



Incomplete multi-view clustering with multiple imputation and ensemble clustering

Guoqing Chao¹ · Songtao Wang² · Shiming Yang¹ · Chunshan Li¹ · Dianhui Chu¹

Accepted: 3 November 2021 / Published online: 28 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Multi-view clustering is an important and challenging task in machine learning and data mining. In the past decade, this topic attracted much attention and there have been many progress achieved in this field. However, in reality, due to different factors such as machine error, sensor failure, multi-view data are mostly incomplete, thus how to deal with this problem becomes a challenge. Some existing works mainly deal with view missing case, which means in certain view of datasets, the whole features of some samples would be lost. In fact, missing value can occur in any position, that is, any value missing case. In that case, there would be some values missed in any view with sheerly random way. We proposed a two-stage algorithm involved multiple imputation and ensemble clustering to deal with multi-view clustering in any value missing case. Multiple imputation is adopted to deal with missing values problem and weighted ensemble clustering is applied to implement multi-view clustering. The experimental comparison on several data sets verified the effectiveness of the proposed method.

Keywords Ensemble clustering · Missing value · Multi-view clustering · Multiple imputation

This article belongs to the Topical Collection: *Special Issue on Multi-view Learning*

Guest Editors: Guoqing Chao, Xingquan Zhu, Weiping Ding, Jinbo Bi and Shiliang Sun

✉ Dianhui Chu
chudh@hit.edu.cn

Guoqing Chao
guoqingchao@hit.edu.cn

Songtao Wang
Songtwang@163.com

Shiming Yang
2190400717@stu.hit.edu.cn

Chunshan Li
lics@hit.edu.cn

¹ School of Computer Science and Technology, Harbin Institute of Technology, 2 West Culture, Weihai, Shandong Province, China

² School of Information Science and Engineering, Harbin Institute of Technology, 2 West Culture, Weihai, Shandong Province, China

1 Introduction

Nowadays, along with data collection techniques, multi-view data has become ubiquitous around us in the real world. For example, a product online can be described by a paragraph of text introduction or several images, or even a short video. In web page classification tasks, the words on the web page itself and words underlying the links pointing to the web page from other pages constitute the two views of the web pages [4, 9, 11]. How to mine the rich information underlying the multi-view data becomes a challenging problem. As an important tool in machine learning and data mining fields, multi-view learning attracted more and more attentions during the past two decades [7, 10, 14, 16–19, 33, 40]. Relying on whether to use the labels, multi-view learning consists of multi-view classification and multi-view clustering. This work will focus on multi-view clustering. Multi-view clustering aims to cluster the subjects into several groups by integrating the multiple view information of the subjects [24, 39].

Besides multi-view clustering, there is a similar technique named ensemble clustering (EC) to mine multi-view information by clustering. Ensemble clustering boosts

Fig. 1 Differences between view missing case and any value missing case. These two pictures just show one view. O_n indicates the n -th sample while F_m indicates the m -th feature

	F_1	F_2	F_3
O_1	0.5	0.3	0.2
O_2	NA	NA	NA
O_3	0.1	0.4	0.6
O_4	0.2	0.1	0.5
O_5	0.3	0.2	0.1

(a) View missing case

	F_1	F_2	F_3
O_1	0.5	0.3	0.2
O_2	NA	0.2	0.3
O_3	0.1	0.4	NA
O_4	0.2	0.1	0.5
O_5	0.3	NA	0.1

(b) Any value missing case

the clustering performance by ensembling many clustering algorithms [1, 13, 15, 27, 34]. When each clustering algorithm works on each view, EC becomes multi-view clustering. Thus, EC is a way to implement multi-view clustering to some degree. EC mainly consists of two steps: generation step and consensus step [35]. Generation step aims to generate many clustering results which are prepared for consensus step to combine and finally obtain a consistent result. When EC works for multi-view clustering, consensus step should be emphasized to achieve a better clustering result. In addition, ensemble strategy is highly popular in many Kaggle competitions. In this paper, we will adopt EC to implement multi-view clustering to take advantage of ensemble learning merits. Up to date, there exist a large number of multi-view clustering works. Most of these works assume that all the multi-view information are available and they rarely deal with the situation where missing values occur in the data. Unfortunately, missing values are pretty common in real world.

Due to the errors occurring in the processes of data collecting, data inputting, data storing and so on, missing data problem are pretty common in multi-view data. It is a significant challenge to cluster the multi-view data with missing values. As for missing value handling techniques, they can be categorized into three groups: deleting the subjects with missing values, imputation, and no handling. For the first group of techniques, after deleting the subjects with missing values, all the remaining subjects have all the information, but when most of the subjects have missing value, the information loss is too heavy for real-world applications. The techniques of the second group can be further split into single imputation and multiple imputation (MI). Single imputation methods include mean imputation, random imputation, regression imputation, EM (expectation maximum) imputation and so on. Compared with single imputation, MI imputes the missing value multiple times to take the missing value uncertainty into consideration, and performs better in most cases. The third group no handling is also a technique to deal with missing value. For instance,

an indication matrix is adopted to deal with the missing views or values in some multi-view clustering works [8, 37]. Among all these missing value handling techniques, MI performs best in most cases and becomes the most popular strategy to deal with missing values.

