centres of clusters is often used to measure the dissimilarity between two clusters in the centres-based clustering algorithms. This measure is robust to noise and computationally efficient, but it cannot reflect the distinction of the boundary between two clusters and cannot assess distance between centres which have different feature spaces. Jaccard measure uses the number of the common objects included by the two clusters to reflect their similarity. The measure does not consider the objects whose cluster labels may be incorrect in a cluster. The number of these objects affects the effectiveness of the measure. Besides, the measure treats the similarity between clusters to be roughly 0 if they belong to the same base clustering, such measure is rather rough but we can capture the relationships between clusters according to other base clusterings. Thus, a new measure method is proposed to compute similarity of clusters in this paper by depending on the relationship between base clusterings.

Each cluster of base clusterings is viewed as a new data sample, and creates a new dataset $C = \{c_1, c_2, \dots, c_m\}$ with $m = k \times (M_1 + M_2 + \dots + M_T)$. The SM $SM_{m \times m}$ is constructed to store the relationship between any pair of clusters. Each element $SM(c_p, c_q)$ in the matrix denotes the similarity of clusters c_p and $c_q(1 \leq p, q \leq m)$ (Wang et al., 2013). Let c_p be a cluster of the base clustering $\pi_i(\pi_i \in \Pi(D))$, i.e., $c_p \in \pi_i$ and c_q be a cluster of the base clustering $\pi_j(\pi_j \in \Pi(D))$, i.e., $c_q \in \pi_j$.

If $i = j$, i.e., c_p and c_q belong to the same base clustering, the similarity of clusters c_p and c_q is computed based on other base clustering $\pi_s(s \neq i, \pi_s \in \Pi(D))$. Let c_r be a cluster of the base clustering π_s and $c_r \cap c_p \neq \emptyset, c_r \cap c_q \neq \emptyset$, the similarity of clusters c_p and c_q based on c_r is

$$SM(c_p, c_q | c_r) = \min \{ P(c_r | c_p), P(c_r | c_q) \} \quad (4)$$

where

$$P(c_r | c_p) = \frac{|c_r \cap c_p|}{|c_p|}, P(c_r | c_q) = \frac{|c_r \cap c_q|}{|c_q|} \quad (5)$$

Combining the similarity based on every cluster in π_s , so the similarity of clusters c_p and c_q based on the base clustering π_s is

$$SM(c_p, c_q | \pi_s) = \sum_{c_r \cap c_p \neq \emptyset, c_r \cap c_q \neq \emptyset, c_r \in \pi_s} SM(c_p, c_q | c_r) \quad (6)$$

Combining the similarity based on every base clustering, and in order to improve accuracy, we assign a weight to every base clustering. So the similarity of clusters c_p and c_q is

$$SM(c_p, c_q) = \frac{\sum_{s \neq i} \omega_s \times SM(c_p, c_q | \pi_s)}{\sum_{s \neq i} \omega_s} \quad (7)$$

where ω_s is the weight for base clustering π_s .

If $i \neq j$, i.e., c_p and c_q belong to different base clusterings, the similarity of clusters c_p and c_q is computed based on other base clustering $\pi_s (s \neq i, j, \pi_s \in \Pi(D))$. Similar to the above process, every cluster in the base clustering π_s is compared to c_p and c_q , respectively. The similarity of clusters c_p and c_q based on the base clustering π_s is computed according to equation (6), then combining the similarity based on every base clustering. Due to different base clusterings that c_p and c_q belong to, the similarity of clusters c_p and c_q is computed based on base clusterings except for π_i and π_j , which is defined as follows

$$SM(c_p, c_q) = \frac{\sum_{s \neq i, j} \omega_s \times SM(c_p, c_q | \pi_s)}{\sum_{s \neq i, j} \omega_s} \quad (8)$$

where ω_s is the weight for base clustering π_s .

Equation (6) considers the similarity of two cluster c_p and c_q on each possible cluster in base clustering π_s to capture the co-occurrence comparison between them (Cost and Salzberg, 1993; Wang et al., 2011). Further, the similarity between the cluster pair (c_p, c_q) can be calculated on top of $SM(c_p, c_q | \pi_s)$ by aggregating all the relative similarity on base clusterings other than π_i and π_j . For the parameter ω_s , in this paper, we simply collect information from the crowd of base clusterings to estimate the quality of a base clustering as the weight. Each base clustering is compared with the other ones in the ensemble and the average opinion is obtained for quality estimation (Huang et al., 2013).

The weight ω_i for base clustering π_i is defined as follows

$$\omega_i = \frac{1}{M-1} \sum_{\pi_j \in \Pi(D), i \neq j} Sim(\pi_i, \pi_j) \quad (9)$$

where $M = M_1 + M_2 + \dots + M_T$, $Sim(\pi_i, \pi_j)$ represents the similarity between the two base clusterings π_i and π_j . In this paper, the normalised mutual information (NMI) (Strehl and Ghosh, 2002) which will be introduced in the next section is used as the similarity measure $Sim(\pi_i, \pi_j)$. The greater agreement of a base clustering from the crowd is, the higher its weight is supposed to be.

3.3 Determining final clustering result

Having obtained a cluster-cluster SM, a graph-based partitioning method is exploited to obtain the metaclustering. This consensus function requires the underlying matrix to be initially transformed into a weighted metalevel graph $G = (V, W)$, where V is a set of vertices representing clusters C in the ensemble, and W denotes a set of weighted edges that can be defined from the SM. Afterward, a graph partitioning algorithm called METIS is used to partition the metalevel graph into k metaclusters, each metacluster contains a set of clusters. By this time, we obtain a clustering result of the dataset C which consists of all clusters in base clusterings.

Each object has a specific association degree to each metacluster. This can be estimated from the number of original clusters to which the object belongs, in the underlying metacluster, because an object can occur multiple times in a metacluster and can also occur in different metaclusters. In order to produce a non-overlapping final clustering, majority voting is performed on the metaclustering. The final clustering is produced by assigning each object to only one metacluster with which it is most frequently associated, i.e., it exists most frequently.

3.4 Algorithm description

Based on the above introduced major steps, an ensemble clustering algorithm for multi-view data (abbreviated as ENCA) is described in Table 2.

