Neural Networks 122 (2020) 279–288

Contents lists available at ScienceDirect

Neural Networks

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

Partition level multiview subspace clustering

Zhao Kang [a], Xinjia Zhao [a], Chong Peng [b], Hongyuan Zhu [c], Joey Tianyi Zhou [d], Xi Peng [e], Wenyu Chen [a,*], Zenglin Xu [a,f,**]

[a] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Sichuan, 611731, China*
[b] *College of Computer Science and Technology, Qingdao University, China*
[c] *Institute for Infocomm Research, A*STAR, Singapore*
[d] *Institute of High Performance Computing, A*STAR, Singapore*
[e] *College of Computer Science, Sichuan University, China*
[f] *Centre for Artificial Intelligence, Peng Cheng Lab, Shenzhen 518055, China*

ARTICLE INFO

Article history:
Received 7 July 2019
Received in revised form 17 September 2019
Accepted 14 October 2019
Available online 6 November 2019

Keywords:
Multi-view learning
Subspace clustering
Partition space
Information fusion

ABSTRACT

Multiview clustering has gained increasing attention recently due to its ability to deal with multiple sources (views) data and explore complementary information between different views. Among various methods, multiview subspace clustering methods provide encouraging performance. They mainly integrate the multiview information in the space where the data points lie. Hence, their performance may be deteriorated because of noises existing in each individual view or inconsistent between heterogeneous features. For multiview clustering, the basic premise is that there exists a shared partition among all views. Therefore, the natural space for multiview clustering should be all partitions. Orthogonal to existing methods, we propose to fuse multiview information in partition level following two intuitive assumptions: (i) each partition is a perturbation of the consensus clustering; (ii) the partition that is close to the consensus clustering should be assigned a large weight. Finally, we propose a unified multiview subspace clustering model which incorporates the graph learning from each view, the generation of basic partitions, and the fusion of consensus partition. These three components are seamlessly integrated and can be iteratively boosted by each other towards an overall optimal solution. Experiments on four benchmark datasets demonstrate the efficacy of our approach against the state-of-the-art techniques.

1. Introduction

Classic clustering methods (Chen et al., 2018; Jain, 2010; Kang, Pan, Hoi and Xu, 2019; Kang, Xu, Wang, Zhu and Xu, 2019; Ng, Jordan, Weiss, et al., 2002; Yang, Shen, Huang, Shen, & Li, 2017) aim to identify underlying group structure in single view data. However, real-world data are often generated from multiple sources (views) (Li, Shao, & Fu, 2018; Sun, Liu, & Mao, 2019; Tang et al., 2018). For instance, documents can have different languages; news can be represented by a combination of texts, images, and videos; an image/video can be described by different visual descriptors such as SIFT, LBP, HOG, and GIST. To conduct clustering with multiview data, the naive way is to treat them as the single-view data by concatenating multiview features directly (Kumar & Daumé, 2011; Liu & Fu, 2018). However, this approach fails to consider view divergence that might prevent various views forming an ideal solution. Consequently, the key problem for multiview clustering is how to effectively integrate the complementary information from different views.

In recent years, numerous multiview clustering techniques have been developed (Chao, Sun, & Bi, 2017). They can be roughly divided into three main types. First, Matrix Factorization (MF) based approaches. In this framework, a common indicator matrix is sought. Many researchers extended the nonnegative MF (NMF) to multiview settings (Huang, Kang, & Xu, 2020; Liu, Wang, Gao and Han, 2013). The classic K-means based multiview clustering methods also belong to this group (Cai, Nie, & Huang, 2013; Chen, Xu, Ye, & Huang, 2013; Huang, Kang, & Xu, 2018). Some kernel-based multiview clustering methods have also been proposed (Liu et al., 2019; Tzortzis & Likas, 2012; Zhou et al., 2019). Second, spectral clustering based approaches (Kang et al., 2019). These methods assume that all the views share the same or similar eigenvector matrix. The representative methods in this category are co-training and co-regularization based multiview clustering (Kumar & Daumé, 2011; Kumar, Rai, & Daume, 2011). Third, subspace clustering based methods. Subspace clustering (Chen, Ye, Xu, & Huang, 2012; Huang, Kang, & Xu, 2019; Kang et al.,

[*] Corresponding author.
[**] Correspondence to: B1-201, Main Building, No.2006, Xiyuan Ave, West Hi-Tech Zone, 611731 Chengdu, Sichuan, P.R.China.
E-mail addresses: cwy@uestc.edu.cn (W. Chen), zlxu@uestc.edu.cn (Z. Xu).

https://doi.org/10.1016/j.neunet.2019.10.010
0893-6080/© 2019 Elsevier Ltd. All rights reserved.

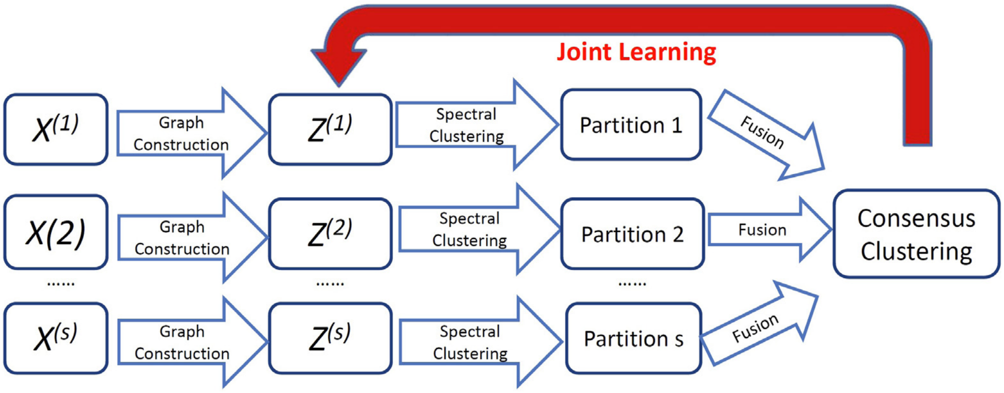


Fig. 1. A flow diagram of our proposed method. We dynamically construct a graph for each view at each iteration, perform spectral clustering to obtain the partition for each view, and integrate the basic partitions to generate the final clustering result.

2019; Kang, Peng, & Cheng, 2017; Peng et al., 2018; Vidal, 2011) simultaneously divides data into multiple subspaces and finds a low-dimensional subspace fitting each group of data points. In order to take advantage of the complementary information from multiple views, varieties of multiview subspace clustering methods have been proposed and achieved great success (Gao, Nie, Li, & Huang, 2015; Zhang, Hu, Fu, Zhu, & Cao, 2017).

Most multiview subspace clustering methods learn a sample affinity graph matrix for each view by deploying features of different views, then a consensus graph Z is built (Gao et al., 2015; Wang et al., 2016; Zhang et al., 2017). Some other approaches directly learn a common graph matrix Z (Abavisani & Patel, 2018). Then the spectral clustering algorithm is implemented on the graph Laplacian constructed from Z to obtain the final clustering result. Therefore, a two-step routine is often adopted. Recently, another class of graph-based multiview clustering methods has also shown impressive performance (Nie, Cai, & Li, 2017; Nie, Li, Li, et al., 2016; Zhan, Zhang, Guan, & Wang, 2017). These methods learn a common affinity graph or fuse different views to one graph based on adaptive neighbors. To be specific, each data point x_i is connected by x_j with a probability s_{ij}, and such probability can be seen as the similarity between them.