To cluster the subjects with incomplete multi-view information, there appeared some incomplete multi-view clustering methods. They can be mainly classified into two groups. The first group uses an indicator matrix \mathbf{M}_{ij} to represent the i th subject is missing in j th view when $\mathbf{M}_{ij} = 1$, and the i th subject appears in j th view when $\mathbf{M}_{ij} = 0$ [26, 36, 37]. The second group imputes the missing value first and then runs general multi-view clustering algorithms. It is noted that the first group of methods are intended to deal with view missing case while the second group of methods aim to deal with any value missing case. Examples to explain the differences between two cases of missing value have been displayed in Fig. 1. View missing cases refers to the situation where the whole view will be missed for one subject while any value missing cases refer to the situation where any feature of any view can be missed for one subject. View missing cases can be considered as one special form of any value missing cases thus any value missing cases themselves are more difficult to tackle. To date, most of the existing incomplete multi-view clustering algorithms deal with view missing case. In this paper, we aim to deal with clustering problems in the general any value missing cases. We explore a two-stage method: the first stage is used to deal with missing values while the second stage is used to ensemble the clustering results from multiple views. To deal with these two problems, we adopted the popular MI to handle missing values and we explored different MI methods to choose the best and most stable one and then by considering the contributions from different views are different, we adopted a view weighting strategy [3, 20] to ensemble. We name our incomplete multi-view clustering with Multiple Imputation and Ensemble Clustering as MIEC. We compared it with the existing incomplete multi-view clustering algorithms

including state-of-the-art ones and the results verified the effectiveness of our proposed method.

The contributions of this paper are listed as follows:

- This paper combined multiple imputation and ensemble clustering to implement incomplete multi-view clustering for the first time.
- Compared with the existing incomplete multi-view clustering methods those handle view missing case, the proposed MIEC can deal with more general data missing problem: any value missing case.
- To further boost the clustering performance, view weights are taken into account in the proposed method.
- The experimental comparison with seven other algorithms on five data sets verifies the effectiveness of the proposed method.

The rest of this paper is organized as follows. Section 2 introduces the related works including MI and EC. Section 3 details the proposed MIEC. Section 4 shows the experimental setting and its results. Finally, conclusions are given in Section 5.

2 Related works

There are closely related works to our proposed incomplete multi-view clustering algorithm. They are MI and EC. We'll introduce them briefly.

2.1 Multiple imputation

MI produces m complete data sets from the incomplete data by imputing the missing data m times with some reasonable method. In statics, there are two techniques to implement MI:

1. Joint Multivariate Normal Distribution MI: This technique assumes that the incomplete data follows a multivariate normal distribution. However, this technique will be inappropriate when the incomplete dataset doesn't follow a multivariate normal distribution.
2. Conditional MI: Conditional MI provides many methods (different distribution models) to users and they follow an iterative procedure to implement modeling the conditional distribution of a certain variable given the other variables. It allows users to be more flexible to select appropriate distribution models assumed for incomplete datasets. A popular package named Multivariate Imputation by Chained Equations (MICE) [5] belongs to conditional MI.

The process of MICE package to MI is as follows:

At the first step, the MICE package creates several complete datasets. It assumes that the whole dataset follows

a specific distribution selected by users, and fills in the blank with reasonable value following the distribution. Considering the uncertainty of the predicted values, MICE creates multiple complete datasets.

At the second step, the Ordinary Least Squares (OLS) regression is run and a different regression coefficient for each dataset is obtained to reflect the effect of imputation according to some basic value such as mean, variance, etc. Due to the uncertainty of the imputed data, the analysis results will be considered as the basis of selecting the most reasonable value. The analysis results are stored in a *mira* (multiply imputed repeated analysis) object class.

Finally, all the analysis results of imputed datasets are pooled together into one complete dataset. With the parameters in the *pool()* users select, means, variances, and other baselines are calculated to obtain the final imputed complete dataset.

2.2 Ensemble clustering

Ensemble technique was developed to boost classification performance when it appeared in the beginning. Inspired by this idea, we investigated the potential of implementing ensemble technique to deal with clustering problem. Generally, EC aims to produce a better clustering result by combing multiple clustering models than the result produced by individual clustering model. It can be divided into two steps: generation step and consensus step [35]. Generation step will generate many clustering models to feed consensus step to combine. There are many techniques to achieve the generation step: different clustering algorithms, different parameters initialization, different object representations and so on. Consensus step is the main step in EC algorithm. How to design an appropriate consensus function is the key to the success of EC. Thus researchers mainly work on the consensus step. There are also two ways to implement consensus step: objects co-occurrence and median partition.

For the objects co-occurrence way, Fred and Jain [21, 22] encode the information of the base clusterings as a co-association matrix whose each entry indicates the frequency according to which two objects are clustered together and then treat the co-association matrix as a similarity or distance matrix to compute the final clustering. Based on co-association matrix, one graph-based EC method is proposed in [32] and in the same paper, two other graph-based EC ideas are figured out. In addition, relabeling and voting are also popular methods to implement consensus function.

Median partition is defined as the partition that maximizes the similarity with all partitions in the cluster ensemble, thus it is generally formulated as an optimization problem. The difficulty of median partition is to define the similarity measure, thus many similarity measurement

functions or distance functions are proposed to solve the problem, such as Mirkin distance [28], rand index [29], Jaccard coefficient [2] and so on.

Among all these successful EC algorithms, works [3, 20] took into account the different contributions of many base clustering models to the final results and eventually demonstrated superior performance. Although they lack the ability to deal with missing values directly, it can be an appropriate candidate to implement multi-view clustering in complete data settings.

2.3 Combination of MI and EC

One of the most significant advantage of multiple imputation is that it can recover the incomplete datasets without caring about the form of incompleteness. Given that most of existing multi-view clustering algorithms are confined to deal with view missing datasets, the introduction of multiple imputation to clustering could bring about a great improvement of the performance of algorithms. Furthermore, multiple imputation yields multitude of datasets and ensemble clustering is exactly good at processing large-scale datasets simultaneously. Therefore, combination of multiple imputation and ensemble clustering would be a brand-new and effective approach to implement clustering on incomplete multi-view datasets.