Table 2 An ensemble clustering algorithm for multi-view data

Input: a multi-view dataset $D = \{D^1, D^2, \dots, D^T\}$, the number of clusters k in the final clustering.
Output: final clustering.
1 Respectively execute the K-prototypes clustering algorithm $M_t (1 \leq t \leq T)$ times on each view D^t to get a set of base clusterings $\Pi(D)$.
2 Calculate the cluster-cluster SM $SM_{m \times m}$ according to equation (8).
3 Utilise METIS to conduct metaclustering based on the SM $SM_{m \times m}$.
4 Perform majority voting on the metaclustering to produce a final clustering $\pi^*(D)$.

The following is the time complexities in the ensemble stage of the ENCA algorithm. The time complexity of constructing cluster-cluster SM is $O(m^3)$, where m is the number of all clusters. We use METIS for partition with time complexity being $O(m^2)$ and the computational cost of assigning the objects to the corresponding clusters based on

majority voting is $O(kn)$. Since $k \ll n$, the overall time complexity of the proposed algorithm is $O(m^3) + O(n)$.

4 Experimental evaluations

In this section, we conduct effectiveness and efficiency tests to analyse the performance of the proposed algorithm (ENCA) on some multi-view datasets in terms of three benchmark evaluation measures.

4.1 Investigated datasets

ENCA is tested on seven real multi-view datasets, including two real multi-view UCI datasets (UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml>), i.e., multiple features (MF) dataset and image segmentation (IS) dataset, and Reuters RCV1/RCV2 Multilingual (Reuters) dataset (Amini et al., 2009). The details of these datasets are summarised in Table 3.

Table 3 Detail description of datasets

<i>Datasets</i>	<i>Sizes</i>	<i>Views</i>	<i>Dimensions</i>	<i>Clusters</i>
MF	2,000	Mfeat-fou view	76	10
		Mfeat-fac view	216	
		Mfeat-kar view	64	
		Mfeat-pix view	240	
		Mfeat-zer view	47	
		Mfeat-mor view	6	
IS	2,310	Shape view	9	7
		RGB view	10	
Rt.SP-EN	12,342	SP view	11,547	6
		EN view	21,530	
Rt.EN-FR	18,758	EN view	21,531	6
		FR view	24,892	
Rt.IT-SP	24,039	IT view	15,506	6
		SP view	11,547	
Rt.FR-GR	26,648	FR view	24,893	6
		GR view	34,278	
Rt.GR-IT	29,953	GR view	34,279	6
		IT view	15,505	

4.2 Evaluation criteria

In order to evaluate the effectiveness of the clustering algorithms, we have adopted the following three popular external criteria: clustering accuracy (CA) (Liang et al., 2012a), adjusted rand index (ARI) (Liang et al., 2012a), and NMI (Strehl and Ghosh, 2002),

which measure the agreement of the clustering results produced by an algorithm and the ground truth.

Suppose that $C = \{C_1, C_2, \dots, C_I\}$ and $P = \{P_1, P_2, \dots, P_J\}$ represent the clustering results and pre-defined classes of the dataset with n objects, respectively. I and J are the number of clusters C and classes P ; n_{ij} is the number of common objects of cluster C_i and pre-defined class P_j ; n_i is the number of objects in cluster C_i ; and n_j is the number of objects in class P_j . Then the three popular external criteria are as follows.

- *Clustering accuracy (CA)*. CA measures the percentage of correctly classified objects in the clustering solution compared to pre-defined class labels. The CA is defined as

$$CA = \frac{\sum_{i=1}^I \max_{j=1}^J n_{ij}}{n} \quad (10)$$

- *ARI*. ARI takes into account the number of objects that exist in the same cluster and different clusters (Hubert and Arabie, 1985). The ARI is defined as

$$ARI = \frac{\sum_{i=1}^I \sum_{j=1}^J \binom{n_{ij}}{2} - \eta}{\frac{1}{2}(\rho + \nu) - \eta} \quad (11)$$

$$\rho = \sum_{i=1}^I \binom{n_i}{2}, \nu = \sum_{j=1}^J \binom{n_j}{2}, \eta = \frac{2\rho\nu}{n(n-1)} \quad (12)$$

- *NMI*. This is one of the common external clustering validation metrics, which estimates the extent of the clustering structure with the external classification information of the data. Thus, NMI is defined as

$$NMI = \frac{\sum_{i=1}^I \sum_{j=1}^J n_{ij} \log \frac{mn_{ij}}{n_i n_j}}{\sqrt{\sum_{i=1}^I n_i \log \frac{n_i}{n} \sum_{j=1}^J n_j \log \frac{n_j}{n}}} \quad (13)$$

The maximum value of the three external criteria is one. If the clustering structure is close to the true class structure, then the values of them are high. The higher the values of the three measures for a clustering result, the better the clustering performance is.

4.3 Experiment design

To fully evaluate the performance of the proposed method, it is compared to six ensemble clustering methods, i.e., CO + SL, CO + CL, CO + AL, CSPA, HGPA, and MCLA. The first three algorithms are based principally on the pairwise similarity among objects (Fred

and Jain, 2005). Given a cluster ensemble of multiview dataset $\Pi(D)$, a $n \times n$ SM (CO) is constructed by $CO(d_i, d_j) = \frac{1}{M} \sum_{t=1}^T \sum_{m=1}^{M_t} S_m^t(d_i^t, d_j^t) (1 \leq i, j \leq n)$, where $CO(d_i, d_j) \in \{0, 1\}$ represents the similarity measure between objects d_i, d_j . In addition, $S_m^t(d_i^t, d_j^t) = 1$ if $v_m^t(d_i^t) = v_m^t(d_j^t)$, and $S_m^t(d_i^t, d_j^t) = 0$ otherwise. Note that $v_m^t(d_i^t)$ denotes the cluster label to which d_i^t belongs in the base clustering π_m^t . The single-linkage (SL), complete-linkage (CL) and average linkage (AL) agglomerative hierarchical clusterings are applied to this matrix to yield the final partition $\pi^*(D)$.

