Neurocomputing 402 (2020) 148–161

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# An overview of recent multi-view clustering

Lele Fu[a,b], Pengfei Lin[a,b], Athanasios V. Vasilakos[a,b], Shiping Wang[a,b,*]

[a] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China
[b] Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China

ARTICLE INFO

*Article history:*
Received 2 December 2019
Revised 11 February 2020
Accepted 27 February 2020
Available online 16 March 2020

Communicated by Dr. Nianyin Zeng

*Keywords:*
Machine learning
Unsupervised learning
Multi-view clustering
Graph-based clustering
Space learning

ABSTRACT

With the widespread deployment of sensors and the Internet-of-Things, multi-view data has become more common and publicly available. Compared to traditional data that describes objects from single perspective, multi-view data is semantically richer, more useful, however more complex. Since traditional clustering algorithms cannot handle such data, multi-view clustering has become a research hotspot. In this paper, we review some of the latest multi-view clustering algorithms, which are reasonably divided into three categories. To evaluate their performance, we perform extensive experiments on seven real-world data sets. Three mainstream metrics are used, including clustering accuracy, normalized mutual information and purity. Based on the experimental results and a large number of literature reading, we also discuss existing problems in current multi-view clustering and point out possible research directions in the future. This research provides some insights for researchers in related fields and may further promote the development of multi-view clustering algorithms.



## 1. Introduction

Clustering is a relatively traditional problem that has existed for a long time. In the information age, the amount of data is exploding exponentially, so it is more necessary to rationally classify similar objects to reduce the degree of data confusion, which helps the researchers more easily distinguish its inherent logic. At present, clustering algorithms are widely used in data mining [1,2], computer vision [3–6], pattern recognition [7,8] and other fields [9–12]. Some well-known clustering algorithms such as k-means, spectral clustering etc. play an increasingly important role.

After years of development, the research of traditional single view clustering has almost reached the bottleneck. The main reason for this situation is that the data sets only describe objects from single aspect, so that they do not accurately grasp the comprehensive information of the objects. With the rapid development of multimedia technology, the acquisition of data is not as tight as before. Multi-view data has begun to emerge in large numbers, which means that the same objects are described from different perspectives. For example, news can be reported in multiple languages, and the same people can be photographed by cameras from diverse directions. Although the final data may differ in form, they all represent the same thing. Thus, the applications of multi-view learning [13–15] in clustering problems produce many novel multi-view clustering algorithms for multi-view data.

The most primitive multi-view clustering is to simply stitch all data features and then use them for clustering. This obviously does not take advantage of the information complementarity between different views, but also does not have any interpretability. Then, some new methods [16,17] based on hybrid models and expectation maximization (EM) algorithms [18] were proposed, whose core ideas are to learn a model for each cluster and integrate them into a unified model. Due to the simplicity of k-means, many researchers have tried to construct models for multi-view data based on its theory, including papers [19–21]. In current work, the multi-view clustering algorithms derived from spectral clustering and other graph-based methods account for a large part. In these works [22–25], how to combine the data of different views according to their actual contribution level becomes the focus. After all views are merged, some works modify the traditional spectral clustering framework. In other words, they are not committed to obtaining a uniform similarity matrix but learning a final indicator matrix, which is used directly for clustering. Subspace clustering [26–28] can effectively reduce the dimension of data, which has also been noticed in the field of multi-view clustering due to this unique nature, the works [29–33] are to learn the subspace representation of data for clustering with the guidance of its idea. Multi-view clustering based on non-negative matrix factorization [34–38] is designed to learn the indicator matrix, whose mathematical form is very similar to multi-view subspace clustering, the

* Corresponding author at: College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China.
*E-mail address:* shipingwangphd@163.com (S. Wang).

https://doi.org/10.1016/j.neucom.2020.02.104
0925-2312/

distinction is that the dimensions and uses of the final learned matrices are different from the former. Besides, the fusion of different data features should take into account that the gap between them may be very large in reality, the work [39] projects the original data from diverse views into the same lower dimensional data space, in which Canonical Correlation Analysis (CCA) is employed to measure the correlation errors, and then groups the data with some conventional clustering algorithms. On the contrary, the papers [3,40] proposed to adopt the kernel version of CCA to project data into high-dimensional space.

Currently, the reviews on multi-view clustering are very rare, so our work is very necessary. In conclusion, the contributions of this paper are as follows:

- The algorithms we introduce are all from 2016 onwards, aiming at the latest developments in the field of multi-view clustering, filling in gaps in other reviews over time.
- Each algorithm is described more detailedly in order to better express their core ideas. In addition, we divide the methods into graph based model, space learning based model and binary code learning based model, which are easy to grasp the similarities and differences between these algorithms.
- We use seven data sets to test all algorithms, and apply three metrics to evaluate them. Finally, we present the final results in tabular form, and provide some visualized results to facilitate the understanding of algorithm performance more intuitively.
- Based on extensive literature reading and experimental results, we point out future potential research directions for current challenges in the field of multi-view clustering, in order to provide some perspectives for researchers interested in this field.

The following sections are arranged as follows. In Section 2, we introduce eight multi-view clustering algorithms, showing their core formulas and main steps. In Section 3, data sets, evaluation metrics, experimental results and visualization are presented. In Section 4, current challenges and future research directions are discussed. In Section 5, we summarize the whole paper.

## 2. Overview of recent multi-view clustering algorithms

To begin with, the meanings of some commonly used mathematical symbols need to be hereby stipulated. $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^m\}$ represents a data set with $m$ views, where $\mathbf{X}^v \in \mathbb{R}^{d^v \times n}$. A few algorithms require the data set input dimension to be $n \times d^v$, we will emphasize this difference in these algorithms. $\{\mathbf{x}_i^v\}_{i=1}^n$ is the set of samples in the $v$th view. Identity matrix and the column vector in which each element is 1 are represented by **I, 1** respectively, which has different dimensions in specific algorithms. **0** denotes a column vector or a matrix with all elements of 0 in diverse scenes. $\mathbf{L_S}$ denotes the Laplacian matrix constructed by the similarity matrix **S** and $Tr(\cdot)$ denotes the trace of a matrix.

Next, we introduce eight multi-view clustering algorithms according to the classification method of graph-based model, space-learning-based model and binary-code-learning-based model, respectively.

### 2.1. Graph-based model

Graph-based clustering algorithm is one of the most popular methods at present. It revolves around the goal of constructing the similarity matrix of data, and then adopts the typical spectral clustering algorithm or some other methods to carry out the final label distribution. The construction of the graph-based algorithm model is also concerned by the multi-view clustering field.

#### 2.1.1. AMGL

In graph-based multi-view clustering, how to merge multiple views is the key, and the core of this step is to assign an appropriate weight to each view. Some existing works [39,41] do not mind about the importance of different views, and others [22,24,42] learn the weight of each view by introducing additional hyper parameters, but the influence of hyper parameters selection on the clustering effect cannot be underestimated. Thus, Nie et al. [43] proposed the Parameter-Free Auto-Weighted Multiple Graph Learning (AMGL), which implements the automatic allocation of weights by modifying the traditional spectral clustering model and does not require any hyper parameters. Furthermore, the general framework in this paper can be applied to both multi-view clustering and semi-supervising classification, while the latter is not concerned in our paper.

In spectral clustering, the final objective function is written as

$$\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \mathrm{Tr}\left(\mathbf{F}^T\mathbf{L_S}\mathbf{F}\right). \tag{1}$$

Based on the above formula, the authors proposed AMGL, whose mathematical expression is as follows:

$$\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \sum_{v=1}^{m} \sqrt{\mathrm{Tr}\left(\mathbf{F}^T\mathbf{L}_\mathbf{S}^v\mathbf{F}\right)}. \tag{2}$$

In this formula, no definition of weight factors appears, but after constructing the Lagrange function of Eq. (2), further finding partial derivative for **F** and setting the derivative to zero, the weight factor $w^v$ will be included in the formula, the pivotal two steps are shown below:

$$\sum_{v=1}^{m} \sqrt{\mathrm{Tr}\left(\mathbf{F}^T\mathbf{L}_\mathbf{S}^v\mathbf{F}\right)} + \mathcal{G}(\Lambda, \mathbf{F}). \tag{3}$$

$$\sum_{v=1}^{m} w^v \frac{\partial \mathrm{Tr}\left(\mathbf{F}^T\mathbf{L}_\mathbf{S}^v\mathbf{F}\right)}{\partial \mathbf{F}} + \frac{\partial \mathcal{G}(\Lambda, \mathbf{F})}{\partial \mathbf{F}} = 0. \tag{4}$$

Here $\Lambda$ indicates the Lagrange multiplier, and the second item in Eq. (3) represents the formalized term according to the constraint on **F**. The most fascinating thing is that after derivation, it is apt to find that the mathematical expression of $w^v$ is in the following form:

$$w^v = 1/(2\sqrt{Tr\left(\mathbf{F}^T\mathbf{L}_\mathbf{S}^v\mathbf{F}\right)}). \tag{5}$$

However, $w^v$ is evidently not a fixed value and varies with the change of **F**. But when it is regarded as constant, Eq. (2) is converted to the following problem:

$$\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \sum_{v=1}^{m} w^v \mathrm{Tr}\left(\mathbf{F}^T\mathbf{L}_\mathbf{S}^v\mathbf{F}\right), \tag{6}$$

the above equation is used to calculate **F**. And then the value of $w^v$ is also updated according to Eq. (5), so that the optimal values of both can be obtained through an iterative process. At the same time, we can also see that if a view contributes a lot, the $\mathrm{Tr}\left(\mathbf{F}^T\mathbf{L}_\mathbf{S}^v\mathbf{F}\right)$ will be small, and the corresponding value of $w^v$ will become large, which is in line with the actual situation.

In order to highlight the difference between AMGL and the model of automatic learning weights that requires extra hyper parameters, it is advisable to write the objective function of the latter and compare it with the former.

$$\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}, w} \sum_{v=1}^{m} \left(w^v\right)^\gamma \mathrm{Tr}\left(\mathbf{F}^T\mathbf{L}_\mathbf{S}^v\mathbf{F}\right)$$
$$\text{s.t.} \sum_{v=1}^{m} w^v = 1, w^v \geq 0. \tag{7}$$

Here $\gamma$ is the so-called hyper parameter, its value is generally set to non-negative and is utilized to maintain the smooth of

the weights allocation. In the actual algorithm operation, its subtle changes may have a great impact on the performance of the algorithm. Apparently, there are no extra parameters in the AMGL model, and the optimal $w^v$ and $\mathbf{F}$ can be learned. Also, the calculation formula of $w^v$ shows that it is not completely independent, but closely related to the value of $\mathbf{F}$. The main steps of AMGL are shown in Algorithm 1.

**Algorithm 1** Parameter-Free Auto-Weighted Multiple Graph Learning

**Input:** $\mathbf{X} = \left\{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^m\right\}$, $\mathbf{X}^v \in \mathbb{R}^{n\times d^v}$, number of clusters k.
**Output:** Indicator matrix $\mathbf{F}$.
1: Initialize the weight of each view $w^v = \frac{1}{m}$; Calculate the Laplacian matrix $\mathbf{L}_\mathbf{S}^v$ corresponding to each view; Calculate $\mathbf{L}_\mathbf{S} = \sum_{v=1}^m w^v \mathbf{L}_\mathbf{S}^v$;
2: **while** not convergent **do**
3: Compute $\mathbf{F}$ via Eq. (6) and the 2 to $k+1$ smallest eigenvalues of $\mathbf{L}_\mathbf{S}$;
4: Update $w^v$ via Eq. (7);
5: **end while**

*2.1.2. SwMC*

It has been always a challenge to assign appropriate weights to multiple views in graph-based multi-view clustering. Some solutions have been proposed in papers [44–46], but they were either realized by human intervention or by prior knowledge, which does not guarantee that the distribution results are consistent with the actual contribution of each view. Nie et al. [47] proposed a feasible solution to address this problem, which automatically assigns appropriate weights to different views and is named Self-weighted Multiview Clustering (SwMC). Meanwhile, in view of the traditional graph-based clustering method execution process that once the target graph is solved, some simple clustering algorithms such as k-means are needed to assign each point to a specific cluster, SwMC can omit this step to mitigate the instability caused by the application of additional clustering methods.