3 Incomplete multi-view clustering with multiple imputation and ensemble clustering

In this section, we present our MIEC method in detail. It is split into two stages: MI and EC. We utilize MI to fill the incomplete datasets to obtain a set of complete datasets, and then perform the EC algorithm on the complete datasets to accomplish the clustering.

3.1 Multiple imputation

Compared with single imputation, MI takes the uncertainty of the missing value into consideration and fill in more robust missing values. Especially in statistics, MI has become the most popular and effective technique to deal with missing values. Just as we introduced in Section 2.1, there are many algorithms to implement MI. In the popular R package MICE [5], there are some common algorithms: Sample, RF(Random Forest), Cart, Misdatouch, et al.

These algorithms' performance vary on different datasets, thus it is necessary to select the most effective algorithms roughly. We donot need to identify the most effective one precisely, because its performance will also

depend on the EC strategy. The selection of the algorithms of MI can be found in Section 4.6.2

3.2 Ensemble clustering

As we have known that EC consists of two stages: generation and consensus function. In generation step, after we obtained the set of complete datasets derived from performing MI on one incomplete dataset, we could choose some clustering algorithms such as k-means to perform on each view of the complete datasets, respectively. Finally, each view of each complete dataset corresponds to one result which is a co-association matrix [23].

$X^v = \{X_1^v, X_2^v, \dots, X_{n_v}^v\} \in R^{d_v * n_v}$ denotes the data matrix of the v -th view where d_v is the number of the features and n_v is the number of samples. The ground truth $Y(i) = \{1, 2, \dots, k\}$ denotes which cluster the i -th sample belongs to. After generation step, we obtained many clustering results from each imputation completion of each view. Let V_1, V_2, \dots, V_M be the M cluster model for M views and Y_m be a cluster number assigned to a : $V_m(a) = Y_m$, where Y_m belongs to $1, 2, \dots, k$, and k denotes the number of clusters assigned by this result. Each cell of co-association matrix has the value:

$$h_m(i, j) = \delta(V_m(i), V_m(j))$$

where $\delta(a, b)$ is 1, if $a = b$, 0 otherwise.

For any pair of object(a, b), the true value of the matrix is:

$$Z = \delta(Y(a), Y(b))$$

In order to gather the information of all the results, we introduce a weighted method proposed by [3].

$$H = \sum_{m=1}^M \alpha_m \frac{1}{L_m} \sum_{l=1}^{L_m} h_m \quad (1)$$

where $\alpha_m \geq 0$ is weight of the m th view, M denotes the number of views, L_m denotes the number of iteration.

Let's introduce a probability model to accomplish the optimization of H.

The clustering accuracy can be represented as a probability:

$$P[h_m = 1 | Z = 1] = q_m \quad (2)$$

According to the Bayesian estimate of partitioning probability

$$\tilde{P}_m^B = \frac{\sum_l \tilde{h}_{l,m} + 1}{L_m + 2} \quad (3)$$

$$\tilde{q}_m^B = \max(\tilde{P}_m^B, 1 - \tilde{P}_m^B), m = 1, 2, \dots, M. \quad (4)$$

For a pair of sample (a, b), considering the margin of clustering ensemble :

$$mg = \{\text{weighted number of vote of } Z - \text{weighted number of against of } Z\}$$

which equals to

$$mg = \sum_m \alpha_m \frac{1}{L_m} \sum_l I[h_m = Z] - I[h_m \neq Z]. \quad (5)$$

where $Z \in \{0, 1\}$, it is proven that [3] the probability of wrong prediction of Z :

$$P_{err} < \frac{Var[mg(H, Z)]}{(E[mg(H, Z)])^2} \quad (6)$$

where $E[mg(H, Z)]$ is the expectation, and $Var[mg(H, Z)]$ is the variance of margin.

$$E[mg(H, Z)] = 2 \sum_m \alpha_m q_m - 1$$

$$Var[mg(H, Z)] = 4 \sum_m \frac{\alpha_m^2}{L_m} q_m (1 - q_m)$$

The optimization is altered to find the α_m when P_{err} is minimized. The final solution is

$$\alpha_m = \frac{\frac{L_m}{q_m(1-q_m)}}{\sum_m \frac{L_m}{q_m(1-q_m)}}, m = 1, 2, \dots, M \quad (7)$$

Given that we obtained the optimal H as co-association matrix, we perform the hierarchical agglomerative clustering to get the final cluster result. Hierarchical agglomerative clustering is a bottom-up clustering algorithm and we select “distance” criterion in our experiments. The algorithm treats each sample as a singleton cluster and then merge pairs of cluster to form flat cluster so that the original observations in each cluster have no greater cophenetic distance than “threshold” which is a parameter needed to be tuned.

Algorithm 1 MIEC: multiple imputation and ensemble clustering.

Input: Data matrices of incomplete datasets X^1, X^2, \dots, X^v with v views, the number of clusters k , the number of pieces of multiple imputation m .