The last three algorithms are graph-based ensemble clustering algorithms (Strehl and Ghosh, 2002). CSPA constructs an undirected weighted graph based on the CO matrix, and then METIS is used to partition the graph into k clusters. HGPA constructs a hypergraph, where each object is a vertex, and each cluster is a hyperedge. HMETIS (Karypis et al., 1999) is applied to partition the hypergraph into k unconnected components. MCLA is similar to our proposed algorithm, and the only difference is that similarity between any two clusters is computed using the binary Jaccard measure.

All the experiments are conducted in MATLAB R2014b 64bit edition on a workstation (Windows 64 bit, Intel(R) Xeon(R) 2.60 GHz E5-2650 CPU, 128 GB of RAM, Windows Server 2008 operating system). The proposed algorithm and the comparative algorithm are coded in MATLAB 8.4 programming language. The performance of each ensemble clustering algorithms is measured by the average CA, ARI, and NMI after performing the same approach ten times.

4.4 Experimental results

4.4.1 Effectiveness analysis

Based on three benchmark evaluation measures, the performance of different ensemble clustering algorithms over examined datasets are shown in Tables 4–6 when each ensemble clustering algorithm corresponds to 30 clustering solutions on each view. For each dataset, the highest two index values are highlighted in *italics*. The results shown in these tables indicate the superior effectiveness of ENCA, as compared to other ensemble clustering algorithms included in this experiment. No matter what kind of index is chosen as evaluation measure, ENCA is the leader on multi-view datasets, by producing five best quality output clusterings on seven datasets. CSPA does well and is the most effective among compared algorithms, by producing two best quality final clusterings. Another important discovery is that ENCA achieves comparable or better performance than the other algorithms on all the datasets. Although it is not leading on several datasets, it never produces very bad clusterings and its index values are even close to the best in most cases. In addition, the average index values of ENCA on all the datasets are also remarkable as shown in the same tables. Also in all cases the variance is almost equal to zero (to the fourth decimal digit) that shows the stability of the proposed algorithm.

Table 4 Clustering results of different algorithms with respect to CA

<i>Datasets</i>	<i>ENCA</i>	<i>CO + SL</i>	<i>CO + CL</i>	<i>CO + AL</i>	<i>CSPA</i>	<i>HGPA</i>	<i>MCLA</i>
MF	0.9255 ± 0.0001	0.1045 ± 0.0000	0.6330 ± 0.0024	0.7480 ± 0.0015	0.9225 ± 0.0010	0.1040 ± 0.0472	0.9095 ± 0.0002
IS	0.6416 ± 0.0005	0.4286 ± 0.0000	0.4753 ± 0.0047	0.6242 ± 0.0005	0.6126 ± 0.0006	0.3978 ± 0.0033	0.5563 ± 0.0006
Rt.SP-EN	0.5225 ± 0.0006	0.4052 ± 0.0000	0.5005 ± 0.0007	0.4827 ± 0.0009	0.5455 ± 0.0137	0.4589 ± 0.0011	0.4442 ± 0.0020
Rt.EN-FR	0.5373 ± 0.0000	0.3962 ± 0.0005	0.5017 ± 0.0031	0.5276 ± 0.0033	0.4896 ± 0.0000	0.3024 ± 0.0022	0.3891 ± 0.0039
Rt.IT-SP	0.5032 ± 0.0002	0.2082 ± 0.0000	0.3933 ± 0.0010	0.3787 ± 0.0003	0.5076 ± 0.0000	0.3723 ± 0.0009	0.4030 ± 0.0011
Rt.FR-GR	0.5279 ± 0.0006	0.1879 ± 0.0000	0.4960 ± 0.0007	0.4795 ± 0.0008	0.5229 ± 0.0000	0.3071 ± 0.0226	0.4534 ± 0.0058
Rt.GR-IT	0.5550 ± 0.0001	0.1707 ± 0.0000	0.4955 ± 0.0032	0.4827 ± 0.0023	0.5403 ± 0.0000	0.3476 ± 0.0261	0.4499 ± 0.0025
Average value	0.6019	0.2716	0.4993	0.5319	0.5916	0.3272	0.5151

Table 5 Clustering results of different algorithms with respect to ARI

<i>Datasets</i>	<i>ENCA</i>	<i>CO + SL</i>	<i>CO + CL</i>	<i>CO + AL</i>	<i>CSPA</i>	<i>HGPA</i>	<i>MCLA</i>
MF	0.8421 ± 0.0002	0.0000 ± 0.0000	0.4205 ± 0.0086	0.6982 ± 0.0012	0.8403 ± 0.0015	-0.0042 ± 0.0385	0.8182 ± 0.0006
IS	0.5121 ± 0.0005	0.2363 ± 0.0000	0.2447 ± 0.0087	0.4656 ± 0.0005	0.4624 ± 0.0004	0.2189 ± 0.0040	0.3304 ± 0.0016
Rt.SP-EN	0.1701 ± 0.0004	-0.0116 ± 0.0000	0.1222 ± 0.0058	0.1004 ± 0.0006	0.2059 ± 0.0191	0.1255 ± 0.0050	0.1345 ± 0.0033
Rt.EN-FR	0.2348 ± 0.0000	0.0164 ± 0.0000	0.1476 ± 0.0056	0.1697 ± 0.0048	0.1604 ± 0.0000	0.0108 ± 0.0007	0.0503 ± 0.0049
Rt.IT-SP	0.2452 ± 0.0001	0.0000 ± 0.0000	0.1592 ± 0.0006	0.1512 ± 0.0001	0.2467 ± 0.0000	0.1374 ± 0.0007	0.1845 ± 0.0013
Rt.FR-GR	0.2690 ± 0.0006	0.0000 ± 0.0000	0.2333 ± 0.0008	0.2131 ± 0.0007	0.2613 ± 0.0000	0.0894 ± 0.0127	0.1893 ± 0.0044
Rt.GR-IT	0.3116 ± 0.0004	0.0001 ± 0.0000	0.2861 ± 0.0022	0.2718 ± 0.0009	0.2973 ± 0.0000	0.1364 ± 0.0150	0.2660 ± 0.0016
Average value	0.3693	0.0345	0.2305	0.2957	0.3535	0.1020	0.2819