Despite the significant progress in multiview clustering brought by the above automatic graph learning based approaches, yet the challenge of fully making use of the richness and complementarity of multiview information leaves space for further improving the clustering results. Orthogonally to achieving the multiview consensus graph, in this paper, we manage to fuse multiview information in partition level based on two intuitive assumptions and reach the consensus clustering. In other words, we integrate multiview information through finding a shared clustering from multiple clustering results of views.

The move from the perspective of the data space to the natural space for clustering, i.e., the space of all partitions, this change in paradigm accompanies a number of advantages. First, the interaction between individual partitions from each view and the final result is incorporated. Second, it inherits the robustness and empirical good performance of ensemble learning. Third, integrating multiview information in partition level rather than graph construction stage enhances the representation ability of multiview clustering methods. For current methods, once the common graph is constructed, the final result is fixed. However, some views might contain an irrelevant or noisy representation that can severely damage the consensus graph and lead to degraded performance. More often than not, the similarities between samples may be manifested differently by different views. For example, two video clips that present the same content but in different languages, their audio content will be different. For our method, we directly fuse multiple partitions into an integrated one since the premise for multiview clustering is that there exists a shared cluster structure.

Beyond fusing basic partitions, we further combine it with graph construction. Consequently, a unified framework which integrates graph construction, spectral clustering, and consensus clustering is established. Based on an iterative optimization strategy, the high-quality consensus clustering is employed to guide the graph construction and the updating of basic partitions, which later contributes to a new consensus partition. Fig. 1 gives the illustration of our method. In summary, the main contributions of this work are two-fold:

- We propose to fuse multiview information in the partition level. A novel fusion mechanism is developed to find the consensus partition and assign weights to each basic partition.
- This paper presents a unified multiview clustering framework which simultaneously learns a graph for each view, a partition for each view, and a consensus partition. By leveraging the inherent interactions between these three subtasks, they can be boosted by each other. Extensive experiments on benchmark datasets validate the superiority of our model.

2. Related work

2.1. Notation summary

Throughout this paper, matrices are denoted as capital letters and vectors are written as lower case letters. For an arbitrary matrix $A \in \mathbb{R}^{m\times n}$, $A_{i,:}$ and $A_{:,j}$ denote the ith row and jth column of A, respectively. The ℓ_2-norm of vector x is represented by $\|x\| = \sqrt{x^T \cdot x}$, where T is the transpose operator. $Tr(A)$ is the trace of A. The Frobenius norm of A is defined as $\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}$. $A \geq 0$ indicates all entries of A are nonnegative. I is the identity matrix with a proper size.

2.2. Subspace Clustering (SC)

Given a dataset $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m\times n}$, SC can learn the affinity graph matrix Z by the so-called “self-expressiveness” property, which states that each data point can be represented as a linear combination of other points. More precisely, for $\forall x_i$, it can be reconstructed as follows

$$x_i = \sum_j x_j z_{ij} \quad s.t. \quad z_{ij} \geq 0, \tag{1}$$

where reconstruction coefficient z_{ij} behaves like the similarity between x_i and x_j.

Based on Eq. (1), a series of SC methods have been developed under the framework

$$\min_{Z} \|X - XZ\|_F^2 + \alpha \mathcal{R}(Z) \quad s.t. \quad Z \geq 0, \tag{2}$$

where $\alpha > 0$ is the trade-off parameter and $\mathcal{R}(Z)$ is some regularization function, which varies in different algorithms (Elhamifar & Vidal, 2013; Li, Liu, Tang, & Lu, 2015; Liu et al., 2013; Peng, Lu, Yi and Tang, 2018; Zhang et al., 2019). In this paper, we simply use the Frobenius norm. Once Z is obtained, spectral clustering is performed to obtain the final clustering results, i.e.,

$$\min_{F, F^T F = I} Tr(F^T L F), \tag{3}$$

where graph Laplacian $L = D - Z$ with diagonal matrix D defined as $d_{ii} = \sum_j z_{ij}$, $F \in \mathbb{R}^{n \times c}$ is the cluster indicator matrix, and c is number of clusters. Constructing a powerful graph that can effectively depict the intrinsic connection of data points is the critical step to make the SC algorithms achieve promising performance.

2.3. Multiview subspace clustering

Let $X = [X^1; X^2; \cdots; X^v] \in \mathbb{R}^{d \times N}$ represent the multiview data matrix which consists of v different views, where $X^s \in \mathbb{R}^{d_s \times N}$ denote the sth view data matrix and d_s is the dimension of the data in the sth view. Cao, Zhang, Fu, Liu, and Zhang (2015) and Gao et al. (2015) propose to learn a graph on individual view by solving

$$\min_{Z^s} \sum_s \|X^s - X^s Z^s\|_F^2 + \alpha f(Z^s), \tag{4}$$

where f denotes some regularization term of Z^s. Then the simplest way, i.e., taking the average of individual Z^s, is utilized to achieve consensus graph Z (Cao et al., 2015; Wang et al., 2016). This approach does not take full advantage of complementary information.

Other methods just consider the consistency graph of all views and their objective function can be written as Abavisani and Patel (2018) and Zhuge et al. (2017)

$$\min_{Z} \sum_s \|X^s - X^s Z\|_F^2 + \alpha \mathcal{R}(Z), \tag{5}$$

One limitation of this approach is that one common Z is hard to preserve the flexible local manifold structures for all views (Wang et al., 2016).

We can observe that the above approaches seek to fuse multiview information in the data space by construction a shared graph Z. We argue that fusion in this early stage might fail to obtain the optimal clustering result since some information might get lost during this process. For instance, many real-world data are often contaminated due to noise or outliers, which leads to a poor quality graph. Consequently, degraded clustering performance is generated. Multiview clustering searches for the clusters agreeing across all the views. Hence it makes more sense to fuse multiview information in partition level since each partition will capture the intrinsic cluster structure. Moreover, it is easier to find an agreement in partition space than the data space since the more informative partitions are deployed. Besides, when the number of views increases, there is little common space shared by all the views in the data space. For our method, if some partition is not suitable, its contribution can be easily controlled by assigning a small weight, so as to prevent it from adversely affecting consensus clustering.

2.4. Multiview Ensemble Clustering (MVEC)

Recently, Tao et al. proposed MVEC (Tao, Liu, Li, Ding, & Fu, 2017) method. To deal with multiview data, MVEC adopts an ensemble way. It generates basic partitions for each view and integrates them to reach an agreement. It differs from our method in several aspects. First, a different approach is used to produce the basic partitions. For each view, a number of partitions are generated by the random parameter selection strategy. In contrast, only one partition exists for each view in our method. Second, a low-rank and sparse decomposition technique is deployed to explore the connection among views and detect the noises in each view. We develop a straightforward way to integrate the basic partitions in this paper. Third, the generation of basic partitions and fusion of basic partitions are conducted in a two-step approach. On the contrary, our model is a unified framework. Thus, the high-quality consensus partition is iteratively used to guide the generation of basic partitions, which later contributes to a new consensus partition. In summary, our method is totally different from MVEC. The experimental results also demonstrate the superiority of our proposed technique compared to MVEC.