The reason why SwMC can skip the postprocessing process is because it applies the Constrained Laplacian Rank (CLR) method [48] to multi-view clustering. CLR learns a new but more reliable similarity matrix $\mathbf{S}$ by introducing a matrix rank limit, which can be utilized for clustering directly. This sentence is expressed by a mathematical formula as shown:

$$\min_{\mathbf{s}_i\mathbf{1}_n=1, s_{ij}\geq 0, \text{rank}\,(\mathbf{L}_\mathbf{S})=n-c} \|\mathbf{S}-\mathbf{A}\|_F^2, \tag{8}$$

where $\mathbf{A}$ is the similarity matrix computed from original data. When employing CLR to multi-view clustering, it introduces a hyper parameter to increase constraints to get a better solution. Thus, the objective is written as

$$\min_{w^v,\mathbf{S}} \sum_{v=1}^m w^v \|\mathbf{S}-\mathbf{A}^v\|_F^2 + \gamma\|\mathbf{w}\|_2^2$$
$$\text{s.t. } w^v \geq 0, \mathbf{w}^T\mathbf{1}_m = 1, s_{ij}\geq 0, \mathbf{s}_i\mathbf{1}_n = 1, \text{rank}\,(\mathbf{L}_\mathbf{S}) = n-c, \tag{9}$$

where $\mathbf{A}^v$ indicates the similarity matrix corresponding to the $v$th view, $\mathbf{w}$ is a column vector consisting of $w^1, w^2, \ldots, w^m$ and the value of $\gamma$ is specified to be greater than 0. Furthermore, the weight distribution is ensured to be smooth under the constraint of the last item. However, the clustering result is extremely dependent on the value of $\gamma$, that is, too large and too small value will directly affect the assignment of weights, resulting in a decrease in clustering accuracy. Therefore, Nie et al. [47] further proposed a new algorithm model to remove the hyper parameter $\gamma$, namely SwMC. The new objective function is presented as follows:

$$\min_{s_{ij}\geq 0, \mathbf{s}_i\mathbf{1}_n=1, \text{rank}\,(\mathbf{L}_\mathbf{S})=n-c} \sum_{v=1}^m \|\mathbf{S}-\mathbf{A}^v\|_F. \tag{10}$$

This formula is elegant and concise. More subtly, we do not see the definition of weights in this equation. Nevertheless, after the derivation of the Lagrange multiplier method, the above formula is tuned to be the following form:

$$\min_{s_{ij}\geq 0, \mathbf{s}_i\mathbf{1}_n=1, \text{rank}\,(\mathbf{L}_\mathbf{S})=n-c} \sum_{v=1}^m w^v \|\mathbf{S}-\mathbf{A}^v\|_F. \tag{11}$$

Here $w^v = 1/(2\|\mathbf{S}-\mathbf{A}^v\|_F)$ and is considered to be fixed for solving $\mathbf{S}$. We should notice that the value of $w^v$ is naturally updated when $\mathbf{S}$ is calculated. The authors have proved that SwMC is convergent, so the optimal solutions of $\mathbf{S}$ and $w^v$ can be obtained after an iterative process. The general steps of the method are summarized as shown in Algorithm 2.

**Algorithm 2** Self-weighted Multi-view Clustering

**Input:** $\mathbf{A} = \left\{\mathbf{A}^1, \mathbf{A}^2, \cdots, \mathbf{A}^m\right\}$, $\mathbf{A}^v \in \mathbb{R}^{n\times n}$, number of clusters k.
**Output:** Similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$.
1: Initialize $w^v = \frac{1}{m}$ for each view;
2: **while** not convergent **do**
3: Compute $\mathbf{S}$ by solving Eq. (11);
4: Update $w^v$ by utilizing $w^v = 1/(2\|\mathbf{S}-\mathbf{A}^v\|_F)$;
5: **end while**

*2.1.3. MLAN*

Noise data is inevitably presented in data sets, which has a negative impact on the construction of similarity graph in graph-based multi-view clustering. Obtaining a reliable graph is important for safeguarding clustering effects, Nie et al. [49] proposed Multi-View Clustering and Semi-Supervised Classification with Adaptive Neighbours (MLAN), which can gain the optimal graph utilized to implement the final clustering through learning local manifold structure. In addition, MLAN automatically assigns an appropriate weight to each view to integrate all views.

How to solve the optimal solution of similar matrix is the core process of this paper. Suppose a data set $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ and all data points are preprocessed by the method proposed in [50]. Then, considering the learning of the local structure, Nie et al. [49] constructed the formula to solve the matrix $\mathbf{S}$:

$$\min_{\mathbf{s}_i\in\mathbb{R}^{n\times 1}} \sum_{i,j}^n \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 s_{ij} + \alpha\|\mathbf{S}\|_F^2$$
$$\text{s.t.} \quad \forall i, \mathbf{s}_i^T\mathbf{1} = 1, 0 \leq s_{ij} \leq 1, \text{rank}\,(\mathbf{L}_\mathbf{S}) = n-c. \tag{12}$$

Here the second item guarantees to avoid such situation that only the similarity value of the point closest to $\mathbf{x}_i$ is set to 1 and the remaining similarity values are set to 0. Furthermore, to better understand the constraint of rank $(\mathbf{L}_\mathbf{S}) = n-c$, we have to know this theorem [51]: The multiplicity $c$ of eigenvalue 0 of Laplacian matrix $\mathbf{L}_\mathbf{S}$ (non-negative) represents the number of components connected in the graph of the similarity matrix $\mathbf{S}$.

When the adaptive local structure learning applied for multi-view data, the authors tuned Eq. (12):

$$\min_{\mathbf{S}} \sum_v \sqrt{\sum_{i,j} \left\|x_i^v - x_j^v\right\|_2^2 s_{ij}} + \alpha\|\mathbf{S}\|_F^2$$
$$\text{s.t. } \mathbf{s}_i^T\mathbf{1} = 1, 0 \leq s_{ij} \leq 1, \ \text{rank}\,(\mathbf{L}_\mathbf{S}) = n-c. \tag{13}$$

According to Eq. (13), what needs to be emphasized in particular is that the algorithm will only update the same similarity matrix $\mathbf{S}$ from beginning to end, which satisfies the consistency between different views.

Based on the solution of the Lagrange multiplier algorithm, Eq. (13) is transformed into the following form:

$$\min_{\mathbf{S}} \sum_{v} w^v \sum_{i,j} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij} + \alpha \|\mathbf{S}\|_F^2$$
$$\text{s.t. } \mathbf{s}_i^T \mathbf{1} = 1, 0 \le s_{ij} \le 1, \text{rank}\left(\mathbf{L}_{\mathbf{S}}\right) = n - c, \tag{14}$$

where $w^v = 1/2\sqrt{\sum_{i,j}\left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}}$ and it is assumed to be a fixed value to solve for $\mathbf{S}$.

In order to meet the constraint of rank $(\mathbf{L}_{\mathbf{S}}) = n - c$, the authors employed the Ky Fan's Theorem [52] to Eq. (14)

$$\min_{\mathbf{S},\mathbf{F}} \sum_{v}^{m} w^v \sum_{i,j} \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij} + \alpha \|\mathbf{S}\|_F^2 + 2\lambda \,\text{Tr}\left(\mathbf{F}^T \mathbf{L}_{\mathbf{S}} \mathbf{F}\right)$$
$$\text{s.t. } \mathbf{s}_i^T \mathbf{1} = 1, 0 \le s_{ij} \le 1, \mathbf{F}^T \mathbf{F} = \mathbf{I}. \tag{15}$$

Here $\mathbf{F}$ is a $n \times c$ indicator matrix, which consists of the eigenvectors corresponding to the first $c$ smallest eigenvalues of the matrix $\mathbf{L}_{\mathbf{S}}$. The last item guarantees that the first $c$ minimum eigenvalues of $\mathbf{L}_{\mathbf{S}}$ are minimized, so as to tend to zero. Then, they adopted the strategy of fixing certain variables while updating other variables. When $\mathbf{S}$ is fixed, the value of $w^v$ can be computed easily, and the optimal solution of $\mathbf{F}$ can also be obtained by following equation:

$$\min_{\mathbf{F} \in \mathbb{R}^{n \times c}, \mathbf{F}^T \mathbf{F} = \mathbf{I}} \text{Tr}\left(\mathbf{F}^T \mathbf{L}_{\mathbf{S}} \mathbf{F}\right). \tag{16}$$

When $w^v$ and $\mathbf{F}$ are solved, the attention is transferred to the solution for $\mathbf{S}$. After derivation of mathematical formulas, the problem is written as follows:

$$\min_{\mathbf{s}_i} \quad \left\|\mathbf{s}_i + \frac{1}{2\alpha}\mathbf{d}_i\right\|_2^2 \quad \text{s.t. } \mathbf{s}_i^T \mathbf{1} = 1, 0 \le s_{ij} \le 1. \tag{17}$$

Here we should note the meaning of the parameter $\alpha$, it represents the number of neighbors at a certain point, which is constantly changed until the algorithm converges. $\mathbf{d}_i$ is a vector of $n \times 1$, whose the $j$th element $d_{ij}$ is equal to $d_{ij}^x + \lambda d_{ij}^f$. Certainly, computing rules $d_{ij}^x$ and $d_{ij}^f$ are defined beforehand in the original paper. Nie et al. [49] have proved that MLAN will eventually converge in algorithm execution. The overall steps of MLAN are shown in Algorithm 3.

**Algorithm 3** Multi-view Learning with Adaptive Neighbours

**Input:** $\mathbf{X} = \left\{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^m\right\}$, $\mathbf{X}^v \in \mathbb{R}^{n \times d^v}$, number of clusters k, parameter $\lambda$.

**Output:** Similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$.

1: Initialize the weight coefficient $w^v = \frac{1}{m}$, and $\mathbf{s}_i$ can be initialized by following formula:$\min_{\mathbf{s}_i^T \mathbf{1} = 1, 0 \le s_{ij} \le 1} \sum_{j=1}^{n} \left(\frac{1}{w^v} \sum_v \left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij} + \alpha s_{ij}^2\right)$;

2: **while** not convergent **do**

3: Update $w^v$ by $w^v = 1/2\sqrt{\sum_{i,j}\left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2 s_{ij}}$;

4: Update $\mathbf{F}$ by solving Eq. (16);

5: Update $\mathbf{s}_i$ by solving Eq. (17);

6: **end while**

### 2.2. Space-learning-based model

Raw data perhaps does not have distinct clustering features in the original space, and projecting it into another space may have a significant effect. Space learning is dedicated to reconstructing data in an ideal space to achieve better clustering of data. In recent years, the applications of space learning to multi-view clustering have emerged in large numbers.

#### 2.2.1. ECMSC

In order to take full advantage of the information complementarity between different views and ensure the consistency of the final indicator matrix, Wang et al. [53] proposed the method named Exclusivity-Consistency Regularity Multi-view Subspace Clustering (ECMSC). Although some works [34,54,55] fuse all views to learn a final representation, they do not mine and harness the information complementarity between views. ECMSC makes the actual effect of information complementarity by enhancing the exclusivity of subspace representations between different views. Moreover, for overcoming the loss of clustering effects caused by the separate implementation of subspace learning and spectral clustering such as the works [31,32,55], the authors combined the two processes into a unified framework that learns a final indicator matrix, which ensures the principle of consistency.

Unlike the value-aware Hilbert-Schmidt Independence Criterion (HSIC) to be described in the following, the authors used a position-aware calculation to measure exclusivity between different subspace representations. As described in the paper, this measurement is easier to control the size of the element values and is seamless to be compatible with the SSC [56] framework. Specifically, the calculation formula is defined as $\mathcal{H}(\mathbf{U}, \mathbf{V}) = \|\mathbf{U} \odot \mathbf{V}\|_0 = \sum_{i,j}\left(u_{ij} \cdot v_{ij} \ne 0\right)$, where $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{V} \in \mathbb{R}^{n \times n}$ and $\odot$ denotes the Hadamard product. Then, we can clearly know that if an element in $\mathbf{U}$ is not equal to 0, the corresponding element in $\mathbf{V}$ will approach 0, which makes information complementarity available. Since $\ell_0$-norm is non-convex and discrete, it in the above equation is relaxed to $\ell_1$-norm to make the calculation convenient. Thus, in the multi-view case, the mathematical expression that makes different subspace representations as mutually exclusive as possible is presented as follows:

$$\min_{\mathbf{F}, \mathbf{Z}^v} \sum_{v=1}^{m} \left(\|\mathbf{E}^v\|_1 + \lambda_1 \|\mathbf{Z}^v\|_1 + \lambda_2 \sum_{w \ne v} \|\mathbf{Z}^v \odot \mathbf{Z}^w\|_1\right)$$
$$\text{s.t. } \forall v, \mathbf{X}^v = \mathbf{X}^v \mathbf{Z}^v + \mathbf{E}^v, \text{diag}\left(\mathbf{Z}^v\right) = \mathbf{0}. \tag{18}$$

Here the norm selection for the error term $\mathbf{E}^v$ depends on prior knowledge, the authors chose to use $\ell_1$-norm to deal with the sparse corruptions. For the same reason, the second item makes $\mathbf{Z}^v$ sparse. And the constraint diag $(\mathbf{Z}^v) = \mathbf{0}$ guarantees that each data point can only be represented as a combination of the remaining points except itself.