- 1: Implement multiple imputation to turn the datasets X^1, X^2, \dots, X^v into m pieces complete data matrices $X_1^1, X_2^1, \dots, X_m^1, X_1^2, \dots, X_m^v$;
- 2: Perform k-means on each view of each complete data matrix obtained from Step 1;
- 3: Transform the results obtained from Step 2 into a co-association matrices h_m ;
- 4: Compute the Bayesian estimate of partitioning probability matrix $\tilde{P}_B^1, \tilde{P}_B^2, \dots, \tilde{P}_B^v$ and $\tilde{q}_B^1, \tilde{q}_B^2, \dots, \tilde{q}_B^v$ for each view of co-association matrices of m pieces data matrices;
- 5: Compute the weight α_m for each view based on (7);
- 6: Compute the weighted co-association matrix with (1);
- 7: Perform the hierarchical agglomerative clustering on the weighted co-association matrix to obtain the final assignment of the samples to k clusters;

Output: Assignment of samples to k clusters.

4 Experiment

4.1 Datasets

We choose five complete datasets to perform our algorithm and make comparisons with seven other algorithms. The brief descriptions of the datasets are as follows:

- **LEAVES**¹ consists of 1600 samples containing 100 features in each of three views. This dataset is derived from UCI Machine Learning Repository. 16 leaf samples for each of 100 plant species are collected, and for each sample, a shape descriptor, fine scale margin and texture histogram are given as three different views.
- **ORL**² is the data set of the face images from 40 distinct subjects in AT&T Lab. There are ten different images for each of 40 subjects. Each image is 92×112 pixels, with 256 grey levels per pixel. We used a subset with 10 subjects.
- **COIL10** is subsampled from Columbia Object Image Library (COIL100) [6, 30], which contains 32×32 gray scale images of 100 objects viewed from varying angles. The objects are placed on a motorized turntable against a black background. The turntable is rotated through 360 degrees to vary object pose with respect to a fixed color camera. Images of the objects are taken at pose intervals of 5 degrees. This corresponds to 72 poses per object.
- **YALE** [25] consists of 165 samples. Each of them contains 4096, 3304, 6075 features in the three views. This dataset includes 165 images of raw pixel, with different conditions, lighting conditions, facial expression, illumination, etc.
- **UCIDIGIT**³ consists of 2000 samples with 64,76,216 features in each of three views. This dataset is derived from UCI Machine Learning Repository. It represents features of handwritten numerals extracted from a collection of Dutch utility maps. There are 200 samples for each of 10 classes in the dataset. In the case of our experiment, we only choose three feature sets: 76 Fourier coefficients of the character shapes, 216 profile correlations, 64 Karhunen-Love coefficients as three views.

Considering that some of these datasets contain too many features for each view, we perform PCA (principal component analysis) on YALE, UCIdigit and ORL to pick up a group of the most important features in order to keep as much information as possible. These processed datasets

¹<https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set>

²www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

³<http://archive.ics.uci.edu/ml/datasets/Multiple+Features>

Table 1 Summary of the complete datasets

dataset	# samples	# views	# features in each view	# clusters
LEAVES	1600	3	100,100,100	100
YALE	165	3	125,125,125	6
COIL	1440	3	30,19,30	10
UCIDIGIT	2000	3	64,76,80	20
ORL	400	4	125,125,125,125	40

kept over 95% information via PCA. The details of datasets are listed in Table 1.

4.2 Dataset processing

Since we aim to deal with incomplete multi-view clustering, with the complete datasets, we will miss some values to obtain the incomplete datasets to work with. As aforementioned in Introduction, there are two missing value cases: any value missing and view missing. Accordingly, two missing mechanisms are designed to produce them. In the first mechanism, datasets are processed to miss some data in completely random way. In the second mechanism, datasets are processed in the following rules that a specific ratio of samples will lose all the features in some views while other views will be complete. We conduct this procedure in each view separately. The detailed procedure of the two mechanisms are as follows:

For the first mechanism, we generated each indicator matrix with the same size of dataset matrix X^v (v indicates v -th view), each entry is generated from a uniform distribution on (0,1), and then we replaced those elements below the threshold 0.05 (0.1 or 0.2) with NAN and other elements with 1. In this way, an indicator matrix A^v is obtained for each view. Element-wise product of the original dataset matrix X^v and the indicator matrix A^v gets the incomplete multi-view dataset. It is noted that the threshold represents the ratio of missed elements.

For the second mechanism, we firstly generated a random vector with the length of N (N indicates number of samples) from an uniform distribution on (0,1), and then we replaced those elements below the threshold 0.05 (0.1 or 0.2) with NAN and other elements with 1. In this way, a random vector b^v is obtained from each view, and then a diagonal matrix B^v with diagonal vector as b^v is obtained. If the original dataset is X^v , $B^v \times X^v$ will generate the view missing datasets.

4.3 Compared methods

Corresponding to two missing value cases: view missing and any value missing, we name our method MICE to deal

with them as MICE_VM and MICE_AM. To demonstrate the effectiveness of MICE_VM and MICE_AM, we compare them with the following algorithms.