Table 6 Clustering results of different algorithms with respect to NMI

<i>Datasets</i>	<i>ENCA</i>	<i>CO + SL</i>	<i>CO + CL</i>	<i>CO + AL</i>	<i>CSPA</i>	<i>HGPA</i>	<i>MCLA</i>
MF	0.8592 ± 0.0001	0.0347 ± 0.0000	0.7103 ± 0.0017	0.8170 ± 0.0002	0.8517 ± 0.0004	0.0001 ± 0.0773	0.8443 ± 0.0003
IS	0.6178 ± 0.0006	0.6005 ± 0.0000	0.4600 ± 0.0029	0.5655 ± 0.0003	0.5783 ± 0.0004	0.2749 ± 0.0063	0.4830 ± 0.0012
Rt.SP-EN	0.3320 ± 0.0007	0.0337 ± 0.0002	0.3114 ± 0.0006	0.2975 ± 0.0007	0.3273 ± 0.0343	0.1515 ± 0.0042	0.2283 ± 0.0020
Rt.EN-FR	0.3354 ± 0.0000	0.2227 ± 0.0007	0.2744 ± 0.0018	0.3086 ± 0.0009	0.2647 ± 0.0000	0.0308 ± 0.0022	0.1501 ± 0.0118
Rt.IT-SP	0.3505 ± 0.0000	0.0058 ± 0.0000	0.2808 ± 0.0006	0.2651 ± 0.0001	0.3516 ± 0.0000	0.2023 ± 0.0016	0.3033 ± 0.0013
Rt.FR-GR	0.4461 ± 0.0004	0.0063 ± 0.0000	0.4180 ± 0.0013	0.4231 ± 0.0010	0.3781 ± 0.0000	0.1292 ± 0.0265	0.3890 ± 0.0065
Rt.GR-IT	0.4109 ± 0.0000	0.0200 ± 0.0004	0.3787 ± 0.0005	0.3879 ± 0.0003	0.4183 ± 0.0000	0.1844 ± 0.0255	0.3952 ± 0.0005
Average value	0.4788	0.1320	0.4048	0.4378	0.4529	0.1390	0.3990

Figure 2 The performance of different algorithms with respect to different number of clustering solutions on the mf dataset, (a) CA (b) ARI (c) NMI (see online version for colours)

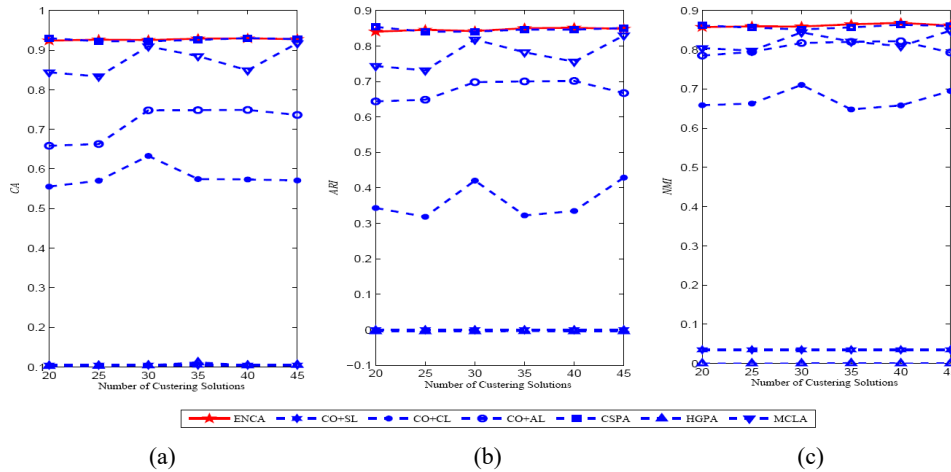
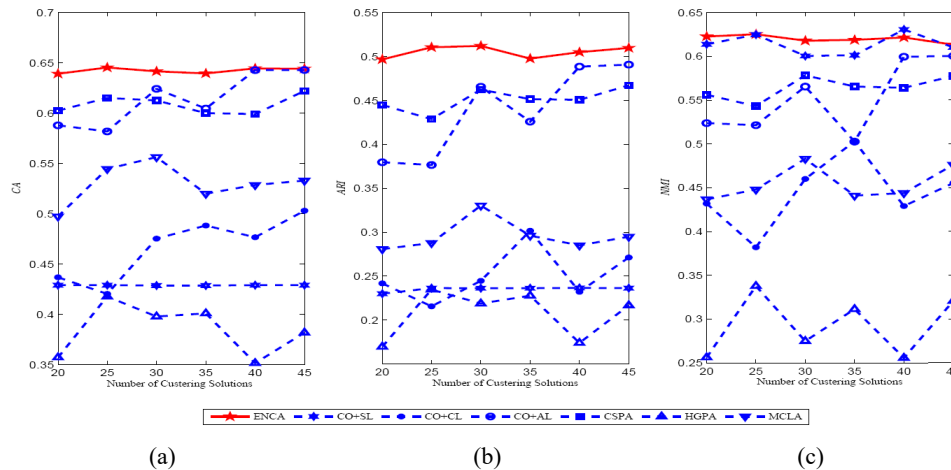


Figure 3 The performance of different algorithms with respect to different number of clustering solutions on the is dataset, (a) CA (b) ARI (c) NMI (see online version for colours)



Figures 2–8 show the performance of different algorithms with respect to different number of clustering solutions on each view. We can find:

- 1 ENCA is less affected by the change of the number of clustering solutions, and generates clustering results of higher quality in terms of the three external criteria in most cases, especially on IS dataset and Rt.EN-FR dataset.
- 2 CSPA achieves comparable performance to ENCA in many cases. Such as on Rt.SP-EN dataset, it sometimes even gets higher index values than ENCA as adopting CA and ARI, but its clustering results are sensitive to the number of clustering solutions.
- 3 Along with the change of the number of clustering solutions, the experiment results of ENCA are relatively stable which shows the robustness of the proposed algorithm.