In subspace learning, after obtaining the desired subspace representation $\mathbf{Z}$, the similarity matrix $\mathbf{S}$ is solved based on it, and then the conventional spectral clustering method is used to acquire the indicator matrix $\mathbf{F}$. The form of the objective function of $\mathbf{F}$ is familiar, it should be noted that for the sake of simplification and consideration of the entire theoretical framework, Wang et al. [53] relaxed the restrictions on $\mathbf{F}$ $\left\{\mathbf{F} \in \{0, 1\}^{n \times k} : \mathbf{F1} = \mathbf{1}, \text{rank}(\mathbf{F}) = k.\right\}$ to $\mathbf{F}^T \mathbf{F} = \mathbf{I}$. Then, the loss function for $\mathbf{F}$ has the following form:

$$\text{Tr}\left(\mathbf{F}^T (\mathbf{D} - \mathbf{S}) \mathbf{F}\right) = \sum_{i,j} \frac{1}{2} s_{ij} \left(\left\|\mathbf{f}_i - \mathbf{f}_j\right\|_2^2\right)$$
$$= \sum_{i,j} \left|z_{ij}\right| \left(\frac{1}{2}\left\|\mathbf{f}_i - \mathbf{f}_j\right\|_2^2\right) = \|\mathbf{Z} \odot \mathbf{\Theta}\|_1, \tag{19}$$

where $\theta_{ij} = \frac{1}{2}\left\|\mathbf{f}_i - \mathbf{f}_j\right\|_2^2$, $\mathbf{f}_i$ and $\mathbf{f}_j$ are two row vectors in $\mathbf{F}$, and $\theta_{ij}$ is an element in the $\mathbf{\Theta}$ matrix. In their paper, the final objective function of spectral clustering is summarized as:

$$\min_{\mathbf{F}} \|\mathbf{Z} \odot \mathbf{\Theta}\|_1, \text{ s.t. } \mathbf{F}^T \mathbf{F} = \mathbf{I}. \tag{20}$$

The fusion of subspace learning and spectral clustering has been proved by the work [57] to upgrade clustering accuracy for single view cases. Based on this inspiration, the authors extended

this theory to multi-view cases, which is to add Eq. (20) corresponding to each view. In this way, the consistency of the final indicator matrix $\mathbf{F}$ between all views can be satisfied. The formula is written as:

$$\min \sum_{v=1}^{m} \|\mathbf{Z}^v \odot \Theta\|_1, \ \text{s.t.}\ \mathbf{F}^T\mathbf{F} = \mathbf{I}. \tag{21}$$

So far, the model of ECMSC proposed in the paper is ready to come out. Uniting Eqs. (18) and (21), the mathematical expression of ECMSC is shown as follows:

$$\min_{\mathbf{F},\mathbf{Z}^1,\ldots,\mathbf{Z}^m} \sum_{v=1}^{m} \|\mathbf{E}^v\|_1 + \lambda_1\|\mathbf{Z}^v\|_1 + \lambda_2 \underbrace{\sum_{w\neq v} \|\mathbf{Z}^v \odot \mathbf{Z}^w\|_1}_{\text{Exclusivity}} + \lambda_3 \underbrace{\|\mathbf{Z}^v \odot \Theta\|_1}_{\text{Consistency}}$$

$$\text{s.t. } \mathbf{X}^v = \mathbf{X}^v\mathbf{Z}^v + \mathbf{E}^v, \text{diag}\left(\mathbf{Z}^v\right) = \mathbf{0}, \mathbf{F}^T\mathbf{F} = \mathbf{I}, \tag{22}$$

where $\lambda_1$, $\lambda_2$ and $\lambda_3$ denote the balance factors of the corresponding item.

To solve the above objective function, the authors split the problem into two sub-problems. First, based on the initialized $\mathbf{F}$, the Alternating Direction Method of Multipliers (ADMM) algorithm [58] is used to find the optimal solutions of $\mathbf{Z}^v$ and $\mathbf{E}^v$ (see the original paper for details of the calculation process), then $\mathbf{Z}^v$ and $\mathbf{E}^v$ are fixed to calculate the value of $\mathbf{F}$. The details of the algorithm are shown in Algorithm 4.

**Algorithm 4** Exclusivity-Consistency Regularized Multi-view Subspace Clustering

**Input:** $\mathbf{X} = \left\{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^m\right\}$, $\mathbf{X}^v \in \mathbb{R}^{d^v \times n}$, number of clusters k.
**Output:** Indicator matrix $\mathbf{F}$.

```
Initialize Θ = 0, λ1, λ2, λ3, t = 0;
while not convergent do
    for each view v ∈ m do
        Given F, obtain Z^v and E^v via ADMM algorithm;
    end for
    Fix all the (Z^v, E^v), solve the problem min_F tr(F^T M F) s.t.
    F^T F = I to obtain F, where M = Σ_v (D^v − S^v);
    Check the condition for the end of the loop:
    ||Θ_{t+1} − Θ_t||_∞ < 1;If not satisfied, set t = t + 1;
end while
```

### 2.2.2. LMSC

Considering the nature of information complementarity between multi-view data and the remarkable effect of self-representation applied in subspace clustering in recent years, Zhang et al. [59] proposed a novel multi-view subspace clustering method: Latent Multi-view Subspace Clustering (LMSC), which makes full use of these two conditions and reconstructs data to gain the latent representation, then mines the subspace representation of data based on this representation. Moreover, the authors combined these two processes into a unified algorithm framework, and then optimized the problem through the Augmented Lagrangian Multiplier with Alternating Direction Minimization (ALM-ADM) [58] method. Meanwhile, the authors also had taken into account the impact of noise data on the algorithm, and come up with concrete solution to this problem.

Same as described in the paper [60], the authors also made the assumption in this paper that multi-view data can be understood as mapping from the same latent representation according to different mapping relationships. Based on the hypothesis, the biggest difference between LMSC and other multi-view subspace algorithms such as [31–33] is that the subspace representation is reconstructed after all views are fused instead of being reconstructed based on single view, the purpose of this operation is to integrate the fragmentary information contained in all views to represent the data more completely and essentially.

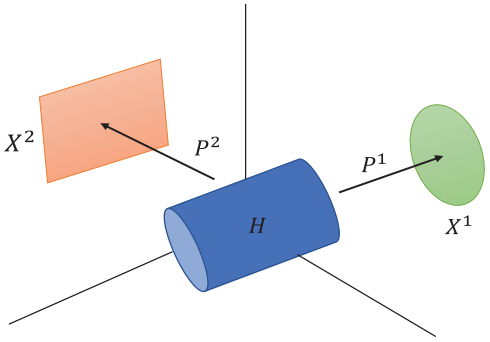



**Fig. 1.** Demonstration of multi-view latent representation proposed by Zhang et al. [59]

In order to describe the relationship between the original data and the latent representation wanted to find, it needs to introduce a set of variables $\{\mathbf{P}^1, \ldots, \mathbf{P}^m\}$, where $\mathbf{P}^v \in \mathbb{R}^{d^v \times r}$ is a mapping matrix corresponding to each view. Then the product of $\mathbf{P}^v$ multiplied by the latent representation $\mathbf{H} \in \mathbb{R}^{r\times n}$ can be roughly regarded as the data matrix of the relevant view, here the value of $r$ needs to be set in advance and the relationship among $\mathbf{H}$, $\mathbf{P}^v$ and $\mathbf{X}^v$ is shown in Fig. 1. The mathematical formula is written accordingly:

$$\min_{\mathbf{P},\mathbf{H}} L_h(\mathbf{X}, \mathbf{PH})$$
$$\text{with } \mathbf{X} = \begin{bmatrix} \mathbf{X}^1 \\ \cdots \\ \mathbf{X}^m \end{bmatrix} \text{ and } \mathbf{P} = \begin{bmatrix} \mathbf{P}^1 \\ \cdots \\ \mathbf{P}^m \end{bmatrix}, \tag{23}$$

where $\mathbf{X}$ and $\mathbf{P}$ are matrices vertically spliced by $\{\mathbf{X}^1, \ldots, \mathbf{X}^m\}$ and $\{\mathbf{P}^1, \ldots, \mathbf{P}^m\}$, respectively. $L_h(\cdot, \cdot)$ indicates the loss function for the potential representation. It can be seen that this method is unique compared to most multi-view fusion methods, in which all views are fused via weight coefficients.

Thus, Zhang et al. [59] used $\mathbf{H}$ in Eq. (23) as the authentic representation of data features and applied it to subspace clustering, in other words, it was to explore the appropriate subspace representation $\mathbf{Z}$ in $\mathbf{H}$. They considered this problem for following equation:

$$\min_{\mathbf{Z}} L_r(\mathbf{H}, \mathbf{HZ}) + \alpha\Omega(\mathbf{Z}). \tag{24}$$

The $L_r(\cdot, \cdot)$ represents the objective function for the solution of $\mathbf{Z}$, $\Omega(\cdot)$ is intended to regularize $\mathbf{Z}$, and the scalar $\alpha > 0$ is to equilibrate the regularization. Certainly, we should note that the form of Eq. (24) draws on the content mentioned in works [46,61,62].

As mentioned above, in order to achieve the fusion of subspace clustering and latent representation learning, the authors added Eq. (23) and Eq. (24) after introducing additional parameters $\lambda_1$ and $\lambda_2$, which are for balancing the three terms. At the same time, they took into account the impact of noise data by using $\ell_{2,1}$-norm, the eventual objective is written as

$$\min_{\mathbf{P},\mathbf{H},\mathbf{Z},\mathbf{E}_h,\mathbf{E}_r} \|\mathbf{E}_h\|_{2,1} + \lambda_1\|\mathbf{E}_r\|_{2,1} + \lambda_2\|\mathbf{Z}\|_*$$
$$\text{s.t. } \mathbf{X} = \mathbf{PH} + \mathbf{E}_h, \mathbf{H} = \mathbf{HZ} + \mathbf{E}_r \text{ and } \mathbf{PP}^T = \mathbf{I}. \tag{25}$$

Here $\|\cdot\|_{2,1}$ not only enhances the robustness of the algorithm to noise, but also promotes the columns of a matrix sparsity. $\|\cdot\|_*$ indicates the matrix nuclear norm and makes the matrix $\mathbf{Z}$ low-rank to avert trivial solution. The constraint on $\mathbf{P}$ is to avoid the situation where $\mathbf{H}$ will tend to zero during the calculation process. Now re-examine Eq. (25), we can clearly understand that the first

item of the formula is to combine multiple views to learn the latent representation $\mathbf{H}$, the second one is based on $\mathbf{H}$ to learn the subspace representation $\mathbf{Z}$, and the last one ensures that the solution of $\mathbf{Z}$ is more normal.

Furthermore, for making the columns of $\mathbf{E}_h$ and $\mathbf{E}_r$ possess the same magnitude values, the authors simply stitched the matrices $\mathbf{E}_h$ and $\mathbf{E}_r$ together vertically. But this method is still valid, which has been widely proved in other works. Then, Eq. (25) is converted to the following form:

$$\begin{aligned}&\min_{\mathbf{P},\mathbf{H},\mathbf{Z},\mathbf{E}_h,\mathbf{E}_r} \|\mathbf{E}\|_{2,1} + \lambda\|\mathbf{Z}\|_* \\ &\text{s.t. } \mathbf{X} = \mathbf{PH} + \mathbf{E}_h, \mathbf{H} = \mathbf{HZ} + \mathbf{E}_r, \\ &\mathbf{E} = [\mathbf{E}_h; \mathbf{E}_r] \text{ and } \mathbf{PP}^T = \mathbf{I}. \end{aligned} \tag{26}$$

For optimization of Eq. (26), the authors used ALM-ADM algorithm to solve it, the whole process of LMSC is demonstrated in Algorithm 5, where $\mathbf{J}$, $\mathbf{Y}_1$, $\mathbf{Y}_2$, $\mathbf{Y}_3$ are intermediate variables and $\mu$, $\rho$, $\varepsilon$ are integral hyper parameters.

**Algorithm 5** Latent Multi-view Subspace Clustering

**Input:** $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^m\}$, $\mathbf{X}^v \in \mathbb{R}^{d^v \times n}$, number of clusters k, parameter $\lambda$, dimension $r$ of the latent representation $\mathbf{H}$.