3. Proposed method

First, we need to obtain the partition for each view. Unlike many existing subspace clustering methods using a two-step approach, we combine graph construction and spectral clustering. Based on Eq. (4), we have

$$\begin{aligned} \min_{Z^s, F_s} & \sum_s \|X^s - X^s Z^s\|_F^2 + \alpha \|Z^s\|_F^2 + \beta Tr(F_s^T L^s F_s) \\ & s.t. \quad F_s^T F_s = I, \quad Z^s \geq 0, \end{aligned} \tag{6}$$

where $F_s \in \mathbb{R}^{N \times c}$ is the partition result for view s. With these basic partitions F_s, how do we integrate them to find a consensus clustering? To address this key problem, we propose two intuitive assumptions: (i) each partition is a perturbation of the consensus clustering; (ii) the partition that is close to the consensus clustering should be assigned a large weight. Next, we need to express them in mathematical language.

Foremost, we need to define distances between partitions. Different from classification or regression, the cluster indicator matrix for each view is not unique. In general, for each unique clustering with c clusters, there are $c!$ (c factorial) equivalent representations. Therefore, it is not correct to directly apply the Euclidean distance to measure the difference between different F_s. To circumvent this obstacle, we can use $F_s F_s^T$ instead. In essence, it represents the similarities among all data points in s view. It is easy to understand that it is invariant with respect to $c!$ permutations. Then we can measure the disagreements between partitions in terms of similarities among samples. If two similarity matrices are close, their corresponding clustering results should be similar. Based on these, in this paper, we design the following partitions fusion objective function

$$\min_{Y \in \mathbb{R}^{N \times c}, Y^T Y = I} \sum_s w_s \|YY^T - F_s F_s^T\|_F^2, \tag{7}$$

where Y is the consensus cluster indicator matrix and w_s is the weight for view s. As a result, the consensus clustering resides in some partitions' neighborhood. To some extend, uncertainties or errors are allowed in individual partitions. This enhances the representation ability of the consensus clustering. The smaller the squared distance between a partition and the consensus clustering Y is, the better the partition is, the larger the weight is.

To avoid solving w_s and introducing additive hyperparameters for w_s, we set w_s to be stationary at the beginning and update it correspondingly after we update Y according to the following equation

$$w_s = \frac{1}{2\|YY^T - F_sF_s^T\|_F}. \tag{8}$$

In fact, Eq. (8) is nothing but the inverse distance weighting.

Eventually, by combining Eqs. (6) and (7), our proposed Partition level Multiview Subspace Clustering (PMSC) can be formulated as

$$\begin{aligned} \min_{Z^s,F_s,Y} \sum_s & \underbrace{\|X^s - X^sZ^s\|_F^2 + \alpha\|Z^s\|_F^2}_{\text{graph construction}} + \underbrace{\beta Tr(F_s^T L^s F_s)}_{\text{spectral clustering}} \\ & + \underbrace{\gamma w_s\|YY^T - F_sF_s^T\|_F^2}_{\text{partition fusion}} \\ & s.t. \quad F_s^TF_s = I, Z^s \geq 0, Y^TY = I. \end{aligned} \tag{9}$$

Note that L^s is a function of Z^s, which is taken cared of in Eq. (11). The properties of our proposed formulation equation (9) are summarized as follows.

- Orthogonal to existing multiview clustering methods, our proposed method integrates multiview information in partition level. This late fusion is robust to variations in different representations since each view is assumed to share the unique cluster structure.
- From the objective function equation (9), we can see that it seamlessly integrates the graph learning, spectral clustering, and partition fusion. Based on alternating optimization strategy, the high-quality consensus partition is used to guide the graph construction and spectral clustering. This joint learning strategy facilitates to obtain the optimal final solution. The diagram of our method is shown in Fig. 1.
- Eq. (9) is a kind of end-to-end learning. With the original data X^s as input, it finally outputs the cluster label matrix Y.

4. Optimization

To solve the constrained problem in Eq. (9), we design an alternative algorithm. Z^s, F^s, Y can be solved effectively by fixing the others.

Z^s-subproblem: By fixing F^s and Y to constant, we update Z^s by solving

$$\min_{Z^s} \sum_s \|X^s - X^sZ^s\|_F^2 + \alpha\|Z^s\|_F^2 + \beta Tr(F_s^T L^s F_s). \tag{10}$$

Note that Zs are independent for each view, hence we can solve them separately. Moreover, $Tr(F^TLF) = \sum_{ij} \frac{1}{2}\|F_{i,:} - F_{j,:}\|^2 z_{ij}$. For convenience, we write Eq. (10) in equivalent vector form and ignore the subscript/superscript tentatively. We get

$$\min_{Z_{:,i}} \|X_{:,i} - XZ_{:,i}\|^2 + \alpha Z_{:,i}^T Z_{:,i} + \frac{\beta}{2} h_i^T Z_{:,i}, \tag{11}$$

where $h_i \in \mathbb{R}^{N\times 1}$ is a vector with the jth element as $h_{ij} = \|F_{i,:} - F_{j,:}\|^2$. Taking the derivative with respect to $Z_{:,i}$ and setting it to zero, we have

$$Z_{:,i} = (X^TX + \alpha I)^{-1}(X^TX_{:,i} - \frac{\beta}{4}h_i). \tag{12}$$

Note that once parameter α is given, the inverse is fixed in every iteration. Therefore, we only calculate it once.

F_s-subproblem: After dropping all other unrelated terms with respect to F_s, we obtain

$$\min_{F_s, F_s^TF_s=I} \sum_s \beta Tr(F_s^T L^s F_s) + \gamma w_s\|YY^T - F_sF_s^T\|_F^2. \tag{13}$$

Again, F_s can be solved separately for each view, so we ignore superscript/subscript. It yields

$$\min_{F,F^TF=I} Tr(F^TMF) \tag{14}$$

where $M = \beta L - 2\gamma wYY^T + \gamma wI$. It is well-known that the optimal solution is the c eigenvectors of M corresponding to the c smallest eigenvalues.

Y-subproblem: When F_s and Z_s are fixed, we have

$$\min_{Y,Y^TY=I} Tr(Y^TPY), \tag{15}$$

where $P = \sum_s w_s(I - 2F_sF_s^T)$. The solution is the eigenvectors corresponding to the smallest c eigenvalues of P.

We repeat the updates iteratively and stop it if the maximum iteration number 200 is reached or the relative change of Y is less than 10^{-3}. In summary, the entire algorithm of solving (9) is outlined in Algorithm 1. Due to the involvement of matrix inversion and singular value decomposition (SVD), the complexity of our algorithm is bounded by $O(N^3)$ in general. This is similar to several state-of-the-art methods, e.g., Chen et al. (2018), Gao et al. (2015), Kumar and Daumé (2011), Kumar et al. (2011) and Nie et al. (2016). After obtaining Y, we run K-means to compute final discrete indicator matrix.

Algorithm 1: Optimization for PMSC

```
Input: Multiview matrix $X^1, \cdots, X^v$, cluster number $c$,
parameters $\alpha$, $\beta$, $\gamma$.
Output: $Z^s$, $F_s$, $Y$.
Initialize: Random matrix $Fs$, $w_s = 1/v$.
REPEAT
    for view 1 to $v$ do
        Update each column of $Z$ according to (12);
        Solve the subproblem (14);
    end for
    Solve the subproblem (15);
    Update $w_s$ via (8) for each view.
UNTIL stopping criterion is met
```

5. Experiments

5.1. Datasets

For a fair comparison, we manage to use the same datasets that are widely used in comparison methods, e.g., Gao et al. (2015), Kumar and Daumé (2011), Kumar et al. (2011), Nie et al. (2016), Tao et al. (2017) and Xu, Tao, and Xu (2015). The statistics of these datasets are summarized in Table 1.