Figure 4 The performance of different algorithms with respect to different number of clustering solutions on the Rt.SP-EN dataset, (a) CA (b) ARI (c) NMI (see online version for colours)

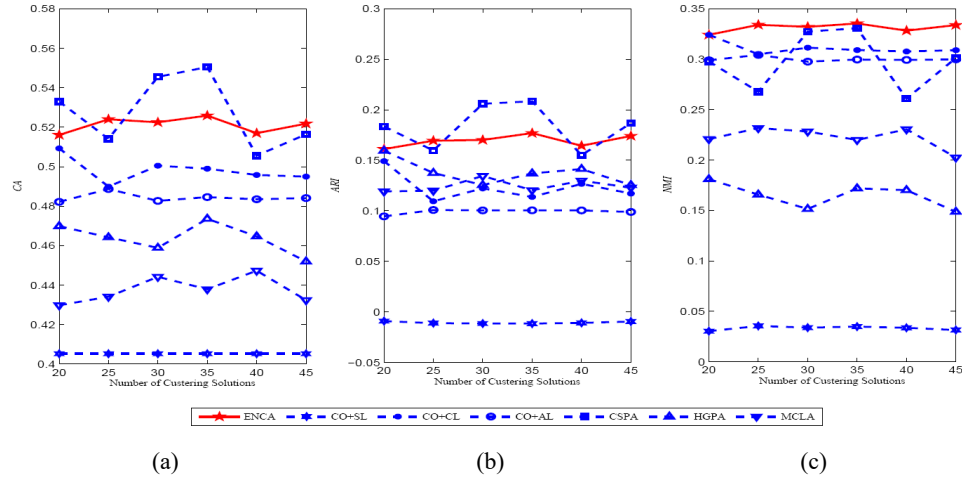
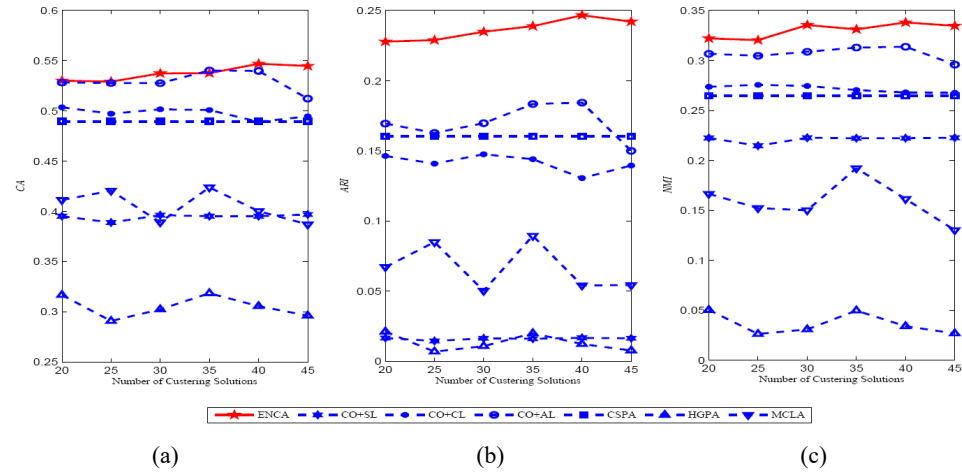


Figure 5 The performance of different algorithms with respect to different number of clustering solutions on the Rt.EN-FR dataset, (a) CA (b) ARI (c) NMI (see online version for colours)



4.4.2 Efficiency analysis

Figure 9 shows the average execution time results in the ensemble stage of ensemble clustering algorithms with respect to different number of clustering solutions on each view. There exist a trade off between the run time and the accuracy. For example, although HGPA and MCLA are the most efficient methods, the results obtained by these two algorithms are not as satisfactory as ENCA and CSPA, which is illustrated in above experiments. That's because HGPA and MCLA also work on cluster level, HGPA only uses clusters to construct hyperedge and MCLA computes similarity between two

clusters by simply using the binary Jaccard measure which is not as complex as ENCA. HGPA and MCLA out perform other ensemble clustering algorithms in terms of running time but sacrificing effectiveness.

Figure 6 The performance of different algorithms with respect to different number of clustering solutions on the Rt.IT-SP dataset, (a) CA (b) ARI (c) NMI (see online version for colours)

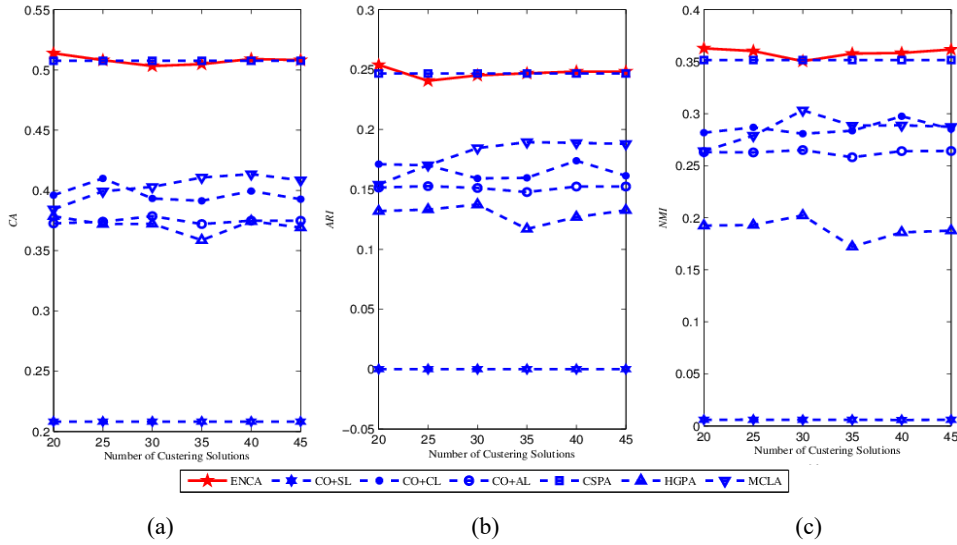


Figure 7 The performance of different algorithms with respect to different number of clustering solutions on the Rt.FR-GR dataset, (a) CA (b) ARI (c) NMI (see online version for colours)