- **Incomplete Multiview Spectral Clustering with Adaptive Graph Learning (IMSC)** [13]: IMSC combines graph learning with spectral clustering to propose a framework to deal with incomplete multi-view clustering. It uses low-rank representation to construct the graph for each view and then conduct multi-view clustering in co-regularization way. It can only deal with view missing datasets.
- **View Variation and View Heredity for Incomplete Multi-view Clustering (V^3H)** [15]: V^3H introduces the concept of variation and heredity of genetics to exploit the multi-view data information. It decomposes the subspace of data into variation matrix and heredity matrix and then integrates unique information of different views to recover incomplete datasets.
- **Perturbation-oriented Incomplete multi-view Clustering (PIC)** [36]: The method aims to tackle the problem of incomplete multi-view clustering. It presents a link between perturbation risk bounds and incomplete multi-view clustering and transfers the missing problem of incomplete datasets to similarity matrix computing and finds the minimization of perturbation risk bounds. However, it can only deal with view missing case.
- **Doubly Aligned Incomplete Multi-view Clustering (DAIMC)** [26]: DAIMC solves the incomplete multi-view clustering problem by introducing a weighted matrix to indicate whether the value in corresponding position is lost in each view and learning a consensus matrix to combine the information from multiple views.
- **Multiple Incomplete Views Clustering (MIC)** [37]: MIC performs weighted nonnegative matrix factorization on each incomplete view and then utilizes the co-regularized way to get the consensus matrix obtained from the process of learning the latent feature matrix of all views, which minimizes the influence of missing data.
- **Partial Multi-View Clustering (PVC)** [31]: PVC could solve the problem of missing data in a dataset, however, it can only handle two-view datasets. According to the method [26], with all the two-view combination of multi-view datasets, PVC is performed and the best result is selected as the final result.
- **Best Single View (BSV)**: It performs non-negative matrix factorization on each single view respectively and then pick the best clustering result as the final result.

4.4 Evaluation metrics

To evaluate the clustering performance, We use four evaluation metrics: accuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI) and F-Scores [12, 38]. Their definitions are given as follows:

- **Accuracy (ACC).** Accuracy discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. It sums up the whole matching degree between all pair class-clusters.

$$ACC = \frac{1}{N} \max(\sum_{X_i, Y_j} N(X_i, Y_j)) \tag{8}$$

where X_i denotes the i -th cluster in the final results, and Y_j is the true j -th class. $N(X_i, Y_j)$ is the number of samples which belong to class j are assigned to cluster i . Accuracy computes the maximum sum of $N(X_i, Y_j)$ for all pairs of clusters and classes, and these pairs have no overlaps. The greater clustering accuracy means the better clustering performance.

- **Normalized Mutual Information (NMI).** For two random variables X and Y , the NMI is defined as:

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}} \tag{9}$$

where $I(X, Y)$ is the mutual information between X and Y , while $H(X)$ and $H(Y)$ are the entropies of X and Y , respectively. Clearly, NMI takes the value from [0, 1], the higher NMI means the better the clustering performance.

- **Adjusted Rand Index.** Adjusted Rand index(ARI) is a corrected-for-chance version of the rand index. ARI assumes the generalized hypergeometric distribution as the model of randomness to deal with the problem of rand index, which is the effect of random labels to evaluation results. Both ARI and random index are calculated by the following four values.

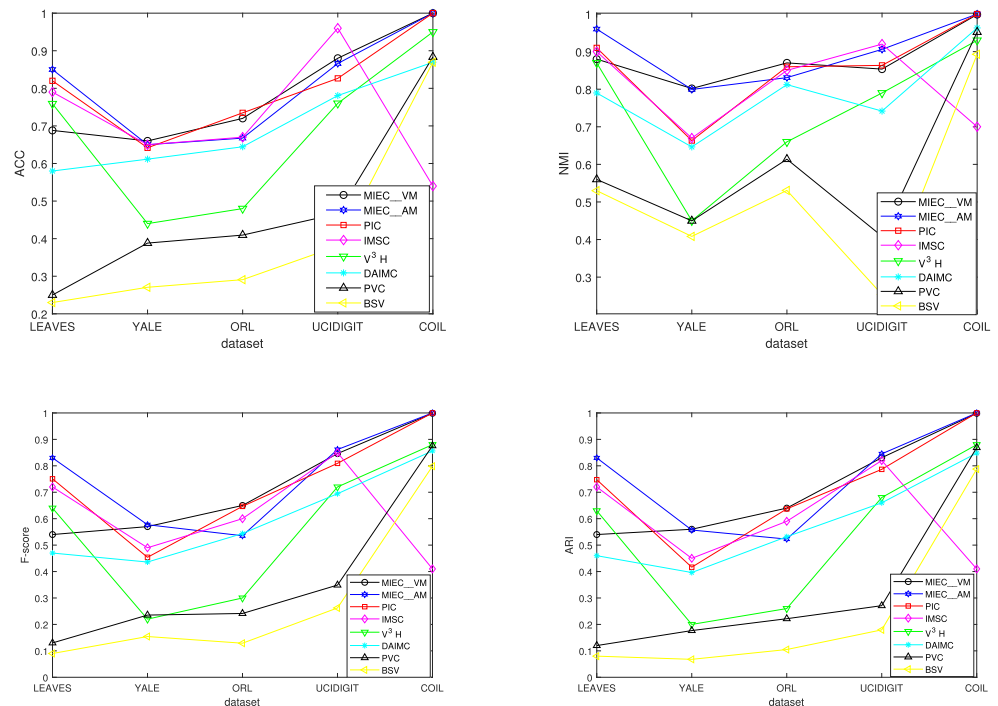
A true positive (TP) decision assigns two similar documents to the same cluster, a true negative (TN) decision assigns two dissimilar documents to different clusters. There are two types of errors we can commit. A false positive (FP) decision assigns two dissimilar documents to the same cluster. A false negative (FN) decision assigns two similar documents to different clusters. Rand index measures the percentage of decisions that are correct.

$$ARI = \frac{2 \times (TP \cdot TN - FN \cdot FP)}{(TP + FN)(TN + FN) + (TP + FP)(FP + TN)} \tag{10}$$

Table 2 Clustering results of nine methods on five datasets with 10% missing ratio. The best one is emphasized with bold font and the second best one is shown underlined