BBC dataset consists of 4 views by splitting each document into four related segments. Each segment contains at least 200 characters and is constituted by consecutive textual paragraphs.

Reuters[1] is a textual dataset written in five different languages. We use the subset that is written in English, while the other 4 views are its corresponding translations in 4 different languages.

Handwritten numerals (HW) dataset consists of 2000 images for 0–9 digit classes, 200 samples for each class. There are six kinds of features that are available.

Caltech101[2] is an object recognition dataset consisting of images. Following previous work (Gao et al., 2015), the widely used 20 classes (**Caltech20**) "Brain, Camera, Face, Ferry, Rhino,

[1] http://archive.ics.uci.edu/ml/datasets.html.

[2] http://www.vision.caltech.edu/ImageDatasets/Caltech101/.

Table 1
Description of the datasets (# features).

View	BBC	Reuters	HW	Caltech20
1	Segment1 (4659)	English (2000)	Profile correlations (216)	Gabor (48)
2	Segment2 (4633)	French (2000)	Fourier coefficients (76)	Wavelet moments (40)
3	Segment3 (4665)	German (2000)	Karhunen coefficients (64)	CENTRIST (254)
4	Segment4 (4684)	Spanish (2000)	Morphological (6)	HOG (1984)
5	–	Italian (2000)	Pixel averages (240)	GIST (512)
6	–	–	Zernike moments (47)	LBP (928)
Data points	145	1200	2000	2386
Classes	2	6	10	20
Type	Text	Text	Image	Image

Pagoda, Snoopy, Wrench, Stapler, Leopards, Hedgehog, Garfield, Binocular, Motorbikes, Windsor Chair, Car-Side, Dolla-Bill, Stop-Sign, Yin-yang, and Water-Lilly" are used. Six kinds of features are extracted and used in the experiment.

5.2. Experiment setup

To evaluate the performance of the presented method, we compare it with several state-of-the-art multiview clustering methods.

- The classic K-means clustering algorithm (KM): It is included as a baseline method. The concatenated features with equal weight are used. In other words, we assume that all the views are of the same importance to the clustering task.
- Co-trained multiview spectral clustering (Co-train) (Kumar & Daumé, 2011): This method learns multiple Laplacian eigenspace via a co-training approach over each individual one.
- Co-regularized multiview spectral clustering (Co-reg) (Kumar et al., 2011): This method utilizes a co-regularization term to make the partitions in different views agree with each other.
- Multiview kernel K-means clustering (MVKKM) (Tzortzis & Likas, 2012): In MVKKM, views are expressed in terms of given kernel matrices and these kernels are integrated by assigning a weight for each kernel.
- Robust multiview K-means clustering (RMKMC) (Cai et al., 2013): This method integrates data's multiple representations via structured sparsity-inducing norm to make it more robust to outliers.
- Multiview clustering with self-paced learning (MSPL) (Xu et al., 2015): MSPL learns the multiview model from easy to complex examples/views. A probabilistic smoother weighting scheme is proposed to define easy and complex.
- Autoweighted multiple graph learning (AMGL) for multiview clustering (Nie et al., 2016): AMGL is a multiview extension of spectral clustering. A weight is automatically learned for each graph which is constructed based on adaptive neighbors.
- Multiview ensemble clustering (MVEC) (Tao et al., 2017): MVEC generates basic partitions for each view and integrates them to reach an agreement. A low-rank and sparse decomposition technique is deployed to explore the connection among views and detect the noises in each view.
- Multiview subspace clustering (MVSC) (Gao et al., 2015): This method simultaneously learns multiple graphs and a shared cluster structure, the latter ensures the consistence among different views.
- Diversity-induced multiview subspace clustering (DiMSC) (Cao et al., 2015): This method learns multiple graphs and their average is taken as the input for spectral clustering. The Hilbert Schmidt Independence Criterion (HSIC) is utilized as diversity regularizer term to explore the complementary information of multiple views.

In addition, we perform normalization on the data so that all the values of each view are in the range $[-1, 1]$ according to Cai et al. (2013). We find the best combination of penalty parameters by grid search and tune them to achieve the best performance for all methods.

5.3. Evaluation metrics

To quantitatively assess our algorithm's performance on the clustering task, we use the popular measures, i.e., accuracy (Acc), Purity, and normalized mutual information (NMI) (Kang, Wen, Chen and Xu, 2019; Peng, Kang, Cai and Cheng, 2018).

Acc discovers the one-to-one relationship between clusters and classes. Let l_i and $\hat{l}_i$ be the clustering result and the ground truth cluster label of x_i, respectively. Then the Acc is defined by

$$Acc = \frac{\sum_{i=1}^{n} \delta(\hat{l}_i, map(l_i))}{n},$$

where n is the total number of samples, delta function $\delta(x, y)$ equals one if and only if $x = y$ and zero otherwise, and map$(\cdot)$ is the best permutation mapping function that maps each cluster index to a true class label based on Kuhn–Munkres algorithm (Chen, Donoho, & Saunders, 2001).

The second evaluation metric that we adopt is the purity, which evaluates the extent to which the most common category in each cluster (Zhao & Karypis, 2001). It is computed as follows:

$$\text{Purity} = \sum_{i=1}^{c} \frac{n_i}{n} P(C_i), \quad P(S_i) = \frac{1}{n_i} max_j(n_i^j),$$

where n_i is the number of points in cluster C_i and n_i^j represents the total number of points that the ith input group is assigned to the jth category. There are c categories in total. It is easy to see that a larger Purity indicated better clustering performance.