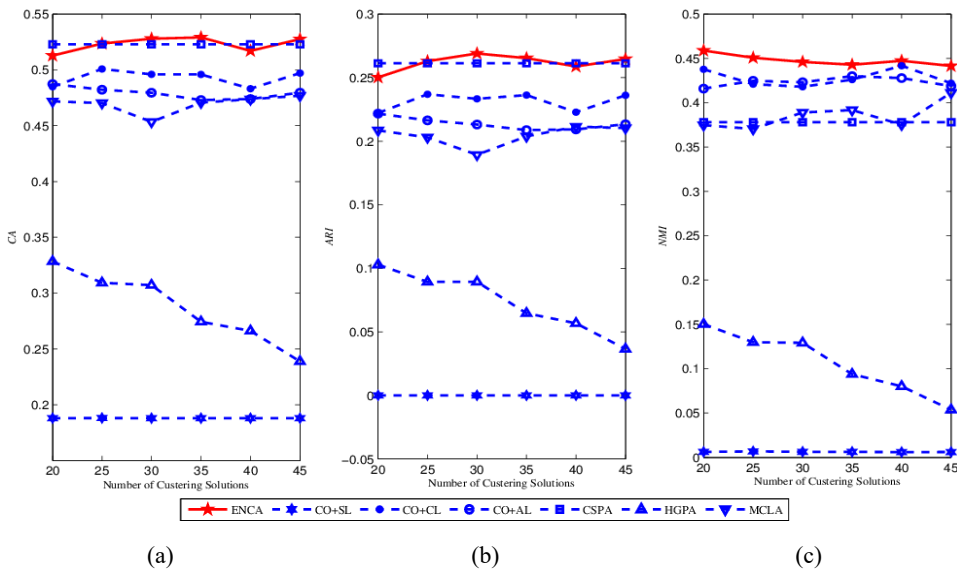
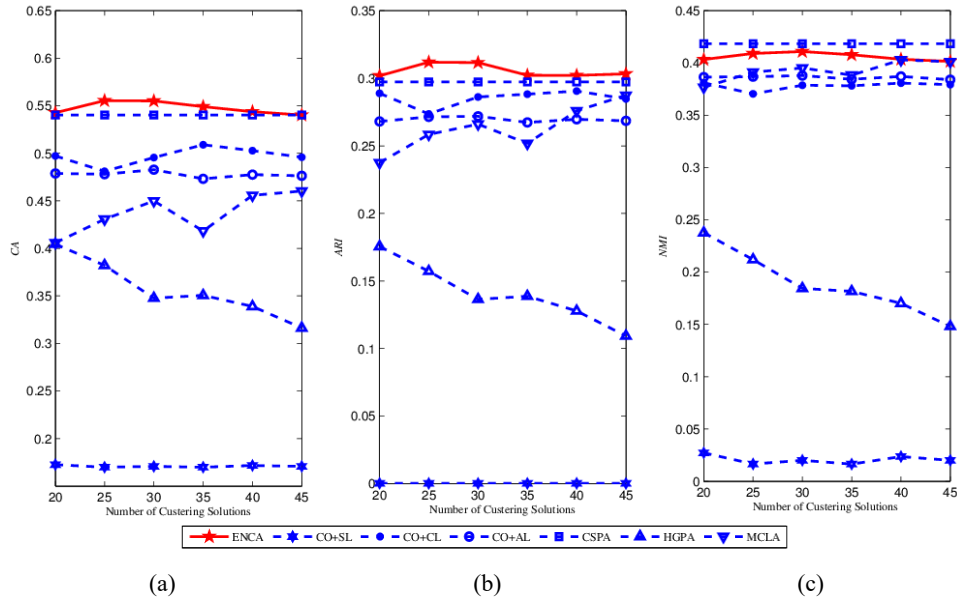


Figure 8 The performance of different algorithms with respect to different number of clustering solutions on the Rt.GR-IT dataset, (a) CA (b) ARI (c) NMI (see online version for colours)



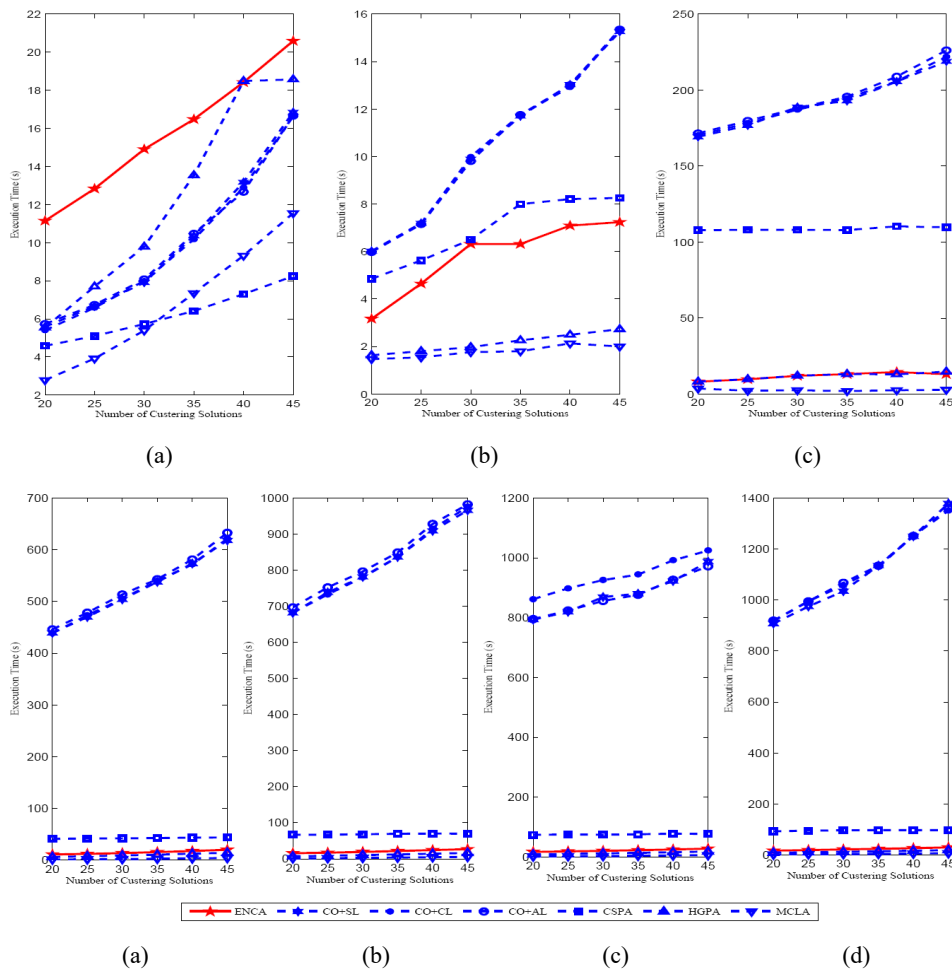
ENCA requires a longer run time, but it obtains better quality of final clusterings, which is more important in many scenarios. CO + SL, CO + CL, and CO + AL are fast on datasets when the data size is small. With the increasing of the data size or number of clustering solutions, execution time of CO + SL, CO + CL, and CO + AL increases sharply while the time consumption of our proposed algorithm is still quite low and the scalability of ENCA is good. Comparing the execution time results of two most accurate algorithms, i.e., ENCA and CSPA, CSPA falls behind for large datasets. An interesting observation to note is that there is only one exception that the execution time of ENCA is higher than CO + SL, CO + CL, CO + AL and CSPA on the MF dataset. The complexities in Section 3.4 indicate that the complexity of ENCA is strongly influenced by the number of clusters k . The complexities of CO + SL, CO + CL, CO + AL and CSPA are influenced by the number of objects n . Compared with other datasets, the number of clusters k on MF is ten, which is the largest, but the number of objects n of the MF dataset is 2,000, which is the smallest. Therefore, it is reasonable that the execution time of ENCA on MF is higher than that of CO + SL, CO + CL, CO + AL and CSPA. In all the other cases, ENCA is faster than them. In conclusion, ENCA is a better choice when the number of objects is very large by taking into account the importance of both the execution time and the accuracy.