Dataset	Method	ACC	NMI	F	ARI
100LEAVES	MIEC_VM	0.688	0.888	0.624	0.620
	MIEC_AM	0.853	0.955	0.832	0.833
	PIC	<u>0.821</u>	<u>0.910</u>	<u>0.750</u>	<u>0.748</u>
	IMSC	0.788	0.901	0.719	0.716
	V ³ H	0.759	0.874	0.637	0.633
	DAIMC	0.590	0.792	0.470	0.464
	MIC	0.576	0.775	0.396	0.389
	PVC	0.247	0.562	0.129	0.118
	BSV	0.230	0.536	0.093	0.079
YALE	MIEC_VM	0.661	0.802	<u>0.563</u>	<u>0.543</u>
	MIEC_AM	0.649	<u>0.799</u>	0.577	0.557
	PIC	0.642	0.663	0.454	0.416
	IMSC	<u>0.660</u>	0.671	0.485	0.450
	V ³ H	0.437	0.453	0.223	0.170
	DAIMC	0.612	0.646	0.436	0.396
	MIC	0.079	0.183	0.122	0.010
	PVC	0.388	0.450	0.235	0.177
	BSV	0.271	0.408	0.154	0.068
ORL	MIEC_VM	<u>0.715</u>	0.877	0.663	0.655
	MIEC_AM	0.668	0.831	0.536	0.523
	PIC	0.735	<u>0.860</u>	<u>0.647</u>	<u>0.637</u>
	IMSC	0.670	0.847	0.600	0.590
	V ³ H	0.475	0.661	0.280	0.260
	DAIMC	0.645	0.812	0.544	0.532
	MIC	0.035	0.205	0.049	0.010
	PVC	0.410	0.614	0.241	0.222
	BSV	0.291	0.531	0.129	0.105
UCIDIGIT	MIEC_VM	0.880	0.854	0.846	<u>0.831</u>
	MIEC_AM	<u>0.866</u>	<u>0.906</u>	0.862	0.845
	PIC	0.827	0.863	0.809	0.787
	IMSC	0.959	0.918	<u>0.850</u>	0.825
	V ³ H	0.758	0.789	0.719	0.677
	DAIMC	0.781	0.742	0.694	0.660
	MIC	0.100	0.009	0.181	0.001
	PVC	0.467	0.409	0.349	0.271
	BSV	0.380	0.254	0.262	0.179
COIL	MIEC_VM	1	1	1	1
	MIEC_AM	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>
	PIC	1	1	1	1
	IMSC	0.535	0.702	0.440	0.407
	V ³ H	0.950	0.934	0.885	0.879
	DAIMC	0.869	0.963	0.856	0.848
	MIC	0.051	0.026	0.094	0.010
	PVC	0.883	0.952	0.876	0.869
	BSV	0.869	0.893	0.799	0.788

Fig. 2 Clustering performance of nine algorithms on five datasets with 10% missing ratio



- **F-score** The F-score is a way of combining precision and recall of the model, it is defined as the harmonic mean of precision and recall.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (11)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (12)$$

$$\text{F-score} = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

For all these four evaluation metrics, the higher value means better performance.

4.5 Experimental results

We perform our MIEC_VM, MIEC_AM and seven baselines on five datasets with 10% missing ratio to compare their performance. Table 2 and Fig. 2 report the performance in four evaluation metrics ACC, NMI, ARI and F-score.

According to Table 2 and Fig. 2, we get the following analysis:

- **Outstanding performance to handle datasets with view missing:** Since other compared algorithms can only deal with view missing case, here we compare our MIEC_VM with other algorithms on the five datasets with 10% missing ratio. According to Fig. 2, our algorithm MIEC_VM outperforms the baselines

on all the datasets except IMSC on UCIDIGIT dataset. One reason why our algorithm shows the best performance is that the method adopts multiple imputation to deal with missing values, which can take the uncertainty of missing value into consideration. These baselines (DAIMC, PVC, BSV) complete the incomplete datasets with the average features, which renders huge deviations. The PIC utilizes a subtle method to transform the incomplete features into incomplete similarity entries which makes use of the values in all views, as a result, PIC performs better than the other baselines. For our algorithms, we use multiple imputation to get many sets of the complete datasets, and then perform k-means clustering algorithm on those imputed datasets. Finally, we get the weighted results in co-association matrix which makes full use of all the existing data as well as lots of imputed data from multiple imputation.

- **Handling the datasets with any value missing:** The significant advantage of our algorithm is that it is capable of handling with the datasets with any missing values (MIEC_AM). By contrast, the baselines could only deal with the datasets with view missing. Meanwhile, according to Table 2, the performance of MIEC_AM is always the best among all the algorithms, which indicates that our MIEC_AM could tackle the datasets with any value missing successfully.

Table 3 The performance of MIEC_VM with different missing ratio. The best one is emphasized with bold font

Dataset	ratio	ACC	NMI	F	ARI
100LEAVES	5%	0.683	0.896	0.628	0.624
	10%	0.688	0.888	0.624	0.620
	20%	0.683	0.893	0.628	0.624
YALE	5%	0.673	0.815	0.604	0.588
	10%	0.661	0.802	0.563	0.543
	20%	0.697	0.796	0.607	0.589
ORL	5%	0.733	0.882	0.666	0.659
	10%	0.715	0.877	0.663	0.655
	20%	0.690	0.872	0.668	0.662
UCIDIGIT	5%	0.828	0.869	0.815	0.794
	10%	0.880	0.854	0.846	0.831
	20%	0.768	0.792	0.725	0.692
COIL	5%	1	1	1	1
	10%	1	1	1	1
	20%	1	1	1	1