**Output:** $\mathbf{Z}$, $\mathbf{H}$, $\mathbf{P}$, $\mathbf{E}$.

1: Initialize $\mathbf{P} = 0$, $\mathbf{E} = 0$, $\mathbf{J} = 0$, $\mathbf{Z} = 0$ $\mathbf{Y}_1 = 0$, $\mathbf{Y}_2 = 0$, $\mathbf{Y}_3 = 0$, $\mu = 10^{-6}$, $\rho = 1.1$, $\varepsilon = 10^{-4}$, $\max_\mu = 10^6$, initialize $\mathbf{H}$ with stochastic values;

2: **while** not convergent **do**

3: Update $\mathbf{P}$ by $\mathbf{P} = \arg\min \frac{\mu}{2}\left\|\left(\mathbf{X} + \frac{1}{\mu}\mathbf{Y}_1 - \mathbf{E}_h\right)^T - \mathbf{H}^T\mathbf{P}^T\right\|_F^2$;

4: Update $\mathbf{H}$ by

$$\begin{gathered}\mathbf{AH} + \mathbf{HB} = \mathbf{C} \\ \text{with } \mathbf{A} = \mu\mathbf{P}^T\mathbf{P}, \mathbf{B} = \mu\left(\mathbf{ZZ}^T - \mathbf{Z} - \mathbf{Z}^T + \mathbf{I}\right) \\ \mathbf{C} = \mathbf{P}^T\mathbf{Y}_1 + \mathbf{Y}_2\left(\mathbf{Z}^T - \mathbf{I}\right) \\ +\mu\left(\mathbf{P}^T\mathbf{X} + \mathbf{E}_r^T - \mathbf{P}^T\mathbf{E}_h - \mathbf{E}_r\mathbf{Z}^T\right);\end{gathered}$$

5: Update $\mathbf{Z}$ by

$$\mathbf{Z} = \left(\mathbf{H}^T\mathbf{H} + \mathbf{I}\right)^{-1}\left[\left(\mathbf{J} + \mathbf{H}^T\mathbf{H} - \mathbf{H}^T\mathbf{E}_r\right) + \left(\mathbf{Y}_3 + \mathbf{H}^T\mathbf{Y}_2\right)/\mu\right];$$

6: Update $\mathbf{E}$ by $\mathbf{E} = \arg\min_{\mathbf{E}} \frac{1}{\mu}\|\mathbf{E}\|_{2,1} + \frac{1}{2}\|\mathbf{E} - \mathbf{G}\|_F^2$;

7: Update $\mathbf{J}$ by $\mathbf{J} = \frac{\lambda}{\mu}\|\mathbf{J}\|_* + \frac{1}{2}\|\mathbf{J} - (\mathbf{Z} - \mathbf{Y}_3/\mu)\|_F^2$;

8: Update $\mathbf{Y}_1$, $\mathbf{Y}_2$, $\mathbf{Y}_3$ by $\begin{cases}\mathbf{Y}_1 = \mathbf{Y}_1 + \mu(\mathbf{X} - \mathbf{PH} - \mathbf{E}_h) \\ \mathbf{Y}_2 = \mathbf{Y}_2 + \mu(\mathbf{H} - \mathbf{HZ} - \mathbf{E}_r) \\ \mathbf{Y}_3 = \mathbf{Y}_3 + \mu(\mathbf{J} - \mathbf{Z})\end{cases}$

9: Update $\mu$ by $\mu = \min\left(\rho\mu; \max_\mu\right)$;

10: Verify that the conditions for the end of the loop are met: $\|\mathbf{X} - \mathbf{PH} - \mathbf{E}_h\|_\infty < \epsilon$, $\|\mathbf{H} - \mathbf{HZ} - \mathbf{E}_r)\|_\infty < \epsilon$ and $\|\mathbf{J} - \mathbf{Z}\|_\infty < \epsilon$;

11: **end while**

#### 2.2.3. MSC_IAS

The high dimensionality of data poses a challenge to clustering algorithms, because it contains a large number of redundant and useless features, which make the construction of the similarity matrix unreliable in graph-based clustering algorithms. If the data has multiple views, it will undoubtedly add one more complicated factor. Wang et al. [63] proposed a novel subspace clustering model for multi-view data, named Multi-view Subspace Clustering with Intactness-Aware Similarity (MSC_IAS). MSC_IAS is capable of generating the similarity matrix by intact space learning [64], which is more dependable for clustering. Once the similarity matrix with intactness-aware is gained, the normalized cuts algorithm (Ncut) [65] is employed on it to realize the final clustering. Concretely, the intact space proposed by authors means a space in which data representation will retain complete data information while the dimension of data will be decreased effectively at the same time. It is because of this characteristic that makes it can contain properties that is critical to build the similarity matrix.

In intact space learning [64], it assumes that the latent intact representation consists of parts of the information contained in each view. From this point of view, all views must be merged to capture the intact space. Supposing that $\mathbf{L} \in \mathbb{R}^{d\times n}$ indicates the intact space, which is composed of column vectors $\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_n$, and $\mathbf{W}^v \in \mathbb{R}^{d^v\times d}$ indicates the linear sample matrix of the $v$th view. Specially, $\mathbf{W}^v\mathbf{L}$ represents a mapping of the intact space. Thus, restoring the intact space is mathematically expressed as the following problem:

$$\begin{aligned}&\min_{\mathbf{L},\mathbf{W}^v} \frac{1}{m}\sum_{v=1}^{m}\|\mathbf{X}^v - \mathbf{W}^v\mathbf{L}\|_F^2 + \lambda_1\|\mathbf{L}\|_F^2 \\ &\text{s.t. } \forall i, \|\mathbf{W}_i^v\|_2 \le 1,\end{aligned} \tag{27}$$

where $\mathbf{W}_i^v$ is the $i$th column vector of matrix $\mathbf{W}^v$, and the constraint on $\mathbf{W}_i^v$ as well as the second item are both for regularizing the corresponding terms. In addition, this formula seems to be very similar to non-negative matrix factorization [66,67] and multi-view dictionary [68], but it should be noted that the model does not include extra non-negative restriction on the learned matrix $\mathbf{L}$, the purpose of which is to make $\mathbf{L}$ have more value space.

Once the solution of intact space is obtained, the construction of similarity matrix based on the intact space becomes the focus of the problem. For the goal of maximizing the dependence of similarity matrix on the intact space, Wang et al. [63] adopted HSIC [69,70] to measure the correlation between $\mathbf{L}$ and $\mathbf{S}$. The inner kernel $\mathbf{K}_1 = \mathbf{L}^T\mathbf{L}$ is applied for the latent intact space $\mathcal{X}$, and the linear kernel $\mathbf{K}_2 = \mathbf{S} - \mathbf{D}$ is employed for the similarity space $\mathcal{S}$. Then, by measuring the HSIC difference between $\mathbf{L}$ and $\mathbf{S}$, it can obtain the formula for the limit of the similarity matrix $\mathbf{S}$:

$$\begin{aligned}\max_{\mathbf{S}} \text{HSIC}(\mathbf{L}, \mathbf{S}) &= \max_{\mathbf{S}} \text{tr}\left(\mathbf{K}_1\mathbf{H}\mathbf{K}_2\mathbf{H}\right) \\ &= \max_{\mathbf{S}} \text{tr}\left(\mathbf{L}^T\mathbf{L}\mathbf{H}(\mathbf{S} - \mathbf{D})\mathbf{H}\right) \\ &= -\max_{\mathbf{S}} \text{tr}\left(\mathbf{L}\mathbf{H}\mathbf{L}_{\mathbf{S}}(\mathbf{L}\mathbf{H})^T\right) \\ &= \min_{\mathbf{S}} \text{tr}\left(\mathbf{L}\mathbf{H}\mathbf{L}_{\mathbf{S}}(\mathbf{L}\mathbf{H})^T\right) \\ s.t. \quad &\mathbf{S}_i^T\mathbf{1} = 1, \mathbf{S}_i \succeq 0,\end{aligned} \tag{28}$$

where they denote $\mathbf{H} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^T$ as a centering matrix to be utilized to center the intact space. Moreover, the significance of this limitation of $\mathbf{S}_i^T\mathbf{1} = 1$ is that the constructed similar points are located in the affine subspace and are concentrated through it.

If based on the assumption that the intact space $\mathbf{L}$ is concentrated, the conclusion that $\mathbf{L}$ is equal to $\mathbf{LH}$ can be obtained. Then according to this verdict, Wang et al. [63] revised the last step in Eq. (28) to the following form:

$$\begin{aligned}&\min_{\mathbf{S}} \lambda_2 \sum_{i=1}^{n}\sum_{j=1}^{n}\left\|\mathbf{L}_i - \mathbf{L}_j\right\|_1\mathbf{S}_{ij} + \gamma\|\mathbf{S}\|_F^2 \\ &\text{s.t. } \mathbf{S}_i^T\mathbf{1} = 1, \mathbf{S}_i \succeq 0.\end{aligned} \tag{29}$$

Here the second item is also to avoid gaining the trivial solution, $\lambda_2$ and $\gamma$ are just two non-negative parameters. It is necessary to elaborate the advantages of the similarity matrix $\mathbf{S}$ solved in this way compared to that solved by using the Gaussian kernel function $\mathbf{S}_{ij} = \exp(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2})$. First, it ensures that the similarity between data points can be adaptively learned and adjusted, then $\ell_1$-norm is more robust to abnormal sample points than using $\ell_2$-norm.

In order to avoid errors caused by the stepwise implementation of both the recovery of intact space and the construction of similar matrix, Wang et al. [63] combined Eq. (27) with Eq. (29) to achieve simultaneous learning of intact spaces and similar matrix.

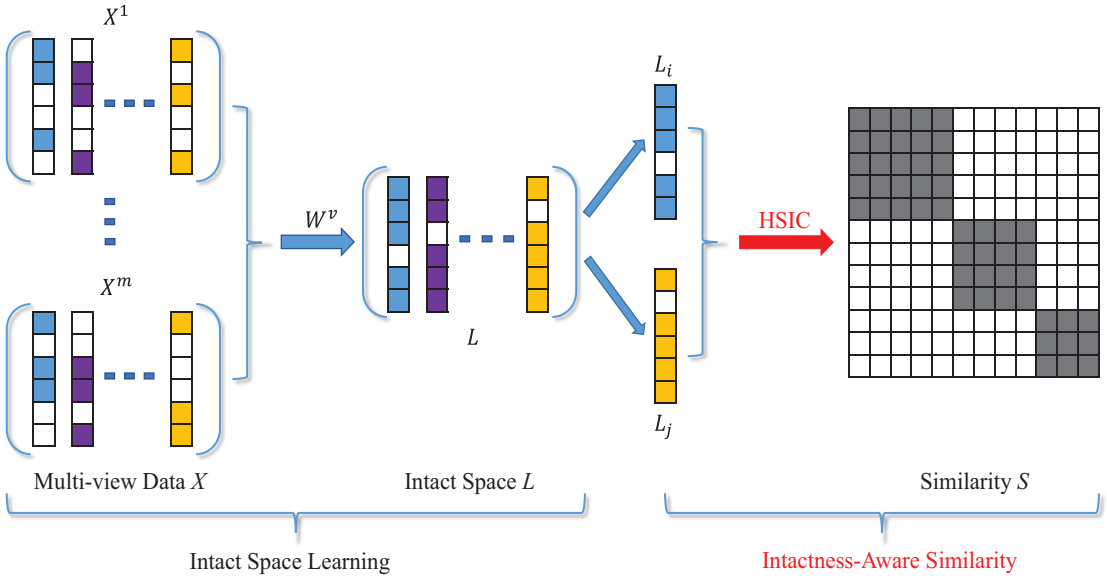



**Fig. 2.** The framework of MSC_IAS proposed by Wang et al. [63].

Here $\lambda_1\|\mathbf{L}\|_F^2$ in Eq. (27) is discarded because $\mathbf{L}$ in Eq. (29) has been regularized. Thus, the complete objective function of MSC_IAS model is written as:

$$\min_{\mathbf{W}^v,\mathbf{L},\mathbf{S}} \frac{1}{m}\sum_{v=1}^{m}\|\mathbf{X}^v-\mathbf{W}^v\mathbf{L}\|_F^2+\lambda_2\sum_{i=1}^{n}\sum_{j=1}^{n}\left\|\mathbf{L}_i-\mathbf{L}_j\right\|_1\mathbf{S}_{ij}+\gamma\|\mathbf{S}\|_F^2$$
$$\text{s.t. } \forall i,\ \|\mathbf{W}_i^v\|_2\le 1,\ \mathbf{S}_i^T\mathbf{1}=1,\mathbf{S}_i\ge 0. \tag{30}$$

Fig. 2 demonstrates the overall framework of MSC_IAS, and Algorithm 6 shows the main steps of MSC_IAS, where $\mathbf{A_S}$, $\mathbf{Q}$, $\mathbf{M}$ and $\mathbf{Z}$ are intermediate variables introduced in the algorithm optimization process.

**Algorithm 6** Multi-view Subspace Clustering with Intactness-Aware Similarity

**Input:** $\mathbf{X}=\{\mathbf{X}^1,\mathbf{X}^2,\cdots,\mathbf{X}^m\}$, $X^v\in\mathbb{R}^{d^v\times n}$, number of clusters k, number of nearest neighbors c, parameter $\lambda_2$, dimension of intact space $d$.

**Output:** Final clustering results.

Initialize $\mathbf{L}$, $\mathbf{W}^v$, $\mathbf{Q}$, $\mathbf{S}$ and $\mathbf{Z}$, $\mu$ = 1.25, $\rho>1$;
**while** not convergent **do**
  Update $\mathbf{L}$ by
  $$\left(\tfrac{2}{m}\textstyle\sum_{v=1}^{m}\mathbf{W}^{vT}\mathbf{W}^v\right)\mathbf{L}+\mathbf{L}\left(\mu\mathbf{A_S}\mathbf{A_S}^T\right)=\tfrac{2}{m}\textstyle\sum_{v=1}^{m}\mathbf{W}^{vT}\mathbf{X}^v+\mathbf{Z}\mathbf{A_S}^T+\mu\mathbf{Q}\mathbf{A_S}^T;$$
  **for** $i=1;\ i<m;\ i++$ **do**
    Update $\mathbf{W}^v$ by $\mathbf{W}^v=\left(\mathbf{X}^v\mathbf{L}^T+\tau(\mathbf{M}-\mathbf{T})\right)\left(\mathbf{L}\mathbf{L}^T+\tau\mathbf{I}\right)^{-1}$;
  **end for**
  Update $\mathbf{Q}$ by $\mathbf{Q}=\text{sign}\left(\mathbf{L}\mathbf{A_S}-\frac{\mathbf{Z}}{\mu}\right)\max\left(\left|\mathbf{L}\mathbf{A_S}-\frac{\mathbf{Z}}{\mu}\right|-\frac{\lambda_3}{\mu}\right)$;
  **for** $i=1;\ i<n;\ i++$ **do**
    Update $\mathbf{S}_i$ by $\mathbf{S}_i=\left(\frac{1+\sum_{j=1}^{k}\tilde{\mathbf{d}}_{ij}^X}{k}\mathbf{1}-\mathbf{d}_i^X\right)_+$;
  **end for**
  Balance $\mathbf{S}$ by $\frac{\mathbf{S}+\mathbf{S}^T}{2}$;
  Update the multipliers $\mathbf{Z}$ by $\mathbf{Z}=\mathbf{Z}+\mu(\mathbf{Q}-\mathbf{L}\mathbf{A_S})$ ; $\mu=\mu\rho$;
**end while**
Employ Ncut [65] on $\mathbf{S}$.