5 Conclusions

Ensemble clustering algorithms for large multi-view datasets are becoming more and more important in real-world applications. In this paper, we introduced a new ensemble clustering algorithm for multi-view data. Our method works on cluster level and the

number of clusters is usually far smaller than the number of objects. Therefore, the efficiency of the proposed algorithm is high for large-scale datasets clustering. To demonstrate the effectiveness and efficiency of the proposed algorithm, a serial of experiments are conducted on several multi view datasets in terms of three benchmark evaluation measures. The results show that the proposed algorithm outperforms the other traditional ensemble clustering algorithms for multi-view data by taking into account the importance of both the execution time and the accuracy.

Figure 9 Running time comparison of different clustering algorithms, (a) MF (b) IS (c) Rt.SP-EN (d) Rt.EN-FR (e) Rt.IT-SP (f) Rt.FR-GR (g) Rt.GR-IT (see online version for colours)



Acknowledgements

This work was supported by National Natural Science Fund of China (Nos. 61432011, 61603230, U1435212), the Natural Science Foundation of Shanxi Province (No. 201601D202039), and the Graduate Education Innovation Project of Shanxi Province (2016SY002).

References

- Amini, M., Usunier, N. and Goutte, C. (2009) 'Learning from multiple partially observed views-an application to multilingual text categorization', in *Proceedings of the 16th International Conference on Neural Information Processing*, pp.28–36, Bangkok, Thailand.
- Bickel, S. and Scheffere, T. (2004) 'Multi-view clustering', in *Proceedings of the 4th IEEE International Conference on Data Mining Workshops*, Washington, USA, pp.19–26.
- Boulis, C. and Ostendorf, M. (2004) *Combining Multiple Clustering Systems*, Vol. 3202, pp.63–74, Springer, Berlin Heidelberg.
- Chung, C.H. and Dai, B.R. (2013) 'A fragment-based iterative consensus clustering algorithm with a robust similarity', *Knowledge and Information Systems*, Vol. 41, No. 3, pp.591–609.
- Cleuziou, G., Exbrayat, M., Martin, L. and Sublemontier, J.H. (2009) 'CoFKM: a centralized method for multiple-view clustering', in *Proceedings of the 9th IEEE International Conference on Data Mining*, pp.752–757, Miami, Florida, USA.
- Cost, S. and Salzberg, S. (1993) 'A weighted nearest neighbor algorithm for learning with symbolic features', *Machine Learning*, Vol. 10, No. 1, pp.57–78.
- Fern, X.Z. and Brodley, C.E. (2003) 'Random projection for high dimensional data clustering: a cluster ensemble approach', in *Proceedings of 20th International Conference on Machine Learning*, pp.186–193, Washington, DC, USA.
- Fern, X.Z. and Brodley, C.E. (2004) 'Solving cluster ensemble problems by bipartite graph partitioning', in *Proceedings of 21st International Conference on Machine Learning*, pp.36–43, Banff, Alberta, Canada.
- Fred, A.L. and Jain, A.K. (2005) 'Combining multiple clusterings using evidence accumulation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 6, pp.835–850.
- Ghosh, J. and Acharya, A. (2011) 'Cluster ensembles', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 1, No. 4, pp.305–315.
- Han, J.W. and Kamber, M. and Pei J. (2011) *Data Mining Concepts and Techniques*, 3rd ed., Elsevier Inc., Netherlands.
- Huang, D., Lai, J.H. and Wang, C.D. (2013) 'Exploiting the wisdom of crowd: a multi-granularity approach to clustering ensemble', in Sun, C., Fang, F., Zhou, Z.H., Yang, W. and Liu, Z.Y. (Eds.): *Intelligence Science and Big Data Engineering. ISClIDE 2013. Lecture Notes in Computer Science*, Vol. 8261, Springer, Berlin, Heidelberg.
- Huang, H.C., Chuang, Y.Y. and Chen, C.S. (2012) 'Multiple kernel fuzzy clustering', *IEEE Transactions Fuzzy System*, Vol. 20, No. 1, pp.120–134.
- Huang, Z.X. (1998) 'Extensions to the k-means algorithm for clustering large data sets with categorical values', *Data Mining and Knowledge Discovery*, Vol. 2, No. 3, pp.283–304.
- Hubert, L. and Arabie, P. (1985) 'Comparing partitions', *Journal of Classification*, Vol. 2, No. 1, pp.193–218.
- Iam-On, N. and Boongoen, T. (2015) 'Comparative study of matrix refinement approaches for ensemble clustering', *Machine Learning*, Vol. 98, Nos. 1/2, pp.269–300.
- Iam-on, N., Boongoen, T. and Garrett, S. (2010) 'LCE: a link-based cluster ensemble method for improved gene expression data analysis', *Bioinformatics*, Vol. 26, No. 12, pp.1513–1519.