4.6 Parameter analysis

4.6.1 The effect of missing ratio

In the above experiments, the parameter missing ratio is fixed to 10%. Herein, we will explore the effect of missing ratio. Tables 3 and 4 present the four metrics of MIEC_VM

Table 4 The performance of MIEC_AM with different missing ratio. The best one is emphasized with bold font

Dataset	ratio	ACC	NMI	F	ARI
100LEAVES	5%	0.849	0.956	0.833	0.831
	10%	0.853	0.955	0.832	0.830
	20%	0.822	0.954	0.821	0.819
YALE	5%	0.673	0.790	0.582	0.559
	10%	0.649	0.799	0.577	0.557
	20%	0.521	0.746	0.456	0.428
ORL	5%	0.675	0.849	0.582	0.572
	10%	0.668	0.831	0.536	0.523
	20%	0.573	0.784	0.461	0.449
UCIDIGIT	5%	0.856	0.892	0.845	0.827
	10%	0.866	0.910	0.862	0.845
	20%	0.860	0.889	0.783	0.825
COIL	5%	1	1	1	1
	10%	1	1	1	1
	20%	0.988	0.990	0.987	0.986

Table 5 The performance of MIEC_VM obtained with different methods of MI. The best one is emphasized with bold font

Method	ACC	NMI	F	ARI
PMM	0.688	0.888	0.624	0.620
RF	0.355	0.666	0.237	0.226
Cart	0.609	0.854	0.513	0.507
Sample	0.645	0.869	0.552	0.547
Midastouch	0.362	0.670	0.251	0.241

and MIEC_AM on all the datasets when the missing ratio is set to 5%, 10%, 20%. We could see that the performance decreases with missing ratio increasing, while the extent of decline is becoming smaller, which means that our algorithm is not sensitive to the missing ratio when missing ratio is less than 20%.

4.6.2 The effect of methods of multiple imputation

In order to explore the effect of the method of MI, we select some popular methods including PMM, Sample, RF, Misdatouch to perform MICE_VM and MICE_AM on dataset LEAVES. Tables 5 and 6 present the results with four clustering metrics ACC, NMI, F, ARI.

It turns out that MIEC_VM performs the best with the method of “PMM”, while MIEC_AM performs the best with the method of “Cart”. Comprehensively, we select “PMM” as our imputation method in all the experiments.

4.6.3 The effect of weighted process

When it comes to ensemble clustering, MIEC_AM and MIEC_VM assign the weights to the co-association matrix according to the correctness of the clustering results obtained from co-association matrix of each single view, and then calculate the weighted average co-association matrix of all the views to perform consensus function. To check the effectiveness of the weighting process, we conduct the comparison experiments on two datasets 100LEAVES and YALE with or without weighting process. The results are shown in Table 7.

Table 6 The performance of MIEC_AM obtained with different methods of MI. The best one is emphasized with bold font and the second best one is shown underlined

Method	ACC	NMI	F	ARI
PMM	<u>0.785</u>	<u>0.929</u>	<u>0.750</u>	<u>0.748</u>
RF	0.438	0.719	0.327	0.319
Cart	0.802	0.937	0.772	0.770
Sample	0.716	0.908	0.677	0.674
Midastouch	0.466	0.729	0.348	0.340

Table 7 The Performance of MIEC with or without Weighting Process. The best one is emphasized with bold font

Dataset	Algorithm	Weighted or not	ACC	NMI	F	ARI
100LEAVES=	MIEC_AM	Without	0.794	0.929	0.753	0.751
	MIEC_AM	With	0.853	0.955	0.832	0.830
100LEAVES=	MIEC_VM	Without	0.626	0.859	0.527	0.522
	MIEC_VM	With	0.688	0.888	0.624	0.620
YALE=	MIEC_AM	Without	0.612	0.743	0.516	0.489
	MIEC_AM	With	0.649	0.799	0.577	0.557
YALE=	MIEC_VM	Without	0.649	0.776	0.561	0.539
	MIEC_VM	With	0.661	0.802	0.563	0.543

In Table 7, we have made a comparison between weighted process and average process. Through the observation, we can find that both MIEC_VM and MIEC_AM with weighting process perform better than those without weighting process, which indicates that the weight assignment improves the performance of our algorithm on all the evaluation metrics.

5 Conclusion

In this paper, we proposed an incomplete multi-view clustering method named MIEC. This method is a two stage algorithm, the first stage adopts the most popular missing value handling technique multiple imputation to complete the multi-view data, the second stage designs a weighted ensemble clustering algorithm to implement multi-view clustering considering the different contributions of multiple views. As far as we know, this is the first time to combine multiple imputation and ensemble clustering to conduct incomplete multi-view clustering. Compared with many existing incomplete multi-view clustering methods who just tackle view missing case, our proposed MIEC can deal with more general case any value missing. Additionally, the experiments on several datasets demonstrated that our proposed methods obtained better performance than existing incomplete multi-view clustering algorithms in terms of four evaluation metrics ACC, NMI, F, ARI. Considering that splitting the method into two stages may result in some gaps, in future work, we are interested in proposing an incomplete multi-view clustering method by conducting multiple imputation and ensemble clustering simultaneously.

Acknowledgment This work is supported by the National Natural Science Foundation of China under Project 61772159, 61902090 and National Key Research and Development Project of China under Project 61832004.