### 2.2.4. COMIC

In current multi-view clustering, more or less parameter settings are required, especially the selection of the number of clusters, which needs to be specified when the variables are initialized in almost all existing algorithms. In order to get rid of the troubles caused by parameter selection, Peng et al. [71] proposed the algorithm called CrOss-view MatchIng Clustering (COMIC), which projects data into a space that takes into account both geometric consistency (GC) and cluster assignment consistency (CAC). More concretely, the purpose of GC is to learn a connection graph in the projection space, in which only sample points belonging to the same class will be connected. And the intention of CAC is to make the connection graphs generated by different views as similar as possible, thereby ensuring that the distribution structures of different views are as identical as possible. Simultaneously, in the proposed model, it automatically learns almost all the required parameters including the number of clusters.

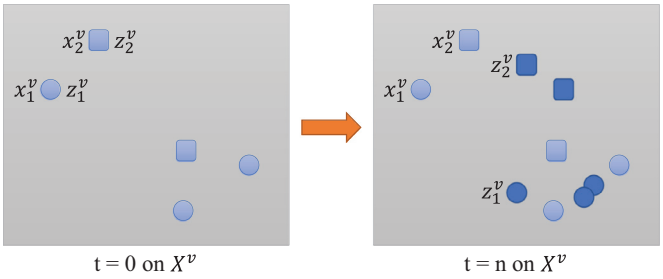



**Fig. 3.** When initializing, $\mathbf{Z}^v=\mathbf{X}^v$. After the iterative process, the data points of the same cluster in the reconstructed data $\mathbf{Z}^v$ will be closer.

Specifically, GC is more focused on learning a favourable connection graph $\mathbf{S}^v$ with the cooperation of the local geometric consistency $\mathbf{W}^v$ for each view, while the reconstructed data representation $\mathbf{Z}^v$ is learned at the same time, which has the same dimensions as the original data points $\mathbf{X}^v$. Fig. 3 shows the meaning of GC, whose idea is similar to the work [72]. What needs special explanation is that $\mathbf{Z}^v$ here does not represent each data point as compactly as the existing methods, it just reconstructs the data according to $\mathbf{S}^v$ to make the data of the same cluster closer. Unlike GC, CAC is introduced to minimize the divergence between different connection graphs. This is indeed different from traditional methods, which chooses to enhance uniformity between connection graphs $\{\mathbf{S}^v\}_{v=1}^m$ of different views rather than the learned representations $\{\mathbf{Z}^v\}_{v=1}^m$. The significance of this approach adopted by

COMIC is that if the similarity of $\{\mathbf{Z}^v\}_{v=1}^m$ is forcibly maximized, it will undoubtedly result in the loss of the original information of data. On the contrary, maximizing the similarity of the former saves the relative position of data points in different views.

As analyzed above, the objective function of COMIC is written as follows:

$$\mathcal{L} = \sum_v \mathcal{L}_1^v + \mathcal{L}_2, \tag{31}$$

where the first item is based on GC and the second item is derived from CAC. The specific formulas of these two items are shown below:

$$\mathcal{L}_1^v = \frac{1}{2}\sum_{i=1}^{m} \underbrace{\|\mathbf{x}_i^v - \mathbf{z}_i^v\|_2^2}_{\text{reconstruction loss}} + \frac{\lambda^v}{2} \underbrace{\sum_{i,j} \mathbf{W}_{ij}^v \left( \left\| \mathbf{S}_{ij}^v \mathbf{z}_i^v - \mathbf{S}_{ji}^v \mathbf{z}_j^v \right\|_2^2 + \mu^v \left( \mathbf{S}_{ij}^v - 1 \right)^2 \right)}_{\text{geometric consistency}} \tag{32}$$

and

$$\mathcal{L}_2 = \frac{1}{2}\sum_{i,j}\sum_{v \neq k} \underbrace{\left( \mathbf{S}_{ij}^v - \mathbf{S}_{ij}^k \right)^2}_{\text{cluster assignment consistency}}. \tag{33}$$

Here $\mathbf{z}_i^v$ is the new representation of the original data $\mathbf{x}_i^v$ after data reconstruction, which satisfies GC and CAC principles concurrently. The reconstruction loss draws on the ideas of convex clustering [73,74], that is, re-learning the representation $\mathbf{Z}^v$ of the raw data $\mathbf{X}^v$ in the neighborhood space. For the calculation of $\mathbf{W}^v$, the authors adopted mutual $k$-nearest neighbors connectivity (m-kNN) to compute it. It is noteworthy that this item $(\mathbf{S}_{ij}^v - 1)$ possesses the following functions. Firstly, as mentioned above, if the connection is built between the two points, then this item will tend to 0 $(\mathbf{S}_{ij}^v \to 1)$, otherwise it will tend to 1 $(\mathbf{S}_{ij}^v \to 0)$. Secondly, the weight of connection graph for each view will be controlled in the range [0,1]. Finally, it is capable to avoid the occurrences of $\mathbf{S}^v = \mathbf{0}$ and $\mathbf{Z}^v = \mathbf{X}^v$.

After optimizing Eq. (31), the final value of $\mathbf{Z}^v$ can be obtained. But at this time, it cannot acquire the label of each data point directly through $\mathbf{Z}^v$, instead $\mathbf{Z}^v$ is used to construct the final cluster graph. First of all, $m$ view-specific connection graphs are built according to the inequality $\|\mathbf{z}_i^v - \mathbf{z}_j^v\|_2 \leq \epsilon^v$, where $\mathbf{z}_i$ and $\mathbf{z}_j$ are connected if their relationship satisfies this inequality. Here $\epsilon^v$ represents the threshold and its value is set to the average length of the shortest 90% edges in $\mathbf{W}^v$. Then, in these $m$ graphs, if the number of connections of two points exceeds half, they are considered to pertain to the same cluster. Pay attention to the objective function of COMIC, which contains two parameters $\lambda^v$ and $\mu^v$. Before the algorithm runs, the value of them are not pre-specified, but are continuously learned and updated according to the established formulas, which also confirms the title of the paper. In addition, the avoidance of selecting the number of clusters benefits from $\mathcal{L}_2$, which pledges that the connected points are deemed to belong to the same cluster. At last, the algorithm steps of COMIC are shown in Algorithm 7.

**Algorithm 7** CrOss-view MatchIng Clustering

**Input:** $\mathbf{X} = \left\{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^m\right\}$, $\mathbf{X}^v \in \mathbb{R}^{d^v \times n}$.
**Output:** Final clustering result.

1: Each row of $\{\mathbf{X}^v\}_{v=1}^m$ is normalized to have a $\ell_2$-norm unit;
2: Construct the similarity graph $\{\mathbf{W}^v\}_{v=1}^m$ by $\mathbf{W}_{ij}^v = \frac{\sum_{k=1}^n n_k^v}{n^v \sqrt{n_i^v n_j^v}}$ and calculate the spectral norm of $\{\mathbf{X}^v\}_{v=1}^m$;
3: Initialize $t = 1$, $\mathbf{Z}^v = \mathbf{X}^v$, $\mathbf{S}^v = \mathbf{1}$;
4: Initialize $\lambda^v$, $\mu^v$, $\epsilon^v$ according to the formulas in the paper;
5: **while** $\left|\mathcal{L}^{t+1} - \mathcal{L}^t\right| \leq 10^{-8}$ or $t \leq 1000$ **do**
6: Update $\mathbf{S}^v$ via $\mathbf{S}_{ij}^v = \frac{\mu^v + \sum_{k \neq v} \mathbf{S}_{ij}^k}{\mu^v + (m-1) + \lambda^v \mathbf{W}_{ij}^v \left\| \mathbf{z}_i^v - \mathbf{z}_j^v \right\|_2^2}$;
7: Update $\mathbf{Z}^v$ via $\mathbf{Z}^v \mathbf{M}^v = \mathbf{X}^v$, where $\mathbf{M}^v = \mathbf{I} + \lambda^v \mathbf{\Omega}$, $\mathbf{\Omega}^v = \sum_{i,j} \mathbf{W}_{ij}^v \left( \mathbf{S}_{ij}^v \right)^2 \left( \mathbf{e}_i^v - \mathbf{e}_j^v \right) \left( \mathbf{e}_i^v - \mathbf{e}_j^v \right)^\top$ and $\mathbf{e}_i^v$ denotes an indicator vector, whose $i$th entry is 1;
8: Update $\lambda^v$ via $\lambda^v = \frac{\|\mathbf{X}^v\|_2}{\|\mathbf{\Omega}^v\|_2}$;
9: Update $t$ via $t + 1$;
10: **end while**

### 2.3. Binary-code-learning-based model

Recently, the advancement has been made in the study of binary code learning, which encodes data features into binary form that is of great significance for reducing data storage and computing time. Applying binary coding learning to multi view clustering is a feasible scheme to improve the speed of algorithm and save storage space.

In existing multi-view clustering algorithms, the executions of them require a considerable amount of storage space and long-time operation to obtain the final result, and the performance of these algorithms will drop dramatically as the size of data sets continues to increase. In view of the above shortcomings, Zhang et al. [75] introduced firstly binary code learning to large-scale multi-view clustering to propose a new method named Binary Multi-view Clustering (BMVC), which incorporates two critical constituents: collaborative discrete representation learning (CDRL) and binary clustering structure learning (BCSL). More concretely, they combined CDRL loss with BCSL loss to construct the final objective. Then, the optimal indicator matrix in the objective function can be gained by iteration.

What is binary coding learning [76]? The core idea is to encode the original features of data into a series of binary codes in a Hamming space of similarity-preserving and low-dimension. BMVC adopts the method to learn the data features in binary form, which can vastly reduce computing time and memory usage. The reason for this effect is that computers are more efficient for binary computing and storage. What's more, compared with [20,77], the distinction of BMVC is that it does not separate the generation of binary coding from the clustering process and is suitable for large-scale data sets.

According to the problem formulation, the data is encoded by the nonlinear RBF mapping

$$\boldsymbol{\phi}\left(\mathbf{x}_s^v\right) = \left[\exp\left(-\|\mathbf{x}_s^v - \mathbf{a}_1^v\|^2/\sigma\right), \ldots, \exp\left(-\left\|\mathbf{x}_s^v - \mathbf{a}_j^v\right\|^2/\sigma\right)\right]^T. \tag{34}$$

Here $\phi(\mathbf{x}_s^v)$ represents a nonlinear embedding that is calculated from the $s$th sample selected in the $v$th view, $\sigma$ indicates the kernel width, and $\left\{\boldsymbol{a}_i^v\right\}_{i=1}^j$ are $j$ anchor samples chosen randomly from the $v$th view.

Then, Zhang et al. [75] computed the binary hash for $\boldsymbol{x}_s^v$ according to CDRL, which is the key step. The function is defined as follows:

$$\mathbf{h}_s^v\left(\boldsymbol{\phi}\left(\mathbf{x}_s^v\right); \mathbf{U}^v\right) = \operatorname{sgn}\left(\mathbf{U}^v \boldsymbol{\phi}\left(\mathbf{x}_s^v\right)\right), \tag{35}$$

where $\operatorname{sgn}(\cdot)$ is an element-wise sign operator and $\mathbf{U}^v \in \mathbb{R}^{l \times m}$ indicates the mapping matrix of the $v$th view.

Considering the complementarity of multi-view data representation, the loss function of CDRL is showed below:

$$\min_{\mathbf{U}^v,\mathbf{b}_s,w}\sum_{v=1}^{m}\left(w^v\right)^r\left(\sum_{s=1}^{n}\|\mathbf{b}_s-\mathbf{h}_s^v\|_F^2+\beta\|\mathbf{U}^v\|_F^2-\gamma\sum_{s=1}^{n}\mathrm{var}\left(\mathbf{h}_s^v\right)\right)$$
$$\text{s.t.}\ \sum_v w^v=1, w^v>0, \mathbf{b}_s\in\{-1,1\}^{l\times 1}. \tag{36}$$

Especially, $w^v$ represents the weight corresponding to the $v$th view, $r$ controls the degree of influence of weights, $\mathbf{b}_s$ is the collaborative binary code of the $s$th sample and $\gamma$ is just a nonnegative constant.