- Jain, A.K. (2010) 'Data clustering: 50 years beyond K-means', *Pattern Recognition Letters*, Vol. 31, No. 8, pp.651–666.
- Karypis, G. and Kumar, V. (1998) 'Multilevel k-way partitioning scheme for irregular graphs', *Journal of Parallel and Distributed Computing*, Vol. 48, No. 1, pp.96–129.
- Karypis, G., Aggarwal, R., Kumar, V. and Shekhar, S. (1999) 'Multilevel hypergraph partitioning: Applications in VLSI domain', *IEEE Transactions Very Large Scale Integration Systems*, Vol. 7, No. 1, pp.69–79.
- Li, G.X., Chang, K.Y. and Hoi, S.C.H. (2012) 'Multi-view semi-supervised learning with consensus', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 24, No. 11, pp.2040–2051.
- Liang, J.Y., Bai, L., Dang, C.Y. and Cao, F.Y. (2012a) 'The k-means-type algorithms versus imbalanced data distributions', *IEEE Transactions on Fuzzy Systems*, Vol. 20, No. 4, pp.728–745.
- Liang, J.Y., Zhao, X.W., Li, D.Y., Cao, F.Y. and Dang, C.Y. (2012b) 'Determining the number of clusters using information entropy for mixed data', *Pattern Recognition*, Vol. 45, No. 6, pp.2251–2265.
- Macqueen, J. (1967) 'Some methods for classification and analysis of multivariate observations', in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp.281–297, Berkeley, California.
- Mimaroglu, S. and Aksehirli, E. (2011) 'DICLENS: divisive clustering ensemble with automatic cluster number', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 9, No. 2, pp.408–420.
- Ng, A.Y., Jordan, M.I. and Weiss, Y. (2001) 'On spectral clustering: analysis and an algorithm', *Advances in Neural Information Processing Systems*, Vol. 14, pp.849–856.
- Nguyen, N. and Caruana, R. (2007) 'Consensus clusterings', in *Proceedings of IEEE 7th International Conference on Data Mining*, pp.607–612, Omaha, Nebraska, USA.
- Pan, S., Wu, J., Zhu, X. and Zhang, C. (2015) 'Graph ensemble boosting for imbalanced noisy graph stream classification', *IEEE Transactions on Cybernetics*, Vol. 45, No. 5, pp.954–968.
- Rana, S., Jasola, S. and Kumar, R. (2013) 'A boundary restricted adaptive particle swarm optimization for data clustering', *International Journal of Machine Learning and Cybernetics*, Vol. 4, No. 4, pp.391–400.
- Strehl, A. and Ghosh, J. (2002) 'Cluster ensembles: a knowledge reuse framework for combining multiple partitions', *Journal of Machine Learning Research*, Vol. 3, pp.583–617.
- Terrades, O.R., Valveny, E. and Tabbone, S. (2009) 'Optimal classifier fusion in a nonbayesian probabilistic framework', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 9, pp.1630–644.
- Topchy, A., Jain, A.K. and Punch, W. (2003) 'Combining multiple weak clusterings', in *Proceedings of IEEE 3rd International Conference on Data Mining*, Melbourne, Florida, USA, pp.331–338.
- Topchy, A.P., Jain, A.K. and Punch, W.F. (2005) 'Clustering ensembles: models of consensus and weak partitions', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 12, pp.1866–1881.
- Tzortzis, G. and Likas, C. (2010) 'Multiple view clustering using a weighted combination of exemplar-based mixture models', *IEEE Transactions on Neural Networks*, Vol. 21, No. 12, pp.1925–1938.
- UCI Machine Learning Repository [online] <http://archive.ics.uci.edu/ml> (17 January 2017).
- Valentini, G. (2007) 'Mosclust: a software library for discovering significant structures in bio-molecular data', *Bioinformatics*, Vol. 23, No. 3, pp.387–389.
- Wang, C., Cao, L., Wang, M., Li, J., Wei, W., and Ou, Y. (2011) 'Coupled nominal similarity in unsupervised learning', in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp.973–978, New York, USA.

- Wang, C., She, Z. and Cao, L.B. (2013) 'Coupled clustering ensemble: incorporating coupling relationships both between base clusterings and objects', in *Proceedings of IEEE 29th International Conference on Data Engineering*, pp.374–385, Brisbane, Australia.
- Wu, O., Hu, W., Maybank, S.J., Zhu, M. and Li, B. (2012) 'Efficient clustering aggregation based on data fragments', *IEEE Transactions on Systems Man and Cybernetics Part B*, Vol. 42, No. 3, pp.913–926.
- Xu, R. and Wunsch, D. (2005) 'Survey of clustering algorithm', *IEEE Transactions on Neural Networks*, Vol. 16, No. 3, pp.645–678.
- Yu, Z. W., Li, L., Wong, H.-S., You, J., Han, G. Q., Gao, Y. J. and Yu, G. X. (2014) 'Probabilistic cluster structure ensemble', *Information Science*, Vol. 267, pp.16-34.
- Yu, Z.W., You, J., Wong, H-S. and Han, G.Q. (2012) 'From cluster ensemble to structure ensemble', *Information Science*, Vol. 198, pp.81–99.
- Zhang, L. and Suganthan, P.N. (2015) 'Oblique decision tree ensemble via multisurface proximal support vector machine', *IEEE Transactions on Cybernetics*, Vol. 45, No. 10, pp.2165–2176.
- Zhao, X.W., Liang, J.Y. and Dang, C.Y. (2017) 'Clustering ensemble selection for categorical data based on internal validity indices', *Pattern Recognition*, Vol. 69, pp.150–168.