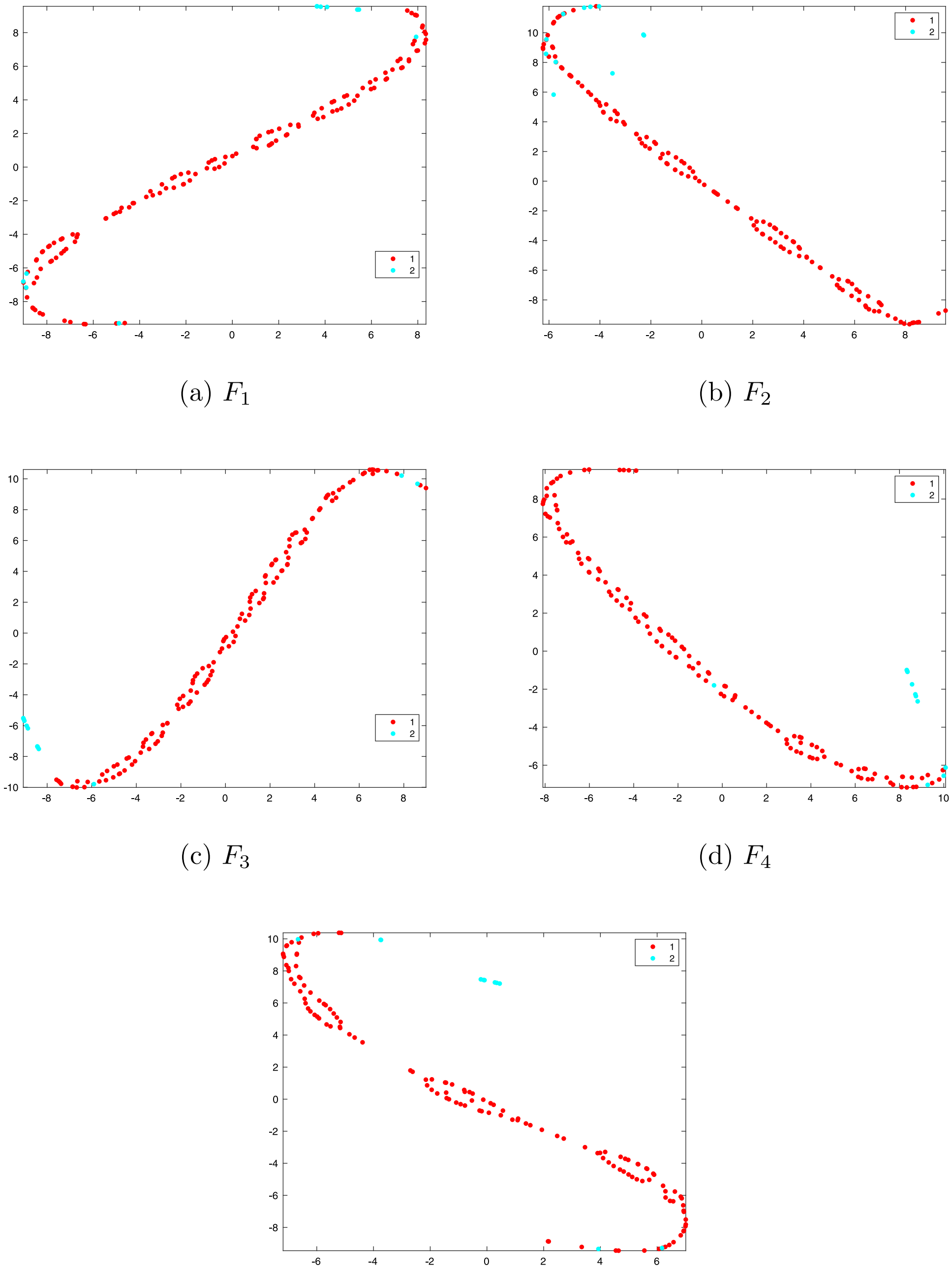
The NMI measures the quality of clustering. Given two sets of clusters L and $\hat{L}$,

$$\text{NMI}(L, \hat{L}) = \frac{\sum_{l \in L, \hat{l} \in \hat{L}} p(l, \hat{l}) \log(\frac{p(l,\hat{l})}{p(l)p(\hat{l})})}{\max(H(L), H(\hat{L}))},$$

where $p(l)$ and $p(\hat{l})$ represent the marginal probability distribution functions of L and $\hat{L}$, respectively, induced from the joint distribution $p(l, \hat{l})$ of L and $\hat{L}$. $H(\cdot)$ is the entropy function. The greater NMI means the better clustering performance.

Each method is repeated 10 times and the mean and standard deviation (std) values are reported. Tables 2–5 show the clustering results on the four benchmark datasets, respectively. In most cases, our proposed PMSC method achieves the best clustering performance in comparison with other state-of-the-art multiview clustering methods. In specific, we have the following observations.

1. Our proposed PMSC method always performs better than the current state-of-the-art multiview subspace clustering



(a) F_1

(b) F_2

(c) F_3

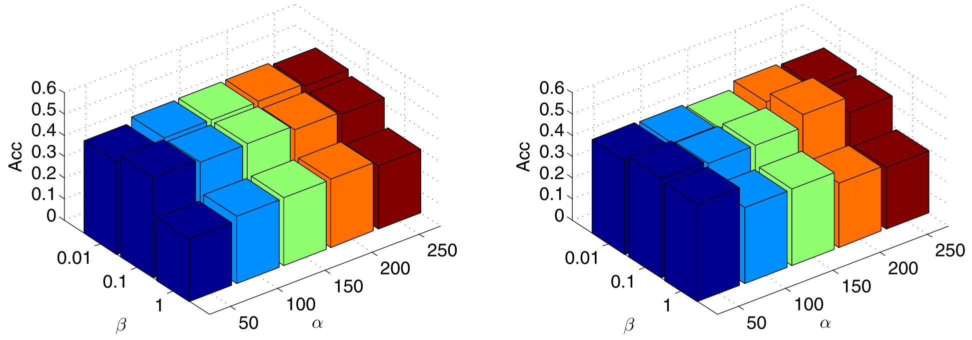
(d) F_4

(e) Y

Fig. 2. The visualization of basic partitions Fs and the consensus partition Y. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

methods MVSC and DiMSC in terms of ACC and NMI. This is mainly due to that our approach integrates the multiview information in the partition level, which is completely different from MVSC and DiMSC since they perform early fusion. In some cases, PMSC reports a lower purity than MVSC and DiMSC.

2. Compared to ensemble clustering method MVEC, our approach shows better accuracy. The improvement is considerable in BBC, Reuters, and HW datasets, i.e., 7%, 9%, 17%, respectively. In terms of NMI, our method also owns a big advantage on the above three datasets. This demonstrates the efficacy of our partition fusion strategy.
3. With respect to AMGL, another type of graph construction based multiview spectral clustering method, our developed PMSC also wins by a very large margin in most cases. For example, on BBC data, the improvement is about 6%, 26%, 5% on ACC, NMI, Purity, respectively; on Reuters, they are 21%, 15%, 40%.
4. Multiview methods often perform better than KM, which merely concatenates all features from different views. This confirms that multiview techniques can effectively explore supplementary information from multiple views to improve the clustering task. However, in some datasets, e.g., BBC and Reuters, multiview extensions of K-means (i.e., MVKKM and RMKMC) even produce worse results than KM. This phenomenon has been observed by some previous researchers (Yang et al., 2013; Zhang, Fu, Liu, Liu, & Cao, 2015).



(a) $\gamma = 0.001$

(b) $\gamma = 0.01$

Fig. 3. Sensitivity analysis of parameters for our method over Caltech20 dataset evaluated with clustering accuracy.

Table 2
Clustering performance on BBC (%).

Method	ACC	Purity	NMI
KM	91.59(0.31)	90.24(0.24)	14.10(1.30)
Co-train	91.27(0.00)	87.57(1.20)	3.50(0.00)
Co-reg	90.90(0.76)	90.78(1.40)	6.80(0.30)
MVKKM	84.00(6.13)	89.01(2.35)	8.30(0.64)
RMKMC	91.31(0.62)	89.67(1.80)	8.00(0.74)
MSPL	80.41(13.24)	90.41(0.00)	10.11(9.48)
AMGL	89.66(0.00)	91.00(0.67)	11.2(0.00)
DiMSC	93.79(0.00)	94.62(0.00)	13.71(0.00)
MVEC	88.97(0.00)	94.48(0.00)	1.03(0.00)
MVSC	91.03(0.00)	95.62(0.00)	0.41(0.00)
PMSC	**95.86**(0.00)	**95.86**(0.00)	**37.42**(0.00)

Table 3
Clustering performance on reuters (%).