In addition to considering the collaborative multi-view representation learning, Zhang et al. [75] attempted to maintain a consistent cluster structure for different views. Thus, they conducted the equation based on BCSL

$$\min_{\mathbf{C},\mathbf{g}_s}\|\mathbf{b}_s-\mathbf{C}\mathbf{g}_s\|_F^2\ \text{s.t.}\ \mathbf{C}^T\mathbf{1}=\mathbf{0}, \mathbf{C}\in\{-1,1\}^{l\times c},$$
$$\mathbf{g}_s\in\{0,1\}^c, \sum_i \mathbf{g}_{is}=1, \tag{37}$$

where $\mathbf{C}$ is the clustering centroids matrix and $\mathbf{g}_s$ is the indicator vector.

The most important point of this paper is to combine CDRL and BCSL. For this purpose, the implementation of BMVC is transformed into the following mathematical formula:

$$\begin{aligned}&\min\mathcal{F}\left(\mathbf{U}^v,\mathbf{B},\mathbf{C},\mathbf{G},w\right)\\&=\sum_{v=1}^{m}\left(w^v\right)^r\left(\left\|\mathbf{B}-\mathbf{U}^v\phi\left(\mathbf{x}^v\right)\right\|_F^2+\beta\|\mathbf{U}^v\|_F^2\right.\\&\left.-\frac{\gamma}{n}\mathrm{tr}\left(\left(\mathbf{U}^v\phi\left(\mathbf{x}^v\right)\right)\left(\mathbf{U}^v\phi\left(\mathbf{x}^v\right)\right)^T\right)+\lambda\|\mathbf{B}-\mathbf{C}\mathbf{G}\|_F^2\right.\\&\text{s.t.}\ \mathbf{C}^T\mathbf{1}=\mathbf{0}, \sum_v w^v=1, w^v>0, \mathbf{B}\in\{-1,1\}^{l\times n},\\&\mathbf{C}\in\{-1,1\}^{l\times c}, \mathbf{G}\in\{0,1\}^{c\times n}, \sum_i \mathbf{g}_{is}=1,\end{aligned} \tag{38}$$

where $\mathbf{B}$ and $\mathbf{G}$ are composed of $\{\mathbf{b}_1,\ldots,\mathbf{b}_n\}$, $\{\mathbf{g}_1,\ldots,\mathbf{g}_n\}$ respectively; $\lambda$ indicates the regularization parameter.

Since minimizing Eq. (38) is an NP-hard problem, Zhang et al. [75] divided it into several sub-problems to solve one by one. That is, they updated one of all parameters and fixed the others.

We should note that the most significant factor for the clustering result is the matrix $\mathbf{G}$. So according to Eq. (38), before solving the optimal value of $\mathbf{G}$, it needs to go through an iterative process to update solutions of **U, B, C**. Particularly, the method named adaptive discrete proximal linearized minimization (ADPLM) was proposed in the optimization of $\mathbf{C}$. Then, after the values of **U, B, C** are updated, the optimal value of each element $\mathbf{g}_{ij}$ of the matrix $\mathbf{G}$ can be gained by

$$\mathbf{g}_{ij}^{p+1}=\begin{cases}1, & j=\arg\min_j H\left(\mathbf{b}_i,\mathbf{c}_j^{p+1}\right)\\0, & \text{otherwise}\end{cases} \tag{39}$$

where $H(\mathbf{b}_i, \mathbf{c}_j)$ is used to calculate the distance between the $i$th binary $\boldsymbol{b}_i$ and the $j$th cluster centroid $\mathbf{c}_j$.

Also, Zhang et al. [75] have proved the convergence of the optimization procedure. Hence, the optimal solution of $\mathbf{G}$ can be obtained through iteration process quickly.

## 3. Experiments

In this section, we conduct some experiments and adopt three mainstream evaluation metrics to observe the performance of above algorithms, and then make analyses based on the real experimental data.

**Table 1**
Description of data sets.

| Data set | # samples | # views | # classes | # features |
|---|---|---|---|---|
| ALOI | 1079 | 4 | 10 | 218 |
| MNIST | 2000 | 3 | 10 | 48 |
| NUS-WIDE | 1600 | 6 | 8 | 1134 |
| MSRC-v1 | 210 | 5 | 7 | 1622 |
| Extended Yale-B | 650 | 3 | 10 | 12554 |
| Caltech101 | 9144 | 6 | 102 | 3766 |
| 3-Sources | 169 | 3 | 6 | 10259 |

### 3.1. Data sets

**ALOI**: ALOI data set contains 110, 250 images of 1, 000 objects, each object has about 100 images. We extract 1, 079 images of 10 objects from it, which includes 4 views of RGB color histograms, HSV color histograms, Color similarity, Haralick features.

**MNIST**: MNIST data set is a well-known collection of handwritten numerals from 0 to 9, from which we select a total of 2, 000 images. Each image has 3 views of IsoProjection, Linear Discriminant Analysis (LDA) and Neighborhood Preserving Embedding (NPE).

**NUS-WIDE**: NUS-WIDE data set consists of 269, 648 images of 81 objects. In our experiments, a total of 1, 600 samples from eight categories are chosen, and each image is represented as 6 different features: CH, CM55, CORR, EDH, WT, BoW.

**MSRC-v1**: MSRC-v1 data set contains 240 images of 8 types of objects. We select seven categories of 210 images to compose the experimental data set. Each image has five feature representations.

**Extended Yale-B**: Extended Yale-B data set contains 2, 414 images of 38 faces. Here, 650 face images of 10 people are selected and the entire data set consists of three views.

**Caltech101**: Caltech101 is composed of 9, 144 images of 102 kinds of objects, and there are 6 views of Gabor, WM, CENTRIST, HOG, GIST, LBP, respectively.

**3-Sources Text**: 3-Sources Text data set is composed of 169 news in three languages, which possesses 6 themes of entertainment, politics, business, sport, health and technology. Table 1 describes the basic information of the seven data sets

### 3.2. Evaluation metrics

In this paper, in order to demonstrate the real performance of each algorithm, we use three common evaluation metrics for evaluating these multi-view clustering methods, which are clustering accuracy (ACC), normalized mutual information (NMI) and purity, respectively.

ACC is used to measure the accuracy between the actual labels and the predict labels obtained by algorithms. Assume a data set $\{\mathbf{x}_i\}_{i=1}^n$, groundtruth $g_i$ and predictive label $p_i$. Then, the calculation formula of ACC is written as:

$$\mathrm{ACC}=\frac{\sum_{i=1}^{n}\delta\left(g_i,\mathrm{map}\left(p_i\right)\right)}{n}, \tag{40}$$

where $\delta(x, y)$ is a judgment function. Specifically, if $x = y$, its value is equal to 1, otherwise it is equal to 0. $map(\cdot)$ represents a permutation mapping that maximizes the matching of groundtruth and predictive labels.

Given two random variables $\mathbf{X}$ and $\mathbf{Y}$, the NMI computation formula between the two terms is written as

$$\mathrm{NMI}(\mathbf{X},\mathbf{Y})=\frac{I(\mathbf{X};\mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}}. \tag{41}$$

Here $I(\cdot;\cdot)$ is the mutual information and $H(\cdot)$ denotes the information entropy. Suppose that the predictive labels are $\tilde{\mathbb{C}}=$

**Table 2**
ACC with diverse multi-view clustering algorithms.

| Data sets | ALOI | MNIST | NUS-WIDE | MSRC-v1 | Yale-B | Caltech101 | 3-Source |
|---|---|---|---|---|---|---|---|
| k-means | 0.4856 | 0.6515 | 0.2550 | 0.4857 | 0.1954 | 0.1024 | 0.3550 |
| SwMC | 0.4171 | 0.7160 | 0.2156 | 0.8714 | 0.3646 | 0.1411 | 0.3313 |
| BMVC | 0.6960 | 0.7985 | 0.3844 | 0.7190 | 0.3985 | 0.2878 | 0.5621 |
| MLAN | 0.5690 | 0.7785 | 0.3100 | 0.6810 | 0.3431 | 0.1885 | 0.3275 |
| AMGL | 0.5542 | 0.8900 | 0.2369 | 0.7476 | 0.4138 | 0.2460 | 0.3254 |
| MSC_IAS | 0.6200 | 0.7515 | 0.2587 | 0.6381 | 0.8369 | 0.2028 | 0.6391 |
| LMSC | 0.7016 | 0.5625 | 0.3306 | 0.6571 | 0.5446 | 0.2352 | 0.6036 |
| ECMSC | 0.6450 | 0.8620 | 0.3000 | 0.7810 | 0.7108 | 0.2647 | 0.3195 |
| COMIC | 0.2623 | 0.2690 | 0.2150 | 0.5095 | 0.1000 | 0.1722 | 0.3314 |

**Table 3**
NMI with diverse multi-view clustering algorithms.

| Data set | ALOI | MNIST | NUS-WIDE | MSRC-v1 | Yale-B | Caltech101 | 3-Source |
|---|---|---|---|---|---|---|---|
| k-means | 0.4873 | 0.6666 | 0.1454 | 0.4168 | 0.1229 | 0.2506 | 0.1226 |
| SwMC | 0.4211 | 0.6592 | 0.1238 | 0.7829 | 0.3516 | 0.1327 | 0.0593 |
| BMVC | 0.6358 | 0.6682 | 0.2170 | 0.7164 | 0.2807 | 0.4858 | 0.4521 |
| MLAN | 0.5659 | 0.7652 | 0.2058 | 0.6299 | 0.3160 | 0.2453 | 0.1237 |
| AMGL | 0.5388 | 0.7844 | 0.1260 | 0.6676 | 0.3757 | 0.3157 | 0.0535 |
| MSC_IAS | 0.6695 | 0.7577 | 0.1405 | 0.6571 | 0.8343 | 0.4009 | 0.4308 |
| LMSC | 0.6741 | 0.5237 | 0.1695 | 0.5833 | 0.5095 | 0.4609 | 0.4433 |
| ECMSC | 0.6110 | 0.7530 | 0.1887 | 0.7129 | 0.7276 | 0.4574 | 0.0725 |
| COMIC | 0.2722 | 0.3677 | 0.2889 | 0.5269 | 0.0000 | 0.5182 | 0.0000 |

$\{\tilde{\mathbf{C}}_i\}_{i=1}^{\tilde{c}}$ and the real labels are $\mathbb{C} = \{\mathbf{C}_j\}_{j=1}^{c}$. Then, the form of NMI can be defined as

$$\mathrm{NMI} = \frac{\sum_{i=1}^{\tilde{c}}\sum_{j=1}^{c}|\tilde{\mathbf{C}}_i \cap \mathbf{C}_j| \log \frac{n|\tilde{\mathbf{C}}_i \cap \mathbf{C}_j|}{|\mathbf{C}_i||\mathbf{C}_j|}}{\sqrt{\left(\sum_{i=1}^{\tilde{c}}|\tilde{\mathbf{C}}_i| \log \frac{|\tilde{\mathbf{C}}_i|}{n}\right)\left(\sum_{j=1}^{c}|\mathbf{C}_j| \log \frac{|\mathbf{C}_j|}{n}\right)}}. \tag{42}$$

Purity measures the degree of each cluster containing a class of data points. Its value can be calculated by

$$\mathrm{Purity} = \sum_{i=1}^{k}\frac{n_i}{n}P(\mathbf{S}_i), P(\mathbf{S}_i) = \frac{1}{n_i}\max_j P\left(n_i^j\right), \tag{43}$$

where $\mathbf{S}_i$ is a specific cluster of size $i$ and $n_i^j$ represents the data point of the $i$th category is assigned to the $j$th cluster. The larger the values of the above three metrics, the better the performance of the algorithm.

### 3.3. Parameter settings

Before the next algorithm performance introduction, it is necessary to declare the parameter selection of these algorithms. In particular, the operation of BMVC requires anchor samples, and the specific number of them in their paper is not explicitly stated. In experiments, we uniformly select twelve percent of all data points in the initial data set as the anchor samples. The various parameters in the rest of algorithms are set according to the best performance in original papers.

### 3.4. Experiment results

This section focuses on the specific performance of the above algorithms on seven public data sets. At the same time, we also consider the comparison with the traditional clustering method k-means, but it cannot be used for the clustering of multi-view data. Therefore, we directly splice the features of these multi-view data sets to form single-view data sets in this paper. Thus, in order to contrast the performance divergence between them for convenience, we list the concrete values of ACC, NMI, Purity in Tables 2–4.

By observing Tables 2–4, we can draw the following conclusions. First of all, the majority of methods have better effects than k-means on most data sets, which prove that using information complementarity between multiple views can actually improve clustering effects. Certainly, some algorithms do not achieve better results on some data sets. For example, on the data set NUS-WIDE, the methods SwMC and AMGL are less effective than k-means. Then, when the data presents a high-dimensional situation, the methods MSC_IAS and LMSC also achieve relatively good results on the data set Extended Yale-B and 3-Sources Text, indicating that these methods can indeed learn a good subspace representation or potential representation to promote clustering. Finally, Fig. 5 depicts the length of time these algorithms run on all data sets, and it is clear that k-means is far ahead in terms of operational efficiency. However, it is easy to understand the reason for this difference. Multi-view clustering algorithms need to consider effective fusion between different views, and the process of fusion will bring about a surge in computational complexity. Nevertheless, it is worth noting that the running time of BMVC is comparable to that of k-means, even the running time is much lower than that of k-means and the performance is also the best of all algorithms on the larger data set Caltech101, which benefits from the introduction of binary code in its algorithm.