References

- Alqurashi T, Wang W (2018) Clustering ensemble method. *Int J Machine Learn Cybern* 10(5):1–18
- Ben-Hur W, Elisseeff A, Guyon I (2002) A stability based method for discovering structure in clustered data. *Biocomputing* 6–17
- Berikov V (2014) Weighted ensemble of algorithms for complex data clustering. *Pattern Recogn Lett* 39(3):99–106
- Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: *Proceedings of the eleventh annual conference on Computational learning theory*, pp 92–100
- Buuren SV, Groothuis-oudshoorn K (2011) Mice: Multivariate imputation by chained equations in r. *J Stat Softw* 45(3):1–67
- Cai D, He X, Han J, S.Huang T. (2011) Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans Pattern Anal Machine Intell* 33(8):1548–1560
- Chang X, Dacheng Tao CX (2007) A survey on multi-view learning arxiv
- Chao G, Sun J, Lu J, Wang AL, Langleben DD, Li CS, Bi J (2019) Multi-view cluster analysis with incomplete data to understand treatment effects. *Inf Sci* 494
- Chao G, Sun S (2016) Alternative multiview maximum entropy discrimination. *IEEE Trans Neural Netw Learn Syst* 27(7):1445–1456
- Chao G, Sun S (2016) Consensus and complementarity based maximum entropy discrimination for multi-view classification. *Inf Sci* 367:296–310
- Chao G, Sun S (2018) Semi-supervised multi-view maximum entropy discrimination with expectation laplacian regularization. *Inform Fusion* 45:296–306
- Chao G (2018) Discriminative k-means laplacian clustering. *Neural Process Lett* 1–13
- Jie W, Yong X, Hong L (2020) Incomplete multiview spectral clustering with adaptive graph learning. *IEEE Trans Cybern* 50:1418–1429
- Sun S, Xie X, Dong C (2019) Multiview learning with generalized eigenvalue proximal support vector machines. *IEEE Trans Cybern* 49:688–697
- Xiang F, Yuchong H, Pan Z (2020) V³h:view variation and view heredity for incomplete multi-view clustering. *IEEE Trans Artif Intell* 1:233–247
- Xie X (2020) Multi-view semi-supervised least squares twin support vector machines with manifold-preserving graph reduction. *Int J Machine Learn Cybern* 11:2489–2499

17. Xie X, Sun S (2019) General multi-view learning with maximum entropy discrimination. *Neurocomputing* 332:184–192
18. Xie X, Sun S (2020) Multi-view support vector machines with the consensus and complementarity information. *IEEE Trans Knowl Data Eng* 32:2401–2413
19. Xie Xijiong, Sun Shiliang (2020) General multi-view semi-supervised least squares support vector machines with multi-manifold regularization. *Inf. Fusion* 62:63–72
20. Domeniconi C, Al-Razgan M (2009) Weighted cluster ensembles: Methods and analysis. *ACM Trans Knowl Discov Data* 2(4):1–40
21. Fred AL, K.Jain A (2002) Data clustering using evidence accumulation. In: *Proceedings of the 16th International Conference of Pattern Recognition*, pp 276–280
22. Fred AL, K.Jain A (2002) Evidence accumulation clustering based on the k-means algorithm. In: *Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition*, pp 442–451
23. Fred ALN, Jain AK (2005) Combining multiple clusterings using evidence accumulation. *IEEE Trans Pattern Anal Machine Intell* 27(6):835–50
24. Guoqing C, Shiliang Sun JB (2021) A survey on multi-view clustering. *IEEE Trans Artif Intell* 2(2):146–168
25. Handong Z, Zhengming Ding YF (2017) Multi-view clustering via deep matrix factorization. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp 2921–2927
26. Hu M, Chen S (2019) Doubly aligned incomplete multi-view clustering. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp 2262–2268
27. Hussain SF, Mushtaq M, Halim Z (2014) Multi-view document clustering via ensemble method. *J Intell Inform Syst* 43(1):81–99
28. Mirkin B (1997) *Mathematical classification and clustering*. J Oper Res Soc 48
29. M.Rand W (1971) Objective criteria for the evaluation of clustering methods. *J Am Stat Assoc* 66(4):846–850
30. Nene SA, Nayar SK, Murase H et al (1996) Columbia object image library (COIL-20) Technical report CUCS-005-96
31. Shao-Yuan Li YJ, Zhou ZH (2014) Partial multi-view clustering. In: *Proceedings of the National Conference on Artificial Intelligence*, vol 3, pp 1968–1974
32. Strehl A, Ghosh J (2003) Cluster ensembles -a knowledge reuse framework for combining multiple partitions. *J Machine Learn Res* 3(3):583–617
33. Sun S (2013) A survey of multi-view machine learning. *Neural Comput Applic* 23(7-8):2031–2038
34. Tao Z, Liu H, Li S, Ding Z, Fu Y (2017) From ensemble clustering to multi-view clustering. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp 2843–2849
35. Vega-Pons S, Ruiz-Shulcloper J (2011) A survey of clustering ensemble algorithm. *Int J Pattern Recognit Artif Intell* 25(3):337–372
36. Wang H, Zong L, Liu B, Yang Y, Zhou W (2019) Spectral perturbation meets incomplete multi-view data. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp 3677–3683
37. Weixiang Shao LH, Yu PS (2015) Multiple incomplete views clustering via weighted nonnegative matrix factorization with l_2, l_1 regularization. In: *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*
38. Xu W, Liu X, Gong Y (2003) Document clustering based on non-negative matrix factorization. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp 267–273
39. Yang Y, Wang H (2018) Multi-view clustering: a survey. *Big Data Mining and Analytics* 1(2):83–107
40. Zhao J, Xie X, Xu X, Sun S (2017) Multi-view learning overview: Recent progress and new challenges. *Inform Fusion* 38:43–54

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.