Method	ACC	Purity	NMI
KM	24.57(4.52)	25.48(4.37)	11.78(5.01)
Co-train	17.00(0.10)	17.15(0.07)	9.40(0.11)
Co-reg	20.62(1.24)	20.95(1.32)	2.33(0.34)
MVKKM	20.48(3.82)	20.65(3.83)	5.77(3.66)
RMKMC	22.42(6.54)	22.55(6.57)	7.21(7.29)
MSPL	24.87(5.98)	28.12(4.97)	11.50(4.28)
AMGL	18.35(0.15)	20.08(0.54)	6.38(1.00)
DiMSC	39.60(1.32)	46.28(1.74)	18.17(0.64)
MVEC	31.08(0.00)	**82.48**(0.09)	11.92(0.01)
MVSC	25.08(0.39)	80.11(5.50)	6.60(0.68)
PMSC	**40.18**(2.32)	60.07(3.56)	**21.83**(1.75)

Table 4
Clustering performance on HW (%).

Method	ACC	Purity	NMI
KM	54.46(5.60)	58.64(2.92)	58.25(0.85)
Co-train	71.42(4.21)	74.86(2.62)	71.06(1.07)
Co-reg	83.38(7.35)	85.17(4.98)	77.97(2.92)
MVKKM	58.81(3.50)	62.40(3.40)	62.91(2.60)
RMKMC	63.04(3.36)	65.74(2.16)	66.57(1.18)
MSPL	68.00(1.12)	68.99(1.17)	70.42(1.95)
AMGL	73.61(10.29)	76.48(8.54)	81.86(4.53)
DiMSC	42.72(1.94)	45.65(0.97)	37.89(0.87)
MVEC	66.93(5.51)	79.95(1.73)	70.69(2.55)
MVSC	79.60(2.54)	87.19(1.48)	73.89(1.93)
PMSC	**83.81**(6.76)	**87.34**(3.07)	**82.05**(2.93)

Table 5
Clustering performance on Caltech20 (%).

Method	ACC	Purity	NMI
KM	31.40(1.30)	60.06(0.38)	37.05(0.41)
Co-train	38.94(2.10)	69.77(1.42)	50.90(1.12)
Co-reg	34.38(0.79)	65.59(1.03)	46.42(0.96)
MVKKM	44.87(2.49)	72.84(0.72)	54.06(1.23)
RMKMC	33.35(1.47)	64.22(0.89)	42.44(0.67)
MSPL	33.49(0.00)	34.24(0.00)	35.80(0.00)
AMGL	52.28(2.91)	67.60(2.31)	56.61(1.93)
DiMSC	33.89(1.45)	37.78(1.35)	39.33(1.16)
MVEC	52.19(4.25)	60.36(3.21)	**59.78**(1.10)
MVSC	44.96(2.06)	50.87(2.35)	45.36(0.88)
PMSC	**52.63**(0.89)	**72.93**(2.57)	48.35(2.82)

5. For Co-train and Co-reg, which are both based on spectral clustering, our method outperforms them on all datasets in terms of ACC and Purity. For NMI, there is one exception. On Caltech20, our NMI score is 48.25, which is lower than 50.90 of Co-train.

In summary, the proposed method PMSC obtains highly competitive performance with state-of-the-art techniques. This verifies the effectiveness of partition level multiview information fusion.

5.4. Experimental results

To better illustrate how our method works, we display the visualized partitions over BBC dataset in Fig. 2. Note that there are only two classes for BBC dataset, so the dimension of F is 145 × 2. We can observe that F_s share a similar cluster pattern, which is consistent with the underlying assumption that multiple views admit the same cluster structure. Thus, we can achieve consensus clustering easier than approaches implemented in feature space, where features or graphs from different views often show differently. However, we notice that different partitions have different orientations. Therefore, we cannot directly measure the differences among partitions based on the Frobenius norm, e.g., $\|Y - F_1\|_F^2$. Instead, Eq. (7) is proposed.

In addition, we can also see that some partitions can distinguish the classes pretty good. For F_3 and F_4, three blue points are in the same line of red points. This is worse in F_1 and F_2. Consequently, our method can find a good clustering as shown in Fig. 2e. Therefore, the proposed partition fusion method is validated.

Furthermore, we take BBC dataset as an example to show the dynamics of weight in Fig. 4. At the beginning, our initialization $w_s = 1/v$ in Algorithm 1 treats each view equally. After the 1st iteration, their values become 0.2514, 0.3334, 0.3345, 0.3390, respectively. Though w_1 has the smallest value, it eventually ranks the 2nd. This demonstrates that our algorithm is robust to initialization. In addition, we show the convergence curves of the proposed algorithm on each dataset in Fig. 5. One can see that they converge very fast.

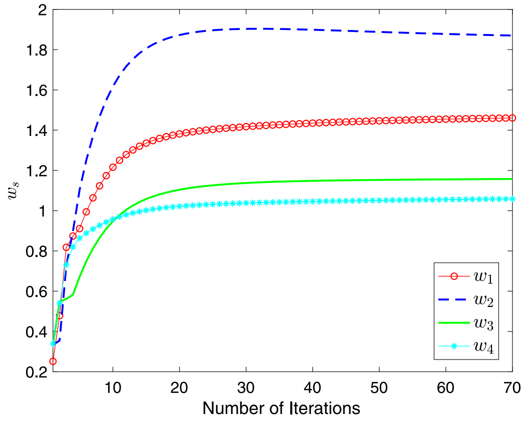


Fig. 4. The evolution of view weights on BBC data.

5.5. Parameter sensitivity analysis

There are three parameters α, β, and γ in our model (9). Grid searching is adopted for all datasets. We show their effects on accuracy using the Caltech20 dataset in Fig. 3. We can see that the results are satisfying and relatively stable for a wide range of parameters.