In order to show the clustering effects more intuitively, the visualizations of experimental results are provided. Here, we select the data set MNIST and display the results of these algorithms on it. Considering all views of the data set, we stitch the three views of MNIST and reduce the final feature dimensionality to 2 dimensions. The results of various multi-view clustering methods are visualized in Fig. 4.

As shown in the Fig. 4, different colors represent different clusters, so it seems that almost all algorithms can clearly separate MNIST into various categories. The distance between diverse clusters is as large as possible and the distance of data points in the same cluster is as small as possible, which is a vital principle of clustering results. From this perspective, the clustering result of LMSC is obviously not very good, which can also be seen by the

**Table 4**
Purity with diverse multi-view clustering algorithms.

| Data set | ALOI | MNIST | NUS-WIDE | MSRC-v1 | Yale-B | Caltech101 | 3-Source |
|---|---|---|---|---|---|---|---|
| k-means | 0.5023 | 0.7750 | 0.2831 | 0.5524 | 0.2123 | 0.2355 | 0.4142 |
| SwMC | 0.4310 | 0.7235 | 0.2369 | 0.8714 | 0.3862 | 0.2057 | 0.3787 |
| BMVC | 0.6960 | 0.8060 | 0.4113 | 0.7762 | 0.4077 | 0.4893 | 0.6805 |
| MLAN | 0.5802 | 0.8255 | 0.3394 | 0.7333 | 0.3462 | 0.3082 | 0.3598 |
| AMGL | 0.5542 | 0.8900 | 0.2375 | 0.7476 | 0.4400 | 0.3659 | 0.3787 |
| MSC_IAS | 0.6701 | 0.8240 | 0.2988 | 0.6952 | 0.8462 | 0.4008 | 0.7041 |
| LMSC | 0.7405 | 0.6150 | 0.3681 | 0.6810 | 0.5446 | 0.4724 | 0.6805 |
| ECMSC | 0.6728 | 0.8620 | 0.3494 | 0.7857 | 0.7246 | 0.4560 | 0.3669 |
| COMIC | 0.3003 | 0.9920 | 0.7288 | 0.9714 | 0.1000 | 0.6363 | 0.3314 |

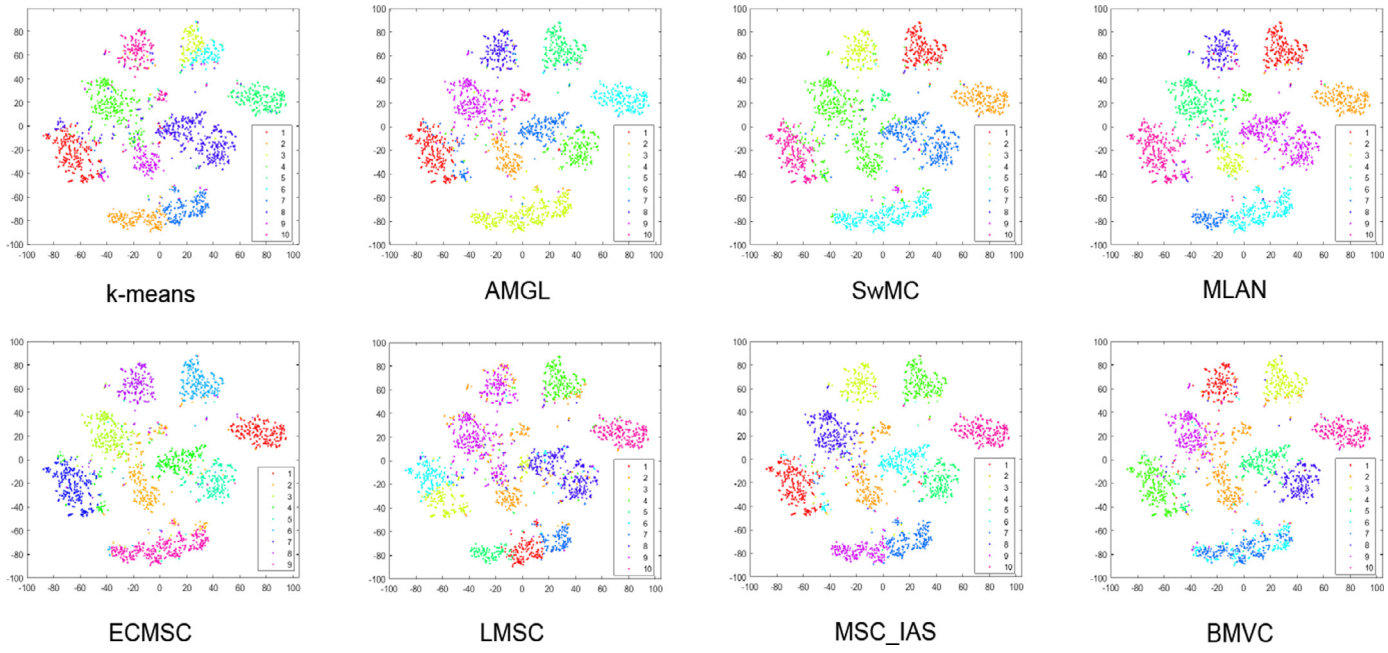



**Fig. 4.** Demonstration of various multi-view clustering algorithms and k-means on the data set MNIST with ten categories, where data points from the same class have the same color.

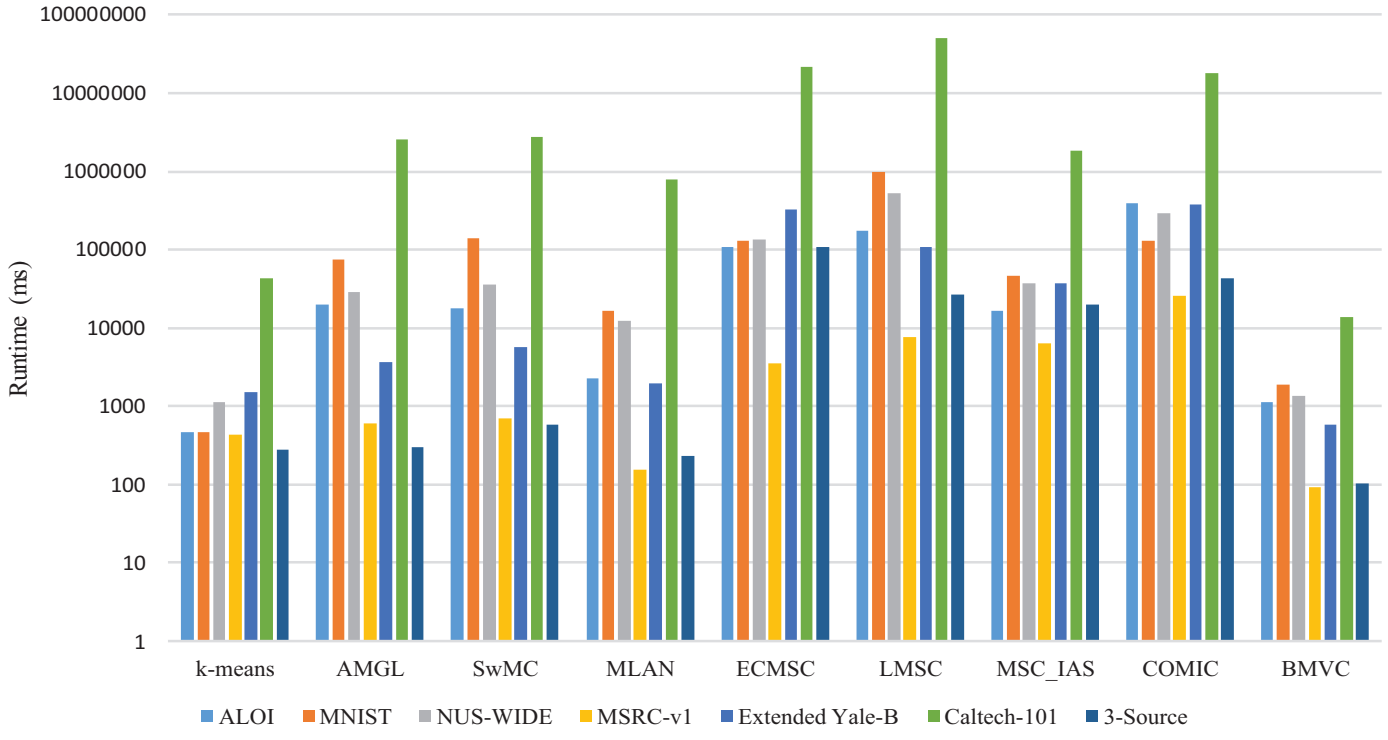



**Fig. 5.** Runtime of different multi-view clustering algorithms on all tested data sets.

numerical values of the metrics. In the meantime, it is worth noting that only k-means, LMSC and MSC_IAS divide the data points into 10 categories, but this does not mean that this division is reasonable. Although some algorithms do not segment enough categories, their effects are apparently gratifying.

## 4. Current challenges and future directions

After the introduction of all algorithms and the analysis of experimental part, it can be seen that there are still some unresolved problems in the field of multi-view clustering, which not only mean challenges for researchers, but also provide some available pointcuts for further research. In general, we have the following points:

- The advantage of multi-view data is that it can describe objects from different angles, so that this representation is more comprehensive and overcomes the shortcomings of singularity and simplification of the data. However, the mining and effective utilization of information complementarity between views still need to be further improved. Currently, the more popular method for multi-view fusion is to assign corresponding weights according to the importance of views and then add them, which is interpretable and easy to understand in mathematical theory, but may not be consistent with the actual situation. In the model of ECMSC, the view fusion method based on position-aware seems to open up another mind worthy of the attention of researchers.
- Since data processing calculations are performed on multiple views one by one, the running time of most methods is undoubtedly increased greatly. The running time on small data sets is acceptable, but when the data sets become larger and larger, the time consumption is intolerable and the effects of algorithms also drop dramatically. The introduction of binary coding in BMVC has greatly improved the operating efficiency and also promoted the performance, which is novel and deserves the attention of researchers for further development.
- When the data is in a high-dimensional situation, the performance of algorithms is also affected. This is because high-dimensional data often has a large amount of redundant information, which not only fails to supplement valid information, but also jeopardizes good data feature representation. Therefore, effective data dimensionality reduction [78] and retention of important data features are particularly important for solving clustering problems of this type of data.
- In current multi-view clustering, most algorithms often only consider the correlation between views from the matrix level, so as to discover the information consistency and information complementarity between different views. It is more reasonable and effective to explore the high-order correlation between perspectives based on the form of tensor. The work [79] has made new attempts in applying tensor to multi-view clustering and achieved good results, which may be an innovative direction worthy of further research.
- The multi-view clustering models are bound to involve more parameters and corresponding parameter restrictions. Therefore, after the algorithm models are constructed, the designs of optimization algorithms for the models are still difficult, which require researchers to continuously try and explore to achieve the desired effects. At present, ingenious transformation of unknown problems into known and solvable problems is a feasible way to optimize the objective function.

## 5. Conclusion and future work

In this paper, we introduced eight multi-view clustering algorithms in recent years and tested them on seven real-world data sets. At the same time, the three metrics (ACC, NMI, Purity) of each algorithm were revealed after running on these data sets. Also, we paid attention to the running time of these algorithms on all data sets, which has important guiding significance for solving practical problems. Summarily, our work is beneficial for researchers to grasp the advantages and weakness of these described algorithms. Furthermore, researchers interested in this field can overcome the defects of current algorithms based on these analyses. In the next work, we are also preparing to explore a new multi-view clustering algorithm that can be adapted to data sets of different sizes and greatly improve the running speed.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

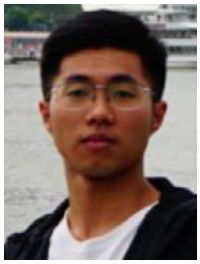
## CRediT authorship contribution statement

**Lele Fu:** Conceptualization, Formal analysis, Methodology, Writing - original draft. **Pengfei Lin:** Conceptualization, Formal analysis, Methodology, Writing - review & editing. **Athanasios V. Vasilakos:** Supervision, Validation, Visualization. **Shiping Wang:** Funding acquisition, Writing - review & editing.

## References

[1] X. He, L. Li, D. Roqueiro, K. Borgwardt, Multi-view spectral clustering on conflicting views, in: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2017, pp. 826–842.
[2] P. Berkhin, A survey of clustering data mining techniques, Grouping Multidimens. Data (2006) 25–71.
[3] L. Houthuys, R. Langone, J.A.K. Suykens, Multi-view kernel spectral clustering, Inf. Fusion 44 (2018) 46–56.
[4] X. Cai, F. Nie, H. Huang, F. Kamangar, Heterogeneous image feature integration via multi-modal spectral clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1977–1984.
[5] C. Lang, G. Liu, J. Yu, S. Yan, Saliency detection by multitask sparsity pursuit, IEEE Trans. Image Process. 21 (2011) 1327–1338.
[6] L. Zhang, Z. Shi, M.-M. Cheng, Y. Liu, J.-W. Bian, J.T. Zhou, G. Zheng, Z. Zeng, Nonlinear regression via deep negative correlation learning, IEEE Trans. Pattern Anal. Mach. Intell. (2019), doi:10.1109/TPAMI.2019.2943860.
[7] S. Kanaanizquierdo, A. Ziyatdinov, A. Pererallua, Multiview and multifeature spectral clustering using common eigenvectors, Pattern Recognit. Lett. 102 (2018) 30–36.
[8] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, S. Yan, MatchinG-CNN meets KNN: Quasi-parametric human parsing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1419–1427.
[9] N. Zeng, Z. Wang, H. Zhang, K.-E. Kim, Y. Li, X. Liu, An improved particle filter with a novel hybrid proposal distribution for quantitative analysis of gold immunochromatographic strips, IEEE Trans. Nanotechnol. 18 (2019) 819–829.
[10] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, Y. Li, A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease, Neurocomputing 320 (2018) 195–202.
[11] N. Zeng, Z. Wang, H. Zhang, Inferring nonlinear lateral flow immunoassay state-space models via an unscented Kalman filter, Sci. China Inf. Sci. 59 (2016) 112–204.
[12] X. Chen, C. Jian, Gene expression data clustering based on graph regularized subspace segmentation, Neurocomputing 143 (2014) 44–50.
[13] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, 2013, ArXiv: Learning.
[14] S. Sun, A survey of multi-view machine learning, Neural Comput. Appl. 23 (2013) 2031–2038.
[15] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: recent progress and new challenges, Inf. Fusion 38 (2017) 43–54.
[16] S. Bickel, T. Scheffer, Multi-view clustering, in: Proceedings of the IEEE International Conference on Data Mining, 2004, pp. 19–26.
[17] G. Tzortzis, A. Likas, Convex mixture models for multi-view clustering, in: Proceedings of the International Conference on Artificial Neural Networks, 2009, pp. 205–214.
[18] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Stat. Soc. Ser. B 39 (1977) 1–22.
[19] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, in: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 2007, pp. 1027–1035.
[20] X. Shen, W. Liu, I. Tsang, F. Shen, Q.S. Sun, Compressed k-means for large-scale clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 2527–2533.

[21] Y. Ding, Y. Zhao, X. Shen, M. Musuvathi, T. Mytkowicz, Yinyang k-means: a drop-in replacement of the classic k-means with consistent speedup, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 579–587.
[22] Y. Li, F. Nie, H. Huang, J. Huang, Large-scale multi-view spectral clustering via bipartite graph, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015, pp. 2750–2756.
[23] Y. Jiang, J. Liu, Z. Li, P. Li, H. Lu, Co-regularized Plsa for multi-view clustering, in: Proceedings of the Asian Conference on Computer Vision, 2012, pp. 202–213.
[24] T. Xia, D. Tao, T. Mei, Y. Zhang, Multiview spectral embedding, IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 40 (2010) 1438–1446.
[25] S. Wang, W. Guo, Sparse multigraph embedding for multimodal feature representation, IEEE Trans. Multimed. 19 (2017) 1454–1466.
[26] R. Vidal, Subspace clustering, IEEE Signal Process. Mag. 28 (2011) 52–68.
[27] H. Kriegel, P. Kröger, A. Zimek, Subspace clustering, Wiley Interdisc. Rev.: Data Mining Knowl. Discov. 2 (2012) 351–364.
[28] S. Wang, W. Pedrycz, Q. Zhu, W. Zhu, Subspace learning for unsupervised feature selection via matrix factorization, Pattern Recognit. 48 (2015a) 10–19.
[29] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, X. Huang, Robust subspace clustering for multi-view data by exploiting correlation consensus, IEEE Trans. Image Process. 24 (2015b) 3939–3949.
[30] Q. Yin, S. Wu, R. He, L. Wang, Multi-view clustering via pairwise sparse subspace representation, Neurocomputing 156 (2015) 12–21.
[31] C. Zhang, H. Fu, S. Liu, G. Liu, X. Cao, Low-rank tensor constrained multiview subspace clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1582–1590.
[32] X. Cao, C. Zhang, H. Fu, S. Liu, H. Zhang, Diversity-induced multi-view subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 586–594.
[33] H. Gao, F. Nie, X. Li, H. Huang, Multi-view subspace clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4238–4246.
[34] J. Gao, J. Han, J. Liu, C. Wang, Multi-view clustering via joint nonnegative matrix factorization, in: Proceedings of the SIAM International Conference on Data Mining, 2013, pp. 252–260.
[35] Y. Yang, F. Shen, Z. Huang, H.T. Shen, X. Li, Discrete nonnegative spectral clustering, IEEE Trans. Knowl. Data Eng. 29 (2017) 1834–1845.
[36] L. Zong, X. Zhang, L. Zhao, H. Yu, Q. Zhao, Multi-view clustering via multi-manifold regularized non-negative matrix factorization, Neural Netw. 88 (2017) 74–89.
[37] S. Wang, W. Guo, Robust co-clustering via dual local learning and high-order matrix factorization, Knowl. Based Syst. 138 (2017) 176–187.
[38] S. Wang, J. Chen, W. Guo, G. Liu, Structured learning for unsupervised feature selection with high-order matrix factorization, Expert Syst. Appl. 140 (2020).
[39] K. Chaudhuri, S.M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, in: Proceedings of the International Conference on Machine Learning, 2009, pp. 129–136.
[40] M.B. Blaschko, C.H. Lampert, Correlational spectral clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[41] D. Niu, J.G. Dy, M.I. Jordan, Multiple non-redundant spectral clustering views, in: Proceedings of the International Conference on Machine Learning, 2010, pp. 831–838.
[42] X. Cai, F. Nie, W. Cai, H. Huang, Heterogeneous image features integration via multi-modal semi-supervised learning model, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1737–1744.
[43] F. Nie, J. Li, X. Li, Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification, in: Proceedings of the International Joint Conferences on Artificial Intelligence Organization, 2016, pp. 1881–1887.
[44] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2014, pp. 2149–2155.
[45] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the International Conference on Machine Learning, 2011, pp. 393–400.
[46] Y. Cheng, R. Zhao, Multiview spectral clustering via ensemble, in: Proceedings of the IEEE International Conference on Granular Computing, 2009, pp. 101–106.
[47] F. Nie, J. Li, X. Li, Self-weighted multiview clustering with multiple graphs, in: Proceedings of the International Joint Conferences on Artificial Intelligence Organization, 2017, pp. 2564–2570.
[48] F. Nie, X. Wang, M.I. Jordan, H. Huang, The constrained Laplacian rank algorithm for graph-based clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016, pp. 1969–1976.
[49] F. Nie, G. Cai, X. Li, Multi-view clustering and semi-supervised classification with adaptive neighbours, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 2408–2414.
[50] Y. Liu, F. Nie, J. Wu, L. Chen, Efficient semi-supervised feature selection with noise insensitive trace ratio criterion, Neurocomputing 105 (2013) 12–18.
[51] B. Mohar, The Laplacian spectrum of graphs, Graph Theory Combinat. Appl. (1991) 871–898.
[52] K. Fan, On a theorem of Weyl concerning eigenvalues of linear transformations, in: Proceedings of the National Academy of Sciences of the United States of America, 1949, pp. 652–655.
[53] X. Wang, X. Guo, Z. Lei, C. Zhang, S.Z. Li, Exclusivity-consistency regularized multi-view subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 923–931.
[54] N. Chen, J. Zhu, E.P. Xing, Predictive subspace learning for multi-view data: a large margin approach, in: Proceedings of Advances in Neural Information Processing Systems, 2010, pp. 361–369.
[55] M.D. Collins, J. Liu, J. Xu, L. Mukherjee, V. Singh, Spectral clustering with a convex regularizer on millions of images, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 282–298.
[56] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2790–2797.
[57] C. Li, R. Vidal, Structured sparse subspace clustering: a unified optimization framework, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 277–286.
[58] Z. Lin, R. Liu, Z. Su, Linearized alternating direction method with adaptive penalty for low-rank representation, in: Proceedings of Advances in Neural Information Processing Systems, 2011, pp. 612–620.
[59] C. Zhang, Q. Hu, H. Fu, P. Zhu, X. Cao, Latent multi-view subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4279–4287.
[60] M. White, X. Zhang, D. Schuurmans, Y. Yu, Convex multi-view subspace learning, in: Proceedings of Advances in Neural Information Processing Systems, 2012, pp. 1673–1681.
[61] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 2765–2781.
[62] H. Hu, Z. Lin, J. Feng, J. Zhou, Smooth representation clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3834–3841.
[63] X. Wang, Z. Lei, X. Guo, C. Zhang, H. Shi, S.Z. Li, Multi-view subspace clustering with intactness-aware similarity, Pattern Recognit. 88 (2019) 50–63.
[64] C. Xu, D. Tao, C. Xu, Multi-view intact space learning, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 2531–2544.
[65] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.
[66] Y. Liu, L. Jiao, F. Shang, An efficient matrix factorization based low-rank representation for subspace clustering, Pattern Recognit. 46 (2013) 284–292.
[67] H. Lu, Z. Fu, X. Shu, Non-negative and sparse spectral clustering, Pattern Recognit. 47 (2014) 418–426.
[68] X. Jing, R. Hu, F. Wu, X. Chen, Q. Liu, Y. Yao, Uncorrelated multi-view discrimination dictionary learning for recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2014, pp. 2787–2795.
[69] A. Gretton, O. Bousquet, A. Smola, B. Schölkopf, Measuring statistical dependence with hilbert-schmidt norms, in: Proceedings of International Conference on Algorithmic Learning Theory, 2005, pp. 63–77.
[70] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, A.J. Smola, A kernel method for the two-sample-problem, in: Proceedings of Advances in Neural Information Processing Systems, 2006, pp. 513–520.
[71] X. Peng, Z. Huang, J. Lv, H. Zhu, J.T. Zhou, COMIC: Multi-view clustering without parameter selection, in: Proceedings of the International Conference on Machine Learning, 2019, pp. 5092–5101.
[72] W. Guo, Y. Shi, S. Wang, A unified scheme for distance metric learning and clustering via rank-reduced regression, IEEE Trans. Syst. Man Cybern. Syst. (2019) 1–12.
[73] T. Hocking, J.P. Vert, F. Bach, A. Joulin, Clusterpath: an algorithm for clustering using convex fusion penalties, in: Proceedings of the International Conference on Machine Learning, 2011, pp. 745–752.
[74] N. Flammarion, B. Palaniappan, F. Bach, Robust discriminative clustering with sparse regularizers, J. Mach. Learning Research 18 (2017) 2764–2813.
[75] Z. Zhang, L. Liu, F. Shen, H.T. Shen, L. Shao, Binary multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2018) 1774–1782.
[76] J. Wang, T. Zhang, J. Song, N. Sebe, H.T. Shen, A survey on learning to hash, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 769–790.
[77] Y. Gong, M. Pawlowski, F. Yang, L. Brandy, L. Bourdev, R. Fergus, Web scale photo hash clustering on a single machine, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 19–27.
[78] S. Wang, W. Zhu, Sparse graph embedding unsupervised feature selection, IEEE Trans. Syst. Man Cybernet. Syst. 48 (2016) 329–341.
[79] J. Wu, Z. Lin, H. Zha, Essential tensor learning for multi-view spectral clustering, IEEE Trans. Image Process. 28 (2019) 5910–5922.



**Lele Fu** received his B.S. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China in 2019. He is currently pursuing the M.E. degree with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His current research interests include machine learning and computer vision.

**Pengfei Lin** is currently pursuing the B.S. degree with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, where he will start pursuing the M.S. degree in 2020. His research interests include data mining, machine learning and computer vision.

**Athanasios V. Vasilakos** is currently a Distinguished Professor with the Computer Science Department, Lulea University of Technology, Sweden. He has authored or co-authored over 600 technical papers in major international journals and conferences, and is the author/co-author of five books and 20 book chapters. His main research interests include cybersecurity, power systems cybersecurity, networking, the IoTs and smart cities, cloud computing, big data analytics, and machine learning. His papers received citations of more than 28400, with h-index= 92. He is also the ISI Highly Cited Researcher (the Highest Scientific Distinction).

Prof. Vasilakos has served as the General Chair and Technical Program Committee Chair for many international conferences. He is also serving/served as an Editor for many leading journals. He is a frequent keynote, panel, and tutorial speaker. Moreover, he is a Consultant to the European Commission.

**Shiping Wang** received his Ph.D. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China in 2014. He worked as a research fellow in Nanyang Technological University from August 2015 to August 2016. He is currently a Full Professor and Qishan Scholar with the College of Mathematics and Computer Science, Fuzhou University. His research interests include machine learning, computer vision and granular computing.