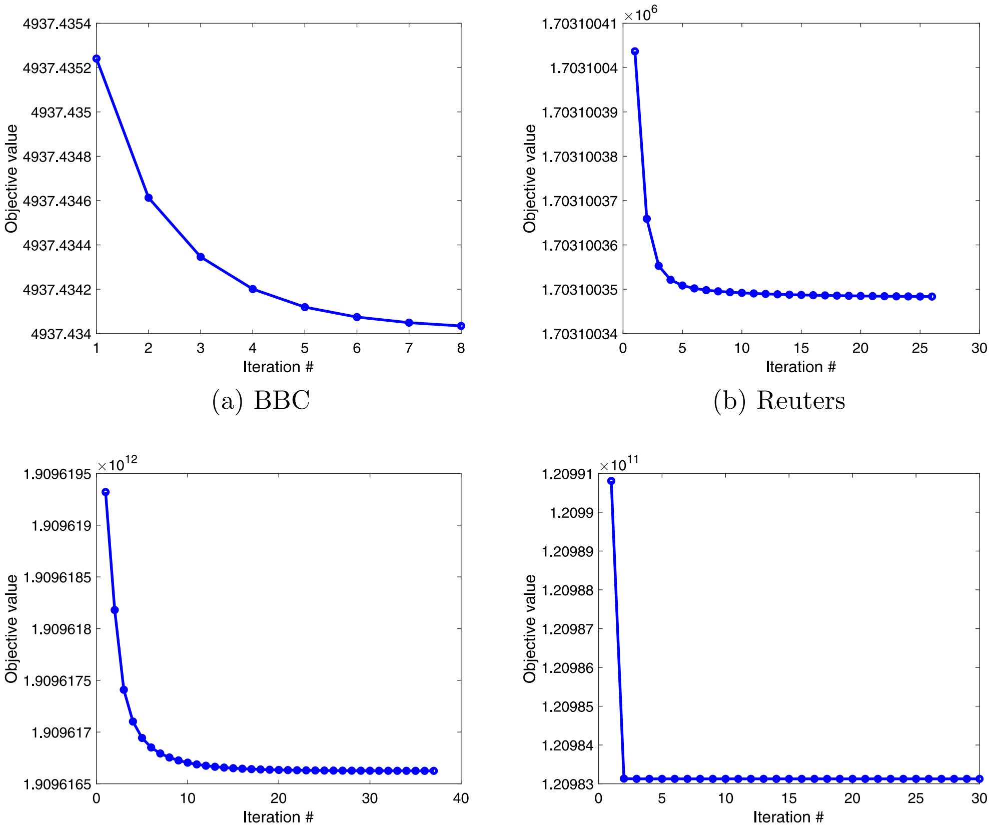
5.6. Robustness study

Orthogonal to existing methods, we integrate multiview information in partition space, which can alleviate the impact of noise to some extent. Unlike data representation which can be easily corrupted by noise, partition space is more robust. Since all views admit the same clustering pattern, we can easily find a good partition from all partitions. To demonstrate this, we construct a new dataset by choosing 24 images for each class from Caltech20. We add different levels of Gaussian noise. Some clean images and corruptions are shown in Fig. 6.

Then we evaluate our algorithm's performance on noisy images. We compare PMSC with AMGL, which provides comparable performance on Caltech20 data as demonstrated in Table 5. The clustering results are presented in Tables 6 and 7. We can notice that our proposed method consistently outperforms AMGL by a large margin. This demonstrates that our method is robust to noise due to the adoption of partition space.

6. Conclusion

In this paper, we propose a unified model for multiview subspace clustering. Different from existing methods, we seek to find



(a) BBC

(b) Reuters

(c) HW

(d) Caltech20

Fig. 5. The convergence curves of objective function (9) on each dataset.

Table 6
Clustering performance on Caltech20 (%).

Noise	Mean = 0, Variance = 0.05			Mean = 0, Variance = 0.1		
Method	ACC	Purity	NMI	ACC	Purity	NMI
AMGL	27.21(1.76)	30.35(1.91)	30.74(1.34)	21.71(1.46)	23.31(1.20)	22.60(1.21)
PMSC	33.14(1.20)	46.54(2.56)	31.49(1.18)	28.65(1.77)	39.81(2.35)	27.89(2.03)

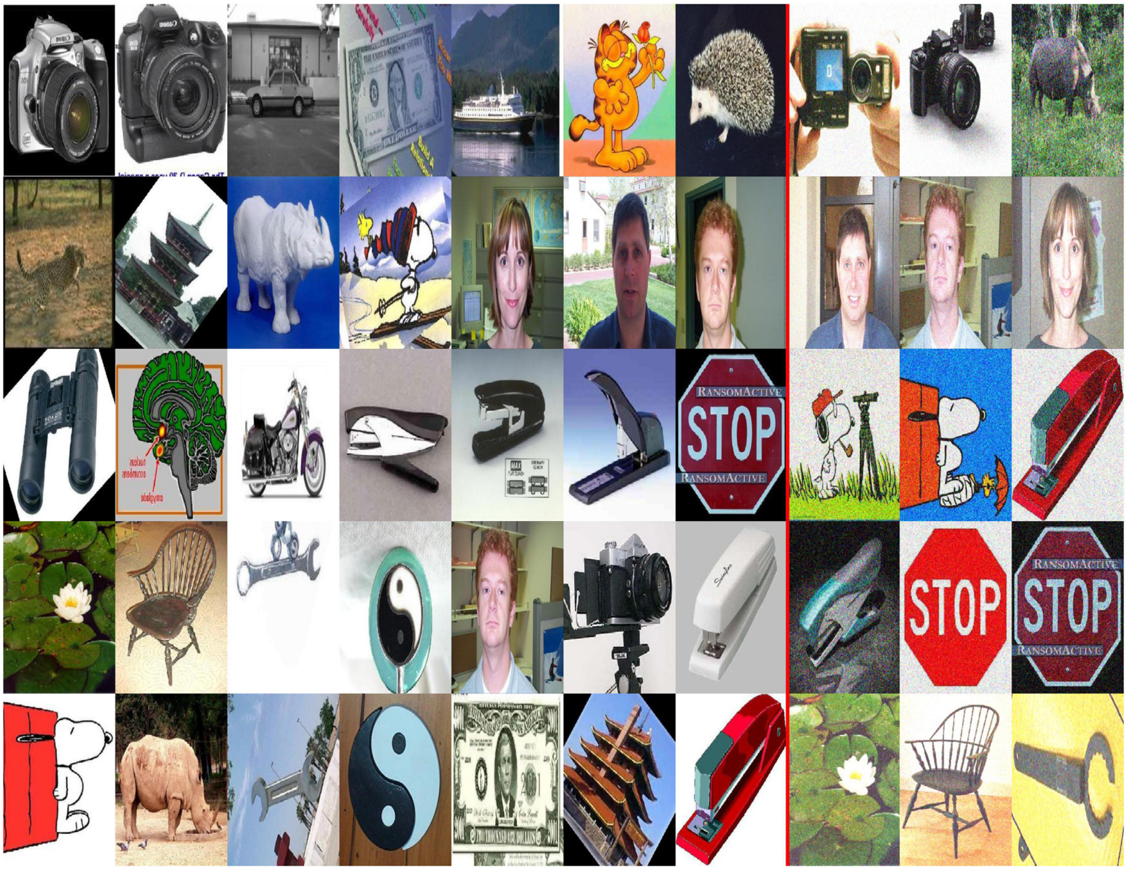


Fig. 6. Sample images of Caltech20. The last three columns display some noisy images.

Table 7
Clustering performance on Caltech20 (%).

Noise	Mean = 0.1, Variance = 0.01			Mean = 0.3, Variance = 0.01		
Method	ACC	Purity	NMI	ACC	Purity	NMI
AMGL	43.04(3.56)	47.27(2.83)	47.97(2.37)	41.85(1.84)	45.85(1.34)	47.16(1.68)
PMSC	46.52(2.23)	59.10(1.74)	48.90(2.08)	50.71(2.32)	59.69(2.09)	52.69(2.55)

a consensus clustering from multiple partitions. That is to say, we obtain a partition for each view and then find a combined clustering with better quality. Compared to directly combine multiview information in the feature space, this partition level fusion approach is in a better position to employ the hidden cluster structure information. When one of the basic partition realizes the ground truth, the consensus clustering can be easily found. Eventually, the graph construction, the generation of basic partitions, and fusion of consensus clustering are implemented in an interactive way, i.e., they are iteratively updated in a mutually promotional way. Real-world data experiments show the effectiveness of our proposed method.

Acknowledgments

This paper was in part supported by Grants from the Natural Science Foundation of China (Nos. 61806045, 61572111, 61772115, 61806135, 61625204, and 61836006) and Fundamental Research Funds for the Central Universities of China (Nos. ZYGX2017KYQD177, YJ201949, 2018SCUH0070, and A030170237 01012).

References

Abavisani, M., & Patel, V. M. (2018). Multimodal sparse and low-rank subspace clustering. *Information Fusion*, *39*, 168–177.

Cai, X., Nie, F., & Huang, H. (2013). Multi-view k-means clustering on big data. In *IJCAI* (pp. 2598–2604).

Cao, X., Zhang, C., Fu, H., Liu, S., & Zhang, H. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 586–594).

Chao, G., Sun, S., & Bi, J. (2017). A survey on multi-view clustering. arXiv preprint arXiv:1712.06246.

Chen, S. S., Donoho, D. L., & Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM Review*, *43*(1), 129–159.

Chen, X., Hong, W. H., Nie, F. N., He, D., Yang, M., & Huang, J. Z. (2018). Directly minimizing normalized cut for large scale data. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, KDD-18* (pp. 1206–1215).

Chen, X., Xu, X., Ye, Y., & Huang, J. Z. (2013). TW-k-means: Automated two-level variable weighting clustering algorithm for multi-view data. *IEEE Transactions on Knowledge and Data Engineering*, *25*(4), 932–944.

Chen, X., Ye, Y., Xu, X., & Huang, J. Z. (2012). A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, *45*(1), 434–446.

Elhamifar, E., & Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2765–2781.

Gao, H., Nie, F., Li, X., & Huang, H. (2015). Multi-view subspace clustering. In *Proceedings of the IEEE international conference on computer vision* (pp. 4238–4246).

Huang, S., Kang, Z., & Xu, Z. (2018). Self-weighted multi-view clustering with soft capped norm. *Knowledge-Based Systems*, *158*, 1–8.

Huang, S., Kang, Z., & Xu, Z. (2019). Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recognition*, 107015.

Huang, S., Kang, Z., & Xu, Z. (2020). Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recognition*, *97*, 107015.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, *31*(8), 651–666.

Kang, Z., Guo, Z., Huang, S., Wang, S., Chen, W., Su, Y., et al. (2019). Multiple partitions aligned clustering. In *IJCAI* (pp. 2701–2707).

Kang, Z., Pan, H., Hoi, S. C. H., & Xu, Z. (2019). Robust graph learning from noisy data. *IEEE Transactions on Cybernetics*, 1–11.

Kang, Z., Peng, C., & Cheng, Q. (2017). Kernel-driven similarity learning. *Neurocomputing*, *267*, 210–219.

Kang, Z., Shi, G., Shi, Huang, S., Chen, W., Pu, X., et al. (2019). Multi-graph fusion for multi-view spectral clustering. *Knowledge-Based Systems*.

Kang, Z., Wen, L., Chen, W., & Xu, Z. (2019). Low-rank kernel learning for graph-based clustering. *Knowledge-Based Systems*, *163*, 510–517.

Kang, Z., Xu, H., Wang, B., Zhu, H., & Xu, Z. (2019). Clustering with similarity preserving. *Neurocomputing*, *365*, 211–218.

Kumar, A., & Daumé, H. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 393–400).

Kumar, A., Rai, P., & Daume, H. (2011). Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems* (pp. 1413–1421).

Li, Z., Liu, J., Tang, J., & Lu, H. (2015). Robust structured subspace learning for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(10), 2085–2098.

Li, S., Shao, M., & Fu, Y. (2018). Multi-view low-rank analysis with applications to outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *12*(3), 32.

Liu, H., & Fu, Y. (2018). Consensus guided multi-view clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *12*(4), 42.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., & Ma, Y. (2013). Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 171–184.

Liu, J., Wang, C., Gao, J., & Han, J. (2013). Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM international conference on data mining* (pp. 252–260). SIAM.

Liu, X., Zhu, X., Li, M., Wang, L., Zhu, E., Liu, T., et al. (2019). Multiple kernel k-means with incomplete kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, *2*, 849–856.

Nie, F., Cai, G., & Li, X. (2017). Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI* (pp. 2408–2414).

Nie, F., Li, J., Li, X., et al. (2016). Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *IJCAI* (pp. 1881–1887).

Peng, X., Feng, J., Xiao, S., Yau, W.-Y., Zhou, J. T., & Yang, S. (2018). Structured autoencoders for subspace clustering. *IEEE Transactions on Image Processing*, *27*(10), 5076–5086.

Peng, C., Kang, Z., Cai, S., & Cheng, Q. (2018). Integrate and conquer: Double-sided two-dimensional k-means via integrating of projection and manifold construction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *9*(5), 57.

Peng, X., Lu, C., Yi, Z., & Tang, H. (2018). Connections between nuclear-norm and frobenius-norm-based representations. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(1), 218–224.

Sun, S., Liu, Y., & Mao, L. (2019). Multi-view learning for visual violence recognition with maximum entropy discrimination and deep features. *Information Fusion*, *50*, 43–53.

Tang, C., Chen, J., Liu, X., Li, M., Wang, P., Wang, M., et al. (2018). Consensus learning guided multi-view unsupervised feature selection. *Knowledge-Based Systems*, *160*, 49–60.

Tao, Z., Liu, H., Li, S., Ding, Z., & Fu, Y. (2017). From ensemble clustering to multi-view clustering. In *Proc. of the twenty-sixth int. joint conf. on artificial intelligence (IJCAI)* (pp. 2843–2849).

Tzortzis, G., & Likas, A. (2012). Kernel-based weighted multi-view clustering. In *Data mining (ICDM), 2012 IEEE 12th international conference on* (pp. 675–684). IEEE.

Vidal, R. (2011). Subspace clustering. *IEEE Signal Processing Magazine*, *28*(2), 52–68.

Wang, Y., Zhang, W., Wu, L., Lin, X., Fang, M., & Pan, S. (2016). Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering. arXiv preprint arXiv:1608.05560.

Xu, C., Tao, D., & Xu, C. (2015). Multi-view self-paced learning for clustering. In *IJCAI* (pp. 3974–3980).

Yang, Y., Shen, F., Huang, Z., Shen, H. T., & Li, X. (2017). Discrete nonnegative spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, *29*(9), 1834–1845.

Yang, Y., Song, J., Huang, Z., Ma, Z., Sebe, N., & Hauptmann, A. G. (2013). Multi-feature fusion via hierarchical regression for multimedia analysis. *IEEE Transactions on Multimedia*, *15*(3), 572–581.

Zhan, K., Zhang, C., Guan, J., & Wang, J. (2017). Graph learning for multiview clustering. *IEEE Transactions on Cybernetics*, (99), 1–9.

Zhang, C., Fu, H., Liu, S., Liu, G., & Cao, X. Low-rank tensor constrained multiview subspace clustering. In *Proceedings of the IEEE international conference on computer vision* (pp. 1582–1590).

Zhang, C., Hu, Q., Fu, H., Zhu, P., & Cao, X. (2017). Latent multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4279–4287).

Zhang, Z., Ren, J., Li, S., Hong, R., Zha, Z., & Wang, M. (2019). Robust subspace discovery by block-diagonal adaptive locality-constrained representation. In *Proceedings of the 27th ACM international conference on multimedia*.

Zhao, Y., & Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis.

Zhou, S., Zhu, E., Liu, X., Zheng, T., Liu, Q., Xia, J., et al. (2019). Subspace segmentation-based robust multiple kernel clustering. *Information Fusion*.

Zhuge, W., Hou, C., Jiao, Y., Yue, J., Tao, H., & Yi, D. (2017). Robust auto-weighted multi-view subspace clustering with common subspace representation matrix. *PloS One*, *12*(5), e0176769.